Methods in Molecular Biology

Series Editor
John M. Walker
School of Life Sciences
University of Hertfordshire
Hatfield, Hertfordshire, AL10 9AB, UK

For further volumes:
http://www.springer.com/series/7651

Molecular Modeling of Proteins

Second Edition

Edited by

Andreas Kukol

University of Hertfordshire, Hatfield, Hertfordshire, UK

Humana Press

Editor
Andreas Kukol
University of Hertfordshire
Hatfield
Hertfordshire, UK

Additional material to this book can be downloaded from http://extras.springer.com

ISBN 978-1-4939-5491-9		ISBN 978-1-4939-1465-4 (eBook)
DOI 10.1007/978-1-4939-1465-4
Springer New York Heidelberg Dordrecht London

Softcover reprint of the hardcover 2nd edition 2015

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Humana Press is a brand of Springer
Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Over the years, molecular modeling and simulation of biomolecules has become an important tool in the molecular biosciences. Initially situated in the realm of specialists with in-depth knowledge of physics and computer science and access to supercomputers, molecular modeling is used increasingly by bioscientists who are mainly interested in investigating biological problems. This development has been supported by improved hardware, such as multi-core processors or graphic processing units, on the one hand, and accelerated sampling algorithms on the other hand that increase the timescale without increasing the demands on the hardware or the calculation time. The purpose of *Molecular Modeling of Proteins* is to provide a theoretical background of various methods available and to enable nonspecialists to apply methods to their problems. Most chapters contain, in addition to a thorough introduction, step-by-step instructions and notes on troubleshooting and how to avoid common pitfalls.

The current second edition of *Molecular Modeling of Proteins* provides some updated chapters and new material not covered in the first edition. The first part describes classical and advanced simulation methods as well as methods to set up complex systems such as lipid membranes and membrane proteins. The second part is devoted to the simulation and analysis of conformational changes of proteins, while Part III covers computational methods for protein structure prediction as well as using experimental data in combination with computational techniques. The final part contains chapters concerning protein–ligand interactions, which are relevant in the drug design process.

The topics cover some long established methods together with the latest developments in the field. The chapters are written by internationally renowned investigators: they include leading developers of popular simulation packages or force fields.

The second edition of *Molecular Modeling of Proteins* is directed at researchers in the physical-, chemical-, and biosciences working in industry and academia, who are interested in applying the methods in their own research. Additionally, the book forms a valuable resource for educators who wish to teach courses about molecular modeling.

Hertfordshire, UK ***Andreas Kukol***

Contents

Part III Protein Structure Determination

Part IV Protein–Ligand Interactions

Contributors

ROMMIE E. AMARO • *Department of Chemistry and Biochemistry, University of California, San Diego, CA, USA*

ALESSANDRO BARDUCCI • *Laboratory of Statistical Biophysics, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland*

JONATHAN BARNOUD • *INSERM, Lyon, France*

PHILIP C. BIGGIN • *Department of Biochemistry, University of Oxford, Oxford, UK*

PETER J. BOND • *Department of Chemistry, The Unilever Centre for Molecular Science Informatics, Cambridge, USA; Department of Biological Sciences, National University of Singapore, Singapore*

MASSIMILIANO BONOMI • *Department of Bioengineering and Therapeutic Sciences and California Institute of Quantitative Biosciences, University of California, San Francisco, CA, USA*

ALEXANDRE M.J.J. BONVIN • *Computational Structural Biology Group, Bijvoet Center for Biomolecular Research, Faculty of Science, Utrecht University, Utrecht, The Netherlands*

ÖZLEM DEMIR • *Department of Chemistry and Biochemistry, University of California, San Diego, CA, USA*

VICTORIA A. FEHER • *Department of Chemistry and Biochemistry, University of California, San Diego, CA, USA*

VYTAUTAS GAPSYS • *Max Planck Institute for Biophysical Chemistry, Göttingen, Germany*

PATRICK C. GEDEON • *Department of Biomedical Engineering, Duke University, Durham, NC, USA*

FRAUKE GRÄTER • *Heidelberg Institute for Theoretical Studies, Heidelberg, Germany*

BERT L. DE GROOT • *Max Planck Institute for Biophysical Chemistry, Göttingen, Germany*

OLGUN GUVENCH • *Department of Pharmaceutical Sciences, University of New England, Portland, ME, USA*

MING-JING HWANG • *Institute of Biomedical Sciences, Academia Sinica, Taipei, Taiwan*

ROBERT L. JERNIGAN • *National Cancer Institute, National Institute of Health, Bethesda, MD, USA; Interdepartmental Program for Bioinformatics and Computational Biology, L.H. Baker Center for Bioinformatics and Biological Statistics, Iowa State University, Ames, IA, USA*

KEJUE JIA • *National Cancer Institute, National Institute of Health, Bethesda, MD, USA; Interdepartmental Program for Bioinformatics and Computational Biology, L.H. Baker Center for Bioinformatics and Biological Statistics, Iowa State University, Ames, IA, USA*

EZGI KARACA • *Computational Structural Biology Group, Bijvoet Center for Biomolecular Research, Faculty of Science, Utrecht University, Utrecht, The Netherlands*

ATAUR R. KATEBI • *National Cancer Institute, National Institute of Health, Bethesda, MD, USA; Interdepartmental Program for Bioinformatics and Computational Biology, L.H. Baker Center for Bioinformatics and Biological Statistics, Iowa State University, Ames, IA, USA*

ANDREAS KUKOL • *School of Life and Medical Sciences, University of Hertfordshire, Hatfield, UK*

MARC F. LENSINK • *Interdisciplinary Research Institute, CNRS USR3078, University Lille1, Villeneuve d'Ascq, France*

HADAS LEONOV • *Max Planck Institute for Biophysical Chemistry, Göttingen, Germany*

WENJIN LI • *Department of Bioengineering, University of Illinois at Chicago, Chicago, IL, USA*

ERIK LINDAHL • *Department of Biochemistry and Biophysics, Stockholm University, Stockholm, Sweden*

PEDRO E.M. LOPES • *Department of Pharmaceutical Sciences, University of Maryland, Baltimore, MD, USA*

GERALD H. LUSHINGTON • *LiS Consulting, Lawrence, KS, USA*

ALEXANDER D. MACKERELL JR. • *Department of Pharmaceutical Sciences, University of Maryland, Baltimore, MD, USA*

JEFFRY D. MADURA • *Department of Chemistry and Biochemistry and Center for Computational Sciences, Duquesne University, Pittsburgh, PA, USA*

SERVAAS MICHIELSSENS • *Max Planck Institute for Biophysical Chemistry, Göttingen, Germany*

LUCA MONTICELLI • *INSERM, Lyon, France*

TIMOTHY NUGENT • *Department of Computer Science, University College London, London, UK*

JUAN R. PERILLA • *Beckman Institute, University of Illinois, Urbana at Urbana-Champaign, IL, USA*

JAN HENNING PETERS • *Max Planck Institute for Biophysical Chemistry, Göttingen, Germany*

JIM PFAENDTNER • *Department of Chemical Engineering, University of Washington, Seattle, WA, USA*

JOÃO P.G.L.M. RODRIGUES • *Computational Structural Biology Group, Bijvoet Center for Biomolecular Research, Faculty of Science, Utrecht University, Utrecht, The Netherlands*

KANNAN SANKAR • *National Cancer Institute, National Institute of Health, Bethesda, MD, USA; Interdepartmental Program for Bioinformatics and Computational Biology, L.H. Baker Center for Bioinformatics and Biological Statistics, Iowa State University, Ames, IA, USA*

GIOVANNI SETTANNI • *Physics Department, Johannes Gutenberg Universität, Mainz, Germany*

JESPER SØRENSEN • *Department of Chemistry and Biochemistry, University of California, San Diego, CA, USA*

ROBERT V. SWIFT • *Department of Chemistry and Biochemistry, University of California, San Diego, CA, USA*

JAMES R. THOMAS • *Department of Chemistry and Biochemistry and Center for Computational Sciences, Duquesne University, Pittsburgh, PA, USA*

WIM F. VRANKEN • *Department of Structural Biology, Structural Biology Brussels, Vrije Universiteit Brussel, Brussels, Belgium*

GEERTEN W. VUISTER • *Department of Biochemistry, University of Leicester, Leicester, UK*

DAVID S. WISHART • *Department of Computing Science, University of Alberta, Edmonton, AB, Canada; Department of Biological Sciences, University of Alberta, Edmonton, AB, Canada*

THOMAS B. WOOLF • *Department of Physiology, John Hopkins University, Baltimore, MD, USA*

ZHONG-RU XIE • *Institute of Biomedical Sciences, Academia Sinica, Taipei, Taiwan*

Part I

Simulation Methods

Chapter 1

Molecular Dynamics Simulations

Erik Lindahl

Abstract

Molecular dynamics has evolved from a niche method mainly applicable to model systems into a cornerstone in molecular biology. It provides us with a powerful toolbox that enables us to follow and understand structure and dynamics with extreme detail—literally on scales where individual atoms can be tracked. However, with great power comes great responsibility: Simulations will not magically provide valid results, but it requires a skilled researcher. This chapter introduces you to this, and makes you aware of some potential pitfalls. We focus on the two basic and most used methods; optimizing a structure with energy minimization and simulating motion with molecular dynamics. The statistical mechanics theory is covered briefly as well as limitations, for instance the lack of quantum effects and short timescales. As a practical example, we show each step of a simulation of a small protein, including examples of hardware and software, how to obtain a starting structure, immersing it in water, and choosing good simulation parameters. You will learn how to analyze simulations in terms of structure, fluctuations, geometrical features, and how to create ray-traced movies for presentations. With modern GPU acceleration, a desktop can perform μs-scale simulations of small proteins in a day—only 15 years ago this took months on the largest supercomputer in the world. As a final exercise, we show you how to set up, perform, and interpret such a folding simulation.

Key words Molecular dynamics, Simulation, Force field, Protein, Solvent, Energy minimization, Position restraints, Equilibration, Trajectory analysis, Secondary structure

1 Introduction

Biomolecular dynamics occur over a wide range of scales in both time and space, and the choice of approach to study them depends on the question asked. In many cases the best alternative is an experimental technique, for instance spectroscopy to study bond vibrations or electrophysiology to study ion channels opening and closing. However, theoretical methods have made huge advances the last few decades, and there are now large domains where modeling and simulation either provide more detail or are more efficient to use compared to setting up a new experiment.

Molecular dynamics simulation is far from the only theoretical method; when the aim is to predict for example the structure

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_1, © Springer Science+Business Media New York 2015

and/or function of proteins (rather than studying the dynamics of a protein) the best tool is normally bioinformatics that detect related proteins from amino acid sequence similarity. Similarly, for computational drug design often it is much more productive to use less accurate but exceptionally fast statistical methods like QSAR (Quantitative Structure–Activity Relationship) instead of spending billions of CPU hours to simulate binding of thousands of compounds.

Traditionally, the role of simulations has been to test if simple theoretical models can predict experimental observations. For example, simulations of ion channels have been useful to explain *why* some ions pass while others are blocked, although the conductivity itself was already known from experiments. Similarly, simulations can provide detail not accessible through experiments, for instance pressure distributions inside membranes. However, this is changing rapidly—simulations have moved far beyond confirming experiments, and today they frequently make predictions about properties such as binding or folding dynamics that are later confirmed in the lab. With ever-increasing computational power this development will not only continue, but it is likely to accelerate significantly the next few years.

From an ideal physics point-of-view, the time-dependent Schrödinger equation should be able to predict all properties of any molecule with arbitrary precision ab initio. However, as soon as more than a handful of particles are involved it is necessary to introduce approximations. In quantum chemistry, one common approximation is to assume that atomic nuclei do not move, and using an implicit representation of solvent. This is obviously not realistic for large biomolecules if we are interested in understanding their motion and sampling of lots of different states, so for most biomolecular systems we instead choose to work with empirical parameterizations of models, for instance classical Coulomb interactions between pointlike atomic charges. The conceptual difference is that quantum chemistry is excellent at describing the electronic structure and enthalpy (potential) of the system, while classical molecular dynamics instead excels at sampling the billions of states a macromolecule will adapt—in particular this means they properly include the entropy part of free energy. These models are not only orders of magnitude faster, but since they have been parameterized from experiments they also perform better when it comes to reproducing observations on microsecond scale (Fig. 1), rather than extrapolating quantum models 10 orders of magnitude. The first molecular dynamics simulation was performed as late as 1957 [1], although it was not until the 1970s that it was possible to simulate water [2] and biomolecules [3].

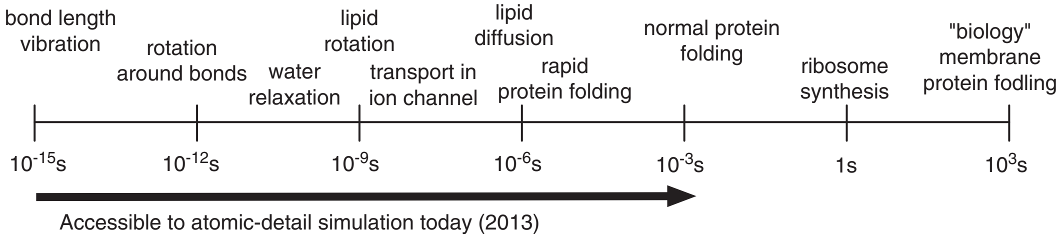


Fig. 1 Range of time scales for dynamics in biomolecular systems. While the individual time steps of molecular dynamics is 1–2 fs, parallel computers make it possible to simulate on microsecond scale, and distributed computing techniques can sample even slower processes, almost reaching milliseconds

2 Theory

Macroscopic properties measured in an experiment are not direct observations, but averages over billions of molecules representing a statistical mechanics *ensemble*. This has deep theoretical implications that are covered in great detail in the literature [4, 5], but even from a practical point of view there are important consequences: (1) It is not sufficient to work with individual structures, but systems have to be expanded to generate a representative ensemble of structures (*see* **Note 1**) at the given experimental conditions, e.g., temperature and pressure—this is one thing that sets classical molecular dynamics apart from quantum chemistry. (2) Thermodynamic equilibrium properties related to free energy, such as binding constant, solubilities, and relative stability cannot be calculated directly from individual simulations, but require more elaborate techniques covered in later chapters—these all rely on entropy. (3) For equilibrium properties (in contrast to kinetic) the aim is to examine the ensemble of structures, and *not* necessarily to reproduce individual atomic trajectories!

The two most common ways to generate statistically faithful equilibrium ensembles are *Monte Carlo* and *Molecular Dynamics simulations*, where the latter also has the advantage of accurately reproducing kinetics of non-equilibrium properties such as diffusion or folding times. However, these methods cannot handle the case where a structure is very far from equilibrium, for instance if two atoms are almost overlapping after building a new side chain. To remove this type of clashes prior to simulation, we typically start with an *Energy Minimization*. This type of minimization is also commonly used to refine low-resolution experimental structures.

All classical simulation methods rely on more or less empirical sets of parameters called *Force fields* [6–9] to calculate interactions and evaluate the potential energy of the system as a function of pointlike atomic coordinates. A force field consists of both the set of equations used to calculate the potential energy and forces from particle coordinates, as well as a collection of parameters used in

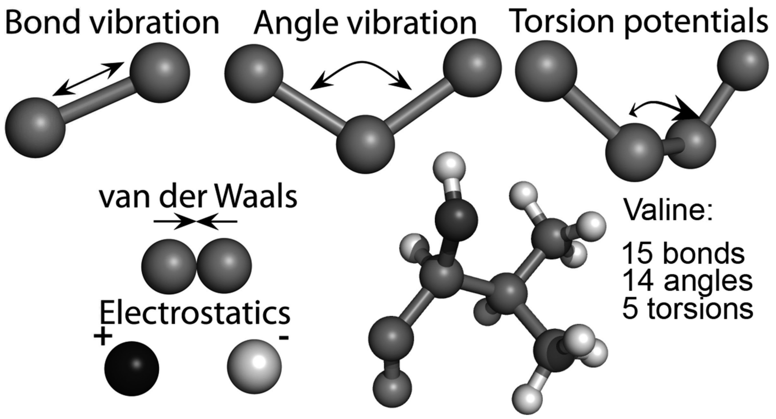


Fig. 2 Examples of interaction functions in modern force fields. Bonded interactions include covalent bond-stretching, angle-bending, torsion rotation around bonds, and out-of-plane or "improper" torsions (not shown). Nonbonded interactions are based on neighborlists and consist of Lennard–Jones attraction and repulsion, as well as Coulomb electrostatics. Even a small amino acid residue contains a large number of interactions, and for a protein there are thousands

the equations. For most purposes these approximations work great, but they cannot reproduce quantum effects such as bond formation or breaking (*see* **Note 2**).

All common force fields subdivide potential functions in two classes. *Bonded interactions* cover stretching of covalent bonds, angle-bending, torsion potentials when rotating around bonds, and out-of-plane "improper torsion" potentials, all which are normally fixed throughout a simulation (Fig. 2). The remaining *nonbonded interactions* between atoms that are merely close in space consist of Lennard–Jones repulsion and dispersion as well as Coulomb electrostatic. These are typically computed from neighborlists updated periodically.

Given the force on all atoms, the coordinates are updated for the next step. For energy minimization, the *steepest descent* algorithm simply moves each atom a short distance in direction of decreasing energy (force is the negative gradient of energy), while molecular dynamics is performed by integrating Newton's equations of motion [10]:

$$\mathbf{F}_i = -\frac{\partial V(\mathbf{r}_1,\ldots,\mathbf{r}_N)}{\partial \mathbf{r}_i}$$

$$m_i \frac{\partial^2 \mathbf{r}_i}{\partial t^2} = \mathbf{F}_i$$

The updated coordinates are then used to evaluate the potential energy again, as shown in the flowchart of Fig. 3.

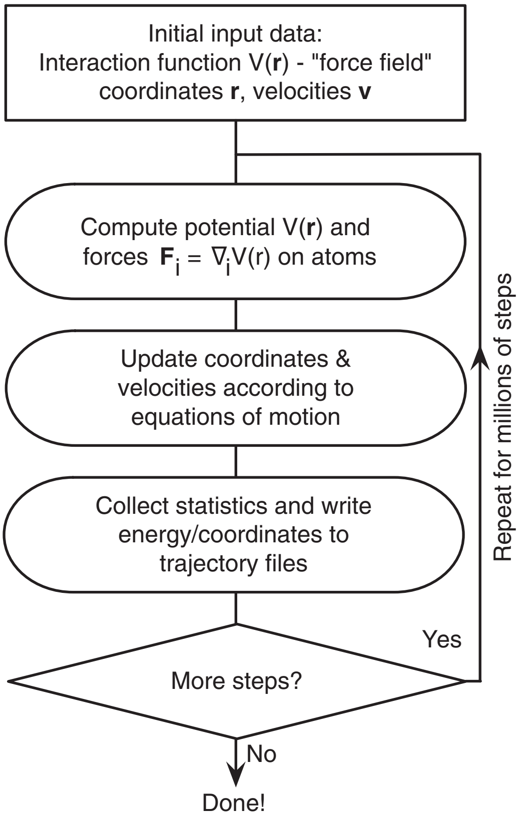

Initial input data:
Interaction function V(**r**) - "force field"
coordinates **r**, velocities **v**
Compute potential V(**r**) and
forces **F**$_i$ = ∇$_i$V(r) on atoms
Update coordinates &
velocities according to
equations of motion
Collect statistics and write
energy/coordinates to
trajectory files
More steps?
Yes
No
Done!
Repeat for millions of steps

Fig. 3 Simplified flowchart of a typical molecular dynamics simulation. The basic idea is to generate structures from a natural ensemble by calculating potential functions and integrating Newton's equations of motion, structures which are then used to evaluate equilibrium properties of the system. A typical time step is in the order of 1 or 2 femtoseconds, unless special techniques are used

Even the smallest chemical sample we can imagine is far too large to include completely in a simulation. Instead, biomolecular simulations normally uses *periodic boundary conditions* to avoid surface artifacts, so that a water molecule that exits to the right reappears on the left in the system; if the box is sufficiently large the molecules will not interact significantly with their periodic copies. This is intimately related to the nonbonded interactions, which ideally should be summed over all neighbors in the resulting infinite periodic system. Simple cutoffs can work for Lennard–Jones interactions that decay very rapidly, but for Coulomb interactions a sudden cutoff can lead to large errors. In the early days of simulation is was common to "switch off" the electrostatic interaction before the cutoff as shown in Fig. 4, but this too has severe artifacts—the current method of choice is to use Particle-Mesh-Ewald summation (PME) to calculate the infinite electrostatic interactions by splitting the summation into short- and long-range parts [11].

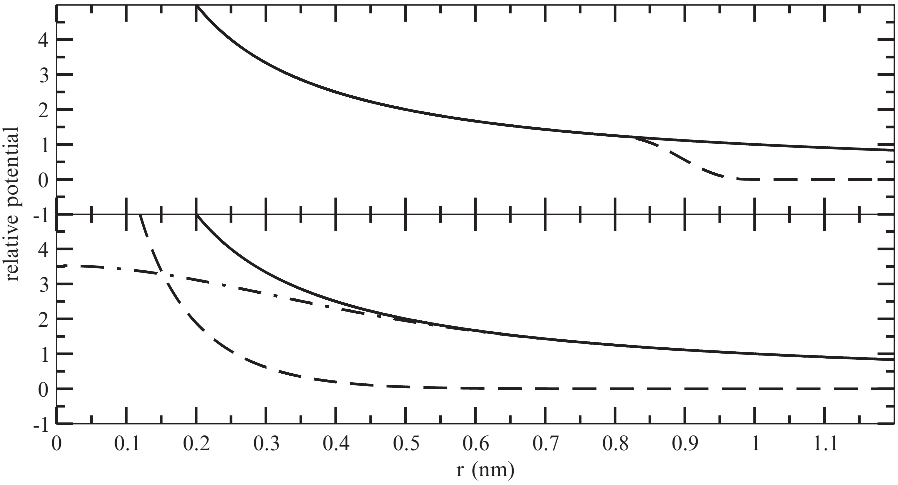


Fig. 4 Alternatives to a sharp cutoff for nonbonded coulomb interactions. *Top*: By switching off the interaction (*dashed*) before the cutoff the force will be the exact derivative of potential, but the derivative (and thus force) will unnaturally *increase* just before the cutoff. *Bottom*: Particle-Mesh-Ewald is an amazing algorithm where the coulomb interaction (*solid*) is divided into a short-range term that is evaluated within a cutoff (*dashed*) and a long-range term which can be solved exactly in reciprocal space with Fourier transforms (*dot-dash*)

For PME, the cutoff is not really a cutoff; it only determines the balance between the two parts, and the long-range part is treated by assigning charges to a grid that is solved in reciprocal space through Fourier transforms.

Cutoffs and rounding errors can lead to drifts in energy, which will cause the system to heat up during the simulation. Even with a theoretically perfect simulation we would run into problems since we typically start from an imperfect structure. As the potential energy of this structure decreases during the simulation, the kinetic energy (i.e., temperature) would increase if the total system energy was constant. To control this, the system is normally coupled to a thermostat that scales velocities during the integration to maintain room temperature. Similarly, the total pressure in the system can be adjusted through scaling the simulation box size, either isotropically or separately in x/y/z dimensions.

The single most demanding part of simulations is the computation of nonbonded interactions, since millions of pairs have to be evaluated for each time step. Extending the time step is thus an important way to improve simulation performance, but unfortunately errors are introduced in bond vibrations already at 1 fs. However, in most simulations these bond vibrations are not of interest per se, and can be removed entirely by introducing *bond constraint* algorithms such as SHAKE [12] or LINCS [13]. Constraints make it possible to extend time steps to 2 fs, and fixed-length bonds are likely better approximations of the quantum mechanical oscillators than harmonic springs (*see* **Note 3**)—and in the final section we will show you how to go even further.

3 Methods

With the basic theory covered, this section will describe how to (1) choose and obtain a starting structure, (2) prepare it for a simulation, (3) create a simulation box, (4) add solvent water, (5) perform energy minimization, (6) equilibrate the structure with simulation, (7) perform the production simulation, and (8) analyze the trajectory data. To reproduce it, you will need access to a Unix/Linux machine (*see* **Note 4**) with a molecular dynamics package installed. While the options and files below refer to the GROMACS program [14], the description should be reasonably straightforward to follow with other programs like AMBER [15], CHARMM [16], or NAMD [17]. It will also be useful to have the molecular viewer PyMOL [18] and Unix graph program Grace installed (*see* **Note 5**).

3.1 Obtaining a Starting Structure

The Bovine Pancreatic Trypsin Inhibitor (BPTI) is a small 58-residue water-soluble protein that inhibits several serine proteases [19]. It was one of the first proteins to be simulated [3], and has often been referred to as a "hydrogen atom" of protein simulation. There are several high-resolution X-ray structures of BPTI [20] in the Protein Data Bank (http://www.pdb.org), and also NMR structures. It was actually the early simulations of BPTI [3] that lead experimentalists to realize that X-ray temperature factors can be used to study the local dynamics of a protein [21]. Choose the entry 6PTI with 1.7 Å resolution [20], and download it as 6PTI.pdb (*see* **Note 6**). Figure 5 shows a cartoon representation of this structure; the small crosses are crystal water oxygen atoms visible in the X-ray experiment (*see* **Note 7**).

3.2 Preparation of Input Data

In addition to the coordinates/velocities that change each step, simulations also need a static description of all atoms and interactions in the system, called *topology*. In GROMACS, this is created from the PDB structure by the program pdb2gmx, which also adds all the hydrogen atoms that are not present in most X-ray structures. For this example we will work with the Amber99SB-ILDN force field, the TIP3P [22] water model (*see* **Note 8**), and accept the default choices for all residue protonation states, termini, disulfide bridges, etc. If you just try the command below right away you will get an error due to the issues with the structure mentioned in **Note 6**. This is common, so it is something you need to learn how to fix. Open the PDB file in an editor and scroll down to residue 57 at the end of the chain. Remove the single nitrogen atom line just before the line starting with "TER"—we simply skip the missing residues. Just after the "TER" line there are also some lines for a phosphate ion (residue "PO4"). To avoid problems with finding parameters for this, remove these five lines too. Now you should be good to go. The command to use is then

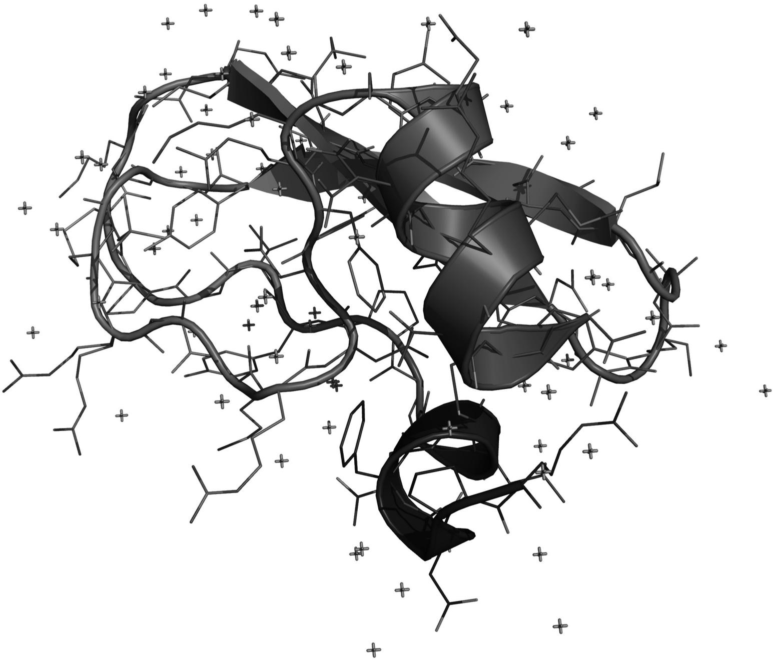


Fig. 5 Cartoon representation of the BPTI structure 6PTI from Protein Data Bank, with side chains shown as sticks. Including hydrogens, the protein contains roughly 800 atoms. Ray-traced image generated with PyMOL

```
pdb2gmx -f 6PTI.pdb -water tip3p
```

You will be prompted for the force field (select "6" for Amber99SB-ILDN), and the command will produce three files: `conf.gro` contains coordinates with hydrogens, `topol.top` is the topology, and `posre.itp` contains a list of position restraints that will be used in Subheading 3.7. For all these programs, you can use the `-h` flag for help and a detailed list of options (*see* **Note 9**).

3.3 Creating a Simulation Box

The default box is taken from the PDB crystal cell, but a simulation in water requires something larger. The box size is a trade-off, though: volume is proportional to the box side cubed, and more water means the simulation is slower. The easiest option it to place the solute in the center of a cube, with for example 0.75 nm to the box sides. We will show up some more advanced alternatives later, but for now this will suffice:

```
editconf -f conf.gro -d 0.75 -o box.gro
```

where the distance (-d) flag automatically centers the protein in the box, and the new conformation is written to the file `box.gro` (*see* **Note 10**).

3.4 Adding Solvent Water

The last step before the simulation is to add water in the box to solvate the protein. This is done by using a small pre-equilibrated system of water coordinates that is repeated over the box, and

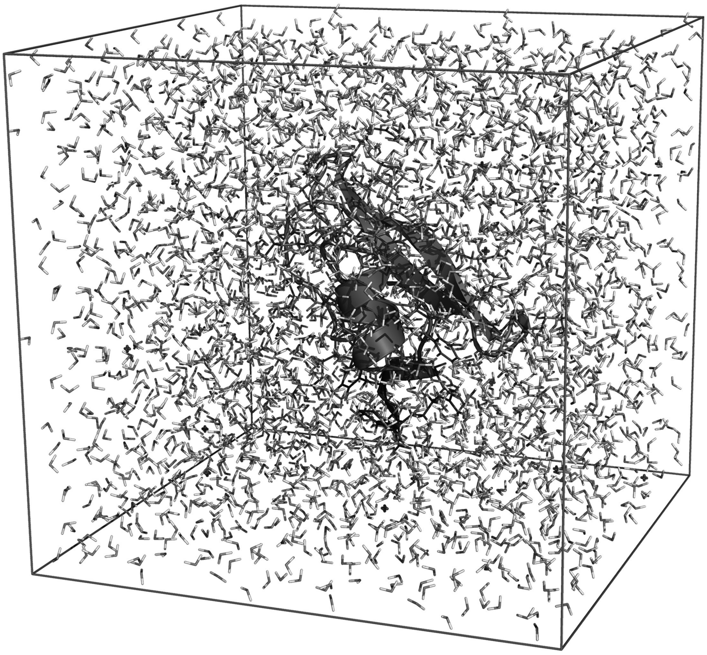


Fig. 6 BPTI solvated in water in a cubic box. Note that there is quite a lot of water, in particular in the box corners

overlapping water molecules removed. The BPTI system will require roughly 3,400 water molecules, which increases the number of atoms significantly. GROMACS does not use a special pre-equilibrated system for TIP3P water since water coordinates can be used with any model—the actual parameters are stored in the topology and force field. In GROMACS, a suitable command to solvate the new box would be

```
genbox -cp box.gro -cs spc216.gro \
     -p topol.top -o solvated.gro
```

The backslash means the entire command should be written on a single line. Solvent coordinates (`-cs`) are taken from an SPC water system [23], and the `-p` flag adds the new water to the topology file. The resulting system is illustrated in Fig. 6.

3.5 Adding Ions

In principle you could use the system as is, but the net charge on the protein is unphysical in an infinite system, and many proteins interact with counterions. There is a GROMACS program to help us with this, but we first need an input file. GROMACS uses a separate preprocessing program `grompp` to collect parameters, topology, and coordinates into a single run input file (`em.tpr`) from which the simulation is then started (this makes it easier to move it to a another computer). Here we are not really going to run anything, so just create an *empty* file called `ions.mdp` and prepare an input file as:

```
grompp –f ions.mdp –p topol.top –c solvated.gro \
–o ions.tpr
```

To neutralize the system and add 100 mM NaCl to the output file `ions.gro`, use the command

```
genion –s ions.tpr –neutral –conc 0.1 \
–p topol.top –o ions.gro
```

3.6 Energy Minimization

The added hydrogens and broken hydrogen bond network in water would lead to quite large forces and structure distortion if molecular dynamics was started immediately. To remove these forces it is necessary to first run a short energy minimization. The aim is not to reach any local energy minimum, so 500 steps of *steepest descent* (as mentioned in the theory section) works very well as a stable rather than maximally efficient minimization. Nonbonded interactions and other settings are specified in a parameter file (`em.mdp`); it is only necessary to specify parameters where we deviate from the default value (this is why we could use an empty file above), for example:

```
------em.mdp------
integrator=steep
nsteps=500
nstlist=10
rlist=1.0
coulombtype=pme
rcoulomb=1.0
vdw-type=cut-off
rvdw=1.0
nstenergy=10
------------------
```

See **Note 11** contains a more detailed description of these settings. Then prepare the input file and run the energy minimization:

```
grompp –f em.mdp –p topol.top –c ions.gro \
–o em.tpr
<lots of output>
mdrun –v –deffnm em
```

The `–deffnm` is a smart shortcut that uses "`em`" as the base filename for all options, but with different extensions. The minimization will complete in a few seconds (*see* **Note 12**).

3.7 Position Restrained Equilibration

To avoid unnecessary distortion of the protein when the molecular dynamics simulation is started, we first perform a 100 ps equilibration run where all heavy protein atoms are restrained to their starting positions (using the file `posre.itp` generated earlier) while the water is relaxing around the structure. As covered in the theory section, bonds will be constrained to enable 2 fs time steps. Other settings are identical to energy minimization, but for molecular dynamics we also control the temperature with the Bussi thermostat [24] (*see* **Note 13**). The settings used are (*see* **Note 14**):

```
------pr.mdp------
integrator=md
nsteps=50000
dt=0.002
nstenergy=1000
nstlist=10
rlist=1.0
coulombtype=pme
rcoulomb=1.0
vdw-type=cut-off
rvdw=1.0
tcoupl=v-rescale
tc-grps=protein water_and_ions
tau-t=0.5 0.5
ref-t=300 300
pcoupl=parrinello-rahman
pcoupltype=isotropic
tau-p=2.0
compressibility=4.5e-5
ref-p=1.0
cutoff-scheme=Verlet
define=-DPOSRES
refcoord_scaling=com
constraints=all-bonds
------------------
```

For a small protein like BPTI it should be more than enough with 100 ps (50,000 steps) for the water to equilibrate around it, but in a large membrane system the slow lipid motions can require several nanoseconds of relaxation. The only way to know for certain is to watch the potential energy, and extend the equilibration until it has converged. Running this equilibration in GROMACS you execute

```
grompp –f pr.mdp –p topol.top –c em.gro \
–o pr.tpr
mdrun –v –deffnm pr
```

This simulation will finish in a few minutes on a GPU-equipped workstation.

3.8 Production Runs

The difference between equilibration and production run is minimal: the position restraints and pressure coupling are turned off (*see* **Note 15**), we decide how often to write output coordinates to analyze (say, every 5,000 steps), and start a significantly longer simulation. How long depends on what you are studying, and that should be decided before starting any simulations. For decent sampling the simulation should be at least ten times longer than the phenomena you are studying, which unfortunately sometimes conflicts with reality and available computer resources. We will perform a 10 ns simulation (5 million steps), which takes about an hour on a GPU workstation. If you are not that patient (or have a slow machine) you can choose a shorter simulation just to get an idea of the concepts, and the analysis programs in the next section can read the simulation output trajectory as it is being produced.

```
------run.mdp------
integrator=md
nsteps=5000000
dt=0.002
nstlist=10
rlist=1.0
coulombtype=pme
rcoulomb=1.0
vdw-type=cut-off
rvdw=1.0
tcoupl=v-rescale
tc-grps=protein water_and_ions
tau-t=0.5 0.5
ref-t=300 300
cutoff-scheme=Verlet
nstxtcout=5000
nstenergy=5000
------------------
```

Prepare and perform the production run as (the extra option to mdrun avoids spending too much time on writing out the current step to frequently):

```
grompp –f run.mdp –p topol.top –c pr.gro –o run.tpr
<output>
mdrun –v –deffnm run –stepout 10000
```

3.9 Trajectory Analysis

3.9.1 Deviation from X-Ray Structure

One of the most important fundamental properties to analyze is whether the protein is stable and close to the experimental structure. The standard way to measure this is the root-mean-square displacement (RMSD) of all heavy atoms with respect to the X-ray structure. GROMACS has a program to do this, as

```
g_rms –s em.tpr –f run.xtc
```

Note that the reference structure here is taken from the input before energy minimization. The program will prompt both for a fit group, and the group to calculate RMSD for—choose "`Protein-H`" (protein except hydrogens) for both. The output will be written to `rmsd.xvg`, and if you installed the Grace program you will directly get a finished graph with

```
xmgrace rmsd.xvg
```

The RMSD is also illustrated in Fig. 7. It increases pretty rapidly in the first part of the simulation, but stabilizes around 0.14 nm,

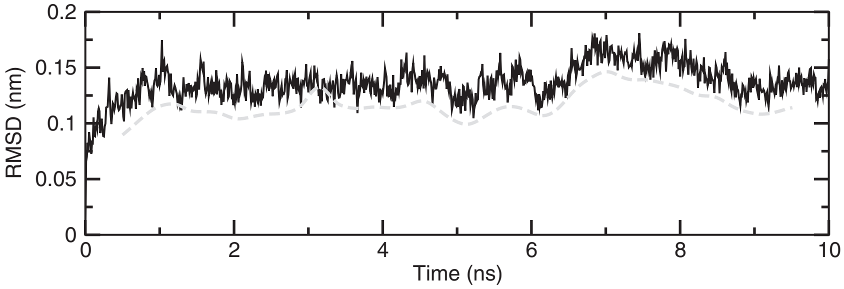


Fig. 7 Instantaneous Root-mean-square displacement (RMSD) of all heavy atoms in Lysozyme during the simulation (*solid*), relative to the crystal structure. To a large extent atoms are vibrating around an equilibrium, so the RMSD of a 1-ns running average structure (*dashed gray*) is a better measure

roughly the resolution of the X-ray structure. The difference is partly caused by limitations in the force field, but also because atoms in the simulation are moving and vibrating around an equilibrium structure. A better measure can be obtained by first creating a running average structure (*see* **Note 16**) from the simulation and comparing the running average to the X-ray structure, which gives a more realistic RMSD around 0.12 nm (*see* **Note 17**).

3.9.2 Comparing Fluctuations with Temperature Factors

Vibrations around the equilibrium are not random, but depend on local structure flexibility. The root-mean-square-fluctuation (RMSF) of each residue is straightforward to calculate over the trajectory, but more important they can be converted to *temperature factors* that are also present for each atom in a PDB file. Once again there is a program that will do the entire job:

```
g_rmsf -s run.tpr -f run.xtc -o rmsf.xvg \
-oq bfac.pdb
```

You can use the group "C-alpha" to get one value per residue. Figure 8 displays both the residue RMSF from the simulation (`xmgrace rmsf.xvg`), as well as the calculated and experimental temperature factors. The overall agreement is reasonable for a protein this small and a short simulation. Longer simulations of larger proteins can fit almost perfectly.

3.9.3 Secondary Structure

Another measure of stability is the protein secondary structure. This can be calculated for each frame with a program such as DSSP [25]. If the DSSP program is installed and the environment variable DSSP points to the binary (*see* **Note 18**), the GROMACS program do_dssp can create time-resolved secondary structure plots. Since the program writes output in a special xpm (X pixmap) format you probably also need the GROMACS program xpm2ps to convert it to postscript:

```
do_dssp -s run.tpr -f run.xtc -dt 50
xpm2ps -f ss.xpm -o ss.eps
```

Use the group "`protein`" for the calculation. Figure 9 shows the resulting output in grayscale, with some unused formatting

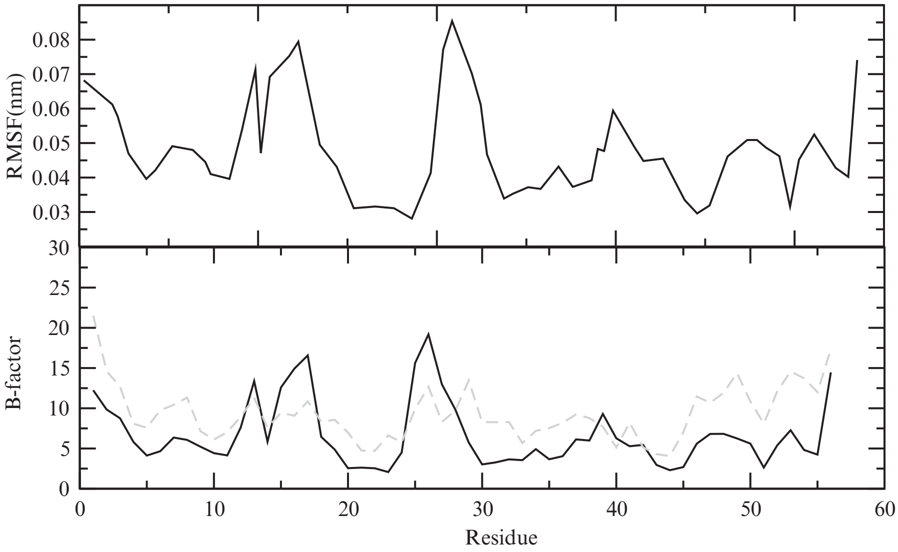


Fig. 8 *Top*: Root-mean-square fluctuations of residue coordinates in the simulation. *Bottom*: The fluctuations can be converted to X-ray temperature factors (*solid*), which agree quite well with the experimental B-factors from the PDB file (*dashed*)

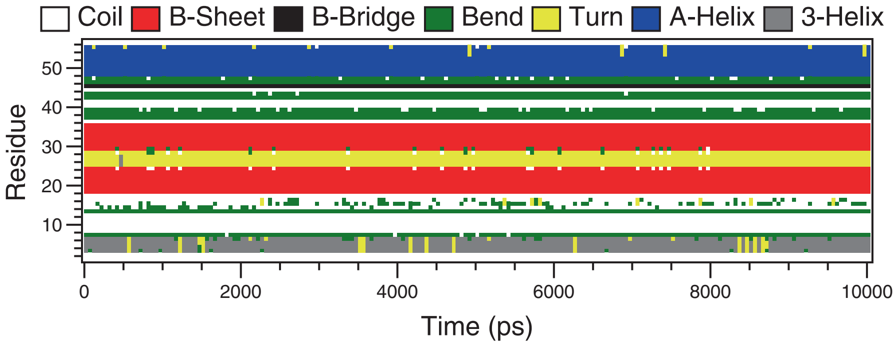


Fig. 9 Local secondary structure in BPTI as a function of time during the simulation, according to the DSSP definition. Note how some elements periodically lose a bit of structure, but it rapidly reforms and the overall structure is quite stable over 10 ns

removed. The DSSP secondary structure definition is pretty tight, so it is quite normal for residues to fluctuate around the well-defined state, in particular at the ends of helices or sheets. For a (long) protein folding simulation, a DSSP plot would show how the secondary structures form during the simulation.

3.9.4 Radius of Gyration and Hydrogen Bonds

There are two more very basic properties that are useful to analyze: The size of the protein defined by the “radius of gyration” and the number of hydrogen bonds. To calculate the radius of gyration, use the command:

```
g_gyrate -s run.tpr -f run.xtc
```

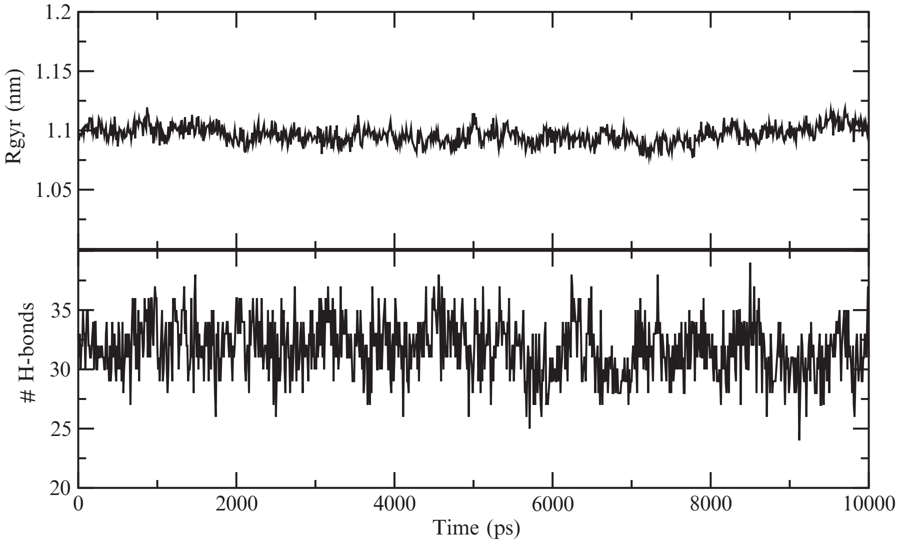


Fig. 10 *Top*: Radius of gyration of BPTI during 10 ns simulation. This is a good measure of how compact a structure is. *Bottom*: Number of hydrogen bonds inside the protein

The result will be written to the file gyrate.xvg, which includes both the overall radius and the radii around the three axes. Similarly, you get the hydrogen bonds with

```
g_hbond -s run.tpr -f run.xtc-num hbnum.xvg
```

Select the group "protein" twice to get all hydrogen bonds between the protein and the protein itself. Figure 10 shows the results for both these analyses (*see* **Note 19**).

3.9.5 Making a Movie

A normal movie uses roughly 30 frames/second, so a 10-s movie requires 300 simulation trajectory frames. To make a smooth movie the frames should not be more than 1–2 ps apart, or it will just appear to shake nervously (*see* **Note 20**). In many cases it makes sense to rerun a shorter trajectory just for the movie, but here we just export a short trajectory from the first 500 ps in PDB format (readable by PyMOL) as

```
trjconv -s run.tpr -f run.xtc \
-e 2500.0 -o movie.pdb
```

Choose the protein group for output rather than the entire system (*see* **Note 21**). If you open this trajectory in PyMOL as "`PyMOL movie.pdb`" you can immediately play it using the VCR-style controls on the bottom right, adjust visual settings in the menus, and even use photorealistic ray-tracing for all images in the movie. With MacPyMOL you can directly save the movie as a quicktime file, and on Linux you can save it as a sequence of PNG images for assembly in another program. Rendering a movie only takes a few minutes, and the final product `bpti.mov` is included with the reference files.

4 Speeding Things up to Solve a Real Problem

Once you have gotten your feet wet you will likely want to approach more realistic problems. One important such problem is folding of small proteins, for instance the Villin headpiece where some mutants fold within a microsecond [30]. This mutant contains a special residue (norleucine), so you will need to copy the entire amber force field directory from the installation directory of Gromacs to your current working directory, place the file `norleucine.rtp` in the amber subdirectory, and also put the file `residuetypes.dat` in your current directory (so GROMACS recognizes the new residue as a protein residue).

The first challenge you are likely to hit is that you need your simulations to run faster to be able to reach relevant time scales. Here we will briefly go through a couple of recommendations that will help you achieve this.

You have probably seen that the problem is the number of steps we must take. One way to improve performance is to make each time step longer, but this is limited by the vibrations in the angles involving hydrogens (try a longer timestep in your files above and see what happens). GROMACS has a feature to remove these vibrations by replacing individual hydrogens with virtual sites. This retains the rotation of for example CH3-groups, but removed the fast vibrations. To enable this, use the `–vsite` option when you run pdb2gmx (or skip it if you want to stick to 2 fs steps):

```
pdb2gmx –f protein.pdb –water tip3p \
–vsite hydrogen
```

This will instantly enable us to take time steps up to 5 fs, which will improve your performance by 150 %. Second, if you look at the box you used for BPTI you will likely see that it would better match the shape of the protein as a more spherical shape. Unfortunately spheres are not periodical, but we can ask GROMACS to use a rhombic dodecahedron box instead, which is at least more spherical than a cube and has only 71 % of the cube volume. That reduces the number of water molecules required to solvate the protein. This is difficult to visualize in three dimensions, but Fig. 11 illustrates in two dimensions how a *hexagonal* cell is more efficient than a square (very useful for membrane simulations). The hexagonal box achieves the same distance between periodic copies as a rectangular box at 86 % of the volume (*see* **Note 22**). Since we really want to push performance, we also accept a very small margin to the box side (-d option):

```
editconf –f conf.gro –bt dodecahedron \
–d 0.3 –o box.gro
```

Finally, although PME does provide state-of-the-art electrostatics, the Villin headpiece is quite well-behaved and there are no large mobile charges in this system. To get a bit of extra performance, this is a case where we can decide to forego PME and use

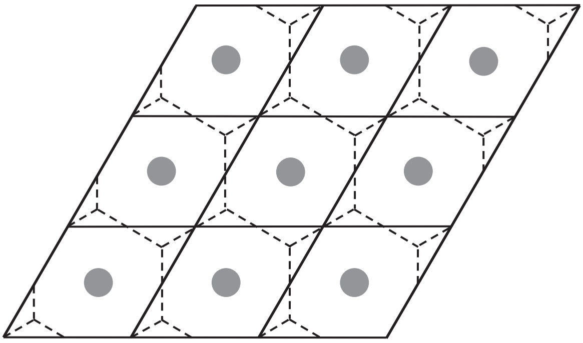


Fig. 11 Two-dimensional example of how a hexagonal box leads to lower volume than a square one, with the same separation distance. In three dimensions, the shape most similar to a sphere is the rhombic dodecahedron

reaction-field electrostatics instead. This difference is easy—just write "reaction-field" instead of "PME" for electrostatics in your mdp files. We also use very short cutoffs (8 Å). The exact files used, including an input structure from the paper [30], are included in a separate directory.

With these settings you should be able to work energy minimization and equilibration of Villin exactly the same way as we did for BPTI, but it will be even faster—on a single Core i7 desktop with a GPU we get almost a microsecond a day. Study what happens with the protein by using the analysis tools you learned above—you should see that it starts to compact and form more hydrogen bonds, but that it takes quite a while for secondary structure to form. To see how far away you are from the native state you can prepare a second TPR file using the native state as reference. However, after a bit of fluctuation, and possible 2–3 days of simulation, you should also be able to reach the native state of Villin.

5 Conclusions

This chapter should hopefully provide a basic introduction to general simulations. An important lesson is that high-quality simulations require a lot of care from the user—just as with experimental techniques the entire result can be ruined by a single sloppy step. Further, recent techniques based on distributed computing and markovian state models have been able to probe dynamics in the millisecond range without extending individual simulations to those scales [31]—this will be covered in much more detail in subsequent chapters presenting metadynamics (Chapter 8) and accelerated MD (Chapter 12). While simulations are advancing rapidly due to the continuous development of faster computers, the field has also been plagued by (published) simulations that have not advanced our knowledge either of simulation methods or biomolecules. Instead of just starting a simulation and hoping for something to happen, you should decide beforehand what you want to study, estimate the timescales necessary or see if it can be accomplished with more advanced methods (e.g., free energy calculations), and not start simulations until you are fairly confident both about sampling, analysis required, and the force field accuracy. Used with caution, molecular dynamics is an amazingly powerful tool, and a great complement to experiments.

6 Notes

1. Most simulations rely on systems being *ergodic*, that is, the time average of the properties of a single molecule on a long simulation should be the same as the instantaneous ensemble average over all molecules in an experimental measurement. This is often (but not always) true, although it assumes our single simulation is sufficiently long, which can be very inefficient to achieve.
2. The standard harmonic bond potentials in molecular simulations will never allow atoms to separate. However, the alternative *Morse* potential is supported in many programs (including GROMACS) and will allow atoms to separate. Still, this is not used very frequently—if your problem involves breaking and forming bonds it is likely a better solution to use a QM/MM simulation.
3. The classical representations can be corrected in a number of ways to make sure that they are faithful representations of the real system. This is discussed in great detail in the first chapter of the GROMACS manual, to which we refer the interested reader. However, the really important thing in modeling is to understand your system and decide in each case what approximations are reasonable. It is easy to add more detail (e.g., by using quantum chemistry), but that automatically means you lose in the other end by not getting as much sampling. The challenge is to strike the right balance for each problem!
4. In general, most computational chemistry programs behave best with the Linux operating system, although it is possible to run GROMACS on Windows. When starting out, you want a standard AMD or Intel desktop. Currently (2013), you will get the best price–performance ratio by investing in a single-socket machine with fastest consumer processor you can buy, for instance Intel Core i7 4770. You can get this for well under $1000. GROMACS and some other codes support GPU acceleration for NVIDIA cards, so to improve performance significantly it is a good idea to add a high-end graphics card such as

GTX780 or GTX TITAN. Beware that this development is even faster than the CPUs, so consult the internet to find up-to-date hardware. Commercial Linux distribution are not required—we typically use the free Ubuntu (http://www.ubuntu.com). If you are hesitant about installing Linux, get an Mac instead.

5. GROMACS is freely available from http://www.gromacs.org. It should be quite easy to install using the step-by-step instructions, and for most common platforms there are finished binary packages (installation might require root access, though). PyMOL is distributed from http://www.PyMOL.org, with binaries for Windows, Linux, and Mac OS X. The MacPyMOL version requires a license after a trial period, but is very much recommended for the better movie export capabilities. Unfortunately, the Grace package is not quite as trivial to install. The distribution site http://plasma-gate.weizmann.ac.il/Grace/ only provides source code, so you might want to perform an internet search for a binary for your platform. Linux RPMs can often be found at http://www.rpmfind.net. Grace uses Motif X11 library, but it compiles fine with the open source clone LessTif, http://www.lesstif.org.

6. For this tutorial pretty much any structures would have been fine too, but some of the pdb-files contain organic molecules that are difficult to model automatically, both in GROMACS and other programs. The key issue is to obtain and validate a topology for your organic molecule before proceeding with the simulation. It is often a good idea to both have a look at the structure in a viewer, and read the text information at the top of the PDB file to see if there are any special issues. For 6PTI, the header mentions that the last residue was not visible at all, and only the nitrogen atom in the second last. If large parts of the protein are inaccurate it might be better to choose a different structure.

7. Sometimes people remove the crystal water to replace it with their own solvent later, but this is usually a bad idea. The reason why they are visible is that these waters are tightly bound to the structure and often form salt bridges, so if they are discarded the structure might distort before new solvent has a chance to equilibrate in these positions. Keep the crystal water!

8. Water is a very special liquid, and actually quite difficult to model accurately. However, biomolecular simulations are usually focusing on the protein/DNA/etc., and thus normally prefer cheap and simple approximate solvent models to the most accurate one. The most common such models are SPC [23] (used with the GROMOS96 force field) and TIP3P [22] (OPLS and Amber force fields), which both represent the water as an entirely rigid molecule with three sites (oxygen & two hydrogens). There are a couple of modified models such

as SPC/E that improve bulk properties, but the standard models are often preferred for interface systems like membranes. TIP4P [26] is a smart model with a fourth interaction site offset from the oxygen, and still reasonably cheap computationally (recommended), while TIP5P [27] with five interaction sites is too expensive for most simulations.

9. `pdb2gmx` can be somewhat picky with the input structures, but that is usually a good thing—it will for instance not accept proteins with missing heavy atoms. If that happens, the best option is to find a better structure, and if that is not possible you can try to build the missing parts with a program like Modeller (http://salilab.org/modeller/). However, if you have to build more than a handful of residues it is doubtful if the resulting structure is accurate enough to simulate. For 6PTI, `pdb2gmx` will also issue a warning about net charge, but that is fine. In general, all GROMACS program try to do both double- and triple-checking of your input, so if you do not get any warning you can be pretty confident about the correctness of your input.

10. All GROMACS programs that write coordinates support a number of different output formats. The default one is `.gro`, simply because it has support for velocities too, but if you want a PDB file to view for example in PyMOL you simply change the output file extension to `.pdb`, when using a gromacs program.

11. We choose a standard cutoff of 1.0 nm, both for the neighborlist generation and the coulomb and Lennard–Jones interactions. `nstlist=10` means it is updated at least every 10 steps, but for energy minimization it will usually be every step. Energies and other statistical data are stored every 10 steps (`nstenergy`), and we have chosen the more expensive Particle-Mesh-Ewald summation for electrostatic interactions. The treatment of nonbonded interactions is frequently bordering to religion. One camp advocates standard cutoffs are fine, another swears by switched-off interactions, while the third would not even consider anything but PME. One argument in this context is that “true” interactions should conserve energy, which is violated by sharp cutoffs since the force is no longer the exact derivative of the potential. On the other hand, just because an interaction conserves energy does not mean it describes nature accurately. In practice, the difference is most pronounced for systems that are very small or with large charges, but the key lesson is really that it is a trade-off. PME is great, but also clearly slower than cutoffs. Longer cutoffs are always better than short ones (but slower), and while switched interactions improve energy conservation they introduce artificially large forces. Using PME is the safe option, but if that is not fast

enough it is worth investigating reaction-field electrostatics, but you should *never* use a plain cutoff for electrostatics. It is also a good idea to check and follow the recommended settings for the force field used.

12. Mdrun will write several output files: `em.edr` is an "energy file" with statistical data (energies, temperature, pressure, etc.). `em.trr` is a trajectory with full coordinates/velocities of the system during the run, and `em.log` a log file. Depending on the parameters (disabled here), it might also write a compressed trajectory with low-precision coordinates only, `em.xtc`.

13. The Bussi thermostat is a great advance for simulations. It is both efficient and avoids excessive fluctuations, and maintains a correct statistical mechanics ensemble. We strongly prefer it over the Nose-Hoover thermostats [28]. For pressure coupling we use the similar Parrinello–Rahman barostat [29]. When your only goal is to get the system to a specific temperature or pressure as quickly as possible without fluctuations, you can also consider the Berendsen weak coupling thermostat/barostat, but these do not provide correct ensembles. For the Bussi thermostat we can use relatively slow coupling times (0.5 ps), and the pressure coupling should be clearly slower than this (2–5 ps).

14. For molecular dynamics simulations the integrator is "`md`." Temperature coupling has been enabled for protein and water separately (to avoid heating the water more than the protein or vice versa), with a 300 K reference temperature. The compressibility is really a symmetric tensor, and by setting the last three elements (off-diagonal) to 0 we disable any box shear deformation. The last line causes `grompp` to include the position restraint file `posre.itp` generated by `pdb2gmx`, which turns on position restraints. Since we are scaling the box with pressure coupling, we also need to adjust the center-of-mass of the reference coordinates for the position restraints with the refcoord_scaling option. Finally, the Verlet cutoff-scheme is a more accurate setting that also enables us to use GPU accelerators in GROMACS.

15. The easiest way to create a running average in GROMACS is to use the `g_filter` program. The command "`g_filter -nf 50 -all -s run.tpr -f run.xtc -ol lowpass.xtc`" will create a lowpass version of the trajectory (cosine averaging over 50 frames), which then can be used as modified input file to the `g_rms` program.

16. In this particular case we just used pressure coupling to get the right density, while the production simulation is performed in a so-called NVT ensemble (constant number of particles, volume, and temperature). For some systems, in particular

membranes and membrane proteins, it is common to enable pressure coupling during the entire simulation in a so-called NPT ensemble.

17. If the RMSD is significantly higher than this, or continuously increasing, there is likely something very wrong. Start over with the PDB file, read the headers carefully and make sure the starting structure is accurate. In the next step, check the different energy terms and RMSD change both during minimization and position restraints. You can also use the `–posrefc` flag with `pdb2gmx` to increase the strength of the position restraints, and extend the equilibration run.
18. The DSSP program can be obtained from http://swift.cmbi.ru.nl/gv/dssp/. The latest version is now freely available for everybody, but it also has a new output format. This new output format is supported by Gromacs version 4.6 and later. Download a precompiled binary if you find a suitable one, or compile the program and install it, e.g., in/usr/local/bin. Set the environment variable with a command like "`export DSSP=/usr/local/bin/dssp`" (bash shell).
19. Modern force fields no longer use special hydrogen bond interactions, partly because it is not necessary and partly because it is difficult to track formation/breaking of hydrogen bonds separately. "Hydrogen bonds" are therefore defined from geometric criteria, typically that the distance between the donor and acceptor atoms should be smaller than 0.35 nm, and the angle donor–acceptor–hydrogen should be below 30 degrees.
20. To visualize slower phenomena such as protein folding, you can use `g_filter` to smooth out motions in longer trajectories. In some cases this can lead to strange artifacts, e.g., when averaging torsion rotation around a bond, but it is usually better than just taking raw trajectory frames with too large spacing.
21. PyMOL loads all frames of the trajectory into memory, so if the water molecules are included it will likely run out of memory when creating graphical representations for over 20,000 atoms repeated in 250 frames. Trajectories restricted to the protein part can thus be much longer.
22. The volume of a rhombic dodecahedron is roughly 71 % of a cube with the same spacing, for a truncated octahedron it is 77 %, and a hexagonal box is 86 % of a rectangular one. These difference can appear small, but 30 % is quite significant when simulations use weeks of supercomputer time, and it is a free lunch after all! However, not all programs support all box shapes.

References

1. Alder BJ, Wainwright TE (1957) Phase transition for a hard sphere system. J Chem Phys 27:1208–1209
2. Rahman A, Stillinger FH (1971) Molecular dynamics study of liquid water. J Chem Phys 55:3336–3359
3. McCammon JA, Gelin BR, Karplus M (1977) Dynamics of folded proteins. Nature 267: 585–590
4. Allen MP, Tildesley DJ (1989) Computer simulation of liquids. Clarendon, New York, NY
5. Frenkel D, Smit B (2001) Understanding molecular simulation. Academic, New York, NY
6. Kaminski GA, Friesner RA, Tirado-Rives J, Jorgensen WL (2001) Evaluation and reparametrization of the OPLS-AA force field for proteins via comparison with accurate quantum chemical calculations on peptides. J Phys Chem B 105:6474–6487
7. MacKerell AD Jr et al (1998) All-atom empirical potential for molecular modeling and dynamics Studies of proteins. J Phys Chem B 102:3586–3616
8. Oostenbrink C, Villa A, Mark AE, van Gunsteren WF (2004) A biomolecular force field based on the free enthalpy of hydration and solvation: the GROMOS force-field parameter sets 53A5 and 53A6. J Comput Chem 25:1656–1676
9. Wang J, Cieplak P, Kollman PA (2000) How well does a restrained electrostatic potential (RESP) model perform in calculating conformational energies of organic and biological molecules? J Comput Chem 21:1049–1074
10. Chandler D (1987) Introduction to modern statistical mechanics. Oxford University Press, New York, NY
11. Essman U, Perera L, Berkowitz M, Darden T, Lee H, Pedersen LG (1995) A smooth particle mesh Ewald method. J Chem Phys 103: 8577–8593
12. Ryckaert JP, Ciccotti G, Berendsen HJC (1977) Numerical integration of the cartesian equations of motion of a system with constraints; molecular dynamics of n-alkanes. J Comp Phys 23:327–341
13. Hess B, Bekker H, Berendsen HJC, Fraaije JGEM (1998) LINCS: a linear constraint solver for molecular simulation. J Comput Chem 18:1463–1472
14. Lindahl E, Hess B, van der Spoel D (2001) GROMACS 3.0: A package for molecular simulation and trajectory analysis. J Mol Model 7:306–317
15. Case DA et al (2005) The Amber biomolecular simulation programs. J Comput Chem 26: 1668–1688
16. Brooks BR et al (1983) CHARMM: a program for macromolecular energy, minmimization, and dynamics calculations. J Comput Chem 4:187–217
17. Phillips JC et al (2005) Scalable molecular dynamics with NAMD. J Comput Chem 26:1781–1802
18. DeLano WL (2002) The PyMOL molecular graphics system. DeLano Scientific, San Carlos, CA, http://www.PyMOL.org
19. Ascenzi P et al (2003) The bovine basic pancreatic trypsin inhibitor (kunitz inhibitor): a milestone protein. Curr Protein Pept Sci 4:231–251
20. Wlodawer A et al (1987) Structure of form III crystals of bovine pancreatic trypsin inhibitor. J Mol Biol 198:469–480
21. Frauenfelder H, Petsko GA, Tsernoglou D (1979) Temperature-dependent X-ray diffraction as a probe of protein structural dynamics. Nature 280:558–563
22. Jorgensen WL, Chandrasekhar J, Madura JD, Impey RW, Klein ML (1983) Comparison of simple potential functions for simulating liquid water. J Chem Phys 79:926–935
23. Berendsen HJC, Postma JPM, van Gunsteren WF (1981) Interaction models for water in relation to protein hydration. In: Pullman B (ed) Intermolecular forces. D. Reidel Publishing Company, Dordrecht, Germany, pp 331–342
24. Bussi G, Donadio D, Parrinello M (2007) Canonical sampling through velocity-rescaling. J Chem Phys 126:014101
25. Kabsch W, Sanders C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22:2577–2637
26. Jorgensen WL, Madura JD (1985) Temperature and size dependence for monte carlo simulations of TIP4P water. Mol Phys 56:1381–1392
27. Mahoney MW, Jorgensen WL (2000) A five-site model for liquid water and the reproduction of the density anomaly by rigid, nonpolarizable potential fuynctions. J Chem Phys 112:8910–8922
28. Nosé S (1984) A molecular dynamics method for simulations in the canonical ensemble. Mol Phys 52:255–268
29. Parrinello M, Rahman A (1981) Polymorphic transitions in single crystals: a new molecular dynamics method. J Appl Phys 52:7182–7190

30. Ensign DL, Kasson P, Pande V (2007) Heterogeneity even at the speed limit of folding: large-scale molecular dynamics study of a fast-folding variant of the villin headpiece. J Mol Biol 374:806–816

31. Voelz VA, Bowman GR, Beauchamp K, Pande VS (2010) Molecular simulation of ab initio protein folding for a millisecond folder NTL9 (1-39). J Am Chem Soc 132: 1526–1528

Chapter 2

Transition Path Sampling with Quantum/Classical Mechanics for Reaction Rates

Frauke Gräter and Wenjin Li

Abstract

Predicting rates of biochemical reactions through molecular simulations poses a particular challenge for two reasons. First, the process involves bond formation and/or cleavage and thus requires a quantum mechanical (QM) treatment of the reaction center, which can be combined with a more efficient molecular mechanical (MM) description for the remainder of the system, resulting in a QM/MM approach. Second, reaction time scales are typically many orders of magnitude larger than the (sub-)nanosecond scale accessible by QM/MM simulations. Transition path sampling (TPS) allows to efficiently sample the space of dynamic trajectories from the reactant to the product state without an additional biasing potential. We outline here the application of TPS and QM/MM to calculate rates for biochemical reactions, by means of a simple toy system. In a step-by-step protocol, we specifically refer to our implementation within the MD suite Gromacs, which we have made available to the research community, and include practical advice on the choice of parameters.

Key words Protein folding, Biochemical reactions, QM/MM, Reactive paths, Rate calculations

1 Introduction

Processes such as biochemical reactions or conformational changes of biomolecules typically occur on timescales beyond those accessible by Molecular Dynamics (MD) simulations at atomistic detail. In many cases, reducing the resolution of the simulation by coarse-graining the biomolecule is not an option, as critical players such as hydrogen bonds or hydrophobic effects involved in the reaction under investigation might be lost or are described at insufficient accuracy.

Purely classical MD simulations at atomistic resolution routinely can reach microsecond time scales. In a few recent cases, millisecond scales were achieved, which allowed the prediction of quantitative rates for the folding of proteins, either by highly parallel distributed computing of many short trajectories or by special purpose high-performance computing to obtain a small number of ultralong trajectories [1–3]. However, the conventionally reached

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_2, © Springer Science+Business Media New York 2015

microsecond time scale is mostly insufficient to sample the process of interest such as a conformational change or (un)folding frequently enough to compute transition rates.

The problem of too short simulation time scales is even larger for the case for chemical reactions, in which covalent bonds are broken or formed. Here, a classical molecular mechanical (MM) description is not sufficient, as it relies on a harmonic potential for a covalent bond, which does not allow dissociation of the bonded atoms. Instead, a quantum mechanical (QM) description is required to treat the change in bonds within the biomolecule accurately. Taking the electronic degrees of freedom into account, however, entails substantially higher computational costs and restricts time scales typically to picoseconds or nanoseconds, very much depending on the theory and basis set of choice. In turn, chemical reactions typically feature high barriers, i.e., rates at the microsecond to millisecond scale.

From a computational point of view, the most interesting quantity for such processes often is the reaction rate. The reason is that, in contrast to a free energy barrier, rates are experimentally directly accessible, and thus a straightforward comparison is possible. Also, the rate is the quantity which is physiologically most relevant, as kinetics determine most of the biological processes. Rates can be obtained from free energy barriers using the Arrhenius or Eyring equations, which requires, however, the assumption of an attempt frequency, the value of which is debated and varies with the nature of the process and the solvent [4, 5]. An elaborate method to directly compute reaction rates is Transition Path Sampling (TPS) [6, 7]. TPS is an algorithm which efficiently searches the space of transition paths between two states. From the ensemble of sampled paths obtained from MD simulations combined with a Monte Carlo sampling scheme, reaction rates can be obtained, without the detour of free energy barriers. TPS can be straightforwardly used for chemical reactions treated with QM or combined QM/MM. It has been proven to be a useful method to obtain quantitative insight into the mechanism of, among others, the reactions catalyzed by lactate dehydrogenase [8] and human purine nucleotide phosphorylase [9]. We have employed QM/MM and TPS to obtain force-dependent rates for a redox reaction, namely, the reduction of a disulfide bond by a small reducing agent, dithiothreitol [10], and for peptide hydrolysis [11].

In this chapter, we outline the basics of TPS, and in particular the calculation of reaction rates based on TPS. We illustrate the methodological details by way of a toy model, namely, three argon atoms in a box of water. While our toy model is, for simplicity, described solely by MM, the same strategy can be employed to a chemical or enzymatic reaction treated by combined QM/MM. For the reader interested in the details of a QM/MM setup for Molecular Dynamics simulations and eventually for TPS, we refer to recent reviews on this subject [12–14], and in combination with TPS to our own work [10]. TPS does not restrict the QM–MM interface in any way. However, as the TPS and rate calculation scheme presented here is based on our implementation into Gromacs [15], only QM/MM features available within Gromacs can be employed along with our implementation. The recent review by Groenhof [14] is the most comprehensive introduction into the current QM/MM implementation within Gromacs, and reviews the available schemes to treat the interactions between the regions described by QM and MM, and to cap the QM region in case of covalent bonds at the QM–MM interface.

2 Theory

Many processes such as chemical reactions or protein folding can be simplified to processes with two stable states that are separated by a single high energy barrier. In Fig. 1a, regions A and B are the two stable states, and the energy barrier is highlighted in between. For chemical reactions, regions A and B represent the reactant and product states, respectively. In this example, the multidimensional space of the system is projected onto two order parameters, R1 and R2, both of which change during the reaction. Examples for order parameters, often distances, angles, or collective coordinates, are given further below. A reactive trajectory (shown as a black solid line) leads to the rare but crucial transition between A and B. The system spends considerably longer times in the two free energy wells of the reactant and product than in the high free energy states between the two. Thus, while the transition of interest might only take a few 100 fs, the dwell time of the system in A or B might be in the microsecond to second time scale. Transition path sampling (TPS) has been developed to enhance the sampling of the rare reactive trajectories, which are otherwise hardly harvested by conventional simulations [6, 7, 16–18].

2.1 Sampling the Transition Path Ensemble

The idea of transition path sampling (TPS) is to sample a new transition path based on an existing (old) one (a transition path refers to a reactive trajectory) with a Monte Carlo procedure, and the new path is made sure to be equally weighted with the old one in the transition path ensemble. In principle, there are many strategies to do this. For illustrating the concept of TPS, we here use the shooting move in a deterministic simulation as an example.

(a) *Defining the probability of a reactive path.* In molecular simulations, the time evolution of a system is represented by an ordered sequence of states, $X(T) \equiv \{X_0, X_{\Delta t}, X_{2\Delta t}, \ldots, X_T\}$ (*see* Fig. 1a, black solid line). Here, Δt is the time increment. $X(T)$ consists of $L = T/\Delta t + 1$ states, and its starting point is X_0.

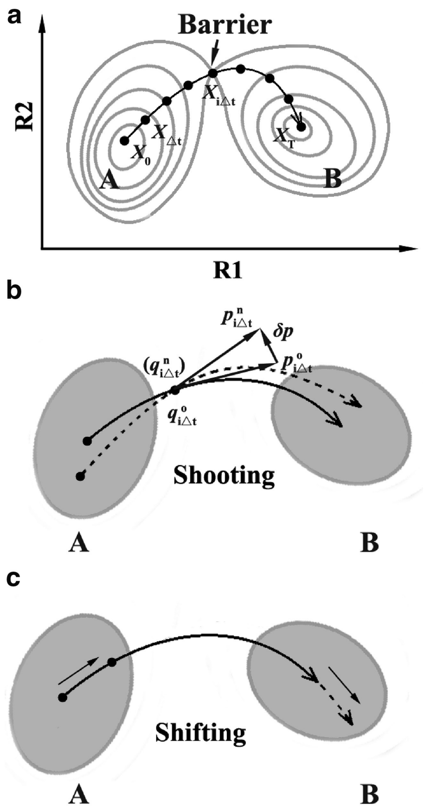


Fig. 1 Schematic description of the free energy landscape of a system and the shooting and shifting moves in TPS. (**a**) A typical free energy landscape of a process is shown with two stable states (labelled with A and B) and a barrier in the *middle*. R1 and R2 are two arbitrary coordinates. A transition pathway (*black solid line*) connecting states A and B is given as well. The transition path is represented by an ordered sequence of states $X(T) \equiv \{X_0, X_{\Delta t}, X_{2\Delta t}, \ldots, X_T\}$. (**b**) An example of shooting moves. The two *filled grey areas* represent the states A and B mentioned above. A state $\{q^o_{i\Delta t}, p^o_{i\Delta t}\}$ is randomly chosen from an old transition path (*solid line*). The momentum $p^o_{i\Delta t}$ is perturbed to be $p_{i\Delta t}{}^n$, where $p_{i\Delta t}{}^n = p_{i\Delta t}{}^o + \delta p$, while the coordinate is unchanged with $q^o_{i\Delta t} = q_{i\Delta t}{}^n$. From the newly generated state $\{q_{i\Delta t}{}^n, p_{i\Delta t}{}^n\}$, a new transition path (*dashed line*) is obtained by evolving the system backward in time to zero and forward in time to T. (**c**) An example of forward shifting moves. A new path is generated by removing a small segment from the beginning of the old path (the starting frame, shown as a *black dot*, moves forward to a new start) and evolving the system forward from the last frame to create a new part with the same length as the removed one (the *dashed line*). Figure adopted from Hierarchical Methods for Dynamics in Complex Molecular Systems, Lecture Notes, Eds. Grotendorst et al, Juelich, 2012” with permission

For deterministic dynamics, the probability of a trajectory equals to the probability of the initial state in a given ensemble, $\rho(X_0)$. Therefore, the probability of trajectory $X(T)$ to be a reactive trajectory is given below:

$$P_{AB}\left(X(T)\right) = h_A\left(X_0\right)\rho\left(X_0\right)h_B\left(X_T\right) / Z_{AB}\left(T\right) \qquad (1)$$

Here, $h_A(X)(h_B(X))$ is the characteristic function of region A(B). $h_A(X)$ equals 1 if state X lies in A, and it equals zero otherwise. $Z_{AB}(T)$ is the normalizing factor, the sum of all the possible reactive trajectories with length T in a given ensemble.

$$Z_{AB}(T) \equiv \int dX_0 h_A(X_0)\rho(X_0)h_B(X_T) \tag{2}$$

(b) *Sampling the transition path ensemble by shooting.* In a transition path ensemble, the distribution of transition paths is given in Eq. 1. To make sure that the correctly weighted transition paths are sampled, the following two probabilities should equal: the probability to generate a new transition path from a old one $P_{gen}(X^o(T) \to X^n(T))$, and the probability to generate the old transition path from the new one $P_{gen}(X^n(T) \to X^o(T))$. In a shooting move, a state $X_{i\Delta t}{}^o$, $i \in [0, L]$, is randomly chosen. Then, a new state $X_{i\Delta t}{}^n$ is generated by adding a small perturbation to $X_{i\Delta t}{}^o$. Here, the superscript o and n refer to the old path and the new path, respectively. Note that a state X consists of the coordinate q and the momentum p, $X = \{q, p\}$, the perturbation can be added to q or/and p. In practice, it is convenient to keep q untouched and change p by δp. As illustrated in Fig. 1b, the selected state $X_{i\Delta t}{}^o = \{q_{i\Delta t}{}^o, p_{i\Delta t}{}^o\}$ in an old transition path (the solid line in Fig. 1b) is changed to $X_{i\Delta t}{}^n = \{q_{i\Delta t}{}^n, p_{i\Delta t}{}^n\}$, where $p_{i\Delta t}{}^n = p_{i\Delta t}{}^o + \delta p$. Starting with $X_{i\Delta t}{}^n$, one can evolve the system backward in time to 0 and forward in time to T, then a new transition path is generated if it initials from region A and ends in region B (the dashed line in Fig. 1b). The probability to generate a new transition path from an old one is the product of four parts, the probability of the old path in the given ensemble, the probability to generate $X_{i\Delta t}{}^n$ from $X_{i\Delta t}{}^o$ ($P_{gen}(X_{i\Delta t}{}^o \to X_{i\Delta t}{}^n)$), the probability of that the new path is reactive, and the probability to accept the new transition path $P_{acc}(X^n(T) \to X^o(T))$.

$$\begin{aligned} P_{gen}\left(X^o(T) \to X^n(T)\right) &= P_{AB}\left(X^o(T)\right) P_{gen}\left(X^o_{i\Delta t} \to X^n_{i\Delta t}\right) h_A\left(X^n_0\right) h_B\left(X^n_T\right) \\ &\quad \times P_{acc}\left(X^o(T) \to X^n(T)\right) \end{aligned} \tag{3}$$

Similarly, for generating the old path from the new one, we have

$$\begin{aligned} P_{gen}\left(X^n(T) \to X^o(T)\right) &= P_{AB}\left(X^n(T)\right) P_{gen}\left(X^n_{i\Delta t} \to X^o_{i\Delta t}\right) h_A\left(X^o_0\right) h_B\left(X^o_T\right) \\ &\quad \times P_{acc}\left(X^n(T) \to X^o(T)\right) \end{aligned} \tag{4}$$

The *detailed balance* of moves in trajectory space requires P_{gen} $(X^o(T) \to X^n(T)) = P_{gen}(X^n(T) \to X^o(T))$, which gives

$$\frac{P_{\text{acc}}\left(X^{\text{o}}(T)\rightarrow X^{\text{n}}(T)\right)}{P_{\text{acc}}\left(X^{\text{n}}(T)\rightarrow X^{\text{o}}(T)\right)}=\frac{P_{\text{AB}}\left(X^{\text{n}}(T)\right)P_{\text{gen}}\left(X_{i\Delta t}^{\text{n}}\rightarrow X_{i\Delta t}^{\text{o}}\right)h_{\text{A}}\left(X_{0}^{\text{o}}\right)h_{\text{B}}\left(X_{T}^{\text{o}}\right)}{P_{\text{AB}}\left(X^{\text{o}}(T)\right)P_{\text{gen}}\left(X_{i\Delta t}^{\text{o}}\rightarrow X_{i\Delta t}^{\text{n}}\right)h_{\text{A}}\left(X_{0}^{\text{n}}\right)h_{\text{B}}\left(X_{T}^{\text{n}}\right)} \tag{5}$$

This condition can be satisfied using a Metropolis criterion [19]

$$P_{\text{acc}}\left(X^{\text{o}}(T)\rightarrow X^{\text{n}}(T)\right)=\min\left[1,\frac{P_{\text{AB}}\left(X^{\text{n}}(T)\right)P_{\text{gen}}\left(X_{i\Delta t}^{\text{n}}\rightarrow X_{i\Delta t}^{\text{o}}\right)h_{\text{A}}\left(X_{0}^{\text{o}}\right)h_{\text{B}}\left(X_{T}^{\text{o}}\right)}{P_{\text{AB}}\left(X^{\text{o}}(T)\right)P_{\text{gen}}\left(X_{i\Delta t}^{\text{o}}\rightarrow X_{i\Delta t}^{\text{n}}\right)h_{\text{A}}\left(X_{0}^{\text{n}}\right)h_{\text{B}}\left(X_{T}^{\text{n}}\right)}\right] \tag{6}$$

Note that the old path is reactive, i.e., $h_{\text{A}}(X_0{}^{\text{o}})=1$ and $h_{\text{B}}(X_T{}^{\text{o}})=1$. Equation 6 can be simplified as

$$P_{\text{acc}}\left(X^{\text{o}}(T)\rightarrow X^{\text{n}}(T)\right)=h_{\text{A}}\left(X_{0}^{\text{n}}\right)h_{\text{B}}\left(X_{T}^{\text{n}}\right)\times\min\left[1,\frac{\rho\left(X_{i\Delta t}^{\text{n}}\right)P_{\text{gen}}\left(X_{i\Delta t}^{\text{n}}\rightarrow X_{i\Delta t}^{\text{o}}\right)}{\rho\left(X_{i\Delta t}^{\text{o}}\right)P_{\text{gen}}\left(X_{i\Delta t}^{\text{o}}\rightarrow X_{i\Delta t}^{\text{n}}\right)}\right] \tag{7}$$

Here, we apply Eq. 1 and the fact that the probabilities of the states on the same path in deterministic dynamics are the same. Although Eq. 7 is obtained based on deterministic dynamics, it can be also inferred based on a general dynamics [18]. In the implementation of shooting moves, a symmetric generation probability is normally ensured, and thus $P_{\text{gen}}(X_{i\Delta t}{}^{\text{o}}\rightarrow X_{i\Delta t}{}^{\text{n}})=P_{\text{gen}}(X_{i\Delta t}{}^{\text{n}}\rightarrow X_{i\Delta t}{}^{\text{o}})$. Specific strategies are always applied to ensure that states $X_{i\Delta t}{}^{\text{o}}$ and $X_{i\Delta t}{}^{\text{n}}$ are within the same microcanonical ensemble, i.e., $\rho(X_{i\Delta t}{}^{\text{o}})=\rho(X_{i\Delta t}{}^{\text{n}})$. Thus, the acceptance probability becomes

$$P_{\text{acc}}\left(X^{\text{o}}(T)\rightarrow X^{\text{n}}(T)\right)=h_{\text{A}}\left(X_{0}^{\text{n}}\right)h_{\text{B}}\left(X_{T}^{\text{n}}\right) \tag{8}$$

This equation states that any new trajectory will be accepted if it initiates from region A and ends in region B.

A new path can be generated by evolving forward from the last frame (forward shifting move, *see* Fig. 1c) or backward from the starting frame (backward shifting move) of the old path to grow a new path segment with a certain length and then deleting a path segment with the same length from the other end to maintain a fixed total length. Such shifting moves can be combined with shooting moves to improve the sampling efficiency.

2.2 Computing Rate Constants

In this section, we explain how to obtain rate constants from the transition path ensemble [17]. Given a system with two stable states A and B, which are separated by a single high energy barrier, molecules transit from one state to the other at equilibrium, while the populations of states remain unchanged. Since such transitions are rare, the time correlation function, $C(t)$, relates to the reaction time of the system ($\tau_{\text{rxn}}\equiv(k_{\text{AB}}+k_{\text{BA}})^{-1}$) via the following formula [20]

$$C(t)\approx\langle h_{\text{B}}\rangle\left(1-\exp\{-t/\tau_{\text{rxn}}\}\right) \tag{9}$$

If the time required for a system to cross the energy barrier and commit to the other stable state (τ_{mol}) is far smaller than the reaction time of the system (i.e., $\tau_{\mathrm{mol}} << \tau_{\mathrm{rxn}}$), $C(t)$ scales linearly in the intermediate time region, and we have

$$C(t) \approx k_{\mathrm{AB}} t, \quad \tau_{\mathrm{mol}} < t << \tau_{\mathrm{rxn}} \tag{10}$$

For a system at equilibrium, $C(t)$ characterizes the conditional probability to find the system in state B at time t, if it was in state A at time zero, and is defined as follows

$$C(t) \equiv \frac{\langle h_{\mathrm{A}}(X_0) h_{\mathrm{B}}(X_t) \rangle}{\langle h_{\mathrm{A}}(X_0) \rangle} \tag{11}$$

Here, $\langle \ldots \rangle$ is the ensemble average of all initial states. In deterministic dynamics, $C(t)$ can be written in terms of the probability of all initial states $\rho(X_0)$:

$$C(t) = \frac{\int \mathrm{d}X_0 \rho(X_0) h_{\mathrm{A}}(X_0) h_{\mathrm{B}}(X_t)}{\int \mathrm{d}X_0 \rho(X_0) h_{\mathrm{A}}(X_0)} \tag{12}$$

Equations 10 and 12 together provide a way to calculate the forward reaction rate constant k_{AB} by molecular simulations. One can simply run a large set of simulations that start in region A and are of the same time length t, and then count the probability of the end state to be in region B, which gives the value of $C(t)$. The derivative of $C(t)$ over time gives the rate constant. However, this apparently involves numerous computational efforts.

If region B can be defined by an order parameter $\lambda(X)$, and the distribution of the end states, i.e., $X(t)$, along the order parameter $P(\lambda, t)$ is known, $C(t)$ is simply the integral of $P(\lambda, t)$ along λ over region B.

$$C(t) = \int_{\lambda\_\mathrm{min}}^{\lambda\_\mathrm{max}} \mathrm{d}\lambda P(\lambda, t). \tag{13}$$

Here, λ_min and λ_max are the lower and upper bound of region B along λ. $P(\lambda, t)$ is given by

$$P(\lambda, t) = \frac{\int \mathrm{d}X_0 \rho(X_0) h_{\mathrm{A}}(X_0) \delta\left[\lambda - \lambda(X(t))\right]}{\int \mathrm{d}X_0 \rho(X_0) h_{\mathrm{A}}(X_0)}, \tag{14}$$

where $\delta(X)$ is Dirac's delta function. $P(\lambda, t)$ can be divided into several overlapped windows, and its distribution in each window can be estimated separately. The distribution of $P(\lambda, t)$ over the whole range of λ is then obtained by connecting all windows. In each window, transition path sampling can be applied to enhance the sampling of paths that connect region A and the window region. Therefore, computational efforts to compute $C(t)$ are dramatically reduced.

The above-mentioned method can only compute $C(t)$ in time t at a time, and the evaluation of k_{AB} requires $C(t)$ at different times to be evaluated. Therefore, it is laborious. Fortunately, $C(t)$ can be factorized to be written as [18]

$$C(t)=\frac{\langle h_{\mathrm{B}}(t)\rangle_{\mathrm{AB}}}{\langle h_{\mathrm{B}}(t')\rangle_{\mathrm{AB}}}C(t'),\quad 0<t<T \tag{15}$$

where $\langle\ldots\rangle_{\mathrm{AB}}$ denotes an average on the ensemble of the reactive paths, which start in region A and visit region B within the time length of T. T is the time length of the transition path. $\langle h_{\mathrm{B}}(t)\rangle_{\mathrm{AB}}$ is then the proportion of reactive paths whose configuration at time t belongs to region B, and can be estimated by a single transition path sampling run. Only $C(t')$, the $C(t)$ at time $t'(t'<T)$, is needed to be evaluated.

Combining Eqs. 10 and 15, the rate constant is given by

$$k_{\mathrm{AB}}=\frac{\mathrm{d}\langle h_{\mathrm{B}}(t)\rangle_{\mathrm{AB}}/\mathrm{d}t}{\langle h_{\mathrm{B}}(t')\rangle_{\mathrm{AB}}}\times C(t'),\quad \tau_{\mathrm{mol}}<t<<\tau_{\mathrm{rxn}} \tag{16}$$

$\mathrm{d}\langle h_{\mathrm{B}}(t)\rangle_{\mathrm{AB}}/\mathrm{d}t$ should show a plateau in the intermediate time range.

3 Materials

A GROMACS-4.0.7 package [15] with a TPS implementation can be downloaded from http://wenjin.people.uic.edu/download/Gromacs4_tps_patch.tar.gz, which is implemented by Dr. Wenjin Li and currently maintained by him as well (*see* **Note 1**). The package can be installed by following the installation instructions of the original GROMACS-4.0.7 version at http://www.gromacs.org. A Linux or Unix system is required for compilation, as well as FFTW libraries.

4 Methods

In this section, we will describe how to (1) establish a toy system, (2) define the stable basins, (3) obtain the $\langle h_{\mathrm{B}}(t)\rangle_{\mathrm{AB}}$ curve, (4) obtain the $P(\lambda,\ t)$ distribution, (5) calculate rate constants, and (6) monitor TPS. All the files necessary to complete this tutorial are available at http://wenjin.people.uic.edu/download/example_3_Ar.tar.gz.

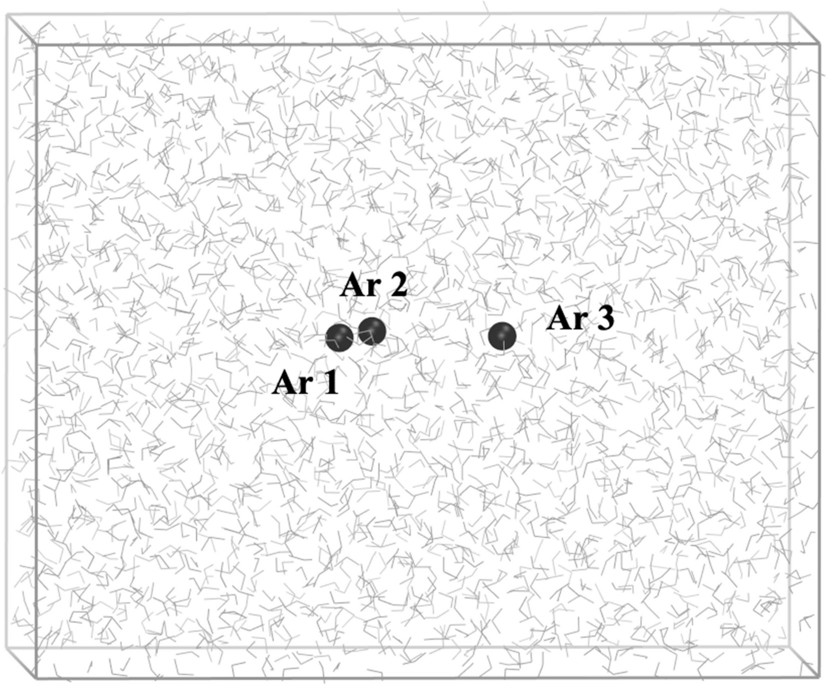


Fig. 2 Simulation setup of the toy system. *Black spheres*: Ar atoms. *Grey lines*: water molecules

To complete this tutorial, we assume the reader to have a basic knowledge of GROMACS and experience in the use of a Linux or Unix operation system.

4.1 A Toy System

We here will illustrate how to use TPS to calculate the rate constant of a rare event with a toy system, which consists of three Ar atoms in a water box (*see* Fig. 2). All three Ar atoms are lying in a line along the *Z*-axis. Atoms 1 and 3 are held by position restraints along the *X*-, *Y*-, and *Z*-axis, while atom 2 is restrained along the *X*- and *Y*-axis, but free to move along the *Z*-axis. Position restrains were switched on by setting *define=-DPOSRES* in the .mdp file, with parameters for position restraints given in *posre.itp*. Atoms 1 and 3 are separated by approximately 1.0 nm. Due to the van der Waals interaction with the other two Ar atoms, atom 2 has two preferred positions (or stable basins). One position is about 0.2 nm, the other 0.8 nm away from atom 1. There is a relatively high barrier between the two minima. Atom 2 can overcome the attraction of one Ar atom and transit from one stable basin to the other. Here, we will estimate the rate of these transitions with TPS. The parameters for van der Waals interaction between two Ar atoms have been modified to unrealistic values (see file *ffoplsaanb.itp*) to increase the barrier between the two minima to make sure that the transition is a rare event (*see* **Note 2**). Therefore, we here are looking at an unphysical toy model to solely focus on the procedure to run TPS with the modified GROMACS package.

4.1.1 Definition of Stable Basins

TPS requires reasonable definitions of the stable basins in terms of one or multiple order parameters. The chosen order parameter should be able to distinguish the two stable basins, but must not necessarily be a good reaction coordinate. Here, the order parameter we choose to distinguish the two minima is $\Delta d = d_{12} - d_{23}$, where d_{12} is the distance between atoms 1 and 2, and d_{23} is the distance between atoms 2 and 3. Then, region A is defined as $-1 < \Delta d < -0.5$ and region B as $0.5 < \Delta d < 1$. The modified Gromacs version includes a section to define these TPS parameters. To define the stable states mentioned above, the parameters are (*see* **Note 3**),

```
========================
tps_npost              = 4
tps_grps1              = a_1 a_2
tps_grps2              = a_2 a_3
tps_dimension          = one
tps_weight_dim          = 1    -1
tps_initial_max         = -0.5
tps_initial_min          = -1
tps_final_max            = 1
tps_final_min            = 0.5
========================
```

Here, *tps_grps1* and *tps_grps2* define the groups to build the order parameters. Currently, the order parameters consist of only distances between two atoms (or two groups of atoms). *tps_grps1* specifies the first group, while *tps_grps2* specifies the second group. The coordinate is the distance between the *i*th group in *tps_grps1* and *tps_grps2*. For example, d_{12} is calculated by the distance between the first group in *tps_grps1* and *tps_grps2*. *a_1*, *a_2*, and *a_3* are the name of atom 1, atom 2, and atom 3 in the index file. *tps_dimension=one* means the two coordinates specified by *tps_grps1* and *tps_grps2* are combined into one parameter using the weights in *tps_weight_dim*. *tps_weight_dim=1 -1* means the order parameter $\Delta d = 1 \times d_{12} + (-1) \times d_{23}$ or $d_{12} - d_{23}$ (*see* **Note 4**). The values of the upper bound and lower bound of region A and region B are given by *tps_initial_max*, *tps_initial_min*, *tps_final_max*, and *tps_final_min*. The number of groups in *tps_grps1* and *tps_grps2* should be the same. *tps_npost* is the total number of groups in *tps_grps1* and *tps_grps2*.

4.2 Obtaining an Initial Transition Path

In TPS, the shooting and shifting moves are based on an initial path, which is not necessary to be physically meaningful, as the subsequent TPS will allow to relax towards more representative paths (*see* **Note 5**). There are many ways to get an initial path. Here, we generate the first path by shooting forward and backward from a structure near the transition state to ensure that we can get an initial reactive path with high probability. Velocities are adapted from a Boltzmann distribution at the given temperature. The setting of the parameters in the TPS section are (the complete parameter file is available in *tps_ini.mdp* provided in the tutorial package),

```
=====   Part of tps_ini.mdp   =====
tps_npost                  = 4
tps_grps1                  = a_1 a_2
tps_grps2                  = a_2 a_3
tps_dimension              = one
tps_weight_dim             = 1    -1
tps_initial_max            = -0.5
tps_initial_min            = -1
tps_final_max              = 1
tps_final_min              = 0.5
tps                        = rand_ini
tps_maxcycle               = 5
tps_maxshoot               = 10
tps_endpoint               = yes
tps_kin_ref                = 100
tps_Temperature            = 300
tps_forward_steps          = 400
tps_backward_steps         = 400
tps_maxframe               = 1
tps_ntrrout                = 1
========================
```

tps=rand_ini defines the attempt to get an initial transition path. *tps_maxcycle* and *tps_maxshoot* specifies the number of sampling circles and the number of samples in each circles. Here, we try 5 × 10 = 50 times to get an initial path. The search will stop immediately if we find one. *tps_endpoint=yes* means the endpoint of the path should be in region B. *tps_forward_steps* and *tps_backward_steps* specify the number of frames saved for forward and backward shooting, respectively. The total frames of the trajectory generated will be the sum of them. Here, the number of frames in the transition path will be 800. The frequency of saving frames in the transition path is defined by *nstxout=10* and the integration step is 2 fs (*see* **Note 6**). Therefore, we will obtain a reactive path with 800 frames, with an interval between each frame of 20 fs and a total length of $t = 16$ ps. *tps_Temperature=300* produces initial atomic velocities from a Boltzmann distribution at a temperature of 300K. *tps_kin_ref* specifies the amount of perturbation to the momenta of the atoms in the frame to which a shooting move is applied. The value will affect the acceptance ratio of shooting moves, with larger perturbations leading to smaller acceptance ratios. *tps_maxframe* is the number of frames in the input trajectory. In this case, *tps_maxframe=1* because the input is a .gro file which contains only one frame. *tps_ntrrout=1* makes the MD code saving every reactive trajectories.

The following input commands initiate the search for an initial path:

```
tar -zxvf example_3_Ar.tar.gz

cd example_3_Ar && mkdir -p tps/initial

grompp -f tps_ini.mdp -c 3_Ar.gro -n index.ndx -p topol.top -o tps/initial/tps.tpr

cd tps/initial &&  mdrun -s tps.tpr -rerun ../../TS.gro -deffnm tps_output
```

The input structure is read via option *-rerun*. *TS.gro* is the structure near the transition region. *3_Ar.gro* is a structure at region A. *index.ndx* and *topol.top* define the index of groups used in the .mdp file and the topology of the system, respectively. Tutorials to prepare these files can be found elsewhere [21] and is out of the scope of this chapter. We therefore provide them in the tutorial package. This run will take about 10 min on a single processor to generate an initial transition path saved as *traj_0.trr*. We rename it as *tps.trr* by executing

```
mv traj_0.trr ../tps.trr
```

4.3 Obtaining the $\langle h_B(t)\rangle_{AB}$ Curve

A requisite to compute the rate constant using TPS is the flux versus time, or the $\langle h_B(t)\rangle_{AB}$ curve, and the probability distribution along an order parameter $P(\lambda, t)$, or specifically $P(\Delta d)$ in this case, which is then used to calculate the value of $C(t)$ at a specific time t (see Theory). With an initial path at hand, we can start TPS to obtain these ingredients for the rate constant calculations. The settings for this purpose are:

```
=====Part of tps.mdp =====
tps_npost           = 4
tps_grps1           = a_1 a_2
tps_grps2           = a_2 a_3
tps_dimension       = one
tps_weight_dim      = 1  -1
tps_initial_max     = -0.5
tps_initial_min     = -1
tps_final_max       = 1
tps_final_min       = 0.5
tps                 = normal
tps_maxcycle        = 150
tps_maxshoot        = 10
tps_maxshift        = 10
tps_endpoint        = no
tps_kin_ref         = 100
tps_reput_length    = 300
tps_maxframe        = 800
tps_ntrrout         = 0
========================
```

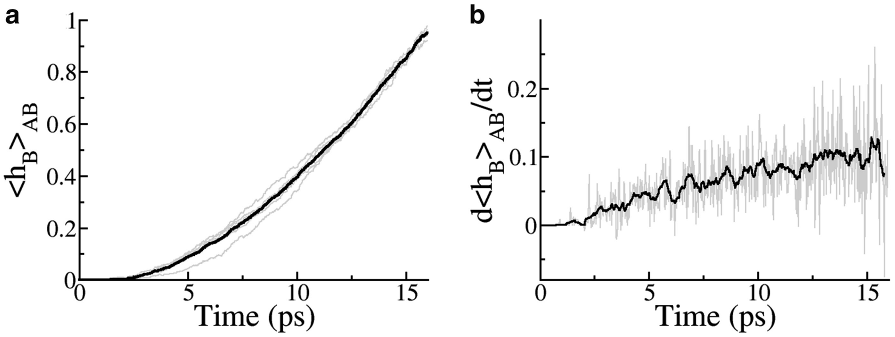


Fig. 3 Results for the $\langle h_B(t)\rangle_{AB}$ curve. (**a**) *Black curve*: the averaged $\langle h_B(t)\rangle_{AB}$ curve. *Grey curves*: the five $\langle h_B(t)\rangle_{AB}$ curves obtained from five independent samplings. (**b**) The derivative of the $\langle h_B(t)\rangle_{AB}$ curve shows a plateau, indicating a length of 16 ps to be sufficient. *Grey*: the derivative of the *black curve* in **a**. *Black*: the smoothed curve of the *grey* one by averaging over five nearby points

tps=normal means that we now perform TPS with an initial path already at hand. *tps_maxcycle* specifies the number of TPS cycles. In each cycle, we perform several shooting moves and shifting moves, and the number of these moves is specified by *tps_maxshoot* and *tps_maxshift*, respectively. *tps_endpoint=no* means the end structure of a reactive trajectory, given the trajectory starts in A, may not be in B, but the structure should reach B at some point within the trajectory (see Theory). *tps_reput_length* specifies the maximum shifting length in shifting moves. *tps_ntrrout =0* means no intermediate reactive trajectories are saved.

Performing the TPS requires executing the following commands:

```
cd ../../ && grompp -f tps.mdp -c 3_Ar.gro -n index.ndx -p topol.top -o tps/tps.tpr

cd tps && mdrun -s tps.tpr -rerun tps.trr -deffnm tps_output
```

We read the initial reactive path again via the option *-rerun*. Here, we run 150 cycles of TPS, with 10 shooting moves and 10 shifting moves in each cycle. In total there are 3,000 TPS runs. This will take about 4 days to complete on a single standard processor. The results of the $\langle h_B(t)\rangle_{AB}$ curve is saved in *hahb.dat*. To obtain an accurate $\langle h_B(t)\rangle_{AB}$ curve, we recommend the reader to run five independent simulations (*see* **Note 7**), and to then combine the resulting five hahb.dat files into one by simple averaging (Fig. 3a). Here, the derivative of $\langle h_B(t)\rangle_{AB}$ reaches a plateau at 13 ps with $d\langle h_B(t)\rangle_{AB}/dt = 0.1\ ps^{-1}$ as shown in Fig. 3b (*see* **Note 8**).

4.4 Obtaining the P(λ, t) Distribution

In the next step, we run TPS in different windows to obtain the distribution of the end points of transition paths (the $P(\Delta d)$ distribution in Eq. 14). Here, we set the length of the trajectory t to be $t' = 6$ ps, with each path containing 300 frames, which is much shorter than 16 ps or 800 frames and saves computational cost (*see* **Note 9**). Therefore, we read $\langle h_{\mathrm{B}}(t')\rangle_{\mathrm{AB}} = 0.14$ from Fig. 3a, as $t' = 6$ ps. In order to get the $P(\Delta d)$ distribution, we divide the configuration space into five windows, which are defined as window 1: $-1 < \Delta d < -0.45$, window 2: $-0.55 < \Delta d < -0.15$, window 3: $-0.25 < \Delta d < 0.25$, window 4: $0.15 < \Delta d < 0.55$, and window 5: $0.45 < \Delta d < 1$. A small overlap between adjacent windows is necessary to merge the distributions of adjacent windows into one. By deleting the segments from the termini of the transition path obtained above (16 ps long), one can easily get an initial path of 6 ps long (*see* **Note 10**). The sampling procedures in each window are similar. For each window, we define the correct region B and adjust *tps_kin_ref* and *tps_reput_length* to maintain a reasonable acceptance ratio. The parameter files for all windows are provided in the tutorial package and are named as *tps_win1.mdp* to *tps_win5.mdp*. We here show the TPS parameters for window 5 as an example to illustrate the procedure.

```
=====Part of tps_win5.mdp =====
tps_npost              = 4
tps_grps1              = a_1 a_2
tps_grps2              = a_2 a_3
tps_dimension          = one
tps_weight_dim         = 1      -1
tps_initial_max        = -0.5
tps_initial_min        = -1
tps_final_max          = 1
tps_final_min          = 0.45
tps                    = normal
tps_maxcycle           = 300
tps_maxshoot           = 10
tps_maxshift           = 10
tps_endpoint           = yes
tps_kin_ref            = 200
tps_reput_length       = 100
tps_maxframe           = 300
tps_ntrrout            = 0
==========================
```

To obtain the endpoint distribution, we need to make sure the endpoints of the transition path to be within the defined region B by setting *tps_endpoint=yes*. The simulation is started as follows:

```
cd ../ && mkdir win5
grompp -f tps_win5.mdp -c 3_Ar.gro -n index.ndx -p topol.top -o win5/tps_win5.tpr
cd win5 && mdrun -s tps_win5.tpr -rerun tps_win5.trr -deffnm tps_output
```

Here, *tps_win5.trr* is the constructed initial transition path. The endpoints of each transition path are saved in *endpoint.dat*.

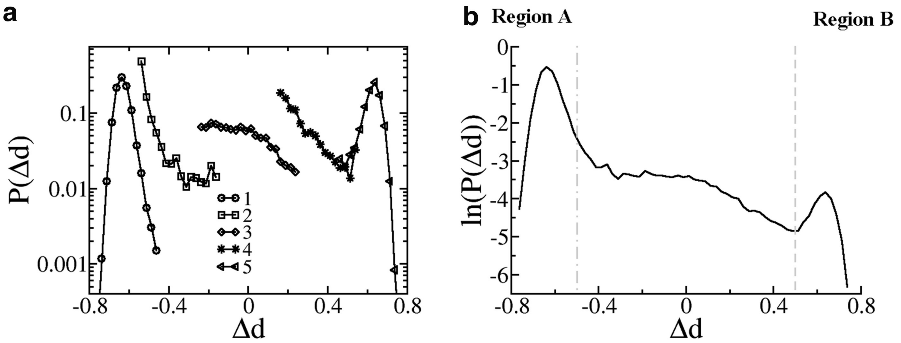


Fig. 4 Calculation of $P(\Delta d)$ through TPS in windows. (**a**) Distribution of $P(\Delta d)$ in different windows. (**b**) The connected distribution of $P(\Delta d)$ over the whole configuration space. *Dashed grey lines*: the boundaries between regions A and B and the transition region

We recommend the reader to run three independent simulations for each window and then collect all the endpoints into a single file of *endpoint.dat*. From the *endpoint.dat* file, the distribution along Δd can be easily obtained. The distributions of in the overlapped regions are the same but weighted differently. Therefore, we can connect all windows by re-weighting them properly. The connected window is then normalized. The distributions in different windows and the normalized distribution are shown in Fig. 4. By integrating the distribution in the range of $0.5 < \Delta d < 1$, we obtain a value of $C(t')$ of 0.00056 (see Theory).

4.5 Calculating the Rate Constant

Given $C(t')$ is 0.00056, $\langle h_B(t')\rangle_{AB}$ is 0.14, and $d\langle h_B(t)\rangle_{AB}/dt$ is 0.1 ps^{-1}, we get a rate constant k of 4.0×10^{-4} ps using Eq. 16 (*see* **Note 11**).

4.6 Monitoring TPS

The *mdrun* command will generate four output files that help to monitor the progress of the sampling: *acc.dat*, *endpoint.dat*, *hahb.dat*, and *summary.dat*. They are explained below:

acc.dat: It summarizes the number of shooting trials, the number of successful shooting trials, the number of shifting trials, and the number of successful shifting trials at each frame. It also includes the acceptance ratio for shooting and shifting.

endpoint.dat: It gives the endpoints of the transition paths in the value of the order parameter, which is used to calculate $P(\lambda, t)$ when *tps_endpoint=yes.*

hahb.dat: It gives the $\langle h_B(t)\rangle_{AB}$ curve when *tps_endpoint=no.*

summary.dat: It summarizes the overall number of shooting and shifting cycles and their acceptance ratio. An example is given below:

```
===============summary.dat==================
The totol TPS cycle is -----------------------------3000
The totol shooting cycle is ------------------------1524
The totol leftshift cycle is -----------------------747
The totol rightshift cycle is ---------------------729
The totol acceptance is --------------------------0.3076667
The acceptance for shooting is --------------------0.0577428
The acceptance for leftshift is ------------------0.5689424
The acceptance for rightshift is ------------------0.5624143
==================================
```

Here, the acceptance for shooting is quite low, around 0.06. Adjusting *tps_kin_ref* and *tps_reput_length* allows to tune the probability of generating a reactive trajectory. A higher acceptance ratio can be achieved by shortening the length of the transition path (*see* **Note 12**). To achieve a better sampling efficiency, the acceptance for shooting it recommended to be about 0.4 [22].

5 Notes

1. The modified GROMACS package supports simulations on only a single CPU and not in parallel, as neither domain decomposition nor particle decomposition are supported in the current implementation.
2. Equation 10 is based on the assumption that the barrier is so high that the time of the actual transition is much smaller than the inverse of the rate constant. Therefore, Eq. 10 is only applicable to systems with high energy barriers, i.e., of several $k_B T$.
3. For many systems, the choice of an order parameter is trivial. One can run a relatively long simulation at the two stable states, and then find an order parameter to distinguish the stable states by inspection of the coordinate spaces that the two simulations sampled at both basins. Usually, an inspection by eye is enough. If not, principle component analysis [23] can assist in identifying an order parameter. Once an order parameter is found, one defines the two stable states according to their distribution of the sampled configuration along the order parameter. Make sure that the two basins are separated and cover the major part of the sampled configurations in that state.
4. One can use multiple coordinates to define regions A and B if the interest is to investigate the mechanism of the transition process rather than the rate constant. If one want to get the rate constant, region A can be defined with multiple coordinates, while region B is preferably defined with a single coordinate, as this reduces the computational expense. If defining region B by multiple coordinates is nevertheless essential, the distribution of $P(\lambda, t)$ is required in the multidimensional

space, which might be feasible but is not recommended. It is generally beneficial to invest some efforts to find a single coordinate to define region B. The coordinates to define regions A and B are not required to be identical.

5. One can generate an initial transition path in many ways, depending on the system under investigation. In general, one can apply a bias to the system to enforce the transition to happen with high probability and short transition times. The bias can be from for example high temperature [24], replica exchange [24], position restraints [10], steering forces [26], conformational flooding [27], or metadynamics [28]. The resulting transition path is a biased transition path, but the bias can be removed gradually [26], or by generating an unbiased path by TPS with shooting moves starting from one frame (e.g., a frame within or close to the transition state ensemble) of the biased path.
6. The number of steps for a simulation defined by *nsteps* in the .mdp file should be larger than the total length of the TPS trajectory. Here, *nsteps* should be no less than 8,000 given the integral timestep of 2 fs.
7. The TPS simulation will generate files with predefined names in the working directory. If one runs multiple independent simulations, it is recommended to start them from separate directories to avoid overwriting output files.
8. In addition to the $\langle h_B(t)\rangle_{AB}$ curve, the simulation will harvest an ensemble of transition paths (one can save the transition paths by setting *tps_ntrrout* to a positive integer to specify the frequency of saving the transition paths). Based on the transition path ensemble, the mechanism of the studied rare events can be elucidated at the atomistic level. Applications include (just to name a few) a reaction catalyzed by lactate dehydrogenase [8], the β-hairpin folding [25], and the chorismate-mutase-catalyzed conversion of chorismate into prephenate [29]. If committor probabilities of the frames in the transition path ensemble are estimated, transition states [10, 29] and reaction coordinates [30] can be identified as well.
9. Choosing a small t' reduces the computational costs of the TPS, but increases the uncertainty of $\langle h_B(t')\rangle_{AB}$. As a compromise, we recommend to choose t' such that $\langle h_B(t')\rangle_{AB}$ is about 0.2. Usually, the cost to calculate $C(t')$ is several times higher than the cost to obtain $\langle h_B(t)\rangle_{AB}$. For this reason, it is worth to invest more efforts into an accurate $\langle h_B(t)\rangle_{AB}$ curve, and to then use a smaller t' to calculate $C(t')$.
10. The initial path for each window to obtain the $\langle h_B(t)\rangle_{AB}$ curve can be obtained following the same procedure as the one for the initial path for the TPS. Alternatively, the initial path for

window 5 should have 300 frames, which can be constructed by shortening the transition paths sampled in the section of obtaining the $\langle h_B(t)\rangle_{AB}$ curve, which comprises 800 frames. Then, the initial path for window 4 can be obtained from one of transition paths sampled in window 5. The initial path for other windows can be taken from one of transition paths from the subsequent window.

11. As a way of validating the computed rates, one can vary t and/or t' as well as the definitions of regions A and/or B to test if the results are quantitatively consistent.
12. It is not necessary to randomly select a frame from the entire transition path to do shooting moves. Specifying the range from which shooting frames are selected allows to increase the acceptance ratio. One possibility is to choose points near the previous shooting point from which a reactive trajectory has been generated. Alternatively, one can choose points only from the barrier region to improve the acceptance ratio. In this case the range to choose shooting points is variable, and the probability to accept a reactive trajectory needs to be modified accordingly [25].

Acknowledgment

We are grateful to the Klaus Tschira Foundation for financial support.

References

1. Lane TJ, Bowman GR, Beauchamp K et al (2011) Markov state model reveals folding and functional dynamics in ultra-long MD trajectories. J Am Chem Soc 133:18413–18419
2. Bowman GR, Pande VS (2010) Protein folded states are kinetic hubs. Proc Natl Acad Sci U S A 107:10890–10895
3. van der Spoel D, Seibert MM (2006) Protein folding kinetics and thermodynamics from atomistic simulations. Phys Rev Lett 96:238102
4. Best RB, Hummer G (2006) Diffusive model of protein folding dynamics with Kramers turnover in rate. Phys Rev Lett 96:228104
5. Popa I, Fernández JM, Garcia-Manyes S (2011) Direct quantification of the attempt frequency determining the mechanical unfolding of ubiquitin protein. J Biol Chem 286:31072–31079
6. Dellago C, Bolhuis PG, Csajka FS et al (1998) Transition path sampling and the calculation of rate constants. J Chem Phys 108:1964
7. Dellago C, Bolhuis PG, Chandler D (1998) Efficient transition path sampling: application to Lennard-Jones cluster rearrangements. J Chem Phys 108:9236
8. Quaytman SL, Schwartz SD (2007) Reaction coordinate of an enzymatic reaction revealed by transition path sampling. Proc Natl Acad Sci U S A 104:12253–12258
9. Saen-Oon S, Quaytman-Machleder S, Schramm VL et al (2008) Atomic detail of chemical transformation at the transition state of an enzymatic reaction. Proc Natl Acad Sci U S A 105:16543–16548
10. Li W, Gräter F (2010) Atomistic evidence of how force dynamically regulates thiol/disulfide exchange. J Am Chem Soc 132:16790–16795
11. Xia F, Bronowska AK, Cheng S et al (2011) Base-catalyzed peptide hydrolysis is insensitive to mechanical stress. J Phys Chem B 115:10126–10132
12. van der Kamp MW, Mulholland AJ (2013) Combined quantum mechanics/molecular mechanics (QM/MM) methods in computational enzymology. Biochemistry 52:2708–2728

13. Steinbrecher T, Elstner M (2013) QM and QM/MM simulations of proteins. In: Monticelli L, Salonen E (eds) Biomolecular simulations. Humana Press, New York, pp 91–124
14. Groenhof G (2013) Introduction to QM/MM simulations. In: Monticelli L, Salonen E (eds) Biomolecular simulations. Humana Press, New York, pp 43–66
15. Hess B, Kutzner C, Van Der Spoel D et al (2008) GROMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation. J Chem Theory Comput 4:435–447
16. Bolhuis PG, Dellago C, Chandler D (1998) Sampling ensembles of deterministic transition pathways. Faraday Discuss 110:421–436
17. Dellago C, Bolhuis PG, Chandler D (1999) On the calculation of reaction rate constants in the transition path ensemble. J Chem Phys 110:6617
18. Dellago C, Bolhuis PG, Geissler PL (2002) Transition path sampling. Adv Chem Phys 123:1–78
19. Metropolis N, Rosenbluth AW, Rosenbluth MN et al (1953) Equation of state calculations by fast computing machines. J Chem Phys 21: 1087
20. Chandler D (1978) Statistical mechanics of isomerization dynamics in liquids and the transition state approximation. J Chem Phys 68:2959
21. Lindahl EL (2008) Molecular dynamics simulations. In: Kukol A (ed) Molecular modeling of proteins. Humana Press, Totowa, NJ, pp 3–23
22. Bolhuis PG, Chandler D, Dellago C et al (2002) Transition path sampling: throwing ropes over rough mountain passes, in the dark. Annu Rev Phys Chem 53:291–318
23. Jolliffe I (2005) Principal component analysis. Wiley Online Library
24. Dellago C, Bolhuis PG, Geissler PL (2006) Transition path sampling methods. In: Computer simulations in condensed matter systems: from materials to chemical biology, vol 1. Springer, Berlin, pp 349–391
25. Bolhuis PG (2003) Transition-path sampling of β-hairpin folding. Proc Natl Acad Sci U S A 100:12129–12134
26. Hu J, Ma A, Dinner AR (2006) Bias annealing: a method for obtaining transition paths de novo. J Chem Phys 125:114101
27. Grubmüller H (1995) Predicting slow structural transitions in macromolecular systems: conformational flooding. Phys Rev E 52:2893
28. Laio A, Gervasio FL (2008) Metadynamics: a method to simulate rare events and reconstruct the free energy in biophysics, chemistry and material science. Rep Prog Phys 71:126601
29. Crehuet R, Field MJ (2007) A transition path sampling study of the reaction catalyzed by the enzyme chorismate mutase. J Phys Chem B 111:5708–5718
30. Ma A, Dinner AR (2005) Automatic method for identifying reaction coordinates in complex systems. J Phys Chem B 109:6769–6779

Chapter 3

Current Status of Protein Force Fields for Molecular Dynamics Simulations

Pedro E.M. Lopes, Olgun Guvench, and Alexander D. MacKerell Jr.

Abstract

The current status of classical force fields for proteins is reviewed. These include additive force fields as well as the latest developments in the Drude and AMOEBA polarizable force fields. Parametrization strategies developed specifically for the Drude force field are described and compared with the additive CHARMM36 force field. Results from molecular simulations of proteins and small peptides are summarized to illustrate the performance of the Drude and AMOEBA force fields.

Key words Force field, Molecular dynamics, Drude polarizable force field, CHARMM, AMOEBA, AMBER, GROMOS, OPLS, NAMD, Electronic polarization

1 Introduction

Classical molecular dynamics (MD) simulations of proteins using empirical force fields have reached a mature state after 35 years of development and are now widely used as tools to investigate their structure and dynamics under a wide variety of conditions. These include studies of ligand binding, enzymatic-reaction mechanisms, protein folding and unfolding, and protein–protein interactions.

Fundamental to such simulations is determination of the time evolution of the system's energy (protein for example) as a function of its atomic coordinates. An accurate description of the energy is thus required, since the lower energy states are expected to be populated. The gradient of the energy function, which is differentiable, is related to the forces acting on individual atoms. In chemistry the set of potential energy functions from which the forces are derived is commonly referred to as a force field (FF). As a result of many years of careful refinement, current additive protein energy functions are of sufficient quality that they may be used predicatively for studying protein dynamics and protein–protein interactions and in pharmacological applications [1]. It is clear that the next major step in advancing protein force field accuracy

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_3, © Springer Science+Business Media New York 2015

requires a different representation of the molecular energy surface. Specifically, the effects of charge polarization must be included, as fields induced by ions, solvent, other macromolecules, and the protein itself will affect electrostatic interactions [2–6].

Our goal here is to provide an update of the newest developments that have occurred in the field of FF-based MD simulations of proteins since our last review was published [7]. Previously, we focused on the functional forms of additive FFs, strategies for parameter optimization, methodologies to perform MD simulations such as pressure and temperature control, and software packages available for MD simulations. Also briefly mentioned were efforts to extend additive FFs to other biomolecules. The FFs detailed were the Amber, CHARMM, GROMOS, and OPLS-AA additive protein FFs, with particular emphasis on CHARMM because of our continuing role in its development. While the present review begins with a brief update on the status of additive protein FFs, here we primarily focus on the latest developments in the inclusion of electronic polarizability into protein FFs. Emphasis is placed on the CHARMM Drude polarizable FF and the polarizable AMOEBA FF, and we direct interested readers to other recent reviews [5, 6, 8, 9]. Also of interest may be new improvements in the Amber family of FFs [10], and, to our knowledge, no new reviews on the OPLS-AA or GROMOS protein FFs have appeared since our previous review in this series.

A general familiarity with molecular mechanics and dynamics and their applications to proteins is assumed. Simulation methods for proteins are well established, with many good textbooks and monographs covering the basics [11–17]. The reader is also referred to Chapter 1 of this volume.

2 Current Status of Additive Force Fields

Since the last review in this series [7], some notable developments have been made to additive FFs for proteins. Below, brief descriptions of the improvements introduced to two of the major additive FFs for proteins, CHARMM and Amber, are given.

2.1 CHARMM Force Field

The CHARMM additive all-atom FF has been in development since the early 1980s [18] and has achieved a substantial degree of completeness with regard to coverage of chemical space. Apart from proteins [19, 20], it supports nucleic acids [21–23], lipids [24–26], and carbohydrates [27–30], allowing simulations on all commonly encountered motifs in biological systems. It has also been extended to cover the wide range of the chemical space required to study compounds common in medicinal chemistry through the CHARMM General FF (CGenFF) [31]. The CHARMM additive FF for proteins recently underwent a significant update that culminated in the C36 version of the FF, as detailed below [20].

Long simulations with the additive C22/CMAP FF [19, 32, 33] had shown that certain fast-folding proteins would reach the native state, when started from a completely unfolded configuration (e.g., Villin headpiece subdomain) [34]. However, significant deficiencies were also found. Examples of problems included misfolding encountered in long simulations of the pin WW domain and differences in the Villin folding mechanism from the experimental results [34, 35]. In the case of the WW domain, free energy calculations showed that the misfolded states had lower free energies than the folded state, confirming that the energy function could be further improved [36]. A number of studies had suggested that such differences could be the result of small inaccuracies in the energy of the backbone, resulting in one structure being favored, and that this behavior could be corrected with minor adjustments to the backbone potential [37–40]. Best et al. reported a revised set of CHARMM all-atom protein FF parameters (C36) that represents a significant improvement in the potential energy surface, while keeping the same functional form [20]. The improvements that were introduced included (1) a new backbone CMAP potential, optimized against experimental data on small peptides and larger, folded proteins and (2) new side-chain dihedral parameters optimized using QM energies for dipeptides, conformational sampling in the model system $(\text{Ala})_4$-X-$(\text{Ala})_4$ [41] and NMR data from unfolded proteins. Other improvements relative to the previous C22/CMAP protein FF included Lennard–Jones (LJ) parameters for aliphatic hydrogens [42], internal parameters for the guanidinium ion [43], and improved parameters for tryptophan [44]. Changes of the backbone and side-chains were done simultaneously, ensuring that in the new FF their contribution to protein structure and dynamics is balanced.

2.2 Amber Force Field

Amber FFs for proteins have been continually improved in recent years and a detailed discussion of the various changes is beyond the scope of this review. Significant revisions have been published, with particular emphasis on important dihedral angles. Simmerling and coworkers [45] introduced changes to the backbone potential in the original Amber ff99 FF by fitting to additional quantum-level data to produce the improved Amber ff99SB FF. Best and Hummer continued along the same line, modifying the backbone potential of the ff99SB and ff03 FFs to obtain a better balance between sampling of helix and coil conformations. The new FFs were named ff99SB* and ff03*, respectively [38]. Modifications of the side-chain torsion potential for four amino acid types in ff99SB was introduced by Lindorff-Larsen et al. originating the ff99SB-ILDN FF [46]. Further enhancements were produced by Li and Bruschweiler based on experimental NMR data, originating the ff99SB-ILDN-NMR FF [47]. To our knowledge, the latest update in the Amber FFs was introduced recently by Neremberg and

Head-Gordon, who included a perturbation to the ϕ backbone dihedral potential to shift the beta–PPII equilibrium. This resulted in improved sampling in water (TIP3P and TIP4P-Ew). Their updates were designated ff99SB-ILDN-Phi [48]. In addition to proteins, the Amber FFs support most common biomolecules. The ff10 FF collection includes the most commonly used variants: the ff99SB protein parameters [45], the BSC0 DNA parameters [49], the Cheatham et al. ion parameters [50, 51], and updated RNA parameters [52, 53]. Carbohydrates are supported through the Glycam FFs [54–56], and phospholipids are supported through the CHARMM FF and the recent Lipid11 FF [57].

3 Polarizable Force Fields for Biomolecules: Current Status

3.1 Drude Polarizable Force Field

Development of the Drude polarizable FF in CHARMM [58] started in 2001 and the capability to simulate the Drude model is now included in NAMD [59], ChemShell QM/MM [60], and the OpenMM suite of utilities for GPUs [61]. Development of the force field first involved implementation of the appropriate integrators to allow computationally efficient extended Lagrangian MD simulations [62]. This was followed by optimization of the first water model, in which a positive charge was assigned to the Drude particle (SWM4-DP) [63]. The SWM4-DP model was reoptimized with a negative charged assigned to the Drude particles, consistent with their representation of the electronic degrees of freedom. The new model, called SWM4-NDP, is the standard polarizable water model of the Drude polarizable FF [64]. It was calibrated to reproduce important properties of the neat liquid at room temperature and pressure such as enthalpy of vaporization, density, static dielectric constant and self-diffusion constant, free energy of hydration and shear viscosity. Concurrently with development of the water model, methodologies to determine electrostatic parameters for the Drude FF were advanced [65].

An early test of the feasibility of molecular dynamics simulations with the Drude polarizable FF was a successful simulation of a DNA octamer in a box of water with sodium counterions [66]. Development of the Drude polarizable FF continued with parametrization of small molecules covering the functional groups commonly found in biomolecules. In 2005, the alkane FF was developed, followed by parametrization of alcohols and aromatic compounds in 2007 [67, 68]. Harder et al. published the first generation of *N*-methyl acetamide (NMA) parameters in 2008 [69]. Noteworthy is the proper treatment of the dielectric constant by the polarizable FF in all systems, a property considered essential for the accurate treatment of, for example, hydrophobic solvation in biomolecules. The Drude polarizable FF was extended to the nitrogen-containing heteroaromatic compounds in 2009 [5]. FF parameters were refitted for ethers by Baker and MacKerell [70], with significant improvements in the reproduction of liquid phase dielectric constants, while maintaining the good agreement of the previous model with all other experimental and quantum mechanical target data [71]. Sulfur containing model compounds were parametrized in 2010 [72]. Other classes of molecules for which Drude empirical FF parameters have been developed are nucleic acid bases [73] and acyclic polyalcohols [74]. Early simulations of dipalmitoylphosphatidylcholine (DPPC) bilayers and monolayers were reported [75], followed by completion of a refined model for DPPC [76].

Significant progress has been made in extending the Drude polarizable FF from small compounds representative of the building blocks encountered in biological polymers to the polymers themselves. The Drude empirical FF applicable to MD simulation studies of peptides and proteins, termed Drude-2013, is covered in Subheading 5, which includes a full account of the results. The optimization of the polypeptide backbone parameters is discussed in detail in Subheading 4.2 and the optimization of side-chain torsions is discussed in Subheading 4.3.

3.2 Amoeba Force Field

AMOEBA is another classical polarizable FF that has achieved the goal of producing a fully functional FF model for proteins [77]. Development of the AMOEBA polarizable FF has been ongoing since 1995 [78] and is based on modeling the electrostatic energy using permanent and induced contributions. Permanent electrostatics originate in atomic multipole–multipole interactions with moments up to the quadrupole located on each atom. The induced contribution is modeled iteratively by generating an induced dipole originated by permanent multipoles and other induced dipoles. Self-consistency is obtained using an iterative scheme, and the Thole model [79] is used to dampen electrostatic interactions at short range.

The AMOEBA FF was initially developed for water [80, 81]. Testing included reproduction of a variety of experimental data and quantum calculations for small clusters, liquid water, and ice. Several liquid phase properties including bulk thermodynamic, transport, and structural measures were tested. These included density, heat of vaporization, self-diffusion coefficient, heat capacity, dielectric constant, and radial distribution functions. Overall, excellent agreement with reference values was obtained, and the model was demonstrated to be applicable to structural properties of two ice forms [81].

Treatment of ions in AMOEBA is described in ref. 82. Absolute solvation free energies for potassium, sodium, and chloride ions in liquid water and formamide have been computed. Simulation results accurately reproduced vacuum QM results, experimental ion-cluster solvation enthalpies, and experimental solvation free energies for whole salts.

The AMOEBA FF has been extended to organic molecules, including alkanes, alcohols, amines, sulfides, aldehydes, carboxylic acids, amides, aromatics, and other small organic molecules [83]. As a validation, the hydrogen bonding energies and structures of gas phase heterodimers with water were evaluated. Liquid self diffusion and static dielectric constants computed from MD simulations with AMOEBA are consistent with experimental values. The FF was further tested by computing the solvation free energy of 27 compounds not included in the parametrization process. It performed well across different environments and phases, yielding an RMS error of 0.69 kcal/mol. Analysis of the dependence of computed hydration free energies for seven small organic molecules with the QM level of theory used to derive atomic multipoles was presented recently [84]. It was concluded that inclusion of diffuse functions in the QM calculation of the atomic multipoles is important. More comprehensive descriptions of the AMOEBA FF have been presented previously and the reader is referred to those publications for additional details [85–87].

4 Parametrization of Polarizable Force Fields

4.1 Generic Parametrization Strategies for the Drude Polarizable Force Field

The quality of FFs is heavily dependent on the quality of the underlying parameters. To obtain parameters of sufficient quality that are capable of producing accurate simulation results, procedures have been developed to target properties such as molecular geometries and vibrations, pure solvent properties, and free energies of solvation, among others during the parametrization. In this section we will describe parametrization of the polarizable Drude FF implemented in CHARMM. Reference to the well-established protocol used to derive CHARMM additive FF parameters will be done whenever a parallel is useful. The general outline of the parametrization process has been described for the CHARMM additive FF in several publications (see refs. 1 and 19 for more details). Note that parameter optimization remains an iterative process in the polarizable FF and several rounds of parametrization are typically performed until a satisfactory level of agreement with target data is obtained.

A common strategy in parameter optimization of biological macromolecules is that parameters are developed for small, representative model compounds and then transferred to the larger macromolecules. The advantages of this approach are: (1) smaller models are easier to treat using both MM and QM methods and (2) more experimental data are available for the smaller systems, including thermodynamic properties of condensed phases, such as heats of vaporization or sublimation and free energies of aqueous solvation. It is crucial to include such data in the parameter optimization process to get an accurate description of the non-bond portion of the FF.

This strategy was also attempted in the development of Drude FF parameters for the protein backbone, but ultimately a more involved procedure was required as detailed in Subheading 4.2.

Parametrization of CHARMM FFs relies on obtaining appropriate intramolecular (bond, angle, dihedral, Urey-Bradley, and improper terms), van der Waals (vdW), and electrostatic parameters that adequately reproduce selected target data. Determination of the electrostatic parameters differs between the additive and the Drude polarizable FFs. In the Drude FFs, in addition to optimization of point charges, which is also required in the additive FF, polarizabilities and Thole factors must also be determined. In the additive CHARMM FF, optimization of point charges is based on a supramolecular approach where the charges are adjusted to reproduce QM HF/6-31G* interaction energies and geometries of the model compound with, typically, individual water molecules. Placement of water molecules at different orientations around the molecule enforces that local electronic polarization is accounted for implicitly, an important feature for accurate reproduction of condensed-phase properties. Additional data often include QM results on dimers and dipole moments of the models. It is well-known that in additive force fields, dipole moments must be overestimated to reproduce condensed phase properties [19, 88].

Other additive biomolecular FFs, most notably Amber, determine atomic partial charges based on reproduction of the QM Electrostatic Potential (ESP), evaluated on grids surrounding model compounds [89–91]. These methods are convenient because charges can be developed quickly for any compound for which the QM ESP can be determined. An extension of ESP methods is inclusion of restraints during fitting, referred to as the restrained ESP (RESP) approach [92]. This overcomes limitations on the determination of charges on buried atoms [93]. It is important to note that partial charges from both supramolecular and ESP approaches are conformation-dependent, requiring care in the selection of appropriate conformations when performing the charge optimization.

Electrostatic parameters of model compounds in the Drude polarizable FF are obtained from restrained fitting to perturbed QM ESP maps on grid points located on concentric Connolly surfaces surrounding the molecule. Often fitting is supplemented with reproduction of the molecular dipole moment and diagonal elements of the polarization tensor [65, 94]. The determination of the atomic polarizabilities and Thole factors [79] requires multiple perturbed ESPs typically calculated at the B3LYP/aug-cc-pVDZ level, with each giving the electronic response of the molecule to a point charge. Perturbing ions are placed mainly along chemical bonds and lone pairs (LPs). This protocol was later extended to incorporate additional lone pair parameters and polarizability anisotropy, and has become the standard in developing electrostatic parameters for small molecules [95]. LPs typically carry the charge of the atom (e.g., N, O, S in proteins) to which they are attached. The associated polarizability and Thole factor are both assigned to the parent atom. Anisotropic polarizability of hydrogen bond acceptors was found to be required to reproduce interactions with ions as a function of orientation. Initial values for the partial atomic charges are taken from the C22 additive all-atom FF, and those for the polarizabilities are based on adjusted Miller's atomic hybrid polarizability (ahp) values [96].

Although gas-phase properties (e.g., dipole moments) are easily reproduced with full atomic polarizabilities, scaling of the polarizabilities has been shown to be necessary to reproduce condensed-phase properties [64]. A scaling factor of approximately 0.7 was found appropriate for the SWM4-DP and SWM4-NDP water models while for other classes of molecules scaling factors range from 0.6 to 1.0, with 1.0 being full polarizability. For instance, scaling factors are 0.7 for primary and secondary alcohols [67], 0.85 for aromatics [68], N-containing heterocycles [94], nucleic acid bases [73] and ethers [97], and 1.0 for alkanes [42]. Other scaling factors are 0.7 for thiols, 0.85 for dimethyl disulfide and 0.6 for ethylmethyl sulfide [72]. A value of 0.724 was recently used in ion parameters [98]. Final optimization of the electrostatic parameters consists of testing the model for reproduction of the pure solvent dielectric constants and adjusting the polarizability scaling if necessary.

Development of parameters to model vdW forces in the Drude FF, which are treated using the Lennard–Jones (LJ) 6–12 term, follows closely the protocol established for the additive FF and will only be briefly outlined here. Jorgensen and coworkers [99, 100] pioneered the use of condensed-phase simulations, usually pure liquids, as the basis for optimization of Lennard–Jones (LJ) parameters that account for both vdW attraction and interatomic repulsion. Typically, once electrostatic parameters are determined, the LJ parameters for a model compound can be adjusted to reproduce experimental pure solvent properties such as heat of vaporization, density, isothermal compressibility, heat capacity, heat of sublimation, lattice geometry, and free energy of aqueous solvation, as available. Although this is an effective method for the fine-tuning of the parameters, there are important issues. One is parameter correlation, such that LJ parameters for different atoms in a molecule and/or the magnitudes of ε_{ij} and $R_{\min}$ on the same atom, can compensate for individual unbalanced values, making it difficult to gauge whether they are balanced relative to one another [101]. To overcome this problem, a method has been developed to determine the relative value of the LJ parameters based on high level QM data [102] with the absolute values being based on scans of ε_{ij} and $R_{\min}$ that reproduce experimental data [103, 104]. This approach requires supramolecular interactions between rare gases and the model compound. Importantly, once satisfactory LJ parameters are obtained for atoms in a class of functional groups, they can often be directly transferred to other molecules carrying those functional groups without additional optimization.

Reproduction of experimental hydration free energies reflects how well the electrostatic and vdW parameters model interactions with bulk water. Recently, in the context of the polarizable Drude FF, it was shown that atom-pair-specific LJ parameters (termed "NBFix" in the context of CHARMM) needed to be used in order to minimize discrepancies between calculated and experimental hydration free energies while simultaneously reproducing pure solvent heats of vaporization and molecular volumes [105].

Optimization of internal parameters is usually done relative to target data that include geometries, vibrational spectra and conformational energies. Geometries are typically optimized at the MP2/6-31G* level (or MP2/6-31 + G* in the case of anions), and vibrational spectra are obtained at the MP2/6-31G* level. Frequencies are scaled using correction factors prescribed by Radom and coworkers [106], and a symbolic potential energy distribution (PED) analysis is performed as proposed by Pulay et al. [107] using the MOLVIB module in CHARMM. This approach has been shown to yield good agreement with experimental geometries for model compounds of complex systems such as proteins, nucleic acid bases, and sugars [22, 108, 109], while being computationally feasible.

4.2 Parametrization of the Polypeptide Backbone in the Drude Force Field

Parameterization of the polypeptide backbone was initially assumed to follow the general rules in use for CHARMM FFs, namely that parameters would be transferable from smaller model compounds. The prototype of the protein backbone, for all residues except glycine and proline, was based on alanine polypeptides. The initial electrostatic model, identified as Drude-NMA, was derived from a combination of electrostatic parameters that included *N*-methyl acetamide (NMA) and ethane, and LJ parameters were also transferred from NMA and ethane. Several rounds of optimization were previously done on NMA: initial parameters were published by Harder et al. [69] and a final set by Lin et al. [110] In the latest model LJ parameters were selected to give acceptable intramolecular hydrogen bond distances in α-helix conformations of alanine polypeptides in addition to allowing reproduction of NMA experimental condensed phase properties [110]. CMAP corrections for alanine dipeptide were also used to allow the (ϕ, ψ) Ramachandran map to reproduce a high-level QM (RIMP2/CBS//RIMP2/cc-pVDZ) surface, where the CBS (complete basis set) extrapolation was obtained from RIMP2/cc-pVTZ and RIMP2/cc-pVQZ single point energies following the prescription of Halkier et al. [111]. The Drude-NMA model was tested by calculating gas phase molecular properties of alanine dipeptide and $(\text{Ala})_5$ in different

Table 1
Gas phase dipole moments of alanine dipeptide and (Ala)$_5$[a], molecular polarizability of alanine dipeptide, and relative energies of (Ala)$_5$

Molecular dipole moment of alanine dipeptide (Debye)								
	αR				C5			
	QM[b]	Drude-NMA	Drude-ALA	Drude-2013	QM	Drude-NMA	Drude-ALA	Drude-2013
M	6.2	5.0	6.4	6.7	4.7	5.8	2.3	2.6
μ_x	1.3	0.1	3.1	3.0	−4.4	−5.6	−1.8	−2.3
μ_y	−1.6	−0.9	−1.7	−1.5	−1.0	−0.9	−0.6	−0.3
μ_z	5.9	4.9	5.4	5.8	1.2	0.9	1.3	1.3
Molecular dipole moment of (Ala)$_5$ (Debye)								
	αR				C5			
	QM[c]	Drude-NMA	Drude-ALA	Drude-2013	QM	Drude-NMA	Drude-ALA	Drude-2013
M	22.0	13.5	22.4	20.8	11.6	24.4	4.5	9.3
Molecular polarizability of alanine dipeptide (Å^3)								
	αR				C5			
	QM[b]	Drude-NMA	Drude-ALA	Drude-2013	QM	Drude-NMA	Drude-ALA	Drude-2013
A_{xx}	13.57	13.40	16.18	15.30	15.49	16.02	19.89	16.07
A_{yy}	12.72	12.60	14.29	14.36	12.06	11.87	13.39	12.78
A_{zz}	11.71	11.03	12.68	9.94	10.35	9.78	11.05	10.39
Relative energies of (Ala)$_5$ (kcal/mol)								
	QM[d]		Drude-NMA		Drude-ALA		Drude-2013	
αR-C5	−6.59		6.21		5.31		−3.89	
αR-PPII	−14.83		−5.77		0.42		−10.17	

[a] (Ala)$_5$ is acetyl-(Ala)$_5$-*N*-methylamide

[b] QM dipole moments and polarizabilities of alanine dipeptide obtained at the B3LYP/aug-cc-pVDZ level with the polarizabilities scaled by 0.85

[c] QM dipole moments for (Ala)$_5$ obtained at the B3LYP/6-31G* level

[d] Single point energies were calculated at the RIMP2/cc-pVTZ//RIMP2/cc-pVDZ level

conformations, such as dipole moments, relative energies, and molecular polarizabilities, and through MD simulations of (Ala)$_5$ in solution [112]. Testing the behavior of (Ala)$_5$ in solution has become common practice in the validation of protein force fields, being used in the development of the C36 additive FF [20] and the AMOEBA polarizable FF [77] (*see* Subheading 5.1) for details.

As alluded to above, direct transfer of Drude-NMA parameters to polypeptides did not yield acceptably accurate results (Table 1).

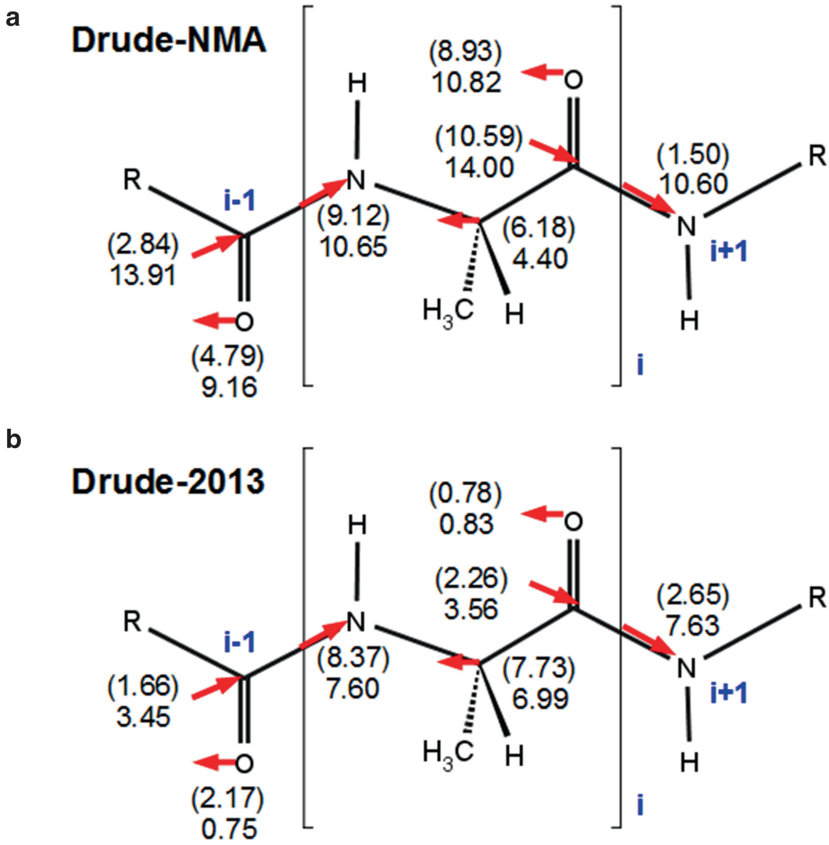


Fig. 1 Illustration of induced dipoles on dipeptide moieties of alanine dipeptide and $(Ala)_5$. Values in *parenthesis* are for alanine dipeptide

Using transferred parameters, the agreement of the computed dipole moments, polarizabilities, and relative energies with target values was poor, in particular for the extended conformations. Tests of NMR *J*-coupling also indicated poor agreement with experiment due to a (ϕ, ψ) distribution that predominantly populated extended C5 conformations.

Included in Fig. 1 are representative orientations and magnitudes of the induced dipoles and separations (pm) of the Drude particle and main atom in a dipeptide section of the alanine dipeptide (values in parenthesis) and $(Ala)_5$. All calculations were done enforcing the C5 conformation for both systems. The separation between the Drude particle and the main atom is a direct measurement of the magnitude of the induced dipole. Using Drude-NMA electrostatic parameters (Fig. 1a), displacement of the Drude particles in $(Ala)_5$ relative to the parent atom are similar on all carbonyl C (labeled C_{i-1}) atoms (~14 pm), and are substantially larger than the displacement for C_{i-1} in alanine dipeptide. C_{i-1} in alanine dipeptide is bound to a methyl group and the polarizing field is much weaker than in the longer polypeptide where C_{i-1} feels the electric field originating from the same amino acid's NH group. The case is similar for the N atoms, with N_{i+1} showing a much stronger induced local dipole in $(Ala)_5$ as compared to the alanine dipeptide. The induced dipole on Cα is smaller on $(Ala)_5$, enhancing the dipole interaction between N_i and C_i. This results in two effects. First, local dipoles associated with the peptide bonds interact with each other enhancing the local dipole moments associated with each peptide bond and, second, the larger dipole strengthens electrostatic interactions with water leading to overstabilization of the C5 conformation. Indeed, a comparison of the dipole moments of acetyl-$(Ala)_5$-*N*-methylamide for the NMA based model with QM data indicated the overall dipole moment of the C5 conformation to be significantly overestimated (Table 1). It was, therefore, hypothesized that the overestimation, which would lead to even more favorable interactions with aqueous solvent, was due to the electrostatic parameter optimization procedure based on NMA alone not defining balanced electrostatic interactions between the individual peptide bonds. Based on this analysis it was concluded that use of larger model compounds allowing communication between adjacent peptide bonds was required in the determination of electrostatic parameters, with the initial candidate being the alanine dipeptide.

Electrostatic parameters based on the alanine dipeptide were determined by averaging the components over five independent sets of parameters obtained from electrostatic potential (ESP) fitting corresponding to the αR, αL, C5, PPII and C7eq conformations. This model is referred to as Drude-ALA in the text below. For each conformation the electrostatic parameter optimization, which included the partial atomic charges, atomic polarizabilities, and atom-based Thole factors, was performed using the FITCHARGE module of CHARMM by fitting to the QM ESP maps as described above. The outcome is electrostatic parameters that better reproduce the change in the ESP associated with electrostatic interactions between the peptides bonds in the different relative orientations. The resulting Drude-ALA model yielded a smaller dipole moment for the C5 conformation for acetyl-$(Ala)_5$-*N*-methylamide (Table 1). Simulations of $(Ala)_5$ in aqueous solution were also performed and compared to Drude-NMA, and while the Drude-ALA model showed improved agreement with experiment, the agreement was still poor as compared to the additive C36 FF. It was found that the PPII region started to be populated, though the C5 conformation still dominated, indicating that the inclusion of electrostatic interactions between the peptide bonds during parameter optimization did improve the quality of the FF. However, those improvements were clearly insufficient, indicating that different target data were needed to obtain a more accurate electrostatic model for the polypeptide backbone.

The inability of Drude-ALA electrostatics to provide a reasonable description of properties of alanine polypeptides in gas-phase and solution prompted development of a third parametrization strategy. The rationale of the new methodology has its roots in the fundamental physics of the Drude model. In the presence of an electric field the position of the Drude particles are optimized while the main atom remains fixed following the Born–Oppenheimer principle. This creates on each atom a local dipole that, although small, is able to interact with neighboring dipoles. The magnitude of each dipole can be controlled by two factors: (1) the atomic polarizability, and (2) the damping of 1–2 and 1–3 interactions through the individual Thole factors. Thus, for polymeric structures, control of the behavior of each dipole is extremely complex. Boundaries cannot be explicitly imposed locally for each atom or groups of atoms as in other polarizable methods since the overall properties are the result of many cross interactions spanning wide regions of the system. As a consequence, an effective methodology of parametrization needed to include enough information on the whole molecule, thus allowing for a balanced set of electrostatic parameters. Furthermore, it was necessary to include information not only in gas phase but also from interactions with water molecules, since water is the preferred medium where most of the MD simulations with Drude oscillators are anticipated to take place. The third optimization method of backbone electrostatic parameters used a Simulated Annealing (SA) protocol [113], yielding the final model, Drude-2013. The target data consisted of an array of QM observables determined for the alanine dipeptide and larger alanine polypeptides. Target data included the polarizability of the alanine dipeptide, relative energies of $(\text{Ala})_5$, dipole moments of alanine dipeptide and $(\text{Ala})_5$, and energetic and structural data for the interaction of the alanine dipeptide with individual water molecules along specific directions. Several conformations of the alanine models were used: αR, C5, and PPII for the relative energies of $(\text{Ala})_5$; C5 and PPII for the interactions of the alanine dipeptide with water; and αR, C5, PPII, and C7eq conformations of the alanine dipeptide for molecular polarizabilities and dipole moments. In addition to the electrostatic parameters, during the SA internal parameters were allowed to vary within a limited range to keep the alanine dipeptide optimized geometries close to the targeted values. SA started with a temperature of 150 K with individual parameters randomly adjusted followed by accepting or rejecting the new parameter set based on the Metropolis criterion, resulting in Monte Carlo Simulated Annealing (MCSA) [114]. The temperature was gradually reduced to near 0 K yielding a selected parameter set for testing in $(\text{Ala})_5$ solution simulations. The error function was the weighted sum of all differences between MM and QM data for all properties mentioned above with various weighting factors. During MCSA fitting, a new CMAP that reproduces the

QM alanine dipeptide (ϕ, ψ) RIMP2/CBS//MP2/6-311G(d,p) surface was generated at each iteration. In addition, empirical adjustments of the CMAP were added to the QM-based surface to improve agreement with conformational sampling of the peptide backbone in peptides and proteins, resulting in the final Drude-2013 model.

While both C36 and Drude-2013 (ϕ, ψ) surfaces have undergone some empirical adjustments, the underlying energy surfaces was based on quantum mechanics, and therefore the overall landscape of the surfaces is similar. Adjustments in the C36 CMAP, which was obtained at the LMP2/cc-pVQZ level, included local optimization of the helical and sheet regions to reproduce subtle features observed in crystallographic survey data [32] followed by subsequent shifting of the helical region to decrease the tendency for the C22/CMAP model to over-populate that conformation [20]. For the Drude-2013 model, the overall sheet region was lowered and the areas between the sheet and helical regions and from $\phi = -90$ to -180 and $\psi = -60$ to 105° were raised.

4.3 Side Chain χ1, χ2 Dihedral Parameter Optimization in the Drude Force Field

Different side chains impact the conformational distribution of the polypeptide backbone, as observed in experimental studies [115–117]. The peptide $(\text{Ala})_4$-X-$(\text{Ala})_4$ has been used before as a model system for χ1, χ2 parameter optimization [46], where X is the amino acid of interest and the backbone conformation is constrained to fully extended, C7eq, PPII, or αR conformations [41]. Those studies indicated that $(\text{Ala})_4$-X-$(\text{Ala})_4$ with either the C7eq or PPII backbone conformation yields aqueous phase conformational properties that mimic those occurring in full proteins. Based on this analysis, χ1, χ2 parameter optimization was performed by initially targeting QM data for the respective side chain dipeptides, with the backbone in the β, αR, and αL conformations. These parameters were then used in Hamiltonian Replica Exchange MD (H-REMD) simulations [118] of $(\text{Ala})_4$-X-$(\text{Ala})_4$ in solution, with χ1, χ2 sampling compared with PDB survey data. Overlap coefficients (OC) [41] for χ1 and χ2 distributions from $(\text{Ala})_4$-X-$(\text{Ala})_4$ in the C7eq conformation and those from a survey of the PDB [119] were computed, with an OC of 1 indicating exact agreement and an OC of 0 indicating no agreement. The extent of overlap for some of the amino acids based on optimization only targeting the QM data was found to be quite good. For example, values of 0.87, 0.88, and 0.87 were obtained for χ1 for Cys, Leu, and Val, respectively, while the OC was 0.92 for χ2 with Leu. Based on the quality of the fit for these residues, additional optimization was not performed. Additional optimization for the remaining residues involved comparison of the computed and target χ1 and χ2 populations of the *gauche+*, *gauche-*, and *trans* rotamers and manually adjusting the corresponding dihedral parameters to improve the level of agreement. After the optimization, significant agreement with the PDB target data was obtained for a number of amino acids, notable examples being Ile, Lys, and Thr. Overall, the final OC values are typically 0.7 or higher, though lower values were also found including Asn χ2, Asp χ1, Gln χ2, and Glu χ1. The final parameters were used for the reported polypeptide and protein simulations. In ref. 120 we present detailed descriptions of the optimization protocol and final results.

4.4 The AMOEBA Force Field and Parametrization of Proteins

Detailed methodology for deriving electrostatic parameters for AMOEBA to allow incorporation of novel molecules has been published [83], and therefore what follows is a brief overview. Determination of permanent atomic multipoles for glycine, alanine, and proline residues was done based on capped acetyl-*X*-*N*-methylamide dipeptides with *X* = Gly, Ala, and Pro. The first step is definition of intramolecular direct polarization groups, which is important because atoms belonging to one group can only polarize atoms outside that group. The group definitions for alanine dipeptide are show in Fig. 2 of ref. 77. For side chains, groups are also selected. The optimization proceeds with assignment of the initial multipole parameters from Distributed Multipole Analysis (DMA) at the MP2/6-311G** level. Initial parameters are then iteratively optimized against the MP2/aug-cc-pVTZ electrostatic potential computed on a set of grid points around the dipeptide compounds. Converged Permanent Atomic Multipoles (PAMs) were determined simultaneously for five local minima: αL, α′, C5, C7a, and C7e conformers.

5 Application of Polarizable Force Fields to Protein Simulations

The year 2013 marked important milestones in the development of polarizable FFs. After years of development, polarizable FFs for peptides and proteins suitable for MD simulations based on classical Drude oscillators (Drude-2013) and the AMOEBA model (AMOEBA-2013) were published. Here, we summarize results of MD simulations with the two FFs.

5.1 Peptide Simulations with C36 Additive, AMOEBA-2013, and Drude-2013 Force Fields

With advances in computing capacity, it has become common to use simulations of oligopeptides in solution to calibrate FF torsional parameters [20, 37, 45, 112, 121–123], since results can be directly compared to experimental nuclear magnetic resonance (NMR) data for corresponding peptides. Conformational distributions in an NMR experiment are reflected in NMR-derived spin–spin coupling (*J*-coupling) constants. Using Karplus relations, *J*-coupling values can be computed from peptide conformations from MD simulations, and the ability to achieve ever-increasing timescales via MD allows for the computational generation of conformational ensembles of a size that can be meaningfully compared with experiment [37, 124].

As an example, simulations of small polypeptides of $(Ala)_3$, $(Ala)_5$, $(Ala)_7$, $(Val)_3$, and $(Gly)_3$ were used by Best et al. to validate the improved CHARMM36 additive FF (C36) [20]. Using this approach, C36 introduced small but significant changes relative to its predecessor, C22/CMAP. In alanine- and valine-based peptides, minima occur at PPII, with C5 and αR being only slightly higher in energy. The additional minima at αL and C7ax are approximately 2–3 kcal/mol higher than the PPII conformation. And while there is only a small difference between $(Ala)_3$, $(Ala)_5$, and $(Ala)_7$, sampling for $(Val)_3$ was significantly different because of the presence of the bulky hydrophobic side chain. Compared with other FFs, AMBER ff99SB9 and ff99SB* are closest to C36, while OPLS/AA [125] is qualitatively different with a minimum at C7eq and Gromos 53a6 FF [126] has two minima near αR and a low-energy transition region between αR and C7ax.

C36 (ϕ, ψ) sampling has also been compared with experimental NMR *J*-coupling. Agreement was very good for the alanine-based peptides and for $(Gly)_3$, and reasonable for $(Val)_3$. The new C36 FF significantly improves over the previous C22/CMAP FF, with improvement coming from decreased sampling of αR conformations and increased sampling of PPII, which is reflected in the *J*-couplings. In ref. 20 C36 was also compared with other FFs outside the CHARMM family (AMBER ff99SB [45], OPLS/AA [125], Gromos 53a6 [126]), showing significantly lower χ^2 values. No direct comparison of C36 with the latest improved Amber FFs has been published, although it is anticipated that C36 will compare very favorably to experimental data based on published results.

Sampling for the unblocked, protonated $(Ala)_5$ peptide has been tested using the AMOEBA-2013 FF [77]. Sampling is similar to C36, with a distinct global minimum located around the PII conformation and two other basins approximately 0.5 kcal/mol higher in free energy in the β-sheet and α-helix regions. Barriers separating the global basin from the two local minima are 1–2 kcal/mol. (ϕ, ψ) sampling was compared with experimental *J*-coupling constants, and values from MD simulations are in excellent agreement with those probed by experiment with a χ^2 value of 1.0

Simulations of $(Ala)_5$ polypeptides were not used to validate the newly developed Drude polarizable FF in CHARMM but rather were explicitly part of the optimization process, particularly for the fine-tuning of the CMAP potential so as to yield acceptable sampling patterns in the tested protein systems. In addition, the GB1 hairpin [41–56], [127, 128] and a dimeric coiled coil (1UOI) [129] were also used as target data for optimization of the Drude-2013 model. Due to the use of multiple small peptides, as well as larger proteins, as target data, sampling of $(Ala)_5$ had to be slightly compromised, yielding χ^2 values larger than 1.0.

Explicit solvent simulations of the GB1 hairpin of 100 ns yielded RMS differences with the Drude model more similar to the crystal structure of the full GB1 protein as compared to the C36 where the RMS difference fluctuated between 2.5 and 3 Å, indicating drift away from the crystal structure. With the dimeric coiled coil (1UOI) [129] RMS analysis showed the overall structure of the coiled coil to deviate more from the crystal structure with the Drude model as compared to C36 (*see* ref. 120 for details). The individual helices in the dimer move relative to each other, while the conformations of the individual helices are well preserved, suggesting the ability of the Drude-2013 model to properly treat the helical secondary structure of the individual monomers. (ϕ, ψ) probability distributions from the simulations supported this conclusion. Thus, the Drude model satisfactorily reproduces the conformational properties of small peptides on the 100 ns time scale, though longer simulations will be required to more rigorously challenge the model.

5.2 Full Proteins

Further validation of the Drude-2013 force field involved explicit solvent MD simulations on ten proteins: 1EJG (crambin), 1P7E (protein GB1 domain), 1MJC (cold-shock protein A), 1UBQ (ubiquitin), 3ZZP (circular permutant of ribosomal protein), 4IEJ (DNA methyl transferase associated protein), 135L (lysozyme), 1IFC (fatty acid binding protein), 3VQF (PDZ domain from tight junction regulatory protein), and 1BYI (dethiobiotin synthase). The proteins are relatively small, typically less than 100 residues, and cover a range of secondary structures.

The stability of each protein was characterized by the value of its backbone RMS deviation (RMSD) relative to the crystal structure. The results summarized in Table 7 and Fig. S2 of the Supplementary Information of ref. 120 showed the RMS differences are typically larger with the Drude model versus C36 additive force field as are the RMS fluctuations. The Drude-2013 model shows additional flexibility compared to the additive model with only one exception, namely ubiquitin (1UBQ). While the Drude model generally appears to have more flexibility than the additive C36 model, NMR analysis indicated that for selected residues with high mobility in C36, the Drude model gave improved agreement with experiment, as shown in Fig. 7 of Lopes et al. [120].

Results with AMOEBA-2013 protein FF [77] have been reported for three of the proteins studied with the Drude-2013 force field. These include crambin (1EJG), ubiquitin (1UBQ), and lysozyme (135L). AMOEBA MD simulations were performed for 30 ns yielding backbone RMSDs in the vicinity of 1 Å, 2 Å, and 2 Å for the three proteins, respectively. At 30 ns of the Drude simulations the corresponding values were 1.1/1.1, 1.9, and 1.9/2.2 Å, where two values are from duplicate simulations.

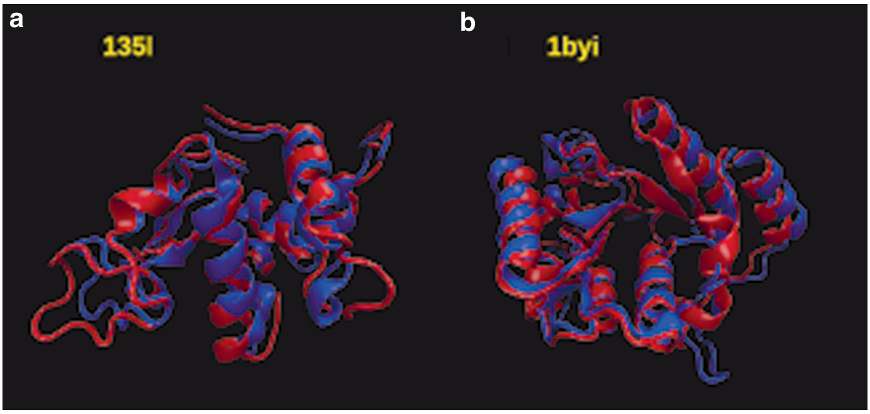


Fig. 2 100-ns snapshots from Drude-2013 simulations (*red*) of lysozyme (135L) and dethiobiotin synthase 1BYI superimposed on the starting crystallographic structures (*blue*)

Although the Drude model showed additional flexibility over the additive C36 model, the overall structures of the proteins are well maintained. Snapshots taken at 100 ns for lysozyme (135L) and dethiobiotin synthase (1BYI) superimposed on the corresponding crystal structures are shown in Fig. 2, showing the overall maintenance of the structures. Consistent with this was the (ϕ, ψ) sampling with the Drude model over all the simulated proteins being similar to a survey of the PDB as well as sampling occurring with C36 (Fig. 6 of Lopes et al.). In addition, the N–H…O=C distance distributions in secondary structures were reasonably reproduced by the Drude model, though there is a tendency towards the distances being slightly longer than distributions from PDB crystal structures (Fig. 5 of Lopes et al.).

Additional analysis involved dipole moments of selected moieties during the MD simulations. These included the peptide bonds in the GB1 hairpin and ubiquitin and tryptophan residues in lysozyme (Fig. 8 of Lopes et al.). In all cases the Drude dipole moments are systematically larger than with the additive model. This indicates that, while partial atomic charges in the additive model are adjusted to overestimate molecular dipole moments, the extent of overestimation is not enough for the protein environment. In addition, the dipole moments of the peptide bonds with the Drude model in sheets are systematically larger than in helices. Finally, significant variations in the dipole moments were observed in the Drude simulations (e.g., >1.5 D for a Trp in lysozyme). Thus, even though the additive models were optimized to yield enhanced dipole moments appropriate for the condensed phase, it does not appear that the overestimation was sufficient based on these initial polarizable calculations. That, as well as the large variation in the dipoles occurring in the Drude model, suggests that the underlying physical forces dictating the overall properties of the peptides and proteins are significantly different in the Drude versus the additive model.

Indeed, the additional flexibility in the Drude model may be due to the inclusion of electronic polarization in the model, allowing for the variability of the local molecular dipoles.

6 Summary

The field of empirical FF based simulations of proteins continues to develop. Since the last publication of a similar review great progress has been made, including the publication of two polarizable force fields for proteins as well as improvements in the AMBER and CHARMM additive protein force fields. Work on other classes of biopolymers has also made significant progress allowing for simulations of heterogeneous systems. As other researchers start using the recently published force fields, in particular the polarizable force fields, limitations will certainly be found and corrections and improvements are expected.

As was emphasized in this review, development of electrostatic parameters in the Drude force field is very complex. It is expected that new optimization algorithms together with more sophisticated target data will lead to significant progress. Polarizable models for other classes of biomolecules based on the Drude oscillator will be published soon for DNA and carbohydrates as well as a wider range of lipids.

While polarizable MD simulations will make a significant contribution to our understanding of protein structure and function it should be emphasized that these models are more sensitive to initial conditions than with an additive FF, and can have polarization catastrophes that will cause simulations to fail. To overcome this it is suggested that systems initially be set up and equilibrated with an additive FF and then converted to the polarizable model. To facilitate this procedure the CHARMM-GUI [130] has been extended to include a new utility, the "Drude Prepper." The Drude Prepper reads equilibrated CHARMM PSF and coordinate files and converts them to Drude format files. This includes the production of inputs for MD simulations using CHARMM or NAMD. This utility will greatly facilitate the application of the Drude model to a range of proteins as well as other systems.

Concerning computational efficiency, the Drude model typically requires the use of a 1 fs integration time step during MD simulations. In addition, there is an approximately twofold overhead associated with the calculation of the polarization contribution to the electrostatics. Thus, the model is approximately fourfold slower than corresponding additive simulations performed with a 2 fs integration time step. However, the NAMD implementation is highly parallelizable [59], which will facilitate simulations of large systems using the Drude model.

Acknowledgement

Financial support from the NIH (GM072558) and computational support from the University of Maryland Computer-Aided Drug Design Center, and the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575, are acknowledged.

References

1. MacKerell AD (2004) Empirical force fields for biological macromolecules: overview and issues. J Comput Chem 25(13):1584–1604
2. Stone AJ (2008) Intermolecular potentials. Science 321(5890):787–789
3. Freddolino PL, Harrison CB, Liu YX, Schulten K (2010) Challenges in protein-folding simulations. Nat Phys 6(10):751–758
4. Warshel A, Kato M, Pisliakov AV (2007) Polarizable force fields: history, test cases, and prospects. J Chem Theory Comput 3(6):2034–2045
5. Lopes PEM, Roux B, MacKerell AD (2009) Molecular modeling and dynamics studies with explicit inclusion of electronic polarizability: theory and applications. Theor Chem Acc 124(1–2):11–28
6. Zhu X, Lopes PEM, MacKerell AD (2012) Recent developments and applications of the CHARMM force fields. Wiley Interdiscip Rev Comput Mol Sci 2(1):167–185
7. Guvench O, MacKerell AD (2008) Comparison of protein force fields for molecular dynamics simulations. In: Kukol A (ed) Molecular modeling of proteins. Humana Press, Totowa, NJ, pp 63–88
8. Lopes PEM, Harder E, Roux B, MacKerell AD (2009) Formalisms for the explicit inclusion of electronic polarizability in molecular modeling and dynamics studies. In: York DM, Lee T-S (eds) Multi-scale quantum models for biocatalysis. Springer, Netherlands, pp 219–257
9. Salomon-Ferrer R, Case DA, Walker RC (2013) An overview of the Amber biomolecular simulation package. Wiley Interdiscip Rev Comput Mol Sci 3(2):198–210
10. Beauchamp K, Lin Y-S, Das R, Pande V (2012) Are protein force fields getting better? A systematic benchmark on 524 diverse NMR measurements. J Chem Theory Comput 8(4): 1409–1414
11. Burkert U, Allinger N (1982) Molecular mechanics. American Chemical Society, Washington, DC
12. McCammon JA, Harvey SC (1987) Dynamics of proteins and nucleic acids. Cambridge University Press, New York
13. Leach AR (2001) Molecular modelling: principles and applications. Prentice Hall, Harlow, England
14. Becker OM (2001) Computational biochemistry and biophysics. M. Dekker, New York
15. Rapaport DC (2004) The art of molecular dynamics simulation. Cambridge University Press, Cambridge, UK
16. Schlick T (2002) Molecular modeling and simulation: an interdisciplinary guide. Springer, New York
17. Satoh A. Introduction to practice of molecular simulation molecular dynamics, Monte Carlo, Brownian dynamics, Lattice Boltzmann, dissipative particle dynamics. http://site.ebrary.com/id/10440534
18. Brooks BR, Bruccoleri RE, Olafson BD, States DJ, Swaminathan S, Karplus M (1983) CHARMM: a program for macromolecular energy, minimization, and dynamics calculations. J Comput Chem 4(2):187–217
19. MacKerell AD, Bashford D, Bellott M, Dunbrack RL, Evanseck JD, Field MJ et al (1998) All-atom empirical potential for molecular modeling and dynamics studies of proteins. J Phys Chem B 102(18):3586–3616
20. Best RB, Zhu X, Shim J, Lopes PEM, Mittal J, Feig M et al (2012) Optimization of the additive CHARMM all-atom protein force field targeting improved sampling of the backbone ϕ, ψ and side-chain χ_1 and χ_2 dihedral angles. J Chem Theory Comput 8(9):3257–3273
21. MacKerell AD, Wiorkiewicz-Kuczera J, Karplus M (1995) An all-atom empirical energy function for the simulation of nucleic acids. J Am Chem Soc 117(48):11946–11975
22. Foloppe N, MacKerell AD (2000) All-atom empirical force field for nucleic acids: I. Parameter optimization based on small molecule and condensed phase macromolecular target data. J Comput Chem 21(2):86–104

23. MacKerell AD, Banavali NK (2000) All-atom empirical force field for nucleic acids: II. Application to molecular dynamics simulations of DNA and RNA in solution. J Comput Chem 21(2):105–120
24. Feller SE, MacKerell AD (2000) An improved empirical potential energy function for molecular simulations of phospholipids. J Phys Chem B 104(31):7510–7515
25. Feller SE, Gawrisch K, MacKerell AD (2001) Polyunsaturated fatty acids in lipid bilayers: intrinsic and environmental contributions to their unique physical properties. J Am Chem Soc 124(2):318–326
26. Klauda JB, Venable RM, Freites JA, O'Connor JW, Tobias DJ, Mondragon-Ramirez C et al (2010) Update of the CHARMM all-atom additive force field for lipids: validation on six lipid types. J Phys Chem B 114(23):7830–7843
27. Kuttel M, Brady JW, Naidoo KJ (2002) Carbohydrate solution simulations: producing a force field with experimentally consistent primary alcohol rotational frequencies and populations. J Comput Chem 23(13): 1236–1243
28. Guvench O, Greene SN, Kamath G, Brady JW, Venable RM, Pastor RW et al (2008) Additive empirical force field for hexopyranose monosaccharides. J Comput Chem 29(15): 2543–2564
29. Hatcher ER, Guvench O, MacKerell AD (2009) CHARMM additive all-atom force field for acyclic polyalcohols, acyclic carbohydrates, and inositol. J Chem Theory Comput 5(5):1315–1327
30. Guvench O, Hatcher E, Venable RM, Pastor RW, MacKerell AD (2009) CHARMM additive all-atom force field for glycosidic linkages between hexopyranoses. J Chem Theory Comput 5(9):2353–2370
31. Vanommeslaeghe K, Hatcher E, Acharya C, Kundu S, Zhong S, Shim J et al (2010) CHARMM general force field: a force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. J Comput Chem 31(4):671–690
32. MacKerell AD, Feig M, Brooks CL (2004) Extending the treatment of backbone energetics in protein force fields: limitations of gas-phase quantum mechanics in reproducing protein conformational distributions in molecular dynamics simulations. J Comput Chem 25(11):1400–1415
33. MacKerell AD, Feig M, Brooks CL (2004) Improved treatment of the protein backbone in empirical force fields. J Am Chem Soc 126(3):698–699
34. Freddolino PL, Schulten K (2009) Common structural transitions in explicit-solvent simulations of villin headpiece folding. Biophys J 97(8):2338–2347
35. Freddolino PL, Liu F, Gruebele M, Schulten K (2008) Ten-microsecond molecular dynamics simulation of a fast-folding WW domain. Biophys J 94(10):L75–L77
36. Freddolino PL, Park S, Roux B, Schulten K (2009) Force field bias in protein folding simulations. Biophys J 96(9):3772–3780
37. Best R, Buchete N-V, Hummer G (2008) Are current molecular dynamics force fields too helical? Biophys J 95(1):L07–L09
38. Best RB, Hummer G (2009) Optimized molecular dynamics force fields applied to the helix-coil transition of polypeptides. J Phys Chem B 113(26):9004–9015
39. Best RB, Mittal J (2010) Balance between α and β structures in ab initio protein folding. J Phys Chem B 114(26):8790–8798
40. Mittal J, Best RB (2010) Tackling force-field bias in protein folding simulations: folding of villin HP35 and Pin WW domains in explicit water. Biophys J 99(3):L26–L28
41. Shim J, Zhu X, Best RB, MacKerell AD (2013) Ala_4-X-Ala_4 as a model system for the optimization of the χ_1 and χ_2 amino acid side-chain dihedral empirical force field parameters. J Comput Chem 34(7):593–603
42. Vorobyov IV, Anisimov VM, MacKerell AD (2005) Polarizable empirical force field for alkanes based on the classical drude oscillator model. J Phys Chem B 109(40): 18988–18999
43. Mason PE, Neilson GW, Enderby JE, Saboungi ML, Dempsey CE, MacKerell AD et al (2004) The structure of aqueous guanidinium chloride solutions. J Am Chem Soc 126(37):11462–11470
44. Macias AT, MacKerell AD (2005) CH/pi interactions involving aromatic amino acids: refinement of the CHARMM tryptophan force field. J Comput Chem 26(14):1452–1463
45. Hornak V, Abel R, Okur A, Strockbine B, Roitberg A, Simmerling C (2006) Comparison of multiple Amber force fields and development of improved protein backbone parameters. Proteins 65(3):712–725
46. Lindorff-Larsen K, Piana S, Palmo K, Maragakis P, Klepeis J, Dror R et al (2010) Improved side-chain torsion potentials for the Amber ff99SB protein force field. Proteins 78(8):1950–1958
47. Li D-W, Bruschweiler R (2011) NMR-based protein potentials. Angew Chem 122(38): 6930–6932

48. Nerenberg P, Head-Gordon T (2011) Optimizing protein–solvent force fields to reproduce intrinsic conformational preferences of model peptides. J Chem Theory Comput 7(4):1220–1230

49. Perez A, Marchan I, Svozil D, Sponer J, Cheatham TE, Laughton CA et al (2007) Refinenement of the AMBER force field for nucleic acids: improving the description of alpha/gamma conformers. Biophys J 92(11): 3817–3829

50. Joung IS, Cheatham TE (2008) Determination of alkali and halide monovalent ion parameters for use in explicitly solvated biomolecular simulations. J Phys Chem B 112(30):9020–9041

51. Joung IS, Cheatham TE (2009) Molecular dynamics simulations of the dynamic and energetic properties of alkali and halide ions using water-model-specific ion parameters. J Phys Chem B 113(40):13279–13290

52. Banas P, Hollas D, Zgarbova M, Jurecka P, Orozco M, Cheatham TE III et al (2010) Performance of molecular mechanics force fields for RNA simulations: stability of UUCG and GNRA hairpins. J Chem Theory Comput 6(12):3836–3849

53. Zgarbova M, Otyepka M, Sponer J, Mladek A, Banas P, Cheatham TE III et al (2011) Refinement of the Cornell et al. Nucleic acids force field based on reference quantum chemical calculations of glycosidic torsion profiles. J Chem Theory Comput 7(9):2886–2902

54. Kirschner KN, Woods RJ (2001) Solvent interactions determine carbohydrate conformation. Proc Natl Acad Sci U S A 98(19): 10541–10545

55. Woods RJ, Dwek RA, Edge CJ, Fraser-Reid B (1995) Molecular mechanical and molecular dynamic simulations of glycoproteins and oligosaccharides. 1. GLYCAM_93 parameter development. J Phys Chem 99(11):3832–3846

56. Kirschner KN, Yongye AB, Tschampel SM, González-Outeiriño J, Daniels CR, Foley BL et al (2008) GLYCAM06: a generalizable biomolecular force field. Carbohydrates. J Comput Chem 29(4):622–655

57. Skjevik ÃGA, Madej BD, Walker RC, Teigen K (2012) LIPID11: a modular framework for lipid simulations using Amber. J Phys Chem B 116(36):11124–11136

58. Brooks BR, Brooks CL III, MacKerell AD Jr, Nilsson L, Petrella RJ, Roux B et al (2009) CHARMM: the biomolecular simulation program. J Comput Chem 30(10):1545–1614

59. Jiang W, Hardy DJ, Phillips JC, Mackerell AD Jr, Schulten K, Roux B (2011) High-performance scalable molecular dynamics simulations of a polarizable force field based on classical Drude oscillators in NAMD. J Phys Chem Lett 2(2):87–92

60. Boulanger E, Thiel W (2012) Solvent boundary potentials for hybrid QM/MM computations using classical drude oscillators: a fully polarizable model. J Chem Theory Comput 8:4527–4538

61. Eastman P, Friedrichs MS, Chodera JD, Radmer RJ, Bruns CM, Ku JP et al (2012) OpenMM 4: a reusable, extensible, hardware independent library for high performance molecular simulation. J Chem Theory Comput 8:461–469

62. Lamoureux G, Roux B (2003) Modelling induced polarizability with drude oscillators: theory and molecular dynamics simulation algorithm. J Chem Phys 119:5185–5197

63. Lamoureux G, MacKerell AD, Roux B (2003) A simple polarizable model of water based on classical Drude oscillators. J Chem Phys 119(10):5185–5197

64. Lamoureux G, Harder E, Vorobyov IV, Roux B, MacKerell AD (2006) A polarizable model of water for molecular dynamics simulations of biomolecules. Chem Phys Lett 418(1–3): 245–249

65. Anisimov VM, Lamoureux G, Vorobyov IV, Huang N, Roux B, MacKerell AD (2005) Determination of electrostatic parameters for a polarizable force field based on the classical Drude oscillator. J Chem Theory Comput 1(1):153–168

66. Anisimov VM, Vorobyov IV, Lamoureux G, Noskov S, Roux B, MacKerell AD (2004) CHARMM all-atom polarizable force field parameter development for nucleic acids. Biophys J 86(1):415A

67. Anisimov VM, Vorobyov IV, Roux B, MacKerell AD (2007) Polarizable empirical force field for the primary and secondary alcohol series based on the classical drude model. J Chem Theory Comput 3(6):1927–1946

68. Lopes PEM, Lamoureux G, Roux B, MacKerell AD (2007) Polarizable empirical force field for aromatic compounds based on the classical drude oscillator. J Phys Chem B 111(11):2873–2885

69. Harder E, Anisimov VM, Whitfield TW, MacKerell AD, Roux B (2008) Understanding the dielectric properties of liquid amides from a polarizable force field. J Phys Chem B 112(11):3509–3521

70. Baker CM, MacKerell AD (2010) Polarizability rescaling and atom-based Thole scaling in the CHARMM Drude polarizable force field for ethers. J Mol Model 16(3): 567–576

71. Vorobyov I, Anisimov VM, Greene S, Venable RM, Moser A, Pastor RW et al (2007) Additive and classical drude polarizable force fields for linear and cyclic ethers. J Chem Theory Comput 3(3):1120–1133

72. Zhu X, MacKerell AD (2010) Polarizable empirical force field for sulfur-containing compounds based on the classical drude oscillator model. J Comput Chem 31(12):2330–2341

73. Baker CM, Anisimov VM, MacKerell AD (2011) Development of CHARMM polarizable force field for nucleic acid bases based on the classical drude oscillator model. J Phys Chem B 115(3):580–596

74. He X, Lopes PEM, MacKerell AD (2013) Polarizable empirical force field for acyclic polyalcohols based on the classical drude oscillator. Biopolymers 99(10):724–738

75. Harder E, MacKerell AD, Roux B (2009) Many-body polarization effects and the membrane dipole potential. J Am Chem Soc 131(8):2760–2761

76. Chowdhary J, Harder E, Lopes PEM, Huang L, MacKerell AD, Roux B (2013) A polarizable force field of dipalmitoylphosphatidylcholine based on the classical drude model for molecular dynamics simulations of lipids. J Phys Chem B 117(31):9142–9160

77. Shi Y, Xia Z, Zhang JH, Best RB, Wu C, Ponder JW et al (2013) Polarizable atomic multipole-based AMOEBA force field for proteins. J Chem Theory Comput 9(9):4046–4063

78. Dudek MJ, Ponder JW (1995) Accurate modeling of the intramolecular electrostatic energy of proteins. J Comput Chem 16(7): 791–816

79. Thole B (1981) Molecular polarizabilities calculated with a modified dipole interaction. Chem Phys 59(3):341–350

80. Ren PY, Ponder JW (2003) Polarizable atomic multipole water model for molecular mechanics simulation. J Phys Chem B 107(24): 5933–5947

81. Ren PY, Ponder JW (2004) Temperature and pressure dependence of the AMOEBA water model. J Phys Chem B 108(35):13427–13437

82. Grossfield A, Ren PY, Ponder JW (2003) Ion solvation thermodynamics from simulation with a polarizable force field. J Am Chem Soc 125(50):15671–15682

83. Ren P, Wu C, Ponder JW (2011) Polarizable atomic multipole-based molecular mechanics for organic molecules. J Chem Theory Comput 7(10):3143–3161

84. Shi Y, Wu C, Ponder JW, Ren P (2011) Multipole electrostatics in hydration free energy calculations. J Comput Chem 32(5): 967–977

85. Ponder JW, Case DA (2003) Force fields for protein simulations, Protein simulations. Academic, San Diego, pp 27–85

86. Ponder JW, Wu C, Ren P, Pande VS, Chodera JD, Schnieders MJ et al (2010) Current status of the AMOEBA polarizable force field. J Phys Chem B 114(8):2549–2564

87. Ren PY, Ponder JW (2002) Consistent treatment of inter- and intramolecular polarization in molecular mechanics calculations. J Comput Chem 23(16):1497–1506

88. Jorgensen WL, Tirado-Rives J (1988) The OPLS potential function for proteins. Energy minimizations for crystals of cyclic peptides and crambin. J Am Chem Soc 110:1657–1666

89. Singh UC, Kollman PA (1984) An approach to computing electrostatic charges for molecules. J Comput Chem 5(2):129–145

90. Chirlian LE, Francl MM (1987) Atomic charges derived from electrostatic potentials: a detailed study. J Comput Chem 8(6): 894–905

91. Merz KM (1992) Analysis of a large data base of electrostatic potential derived atomic charges. J Comput Chem 13(6):749–767

92. Bayly CI, Cieplak P, Cornell WD, Kollman PA (1993) A well-behaved electrostatic potential based method using charge restraints for deriving atomic charges: the RESP model. J Phys Chem 97(40):10269–10280

93. Francl M, Carey C, Chirlian L, Gange D (1996) Charges fit to electrostatic potentials. II. Can atomic charges be unambiguously fit to electrostatic potentials? J Comput Chem 17(3):367–383

94. Lopes PEM, Lamoureux G, Mackerell AD (2009) Polarizable empirical force field for nitrogen-containing heteroaromatic compounds based on the classical Drude oscillator. J Comput Chem 30(12):1821–1838

95. Harder E, Anisimov VM, Vorobyov IV, Lopes PEM, Noskov SY, MacKerell AD et al (2006) Atomic level anisotropy in the electrostatic modeling of lone pairs for a polarizable force field based on the classical Drude oscillator. J Chem Theory Comput 2(6):1587–1597

96. Miller KJ (1990) Additivity methods in molecular polarizability. J Am Chem Soc 112(23): 8533–8542

97. Baker CM, MacKerell AD (2009) Polarizability rescaling and atom-based Thole scaling in the CHARMM Drude polarizable force field for ethers. J Mol Model 16(3): 567–576

98. Yu HA, Whitfield TW, Harder E, Lamoureux G, Vorobyov I, Anisimov VM et al (2010) Simulating monovalent and divalent ions in aqueous solution using a drude polarizable force field. J Chem Theory Comput 6(3): 774–786

99. Jorgensen WL, Madura JD, Swenson CJ (1984) Optimized intermolecular potential functions for liquid hydrocarbons. J Am Chem Soc 106(22):6638–6646

100. Jorgensen WL (1986) Optimized intermolecular potential functions for liquid alcohols. J Phys Chem 90(7):1276–1284

101. MacKerell AD (2001) Atomistic models and force fields. In: Becker O et al (eds) Computational biochemistry and biophysics. Marcel Dekker, Inc., New York, pp 7–38

102. Yin D, MacKerell AD (1996) Ab initio calculations on the use of helium and neon as probes of the van der Waals surfaces of molecules. J Phys Chem 100(7):2588–2596

103. Yin DX, MacKerell AD (1998) Combined ab initio empirical approach for optimization of Lennard-Jones parameters. J Comput Chem 19(3):334–348

104. Chen IJ, Yin D, MacKerell AD (2002) Combined ab initio/empirical approach for optimization of Lennard-Jones parameters for polar-neutral compounds. J Comput Chem 23(2):199–213

105. Baker CM, Lopes PEM, Zhu X, Roux B, MacKerell AD (2010) Accurate calculation of hydration free energies using pair-specific Lennard-Jones parameters in the CHARMM drude polarizable force field. J Chem Theory Comput 6(4):1181–1198

106. Scott AP, Radom L (1996) Harmonic vibrational frequencies: an evaluation of Hartree-Fock, Møller-Plesset, quadratic configuration interaction, density functional theory, and semiempirical scale factors. J Phys Chem 100(41):16502–16513

107. Pulay P, Fogarasi G, Pang F, Boggs JE (1979) Systematic ab initio gradient calculation of molecular geometries, force constants, and dipole moment derivatives. J Am Chem Soc 101(10):2550–2560

108. Foloppe N, Hartmann B, Nilsson L, MacKerell AD (2002) Intrinsic conformational energetics associated with the glycosyl torsion in DNA: a quantum mechanical study. Biophys J 82(3):1554–1569

109. Foloppe N, Nilsson L, MacKerell AD (2001) Ab initio conformational analysis of nucleic acid components: intrinsic energetic contributions to nucleic acid structure and dynamics. Biopolymers 61(1):61–76

110. Lin B, Lopes PEM, Roux B, MacKerell AD (2013) Kirkwood-Buff analysis of aqueous N-methylacetamide and acetamide solutions modeled by the CHARMM additive and Drude polarizable force fields. J Chem Phys 139(8):084509

111. Halkier A, Helgaker T, Jørgensen P, Klopper W, Koch H, Olsen J et al (1998) Basis-set convergence in correlated calculations on Ne, N_2, and H_2O. Chem Phys Lett 286(3–4):243–252

112. Graf J, Nguyen PH, Stock G, Schwalbe H (2007) Structure and dynamics of the homologous series of alanine peptides: a joint molecular dynamics/NMR study. J Am Chem Soc 129(5):1179–1189

113. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680

114. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. J Chem Phys 21(6):1087–1092

115. Shoemaker KR, Kim PS, York EJ, Stewart JM, Baldwin RL (1987) Tests of the helix dipole model for stabilization of α-helices. Nature 326(6113):563–567

116. Shoemaker KR, Kim PS, Brems DN, Marqusee S, York EJ, Chaiken IM et al (1985) Nature of the charged-group effect on the stability of the C-peptide helix. Proc Natl Acad Sci 82(8):2349–2353

117. Padmanabhan S, Marqusee S, Ridgeway T, Laue TM, Baldwin RL (1990) Relative helix-forming tendencies of nonpolar amino acids. Nature 344(6263):268–270

118. Fukunishi H, Watanabe O, Takada S (2002) On the Hamiltonian replica exchange method for efficient sampling of biomolecular systems: application to protein structure prediction. J Chem Phys 116(20):9058–9067

119. Zhu X, Lopes PEM, Shim J, MacKerell AD (2012) Intrinsic energy landscapes of amino acid side-chains. J Chem Inf Model 52(6): 1559–1572

120. Lopes PEM, Huang J, Shim J, Luo Y, Hui L, Roux B et al (2013) Polarizable force field for peptides and proteins based on the classical drude oscillator. J Chem Theory Comput. doi:10.1021/ct400781b

121. Hegefeld WA, Chen S-E, DeLeon KY, Kuczera K, Jas GS (2010) Helix formation in a pentapeptide: experiment and force-field dependent dynamics. J Phys Chem A 114(47): 12391–12402

122. Best RB, Mittal J, Feig M, MacKerell AD (2012) Inclusion of many-body effects in the additive CHARMM protein CMAP potential

results in enhanced cooperativity of α-helix and β-hairpin formation. Biophys J 103(5): 1045–1051

123. Lindorff-Larsen K, Maragakis P, Piana S, Eastwood MP, Dror RO, Shaw DE (2012) Systematic validation of protein force fields against experimental data. PLoS One 7(2): e32131

124. Karplus M (1959) Contact electron-spin coupling of nuclear magnetic moments. J Chem Phys 30(1):11–15

125. Kaminski GA, Friesner RA, Tirado-Rives J, Jorgensen WL (2001) Evaluation and reparametrization of the OPLS-AA force field for proteins via comparison with accurate quantum chemical calculations on peptides. J Phys Chem B 105(28):6474–6487

126. Oostenbrink C, Villa A, Mark AE, Van Gunsteren WF (2004) A biomolecular force field based on the free enthalpy of hydration and solvation: the GROMOS force-field parameter sets 53A5 and 53A6. J Comput Chem 25(13):1656–1676

127. Blanco FJ, Rivas G, Serrano L (1994) A short linear peptide that folds into a native stable [beta]-hairpin in aqueous solution. Nat Struct Mol Biol 1(9):584–590

128. Muñoz V, Thompson PA, Hofrichter J, Eaton WA (1997) Folding dynamics and mechanism of β-hairpin formation. Nature 390(6656): 196–199

129. Schuler B, Eaton WA (2008) Protein folding studied by single-molecule FRET. Curr Opin Struct Biol 18(1):16–26

130. Jo S, Kim T, Iyer VG, Im W (2008) CHARMM-GUI: a web-based graphical user interface for CHARMM. J Comput Chem 29(11):1859–1865

Chapter 4

Lipid Membranes for Membrane Proteins

Andreas Kukol

Abstract

The molecular dynamics (MD) simulation of membrane proteins requires the setup of an accurate representation of lipid bilayers. This chapter describes the setup of a lipid bilayer system from scratch using generally available tools, starting with a definition of the lipid molecule POPE, generation of a lipid bilayer, energy minimization, MD simulation, and data analysis. The data analysis includes the calculation of area and volume per lipid, deuterium order parameters, self-diffusion constant, and the electron density profile.

Key words Lipid bilayer, Molecular dynamics, Simulation, Trajectory analysis, Area per lipid, Volume per lipid, Deuterium order parameter, Self-diffusion constant, Electron density profile

1 Introduction

Molecular simulations of membrane proteins require consideration of the lipid membrane environment. While molecular dynamics (MD) simulations with implicit membrane models have been used successfully [1], for higher accuracy explicit representation of the lipid bilayer is desirable. Furthermore, dependent on the research question, if lipid–protein interactions are a subject of the study, an explicit representation of lipid molecules in unavoidable. Having decided on an explicit representation of lipids, further choice exists between coarse-grained, united-atom, and all-atom lipid models and force fields. In coarse-grained forcefields (covered in Chapter 7 of this book) several atoms are subsumed into one particle, for example in the MARTINI force field [2, 3] four carbon atoms of the aliphatic chain are subsumed into one particle. All-atom force fields usually provide the highest accuracy for the description of lipids and proteins. United-atom force fields subsume nonpolar hydrogen atoms into their adjacent carbon-atoms resulting in a moderate reduction of the number of particles, e.g., for a 1,2-dipalmitoyl-glycero-3-phosphocholine (DPPC) from 130 particles for the all-atom model to 50 particles for the united-atom

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_4, © Springer Science+Business Media New York 2015

model. United-atom models of several lipids have been shown to reproduce the experimentally measured properties of lipid bilayers to reasonable accuracy [4, 5]. Another consideration is the choice of force field for the protein, which is covered in Chapter 3 of this book. As a general principle different force fields should not be mixed, but protein, lipid, and possibly small organic molecules should be represented by the same force field. For practical reasons the choice of the force field is often based on the availability of the desired lipid model or topology for that particular force field.

Once the decision for a particular force field and lipid model for the use in MD simulations of a membrane protein has been made, a simulation of the lipid bilayer should be made and compared with experimental data. The membrane protein simulation may require a mixture of lipids in order to better represent the in vivo environment of a cell membrane, but unfortunately very few experimental data of mixed lipid membrane systems are available and virtually none of lipid membranes with proteins. The computational validation is, therefore, restricted to the individual components of the complete system to be investigated.

In this chapter the setup and MD simulation of a 1-palmitoyl-2-oleoyl-sn-glycero-3-phosphoethanolamine (POPE) bilayer with the united-atom force field GROMOS96 54a7 [6] is described followed by analysis of the simulation and comparison with experimental data. The MD simulation uses the software Gromacs version 4.5 [7, 8]. Similar procedures apply to other MD simulation software and/or force fields.

2 Methods

2.1 Materials

GROMACS version 4.5.3 or higher [8, 9]

CHARMM-GUI (http://www.charmm-gui.org/) via a web browser [10, 11]

GROMOS96 54a7 force field [6, 12] obtained from the 'Automated Topology Builder and Repository' web site (http://compbio.chemistry.uq.edu.au/atb/) [13], move the folder `gromos54a7.ff` directly into `share/top` and Gromacs will recognize the new force field automatically.

Perl interpreter

LINUX operating system

8-core computer workstation

2.2 Lipid Topology

The first step is to obtain or develop a molecular topology for POPE using the desired force field, in this case GROMOS96 54a7. The molecular topology specifies the atom types, atom charges, bonds and angles between atoms. Predefined parameters from the force field are assigned to those atoms, bonds, and angles, for example a predefined bond length and force constant. A good place to obtain lipid topologies for various force fields is Lipidbook (http://lipidbook.bioch.ox.ac.uk/) [14]. For this chapter, we will use a new topology of POPE shown in Fig. 1 based on topologies developed in earlier work for the GROMOS96 53a6 force field [4].

2.3 Lipid Bilayer Setup

The starting coordinates of a lipid bilayer can be conveniently generated with the Membrane Builder of the CHARMM-GUI [10, 11].

1. Using a web browser go to http://www.charmm-gui.org/?doc=input/membrane_only&step=1
2. Under point '3. Length of XY based on' select 'Numbers of lipid components'
3. Into the boxes next to POPE enter 64 for the number of lipids in upper leaflet and 64 for the lower leaflet.
4. Click on 'Next Step: Determine the system size'.
5. An output box appears, with the progress of the calculation. On the next page counter ions can be added. Select 'Add neutralizing ions', which are zero numbers because POPE is neutral. Click on 'Next Step'.
6. Once the calculation has finished click on 'Next Step: Assemble components'.
7. When the calculation has finished, you are taken to the next page, which allows to download the coordinates of the water/lipid bilayer. Right click on 'step5_assembly.pdb' and save it on your local computer for further equilibration with GROMACS (*see* **Note 1**).

2.4 Lipid Bilayer Equilibration

The first step is the construction of a system topology (Fig. 2) and a parameter (mdp-) file for energy minimization (Fig. 3). Additionally the molecular topology (itp-file) for POPE, '`pope_gromos54a7.itp`' shown in Fig. 1 is required. The lipid bilayer obtained from the CHARMM GUI contains a large number of near atomic clashes, which would cause energy minimization to fail. Therefore, we need to adopt a strategy that reduces the almost infinite energy from atomic clashes, by reducing the non-bonded interactions to a very low value as well as setting the emstep value (in Fig. 3) very low. At the same time the number of energy minimization steps is kept low (15 steps in Fig. 3) in order to avoid unusual distortions to the molecules, e.g., due to electrostatic attractions which are not compensated by van der Waals repulsions.

1. A simulation box is defined around the system:

```
ediconf -f step5_assembly.pdb -o step5_assembly.gro -d 0.0
```

```
[ moleculetype ]
; Name   nrexcl
POPE     3

[ atoms ]
;   nr    type   resnr  residu    atom    cgnr        charge         mass
     1     H        1    POPE      H1       0          0.3000   1.0080  ; qtot:0.3
     2     H        1    POPE      H2       0          0.3000   1.0080  ; qtot:0.6
     3     H        1    POPE      H3       0          0.3000   1.0008  ; qtot:0.9
     4      NL      1    POPE      N4       0         -0.2      14.0067 ; qtot:0.7
     5     CH2      1    POPE      C5       0          0.3      14.0270 ; qtot:1.0
     6     CH2      1    POPE      C6       1          0.4      14.0270 ; qtot:1.0
     7      OA      1    POPE      O7       1         -0.8      15.9994 ; qtot:0.54
     8       P      1    POPE      P8       1          1.7      30.9738 ; qtot:2.3
     9      OM      1    POPE      O9       1         -0.8      15.9994 ; qtot:1.5
    10      OM      1    POPE     O10       1         -0.8      15.9994 ; qtot:0.7
    11      OA      1    POPE     O11       1         -0.7      15.9994 ; qtot:0
    12     CH2      1    POPE     C12       2          0.4      14.0270 ; qtot:0.08
    13     CH1      1    POPE     C13       2          0.3      13.0190 ; qtot:0.52
    14      OE      1    POPE     O14       2         -0.7      15.9994 ; qtot:-0.14
    15       C      1    POPE     C15       2          0.7      12.0110 ; qtot:0.56
    16       O      1    POPE     O16       2         -0.7      15.9994 ; qtot:0.0
    17     CH2      1    POPE     C17       3            0      14.0270 ; qtot:
    18     CH2      1    POPE     C18       4            0      14.0270 ; qtot:
    19     CH2      1    POPE     C19       5            0      14.0270 ; qtot:
    20     CH2      1    POPE     C20       6            0      14.0270 ; qtot:
    21     CH2      1    POPE     C21       7            0      14.0270 ; qtot:
    22     CH2      1    POPE     C22       8            0      14.0270 ; qtot:
    23     CH2      1    POPE     C23       9            0      14.0270 ; qtot:
    24     CR1      1    POPE     C24      10            0      13.0190 ; qtot:
    25     CR1      1    POPE     C25      11            0      13.0190 ; qtot:
    26     CH2      1    POPE     C26      12            0      14.0270 ; qtot:
    27     CH2      1    POPE     C27      13            0      14.0270 ; qtot:
    28     CH2      1    POPE     C28      14            0      14.0270 ; qtot:
    29     CH2      1    POPE     C29      15            0      14.0270 ; qtot:
    30     CH2      1    POPE     C30      16            0      14.0270 ; qtot:
    31     CH2      1    POPE     C31      17            0      14.0270 ; qtot:
    32     CH2      1    POPE     C32      18          0.5      14.0270 ; qtot:
    33      OE      1    POPE     O33      18         -0.7      15.9994 ; qtot:
    34       C      1    POPE     C34      18          0.8      12.0110 ; qtot:
    35       O      1    POPE     O35      18         -0.6      15.9994 ; qtot:
    36     CH2      1    POPE     C36      19            0      14.0270 ; qtot:
    37     CH2      1    POPE     C37      20            0      14.0270 ; qtot:
    38     CH2      1    POPE     C38      21            0      14.0270 ; qtot:
    39     CH2      1    POPE     C39      22            0      14.0270 ; qtot:
    40     CH2      1    POPE     C40      23            0      14.0270 ; qtot:
    41     CH2      1    POPE     C41      24            0      14.0270 ; qtot:
    42     CH2      1    POPE     C42      25            0      14.0270 ; qtot:
    43     CH2      1    POPE     C43      26            0      14.0270 ; qtot:
    44     CH2      1    POPE     C44      27            0      14.0270 ; qtot:
    45     CH2      1    POPE     C45      28            0      14.0270 ; qtot:
    46     CH2      1    POPE     C46      29            0      14.0270 ; qtot:
    47     CH2      1    POPE     C47      30            0      14.0270 ; qtot:
    48     CH2      1    POPE     C48      31            0      14.0270 ; qtot:
    49     CH2      1    POPE     C49      32            0      14.0270 ; qtot:
    50     CH3      1    POPE     C50      33            0      15.0350 ; qtot:
    51     CH2      1    POPE     CA1      34            0      14.0270 ; tail2
    52     CH3      1    POPE     CA2      35            0      15.0350; tail2

[ bonds ]
;      ai      aj    funct
        4       5      2     gb_21
        5       6      2     gb_27
        6       7      2     gb_18
        7       8      2     gb_28
        8       9      2     gb_24
        8      10      2     gb_24
        8      11      2     gb_28
       11      12      2     gb_18
       12      13      2     gb_27
       13      14      2     gb_18
       13      32      2     gb_27
       14      15      2     gb_10
       15      16      2     gb_5
       15      17      2     gb_23
       17      18      2     gb_27
       18      19      2     gb_27
       19      20      2     gb_27
       20      21      2     gb_27
       21      22      2     gb_27
       22      23      2     gb_27
       23      24      2     gb_27
       24      25      2     gb_10
```

Fig. 1 The complete topology of POPE in the GROMOS96 54a7 force field

```
      25      26       2     gb_27
      26      27       2     gb_27
      27      28       2     gb_27
      28      29       2     gb_27
      29      30       2     gb_27
      30      31       2     gb_27
      31      51       2     gb_27
      51      52       2     gb_27
      32      33       2     gb_18
      33      34       2     gb_10
      34      35       2     gb_5
      34      36       2     gb_23
      36      37       2     gb_27
      37      38       2     gb_27
      38      39       2     gb_27
      39      40       2     gb_27
      40      41       2     gb_27
      41      42       2     gb_27
      42      43       2     gb_27
      43      44       2     gb_27
      44      45       2     gb_27
      45      46       2     gb_27
      46      47       2     gb_27
      47      48       2     gb_27
      48      49       2     gb_27
      49      50       2     gb_27
       1       4       2     gb_2    ;H-N bond type
       2       4       2     gb_2
       3       4       2     gb_2

[ pairs ]
;  ai    aj funct
    1     6     1
    2     6     1
    3     6     1
    4     7     1
    5     8     1
    6     9     1
    6    10     1
    6    11     1
    7    12     1
    8    13     1
    9    12     1
   10    12     1
   11    14     1
   11    32     1
   12    15     1
   12    33     1
   13    16     1
   13    17     1
   13    34     1
   14    18     1
   14    33     1
   15    19     1
   15    32     1
   16    18     1
   22    25     1    ; pair around double bond
   24    27     1    ; pair around double bond
   32    35     1
   32    36     1
   33    37     1
   34    38     1
   35    37     1

[ angles ]
;  ai    aj    ak funct
       4       5       6       2     ga_15
       5       6       7       2     ga_15
       6       7       8       2     ga_26
       7       8       9       2     ga_14
       7       8      10       2     ga_14
       7       8      11       2     ga_5
       8      11      12       2     ga_26
       9       8      10       2     ga_29
       9       8      11       2     ga_14
      10       8      11       2     ga_14
      11      12      13       2     ga_15
      12      13      14       2     ga_13
      12      13      32       2     ga_13
      13      14      15       2     ga_22
      13      32      33       2     ga_15
      14      13      32       2     ga_13
      14      15      16       2     ga_31
```

Fig. 1 (continued)

```
      14      15      17       2     ga_16
      15      17      18       2     ga_15
      16      15      17       2     ga_35
      17      18      19       2     ga_15
      18      19      20       2     ga_15
      19      20      21       2     ga_15
      20      21      22       2     ga_15
      21      22      23       2     ga_15
      22      23      24       2     ga_15
      23      24      25       2     ga_27 ;double bond
      24      25      26       2     ga_27 ; double bond
      25      26      27       2     ga_15
      26      27      28       2     ga_15
      27      28      29       2     ga_15
      28      29      30       2     ga_15
      29      30      31       2     ga_15
      30      31      51       2     ga_15
      31      51      52       2     ga_15
      32      33      34       2     ga_22
      33      34      35       2     ga_31
      33      34      36       2     ga_16
      34      36      37       2     ga_15
      35      34      36       2     ga_35
      36      37      38       2     ga_15
      37      38      39       2     ga_15
      38      39      40       2     ga_15
      39      40      41       2     ga_15
      40      41      42       2     ga_15
      41      42      43       2     ga_15
      42      43      44       2     ga_15
      43      44      45       2     ga_15
      44      45      46       2     ga_15
      45      46      47       2     ga_15
      46      47      48       2     ga_15
      47      48      49       2     ga_15
      48      49      50       2     ga_15
       1       4       2       2     ga_10
       2       4       3       2     ga_10
       3       4       1       2     ga_10
       1       4       5       2     ga_11
       2       4       5       2     ga_11
       3       4       5       2     ga_11

[ dihedrals ]
;  ai    aj    ak    al funct   phi0     cp     mult
    1     4     5     6     1    gd_29
    4     5     6     7     1    gd_4
    4     5     6     7     1    gd_36
    5     6     7     8     1    gd_29
    6     7     8    11     1    gd_20
    6     7     8    11     1    gd_27
    7     8    11    12     1    gd_20
    7     8    11    12     1    gd_27
    8    11    12    13     1    gd_29
   11    12    13    14     1    gd_34
   11    12    13    32     1    gd_34
   11    12    13    32     1    gd_17
   12    13    32    33     1    gd_34
   12    13    32    33     1    gd_17
   12    13    14    15     1    gd_29
   13    32    33    34     1    gd_29
   13    14    15    17     1    gd_13
   14    13    32    33     1    gd_18
   14    15    17    18     1    gd_40
   15    17    18    19     1    gd_34
   17    18    19    20     1    gd_34
   18    19    20    21     1    gd_34
   19    20    21    22     1    gd_34
   20    21    22    23     1    gd_34
   21    22    23    24     1    0      3.350     1
   21    22    23    24     1    180    1.660     2
   21    22    23    24     1    0      7.333     3
   22    23    24    25     3    2.885  4.17  7.8  4.4  0.0  0.0
   24    25    26    27     3    2.885  4.17  7.8  4.4  0.0  0.0
   25    26    27    28     1    0      3.350     1
   25    26    27    28     1    180    1.660     2
   25    26    27    28     1    0      7.333     3
   26    27    28    29     1    gd_34
   27    28    29    30     1    gd_34
   28    29    30    31     1    gd_34
   29    30    31    51     1    gd_34
   30    31    51    52     1    gd_34
   13    32    33    34     1    gd_29
```

Fig. 1 (continued)

```
   32    33    34    36     1    gd_13
   33    34    36    37     1    gd_40
   34    36    37    38     1    gd_34
   36    37    38    39     1    gd_34
   37    38    39    40     1    gd_34
   38    39    40    41     1    gd_34
   39    40    41    42     1    gd_34
   40    41    42    43     1    gd_34
   41    42    43    44     1    gd_34
   42    43    44    45     1    gd_34
   43    44    45    46     1    gd_34
   44    45    46    47     1    gd_34
   45    46    47    48     1    gd_34
   46    47    48    49     1    gd_34
   47    48    49    50     1    gd_34

[ dihedrals ]
;  ai    aj    ak    al funct
      13      14      32      12       2    gi_2
      15      14      17      16       2    gi_1
      34      33      36      35       2    gi_1
      23      24      25      26       2    gi_1 ; double bond

#ifdef POSRES_LIPID
#include "lipid_posre.itp"
#endif
```

Fig. 1 (continued)

```
; Include forcefield parameters
#include "gromos54a7.ff/forcefield.itp"
#include "pope_Gromacs4_Gromos96.itp"

#ifdef POSRES_P
[ position_restraints ]
; atom  type    fx      fy      fz
   8      1     0.0     0.0     1000.0
; this restraints all P8 atoms
#endif

; Include water topology
#ifdef FLEX_SPC
#include "flexspc.itp"
#else
#include "spc.itp"
#endif

#ifdef POSRES_WATER
; Position restraint for each water oxygen
[ position_restraints ]
;  i funct       fcx        fcy        fcz
   1    1       1000       1000       1000
#endif

[ system ]
; Name
128 POPE + water

[ molecules ]
; Compound        #mols
POPE            128
SOL             2560
```

Fig. 2 The topology of the whole lipid bilayer system composed of 128 POPE molecules and 2560 water molecules. The molecular topologies are read from itp-files via the `#include` commands

```
include                  = -I../top
define                   =

; RUN CONTROL PARAMETERS
integrator               = steep
; number of steps
nsteps = 15
emstep = 0.001
emtol = 200.0
nstcgsteep = 10
; mode for center of mass motion removal
comm-mode                = Linear
; number of steps for center of mass motion removal
nstcomm                  = 1
; group(s) for center of mass motion removal
comm-grps                =

; Selection of energy groups
energygrps               = POPE SOL

; NEIGHBORSEARCHING PARAMETERS
; nblist update frequency
nstlist                  = 10
; ns algorithm (simple or grid)
ns_type                  = grid
; Periodic boundary conditions: xyz (default), no (vacuum)
; or full (infinite systems only)
pbc                      = xyz
; nblist cut-off
rlist                    = 1.2

; OPTIONS FOR ELECTROSTATICS AND VDW
; Method for doing electrostatics
coulombtype              = PME
rcoulomb-switch          = 0
rcoulomb                 = 1.2
; Relative dielectric constant for the medium and the reaction field
epsilon_r                = 1
epsilon_rf               = 1
; Method for doing Van der Waals
vdw-type                 = Cut-off
; cut-off lengths
rvdw-switch              = 0
rvdw                     = 1.2
; Apply long range dispersion corrections for Energy and Pressure
DispCorr                 = EnerPres
; Spacing for the PME/PPPM FFT grid
fourierspacing           = 0.18
; FFT grid size, when a value is 0 fourierspacing will be used
fourier_nx               = 0
fourier_ny               = 0
fourier_nz               = 0
; EWALD/PME/PPPM parameters
pme_order                = 5
ewald_rtol               = 1e-05
ewald_geometry           = 3d
epsilon_surface          = 0
optimize_fft             = yes

; OPTIONS FOR BONDS
constraints              = none
; Type of constraint algorithm
```

Fig. 3 The parameters for initial energy minimization. Note that the values for nsteps (normally thousands) and emstep (normally 0.1) are set very low in this example

2. The coordinate file with simulation box, topology files, and mdp-file are put together into a binary simulation run file (tpr-file) (*see* **Note 2**):

```
grompp -f em1.mdp -c step5_assembly.gro -p
system.top -pp processed.top -o run.tpr
```

   The file `processed.top` contains a complete topology of the system.

3. The van der Waals interactions in the topology are then drastically reduced:

```
RescaleNonBond.pl processed.top 0.005 > em1.
top
```

   The Perl script `RescaleNonBond.pl` is shown in Fig. 4.

4. The new topology `em1.top` is used to run the energy minimization.

```
grompp -f em1.mdp -c step5_assembly.pdb -p
em1.top -o run.tpr
mdrun -s em1.tpr -c em1.gro -v
```

5. Using `em1.gro` as the coordinate file instead of `step5_assembly.gro` (-c option) the **steps 2–4** above are repeated with increasing scale factors of 0.05, 0.1, and 0.5. At the same time the number of energy minimization steps in em1.mdp is increased to up to 40 steps. The exact value of the scale factors and number of energy minimization steps is a matter of trial and error.
6. For the final energy minimization no rescaling of van der Waals interactions is used, the parameter emstep in Fig. 3 is set to 0.01 and the nsteps to 2000. This yields the file `final_em.gro`.

Further equilibration could be performed, such as performing a short MD simulation with position restraints on the phosphorus atoms of the lipid molecules or MD simulations at constant volume (NVT-ensemble). However, we did not find that necessary as lipids tend to self-assemble into their equilibrium position during the MD simulation at constant pressure (NPT-ensemble). For analysis of the equilibrium properties the first few 10th of nanoseconds of an NPT simulation can be discarded.

2.5 MD Simulation Run

The coordinate file of the final energy minimization can be directly used to run a long MD simulation over 100 ns in this example. The parameter file for this simulation is shown in Fig. 5.

```
grompp -f md_100ns.mdp -c final_em.gro -p sys-
tem.top -o run_100ns.tpr
   mdrun -s run_100ns.tpr -x run_100ns.xtc -e
run_100ns.edr -g run_100ns.log -c after_100ns.
gro -v -stepout 2000
```

(*see Note 3*)

The most important output files for further data analysis are the trajectory run_100ns.xtc and the energy file run_100ns.edr, which also contains the data about the system dimension.

```
#!/usr/bin/perl

# scale the [atomtypes], [nonbond_params] and [pairtypes]
# in a Gromacs processed topology by a factor
# Usage: RescaleNonBond.pl <filename> <factor> > <output file>

$file = $ARGV[0];
$factor = $ARGV[1];
open(IN,"$file") || die "cannot open file: $file";

$nonbond_section=0;
$pairtype_section=0;
$atomtype_section=0;
while(<IN>){
  $in = $_;
  if ($in =~ /^.+ atomtypes .+/) {$atomtype_section = 1}
  if ($in =~ /^.+nonbond_params.+/) {$nonbond_section = 1; $atomtype_section=0}
  if ($in =~ /^.+ pairtypes .+/) {$pairtype_section = 1; $nonbond_section=0}

  if ($nonbond_section == 1 && $in =~ /^\s+\S+\s+\S+\s+\S+\s+(\S+)\s+(\S+)/) {
    $c6 = $1;
    $c12 = $2;
    $c6_new = $c6 * $factor; $c12_new = $c12 * $factor;
    $in =~ s/$c6/$c6_new/;
    $in =~ s/$c12/$c12_new/;
  }

  if ($pairtype_section == 1 && $in =~ /^\s+\S+\s+\S+\s+\S+\s+(\S+)\s+(\S+)/) {
    $c6 = $1;
    $c12 = $2;
    $c6_new = $c6 * $factor; $c12_new = $c12 * $factor;
    $in =~ s/$c6/$c6_new/;
    $in =~ s/$c12/$c12_new/;
  }

  if ($atomtype_section == 1 && $in =~
/^\s*\S+\s+\S+\s+\S+\s+\S+\s+\S+\s+(\S+)\s+(\S+)/) {
    $c6 = $1;
    $c12 = $2;
    $c6_new = $c6 * $factor; $c12_new = $c12 * $factor;
    $in =~ s/$c6/$c6_new/;
    $in =~ s/$c12/$c12_new/;
  }

  print $in;
  if ($in =~ /^; Table 2\.5\.2\.1/) {$pairtype_section = 0}
}
```

Fig. 4 The Perl-script ScaleNonBond.pl to rescale the non-bonding interactions of a GROMACS processed topology

```
; VARIOUS PREPROCESSING OPTIONS
include                  = -I../top
define                   =

; RUN CONTROL PARAMETERS
integrator               = md
; Start time and timestep in ps
tinit                    = 0
dt                       = 0.002
nsteps                   = 50000000 ; 100 ns
; For exact run continuation or redoing part of a run
init_step                = 0
; mode for center of mass motion removal
comm-mode                = Linear
; number of steps for center of mass motion removal
nstcomm                  = 10
; group(s) for center of mass motion removal (default=system)
comm-grps                = POPE SOL

nstcalcenergy             = 10

; OUTPUT CONTROL OPTIONS
; Output frequency for coords (x), velocities (v) and forces (f)
; 500 is every picoseconds
nstxout                  = 1000000
nstvout                  = 1000000
nstfout                  = 1000000
; Output frequency for energies to log file and energy file (every 50 ps)
nstlog                   = 25000
nstenergy                = 25000
; Output frequency and precision for xtc file
nstxtcout                = 25000
xtc-precision            = 1000
; This selects the subset of atoms for the xtc file. You can
; select multiple groups. By default all atoms will be written.
xtc_grps                 =
; Selection of energy groups
energygrps               = POPE SOL

; NEIGHBORSEARCHING PARAMETERS
; nblist update frequency
nstlist                  = 10
; ns algorithm (simple or grid)
ns_type                  = grid
; Periodic boundary conditions: xyz (default), no (vacuum)
; or full (infinite systems only)
pbc                      = xyz
; nblist cut-off
rlist                    = 0.9

; OPTIONS FOR ELECTROSTATICS AND VDW
; Method for doing electrostatics
coulombtype              = PME
rcoulomb-switch          = 0
rcoulomb                 = 0.9
; Relative dielectric constant for the medium and the reaction field
epsilon_r                = 1
epsilon_rf               = 1
; Method for doing Van der Waals
vdw-type                 = Cut-off
; cut-off lengths
rvdw-switch              = 0
rvdw                     = 1.4
; Apply long range dispersion corrections for Energy and Pressure
DispCorr                 = EnerPres
; Spacing for the PME/PPPM FFT grid
fourierspacing           = 0.12
; FFT grid size, when a value is 0 fourierspacing will be used
```

Fig. 5 The parameters that were used to run the 100 ns MD simulation of the lipid bilayer system

```
fourier_nx               = 0
fourier_ny               = 0
fourier_nz               = 0
; EWALD/PME/PPPM parameters
pme_order                = 4
ewald_rtol               = 1e-05
ewald_geometry           = 3d
epsilon_surface          = 0
optimize_fft             = yes

; OPTIONS FOR WEAK COUPLING ALGORITHMS
; Temperature coupling
tcoupl                   = v-rescale
; Groups to couple separately
tc-grps                  = POPE SOL
; Time constant (ps) and reference temperature (K)
tau_t                    = 0.1  0.1
ref_t                    = 298  298
; Pressure coupling
Pcoupl                   = Berendsen
Pcoupltype               = semiisotropic
; Time constant (ps), compressibility (1/bar) and reference P (bar)
tau_p                    = 2.0  2.0
compressibility          = 4.5e-5  4.5e-5
ref_p                    = 1.0  1.0
; Random seed for Andersen thermostat
andersen_seed            = 815131

; GENERATE VELOCITIES FOR STARTUP RUN
gen_vel                  = yes
gen_temp                 = 20
gen_seed                 = 168473

; OPTIONS FOR BONDS
constraints              = all-bonds
; Type of constraint algorithm
constraint-algorithm     = Lincs
; Do not constrain the start configuration
continuation             = no
; Use successive overrelaxation to reduce the number of shake iterations
Shake-SOR                = no
; Relative tolerance of shake
shake-tol                = 0.0001
; Highest order in the expansion of the constraint coupling matrix
lincs-order              = 4
; Number of iterations in the final step of LINCS. 1 is fine for
; normal simulations, but use 2 to conserve energy in NVE runs.
; For energy minimization with constraints it should be 4 to 8.
lincs-iter               = 1
; Lincs will write a warning to the stderr if in one step a bond
; rotates over more degrees than
lincs-warnangle          = 30
; Convert harmonic bonds to morse potentials
morse                    = no
```

Fig. 5 (continued)

2.6 Data Analysis

Since this chapter is aimed at providing lipid membranes for membrane proteins, rather than investigating the properties of lipid bilayers in detail, the aim of the data analysis is to establish, if the simulation reproduces experimentally known parameters to reasonable accuracy. Typically the area per lipid and volume per lipid are compared with experiment; for POPE this is available from a study by Rappolt et al. [15]. Deuterium order parameters and self-diffusion constants are available for some lipids. If they are not available for the particular type of lipid, they may be compared with values from other lipids in order to check for errors in the topology or simulation.

2.6.1 Area and Volume per Lipid

These observables are calculated from the size of the simulation box, which is usually arranged in a way that the z-coordinate is perpendicular to the lipid bilayer plane. The area per lipid A_L is then obtained by the area of the simulation box divided by the number of lipid molecules in one leaflet, i.e., $A_L = \text{size}_x \cdot \text{size}_y/64$.

The volume per lipid V_L is obtained as:

$$V_L = \left(V_{box} - V_{water}\right) / 128 = \left(\text{size}_x \bullet \text{size}_y \bullet \text{size}_z - V_{water}\right) / 128$$

The volume of water V_{water} can be obtained by performing an MD simulation of a box of water molecules using the same MD parameters that were used for the lipid/water simulation. This was done and the volume of a water molecule was determined as (0.03058 ± 0.00008) nm^3, leading to a total volume for 2560 water molecules (*see* Fig. 2) of $V_{water} = 78.28$ nm^3.

The size of the simulation box along the trajectory is computed with g_energy:

```
    g_energy -f run_100ns.edr -s run_100ns.tpr
-o box_XYZ.txt
```

In the dialogue select ‘Box-X’, ‘Box-Y’, and ‘Box-Y’ by typing the corresponding numbers. The output file can be imported in a spreadsheet program for calculating the area and volume per lipid as described above. For POPE the change of area and volume per lipid over the 100 ns trajectory is shown in Fig. 6. Averages are calculated for the last 70 ns of the trajectory (Table 1) and show less than 5 % deviation from experimental data.

2.6.2 Lipid Acyl Chain Order Parameters

Order parameters of the lipid acyl chains can be measured from solid-state deuterium NMR spectra. The order parameter for a C-D bond is directly calculated from the measured quadrupolar splittings. It is indicative of the average orientation of the C-D vector with respect to the external magnetic field, normally the z-axis in aligned lipid bilayers. In order to calculate the order parameter for a united-atom force field, the C-D bond vector is

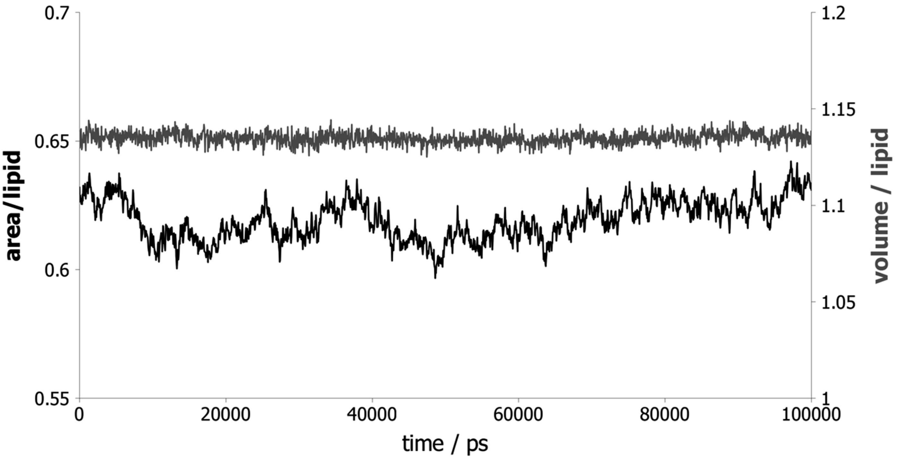


Fig. 6 Area/lipid (*black curve*) and volume/lipid (*grey curve*) over the course of the 100 ns simulation

Table 1
Properties of the POPE lipid bilayer from simulations compared with experimental data

	Simulation	Experiment
Area/lipid	(0.620 ± 0.008) nm^2	0.6025 nm^2 [15]
Volume/lipid	(1.135 ± 0.003) nm^3	1.175 nm^3 [15]
Self-diffusion coefficient	(6.42 ± 0.02) 10^{-8} cm^2/s	8.87 10^{-8} cm^2/s (for POPC) [16]

reconstructed automatically by the g_order tool of the Gromacs software.

1. Prepare an index-file `sn1.ndx` for the *sn1* lipid acyl chain that contains the atom numbers of each carbon at equivalent positions:

```
[C34]
34 86 138 …
[C36]
36 88 140 …
[C37]
37 81 141 …
……
[C50]
```

For the *sn2* lipid acyl chain, the order parameters must be calculated separately for the saturated and unsaturated carbons. The index file for the saturated carbons `sn2.ndx` contains all atoms:

```
[C15]
[C17]
[C18]
…
```

```
[C31]
[CA1]
[CA2]
```

The index file for the unsaturated carbons sn2_unsat.ndx contains index groups for the unsaturated carbons and the two neighbors on each side:

```
[C23], [C24], [C25], [C26]
```

The index groups are made interactively with:

```
make_ndx -f after_100ns.gro -o sn1.ndx
```

This opens an interactive session, in which you create index groups for each acyl chain atom using the add command: ‘`a c34`’, ‘`a c36`’, ‘`a c37`’, and so on. Finally you delete the default groups: ‘`del 0-5`’ and ‘`quit`’.

2. Calculate the order parameters over the last 70 ns of the simulation from the trajectory:

```
g_order -f run_100ns.txt -s run_100ns.tpr -n
sn1.ndx -od deuter_sn1.xvg
- b 30000
g_order -f run_100ns.txt -s run_100ns.tpr -n
sn2.ndx -od deuter_sn2.xvg
- b 30000
g_order -f run_100ns.txt -s run_100ns.tpr -n
sn2_unsat.ndx
-od deuter_sn2unsat.xvg - b 30000
```

3. Using a text editor replace the order parameters for the unsaturated carbons in deuter_sn2.xvg by the corresponding values from deuter_sn2unsat.xvg (*see* **Note 4**).

2.6.3 Lateral Self-Diffusion Coefficient

1. An index file need to be prepared that contains all atoms numbers belonging to lipid molecules:

```
make_ndx -f after_100ns.gro -o lipids.ndx
```

One of the default index group should correspond to the lipid, e.g., ‘2 POPE’. Then you type ‘`keep 2`’ and ‘`q`’ for save and quit.

2. The self-diffusion coefficient can then be calculated with the g_msd tool.

```
g_msd -f run_100ns.xtc -s run_100ns.tpr -n
lipids.ndx
-lateral z -mol diffusion.xvg -o msd.xvg -b
50000
```

A value of (6.42 ± 0.02) 10^{-8} cm^2/s is reported, which is in the right region for lipid diffusion. Note that only the last 50 ns of the trajectory were analyzed in the example above due to computer memory limitations.

2.6.4 Electron Density

The electron density distribution in z-direction (perpendicular to the membrane plane) is one of the best indicators of the experimental accuracy of a lipid membrane simulation, since it can be directly

```
  55
  H1 = 1
  H2 = 1
  H3 = 1
  N4 = 7
  C5 = 8
  C6 = 8
  O7 = 8
  P8 = 15
  O9 = 8
O10 = 8
...
...
...
OW = 8
HW1 = 1
HW2 = 1
```

Fig. 7 An extract of the file `electrons.dat` that specifies the number of electrons for each atom. The first line specifies the number of input lines, which are the sum of 52 lines for lipid atoms and 3 lines for water atoms

obtained from the inverse Fourier-transformation of the small angle X-ray scattering (SAXS) curve. Other parameters derived from SAXS experiments, such as area per lipid (*see* Subheading 2.6.1) require fitting of the SAXS curves to an electron density model of the lipid bilayer, in which the area per lipid is one of many free parameters. It is, therefore, preferable to compare the electron density distribution of the MD simulation directly with the electron density obtained from SAXS curves.

1. A text file (`electrons.dat`) is required that contains the number of electrons associated with each atom in the structure, an extract of the file is shown in Fig. 7.
2. The electron density averaged over the trajectory from 30 to 100 ns is then calculated with the g_density tool:

   ```
   g_density -f run_100ns.xtc -s run_100ns.tpr
   -ei electrons.dat -b 30000 -dens electron -d
   Z -symm -o e_density.xvg
   ```

   The option `-d` specifies the axis normal to the membrane and `-symm` symmetrizes the density around the axis; `-symm` should not be used for asymmetric bilayers.
3. The electron density is contained in the file `e_density.xvg` and plotted in Fig. 8.

3 Notes

1. Clicking further on 'Next Step' provides us with the required input file to run CHARMM MD simulations to equilibrate the bilayer on the local computer. Since we want to use a different force field, we will continue the equilibration with Gromacs.

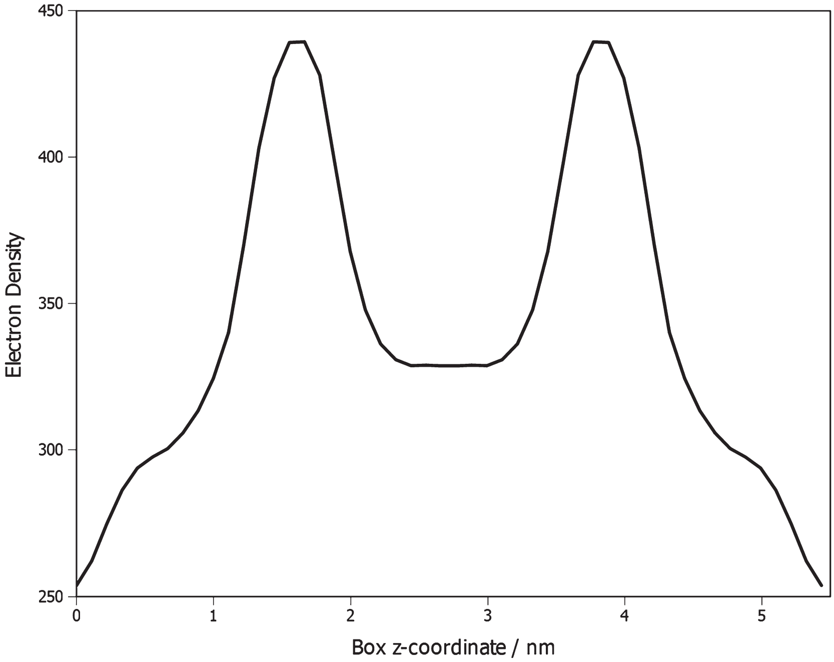


Fig. 8 The electron density in electrons/nm^3 along the z-coordinate of the simulation box

2. The output of grompp usually contains many warnings about non-matching atom names between coordinate file and topology. Grompp terminates with the fatal error of too many warnings. The warnings about non-matching atom names can be safely ignored, but others should be inspected carefully. Then rerun grompp by setting the –maxwarn option.
3. On a standard LINUX workstation the mdrun command would be run in the background: `nohup mdrun … &`

   The output produced by the mdrun command is then available in `nohup.out`.

   In a high-performance cluster environment you need to consult the documentation about how to schedule and run jobs on the cluster.
4. Note that if you specified N atoms in the index file, the order parameters are calculated for N-2 atoms, no order parameter is calculated for the first and the last atom in the index file.

Acknowledgements

This work was supported by the School of Life and Medical Sciences, University of Hertfordshire and has made use of the University of Hertfordshire Science and Technology Research Institute high-performance computing facility. I thank all research groups that made their tools and programs available to the research community.

References

1. Tanizaki S, Feig M (2006) Molecular dynamics simulations of large integral membrane proteins with an implicit membrane model. J Phys Chem B 110(1):548–556
2. Monticelli L, Kandasamy SK, Periole X, Larson RG, Tieleman DP, Marrink SJ (2008) The MARTINI coarse-grained force field: Extension to proteins. J Chem Theory Comput 4(5):819–834
3. Marrink SJ, Risselada HJ, Yefimov S, Tieleman DP, de Vries AH (2007) The MARTINI force field: Coarse grained model for biomolecular simulations. J Phys Chem B 111(27): 7812–7824
4. Kukol A (2009) Lipid Models for United-Atom Molecular Dynamics Simulations of Proteins. J Chem Theory Comput 5(3):615–626
5. Ulmschneider JP, Ulmschneider MB (2009) United Atom Lipid Parameters for Combination with the Optimized Potentials for Liquid Simulations All-Atom Force Field. J Chem Theory Comput 5(7):1803–1813
6. Schmid N, Eichenberger AP, Choutko A, Riniker S, Winger M, Mark AE et al (2011) Definition and testing of the GROMOS force-field versions 54A7 and 54B7. Eur Biophys J Biophys Lett 40(7):843–856
7. Pronk S, Pall S, Schulz R, Larsson P, Bjelkmar P, Apostolov R et al (2013) GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. Bioinformatics 29(7):845–854
8. Hess B, Kutzner C, van der Spoel D, Lindahl E (2008) GROMACS 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. J Chem Theory Comput 4(3):435–447
9. Van der Spoel D, Lindahl E, Hess B, Groenhof G, Mark AE, Berendsen HJC (2005) GROMACS: Fast, flexible, and free. J Comput Chem 26(16):1701–1718
10. Jo S, Kim T, Im W (2007) Automated Builder and Database of Protein/Membrane Complexes for Molecular Dynamics Simulations. Plos One 2(9)
11. Jo S, Lim JB, Klauda JB, Im W (2009) CHARMM-GUI Membrane Builder for Mixed Bilayers and Its Application to Yeast Membranes. Biophys J 97(1):50–58
12. Wang DQ, Freitag F, Gattin Z, Haberkern H, Jaun B, Siwko M et al (2012) Validation of the GROMOS 54A7 Force Field Regarding Mixed alpha/beta-Peptide Molecules. Helvetica Chimica Acta 95(12):2562–2577
13. Malde AK, Zuo L, Breeze M, Stroet M, Poger D, Nair PC et al (2011) An Automated Force Field Topology Builder (ATB) and Repository: Version 1.0. J Chem Theory Comput 7(12): 4026–4037
14. Domanski J, Stansfeld PJ, Sansom MSP, Beckstein O (2010) Lipidbook: A Public Repository for Force-Field Parameters Used in Membrane Simulations. J Membr Biol 236(3): 255–258
15. Rappolt M, Hickel A, Bringezu F, Lohner K (2003) Mechanism of the lamellar/inverse hexagonal phase transition examined by high resolution x-ray diffraction. Biophys J 84(5): 3111–3122
16. Filippov A, Oradd G, Lindblom G (2003) Influence of cholesterol and water content on phospholipid lateral diffusion in bilayers. Langmuir 19(16):6397–6400

Chapter 5

Molecular Dynamics Simulations of Membrane Proteins

Philip C. Biggin and Peter J. Bond

Abstract

Membrane protein structures are underrepresented in the Protein Data Bank (PDB) due to difficulties associated with expression and crystallization. As such, it is one area where computational studies, particularly Molecular Dynamics (MD) simulations, can provide useful additional information. Recently, there has been substantial progress in the simulation of lipid bilayers and membrane proteins embedded within them. Initial efforts at simulating membrane proteins embedded within a lipid bilayer were relatively slow and interactive processes, but recent advances now mean that the setup and running of membrane protein simulations is somewhat more straightforward, though not without its problems. In this chapter, we outline practical methods for setting up and running MD simulations of a membrane protein embedded within a lipid bilayer and discuss methodologies that are likely to contribute future improvements.

Key words Molecular dynamics, Simulation, Computational, Membrane proteins, Ion channels

1 Introduction

Membrane proteins are thought to constitute approximately 30 % of genomes [1]. Furthermore it has been estimated that over half of all drug targets are membrane proteins [2]. However, due to problems associated with expression and crystallization, the number of high-resolution crystal structures is less than 1 % of the total number of structures (*see* http://blanco.biomol.uci.edu/mpstruc/ for a maintained list of membrane protein structures). The situation is further complicated by the fact that many membrane proteins undergo quite large conformational changes in order to complete their function (for example transporter proteins [3–5] which cycle between at least two distinct states). Crystallography will at best only be able to capture a time and space averaged snapshot of these states. Computer simulations on the other hand, and in particular Molecular Dynamics (MD) simulations, are useful tools that in addition to providing information on the stability of a membrane protein can also provide insight into the manner in which these conformational changes can proceed. Thus there has

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_5, © Springer Science+Business Media New York 2015

been a large increase in the application of computer simulation methods to membrane proteins [6, 7] ranging from ion channels [8–10] to outer-membrane proteins [11]. MD can also be used to test hypotheses in both idealized systems where one can explore underlying biophysical principals governing a process [12] through to systems which represent the in vivo systems as close as possible [13].

The field of membrane-protein simulation has matured over recent years and there are now many groups worldwide performing simulations. One reason for the recent increase in the number of research groups is that the computational facilities required to perform membrane-protein simulations are now accessible to more people. As the most common computational method used is MD simulation, we will discuss in this chapter, how to set up and run, a membrane-protein simulation focussing on the more practical aspects. Until recently the setup was rather complicated and required a large amount of interactive input from the researcher. Now, principally due to increases in computational power, the setup and running of such simulations is much simpler. We divide the process up into 4 distinct steps; the preparation of the protein itself, the preparation of the lipid (though only briefly as the main thrust of this chapter is on the setup and running of a membrane protein simulation), the actual insertion and establishing of a stable system and finally, running the simulation. Of these, it is perhaps the preparation of the protein itself which requires the most care and interactive input from the researcher.

2 Theory

The underlying theory for molecular dynamics simulations is covered in Chapters 1 and 3, and therefore in this section we briefly discuss some specific considerations that researchers should bear in mind when performing simulations of membrane proteins. Perhaps the most important of these is the timescale of the problem that is under consideration and the resource that is available. The many different aspects of membrane dynamics span a large timescale ranging from a few picoseconds (for a protein side-chain to rotate) though to minutes and longer (for flip-flop motion of lipids). Indeed, where resources are minimal and only an approximate representation of the bilayer is required, one may be content with using a slab of octane to represent the hydrophobic core of the bilayer [14] or even a hybrid model [15]. Substantial recent efforts have been made to approximate the lipid molecules in a different way by using a coarse-grain approach, where typically 4 atoms are represented by one particle [16]. These methods have become popular as they allow much longer timescale events to be explored, but of course they are less detailed than a fully atomistic simulation.

Typically for simulations of membrane proteins a stable simulation is required, usually reflecting some sort of equilibrium of the system. In these simulations, we have two components to worry about: the protein and the lipid bilayer. Thus some metrics of stability are required. For the protein, the most common of these is the root mean square deviation (RMSD) of Cα atoms from the initial (usually X-ray) starting structure. In the case of the lipid, a good indicator is the mean surface area per lipid [17].

The basic theory underlying the insertion process below is very simple. We place the protein in the bilayer and remove overlapping atoms. We then allow the whole system to relax and equilibrate as the lipids adjust conformation around the protein. The positioning of the protein is still somewhat subjective, especially with respect to its displacement along the membrane normal axis. However, structural bioinformatics analysis [18] has demonstrated that nearly all membrane proteins have two "aromatic girdles" (although not Phenylalanine) separated by about 30 Å. These girdles are thought to interact with the interface region of the lipid bilayer and so provide an approximate indication of how to position the protein. Another approach is to treat the protein as a rigid body and optimize the transfer free energy between water and a hydrophobic slab that represents the location of the lipid bilayer [19–21]. Some of these methods have been developed into web servers where one can obtain a pre-orientated inserted protein (*see* for example Subheading 2.6 in Chapter 17). We discuss towards the end of the chapter how this issue may also be addressed via a coarse-grained simulation approach. If one is not interested in the interaction of the protein with the lipid at the particle level at all, then a more suitable approach may be to use an implicit membrane.

3 Methods

There are obviously two main components to a membrane protein simulation: the actual protein and the lipid bilayer it is to be embedded in. Although we focus on the issues concerning the whole system, it is worth briefly reviewing practical considerations for these individual components.

3.1 Preparation of the Protein

Typically the starting point for the protein will be a structure deposited in the protein data bank (PDB; www.rcsb.org). Often however, these structures will need a certain amount of preparation before production level molecular dynamics (MD) can be run. The most severe of these considerations might be missing atoms, which can range from entire loops to a couple of side-chain atoms. How one deals with this problem depends upon the question one is trying to address with the simulation. For the case where only a few atoms are missing from a small number of side-chains one can manually build in the missing atoms using an interactive modelling program such PyMOL [22] or What-If [23] (*see* **Note 1**). For the more complicated case where whole loops are missing, typically one has to resort to programs which can build random structures which are geometrically correct such as Modeller [24, 25]. Indeed, in some cases, it may be that construction of an entire homology model is required (covered in Chapters 15, 16 and 17). Another related consideration is how to deal with the termini in the structure. Frequently, the structure is not the whole sequence of the protein, and therefore charged termini may not be appropriate. One common procedure has been to build on capping groups that help to best mimic the continuing protein chain (*see* **Note 2**). A simpler approach involves simply protonating the C-terminus and deprotonating the N-terminus.

In all but the very high-resolution structures, one will still have to add hydrogen atoms, as these will not be present in the PDB file. Although this is a very simple process, there are decisions to be made even for this process: (1) First, the choice of force-field is important—in particular, whether it is an all atom, such as the CHARMM parameter sets [26, 27], or a united-atom model in which only polar hydrogens are explicit, as in the GROMOS [28–30] and Berger lipid [31] sets. It is worth bearing in mind that lipid force fields are continually under refinement to improve agreement with experimental data [29, 32, 33]. There are many force fields available for simulating membranes [34], and recent efforts towards systematically comparing them and assessing their relative strengths and weaknesses have been reported [35]. (2) Secondly, the protonation states of ionizable side-chains in proteins must be considered. United-atom force fields will give the benefit of reduced computational effort due to reduction in the number of particles, but all-atom models might be preferred in some cases where greater accuracy is required. Various programs exist to calculate the pK_a of ionisable side-chains (PROPKA [36, 37], H++ [38, 39], WHAT IF [23]), several of which also exist as online servers (*see* **Note 1**). A Graphical User Interface (GUI) has recently been developed as a plug-in for VMD [40] to help interpret the results of PROPKA-based pK_a predictions [41]. Most of the programs rely upon calculating an estimate of the free energy (via the thermodynamic cycle) of protonating the residue within its proteinaceous environment. It may be the case that the protonation state is not important, in which case default ionization states at pH 7 are assumed. However, there are examples where the protonation state may be critical as exemplified by the protonation state of Glu71 in KcsA [42–45]. The position of the hydrogens on histidine residues should also be considered carefully, usually by simple visual inspection to optimize local hydrogen bonding.

Finally, although solvation of the system is generally automated, oxygen atoms from water molecules are often included in protein crystal structures. These reflect low-energy minima for a water molecule and thus it is usual to include these prior to "bulk solvation." Deciding whether a water molecule belongs to the subunit of interest from the PDB file is a problem and usually one simply chooses an arbitrary cutoff within which to include these crystallographic waters in the simulation. A cutoff that we have employed in the past has been 4 Å [46–48]. The remainder of the simulation box is solvated with pre-equilibrated water boxes. Although we are typically interested in the protein–lipid interactions it is important to have adequate solvation of the entire protein (*see* **Note 3**).

3.2 Preparation of the Lipid Bilayer

Not all of the methods below rely on a preformed lipid bilayer. However, one invariably will need a lipid-only system for control purposes so simulation of the pure system should be done at some point. The simulation of lipid bilayers has matured over the past 20–25 years and it is beyond the scope of this chapter to cover this in depth. The interested reader is referred to several excellent reviews [34, 49–54] and Chapter 4 of the current volume. Some groups have generously made equilibrated conformations of some lipid bilayer systems freely available (*see* **Note 4**), which provide a good starting point for the main procedure we outline below. Sometimes it will be necessary to generate a new lipid bilayer system from scratch and one then needs a measure of how stable/good that pure system is before proceeding to insert a protein into it. The most commonly used measure of equilibration/stability is to analyze the mean area per lipid; a quantity for which there frequently exists experimental data for which to make a direct comparison to. Furthermore, if this is incorrect then it is likely that most other properties will also be inaccurate [17].

3.3 Setup of the Protein in the Membrane

We focus here on methods currently used in our laboratory but it is worthwhile briefly mentioning alternative methods. Earlier setup methods were developed with the limitations imposed by available computer power at the time. The approach adopted by Woolf and Roux was to build up lipids around the protein, by placing isolated lipid molecules randomly selected from a library of 2000 conformations. The system was adjusted to remove as many overlaps as possible followed by a constrained minimization procedure [55, 56]. Although one does not start from a preformed lipid-bilayer in this case, the conformations in the library will be derived from a simulation of pure lipids.

A different approach was proposed by Faraldo-Gómez and colleagues [57], who used preformed lipid bilayer as the starting point. Their method relies on creating a cavity in pre-equilibrated lipid bilayer using the solvent-accessible surface area of the protein as a template. Lipid molecules whose head groups fall within the volume are removed whilst remaining lipids are subjected to an ever-increasing force acting perpendicular to the surface of the cavity template until the cavity is empty. The protein itself can then simply be inserted into the cavity. This method has the advantage that a preexisting lipid bilayer can be used and in such a way that non-interfacial lipid molecules are not significantly perturbed, which results in a faster equilibration time. The approach was recently updated to incorporate chemical specificity at the protein–lipid interface, with additional features that enable it to handle complex membrane protein topologies [58]. A key advantage of the new method, named GRIFFIN (GRId-based Force Field INput), is that it is a stand-alone tool, enabling it to work alongside any existing molecular simulation package.

More recently, Wompil Im's group have developed a Web-based membrane-builder, which is tightly coupled to the CHARMM force-field [59]. The interface offers great flexibility and provides users with the option to build either a pure lipid membrane or protein–lipid mixture. The user can also select between two insertion protocols. A similar approach by the same group has been applied to aid researchers build micelles [60].

Another approach within the GROMACS simulation suite, is the so-called g_membed program, which allows the efficient insertion of a membrane protein into a membrane with minimal perturbation [61]. In this procedure the protein (positioned in an equilibrated bilayer) is first shrunk in the x–y plane and overlapping lipids are then removed. The protein is then grown back within a short MD simulation pushing the lipids away as it does so. The result is a system that is relatively unperturbed and thus should require shorter equilibration times. Also developed to work with GROMACS, is the pair of packages LAMBADA and InflateGRO2 [62], which work together to generate a protein–lipid system in an efficient and automated manner.

With the advent of more powerful computers, a more direct approach to the setup has become possible and we will focus on this. We have implemented this procedure using GROMACS [63] in combination with VMD [40], but there is also a plugin available http://www.ks.uiuc.edu/Research/vmd/plugins/membrane/ to do all the whole process in VMD. This plugin however is more useful if you intend to use NAMD [64] as your molecular dynamics program. Both methodologies exploit the fact that enough simulation time is available to adequately equilibrate the system. The procedure is quite interactive and can be summarized by the following steps:

1. Obtain a pre-equilibrated lipid bilayer (*see* **Note 4**).
2. Align protein in the lipid bilayer.
3. Remove overlapping lipid molecules.
4. Equilibrate new system.

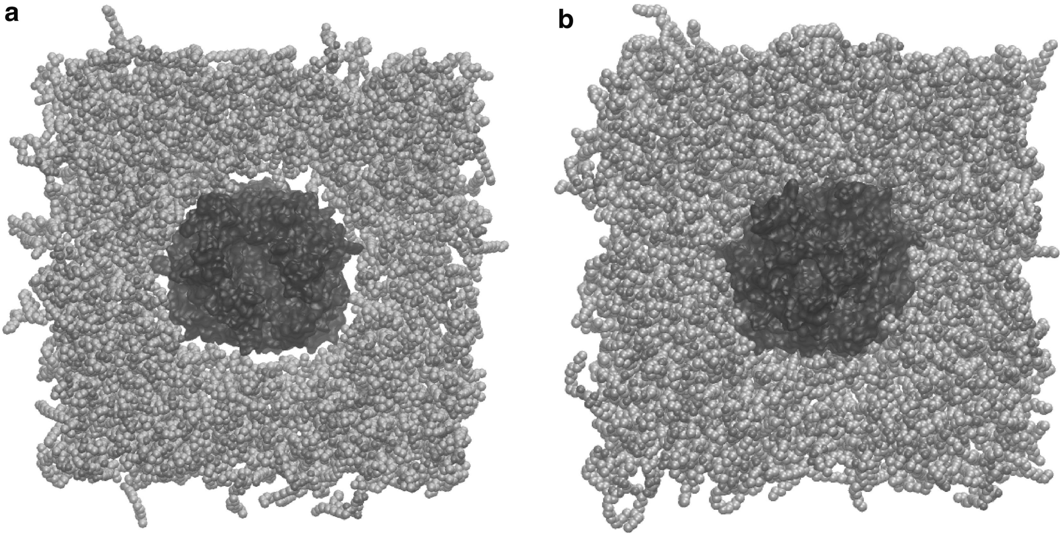


Fig. 1 (**a**) shows the protein BtuB (dark molecular surface), embedded in the bilayer after the removal of overlapping lipids (only protein and lipid are shown in this figure for clarity). Lipid atoms are shown as van der Waals spheres. During the course of the equilibration phase, lipid molecules will move in around the protein as shown in (**b**) which is an equilibrated system

The alignment of the protein with the pre-equilibrated lipid bilayer in this process is essentially something that is done by eye. As mentioned in the introduction, some guidance is afforded by the presence of the tryptophan/tyrosine girdles that are associated with membrane proteins. However, a more objective approach is to make use of the Orientations of Proteins in Membranes (OPM) database [21, 65]. OPM contains coordinates of transmembrane and peripheral proteins and peptides of known structure and their predicted spatial positions with respect to a lipid bilayer (http://opm.phar.umich.edu), based on the Positioning of Proteins in Membrane (PPM) algorithm, which optimizes the transfer energy from water to the membrane. The PPM server (http://opm.phar.umich.edu/server.php) may also be used to position new membrane protein structures [65]. Removal of whole overlapping lipid molecules means that the resulting system will have a vacuum in between the protein and the lipid molecule (*see* Fig. 1a). During the first stage of the equilibration this will be removed as the lipid molecules relax around the protein. Typically for this stage, it is important to keep the protein conformation as close to the starting coordinates as possible. Thus, it is common for positional restraints to be imposed on the protein atoms (or a subset thereof) during this stage. An NPT ensemble (*see* **Note 5**) MD simulation is then performed to allow the lipid molecules to equilibrate around the newly inserted membrane protein (Fig. 1b). During this stage water may penetrate slightly into the vacuum between lipids and the protein. These will be expelled during the course of the simulation as the lipids move towards the protein and the system equilibrates. The length of this equilibration phase is usually determined by monitoring the area per lipid as a function of time. After a period of time (typically between 1 and 3 ns for systems with 512 lipids) one should see this plateau off. This value can be checked against experimental data, although this can be difficult to come by for exactly the same system. Before unconstrained production or further dynamics can be performed it is best to allow the protein to relax in stages. There are many different approaches reported in the literature, which can appear to be rather subjective, but the underlying philosophy is to work back from the backbone of the protein (*see* **Note 6**).

3.4 An Alternative Coarse-Grained Method

A more recent approach has been to simulate the whole system *de novo* from a random arrangement of molecules in the system. Such an approach is made possible via the use of the coarse-grained (CG) methods [16] where small groups of atoms (typically 4) are treated as single particles. Because of the associated reduction in the number of particles, much longer time and length scales can be addressed. This presents the opportunity to simulate large-scale changes in protein–lipid interactions, such as membrane protein bilayer insertion. Thus, the user can, at that point, decide whether the problem requires a switch to a fully atomistic description or whether the CG description is adequate. There are currently ongoing efforts to integrate multi-scale methods into a self-consistent representation [66–69], including hybrid Martini-based approaches [70, 71]. A full discussion of these methods is not possible here, but it seems likely that these methods offer the greatest potential in terms of flexibility across time and length scales for membrane systems.

CG simulations have previously proven useful in modelling the dynamics of lipids and detergents [72–77], DNA [78], proteins [79], and "toy" peptides [16, 80, 81]. A semiquantitative model for lipid systems was devised based on thermodynamic data as well as structural and dynamic properties of atomistic simulations [82]. This model was adapted for application to membrane proteins [83] and similar models have since been developed for related systems by several groups [84]. Marrink and coworkers continue to refine their Martini library of CG parameters for proteins, lipids, and other biomolecules [85], along with Martini-compatible models for polarizable CG water particles [86]. Additionally, improved parameters for more realistic peptide dynamics [87] have also been introduced. A common feature of the basic CG models is that instead of representing every atom in a protein or lipid molecule, approximately four atoms are grouped together into one particle, and are parameterized to capture the hydrophobicity/hydrophilicity, charge and hydrogen-bonding properties of their constituent atoms. Bonds, angles, and the overall backbone secondary/tertiary structure

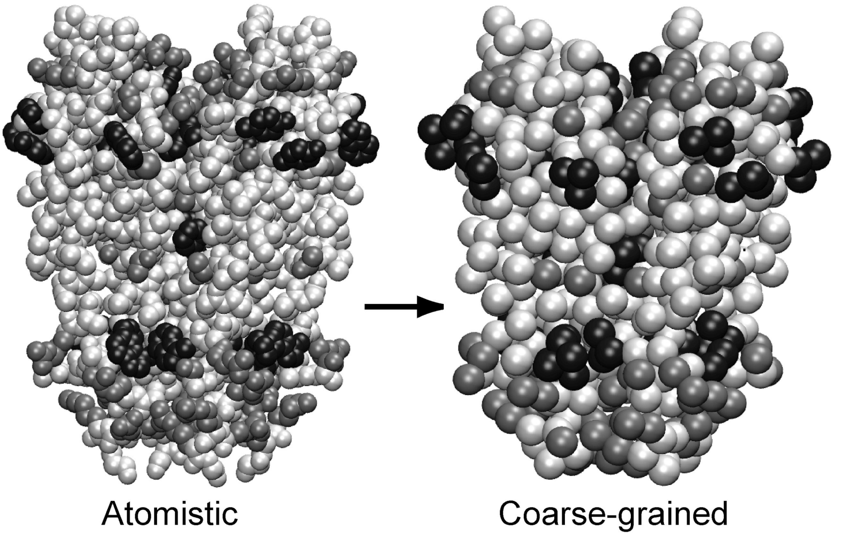


Fig. 2 Illustration of the how the atomistic model translates into the coarse-grained model for the KcsA potassium channel. Aromatic particles are shown as black van der Waals spheres, hydrophobic or backbone particles are shown in *light grey*, and polar/charged particles are shown in *dark grey*

are preserved through either soft harmonic potentials between hydrogen-bonding atoms, through weak dihedral angles, or for larger proteins, elastic network model approaches.

The initial CG model for a protein is normally derived by extracting the coordinates for all Cα atoms and selecting side-chain atoms from the corresponding all-atom protein file. The overall shape and surface area of a lipid or protein molecule is preserved in the model (Fig. 2). Subsequently, lipid molecules taken from a library (derived from for example a pure lipid simulation) are randomly placed in a box containing the protein, before solvation with a pre-equilibrated box of water particles, and neutralizing ions (Fig. 3a). A number of factors must be considered when solvating the protein. Firstly, as with atomistic models, the number of lipid molecules should be such that the bilayer formed is sufficiently large enough to allow plenty of space between the embedded protein and its periodic image within the membrane plane (*see* **Note 3**). This number may be estimated by considering the equilibrium area per lipid of interest. Secondly, a ratio of CG water particles to lipid molecules of ~10–25 should be used to favor the formation of a bilayer, rather than for example hexagonal, micellar, or other nonlamellar phases [82].

Once the system has been prepared, a short energy minimization procedure is carried out, before one or more production runs are performed. A typical such CG simulation will normally be about 2 orders of magnitude faster than the corresponding atomistic simulation, as a consequence of the reduced number of particles, softer potentials (and thus longer MD integration timestep),

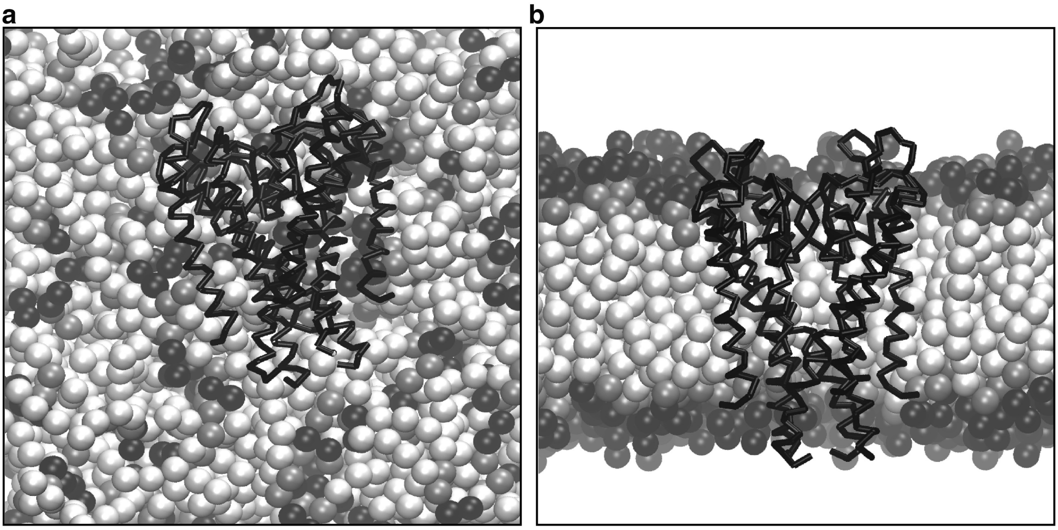


Fig. 3 (**a**) Shows the random starting configuration of the coarse-grained simulation of KcsA with dipalmitoylphosphatidylcholine (DPPC). KcsA is drawn as a black backbone trace. Lipid acyl chain particles are drawn as *light grey* van der Waals spheres, glycerol backbone particles are shown in *dark grey*, and lipid headgroups (including the phosphates) are drawn as *black spheres*. Water molecules are not shown for clarity. (**b**) is the configuration after 200 ns, which clearly shows that the system has evolved into a bilayer arrangement with KcsA embedded within it

and consideration of only short-range non-bonded interactions. Our experience with ~40 different membrane proteins suggests that a period of ~0.1–0.2 μs is normally sufficient to allow the self-assembly of a phospholipid bilayer around the protein of interest. For a typical system of ~6,000 CG particles, this translates to a CPU time of ~1 day on a typical workstation computer [83]. The simulation proceeds via an initial assembly of lipids into a continuous lamellar phase, with a lipid "stalk" which bridges between the bilayer and its periodic image. Eventually this "stalk" is broken to form a defect-free bilayer with the membrane protein correctly inserted (Fig. 3b).

This CG method for bilayer insertion has been successfully applied to a number of membrane proteins [88], and shown to agree in terms of lipid–protein interactions with extended atomistic simulations of an eight-stranded β-barrel, OmpA, a transmembrane α-helical dimer, Glycophorin A [83], and a 12-transmembrane α-helix bundle, LacY [83]. It has also been extensively tested against a number of α-helical membrane peptides and proteins and shown to be in good agreement with experimental data in terms of orientation within the bilayer [83]. Recent successful applications have been reported for a wide range of membrane proteins using the Martini model [89], including large-scale assembly of multiple proteins in membranes, such as for G-protein coupled receptors (GPCRs) [90, 91], and even realistic membrane protein structural transitions, such as mechanosensitive channels [92–94]. Moreover, several groups continue to expand the applicability of CG models, by introducing biologically realistic lipid/protein membrane mixtures [95] that are often crowded [96–98] and sometimes feature functionally important raft assemblies or domains [49, 99].

There is now also a database of all membrane proteins [100] that have been pumped through this procedure, which provides a good initial start point for anyone wanting to begin a membrane protein simulation. The primary advantage of the CG approach to membrane insertion is the elimination of the need for user input, e.g., using the aromatic girdles to guide placement. This is particularly useful for proteins that might be tilted with respect to the bilayer normal, such as the Vpu α-helical fragment from HIV-1 [83]. This is also true of proteins which are nonuniform in their transmembrane distribution, e.g., the coat protein from fd phage which contains an amphipathic in-plane helix thought to reside at the membrane–water interface [83]; monotopic proteins which sit on the surface of the bilayer; or proteins with large extracellular regions such as the multi-domain ABC transporter family. Finally, self-assembly simulations of complex, atypical membrane proteins such as the highly charged voltage sensor domain from a potassium channel reveal that considerable local bilayer deformation may be necessary for insertion, rather than a bilayer of fixed and uniform thickness surrounding the protein [83].

Once the protein has stably inserted within the membrane, it may be desirable to convert from the CG representation back to an atomistic level of detail. The most obvious method for achieving this is to use the CG results as a "rough guide" for positioning the atomistic protein into a bilayer via the method detailed above involving placement into a pre-equilibrated bilayer before removal of overlapping lipid molecules. For example, the peaks in densities of the headgroups along the membrane normal in the atomistic preformed bilayer may be matched up with the CG system, before least-squares fitting the atomistic protein Cα atoms onto the backbone of the CG model, or alternatively using homology modelling techniques. For more complex proteins, such as those which are significantly tilted or which induce local bilayer deformation, a more direct matching of atomistic and CG coordinates may be necessary. Recently, a method has been developed [101] whereby atomistic lipid structures from a pure lipid simulation library are iteratively least-squares fitted onto their CG lipid counterparts, with each of the best matching molecules being retained. Thus, an atomistic bilayer resembling the CG system is gradually built up around the atomistic protein model, which is again obtained by fitting the Cα atoms onto the backbone of the CG protein. Alternatively, Marrink and coworkers presented a promising approach in which atomistic and CG representations of a system are initially coupled via harmonic restraints [102]. Following a simulated annealing protocol, the coupling is gradually removed to achieve the final, relaxed atomistic model.

3.5 Running the Simulation

The last step is to actually run your atomistic simulation. The primary emphasis has been on using parameters and ensembles that best reproduce the properties of lipid bilayers in the absence of proteins. A full review of these considerations is beyond the scope of this chapter, but the interested reader is referred to several articles that discuss sources of error and the best choice of parameters in membrane simulations [17, 103–105].

There are many properties that one could check in the simulation, but probably the most useful is the area per lipid, which gives an indication of molecular packing and the membrane fluidity. It is also a property that is sensitive to simulation set up whilst also being a reasonably reliable indicator that other properties will also be correct. It is important to remember here what your question is—large undulations across large membrane patches will require much longer simulation time than a study of water-headgroup interactions for example.

Finally, there are practical considerations such as disk-space and storage of very large trajectories (*see* **Note 7**), a problem that is presumably going to parallel the increase in computer power.

4 Conclusions

We have discussed two approaches that can be used to set up and perform molecular dynamics simulations of membrane proteins. The advantage of the first atomistic approach is that it is easy to use and generally applicable. A disadvantage of this approach is that to some extent it depends on a subjective positioning of the protein within the bilayer in terms of its overall tilt and its disposition along the bilayer normal. The second approach, via the use of coarse-grain methodologies, allows one to circumvent these problems. The combination of both of these methodologies allows one to explore a wide range of time and length scales with respect to membrane proteins and should provide valuable information on their structure and function.

5 Notes

1. There is also an online server version of the What-If program (http://swift.cmbi.ru.nl/servers/html/index.html) that provides useful tools features to rebuild missing atoms in side-chains. Stereochemical checking tools are also available at this site (useful if you are starting from a model). Similarly, online servers now exist for pK_a calculations of ionisable side-chains,

Table 1
Lipid configurations available for download

PI	URL	Lipids
Scott Feller	http://www.lipid.wabash.edu/	POPC,DOPC, DPPC, SDPC
Helmut Heller	http://heller.userweb.mwn.de/membrane/membrane.html	POPC
Wonpil Im	http://www.charmm-gui.org/?doc=input/membrane	Many combinations possible
Mikko Karttunen	http://www.softsimu.net/downloads.shtml	DMTAP, DMP, DPPC
Peter Tieleman	http://wcm.ucalgary.ca/tieleman/downloads	DPC micelles, POPC, DMPC, DPPC, PLPC
Alexander Lyubartsev	http://people.su.se/~jjm/Stockholm_Lipids/Downloads.html	Various "Stockholm" lipids
Jochen Hub	http://cmb.bio.uni-goettingen.de/downloads.html	Lipid patches with cholesterol
Oliver Beckstein	http://lipidbook.bioch.ox.ac.uk/	Various

including H++ (http://biophysics.cs.vt.edu/H++) and PROPKA (http://propka.ki.ku.dk/).

2. Typical capping groups are an acetyl on the N-terminus (to give CH3-CO-NH2—protein or amidation at the C-terminus to give protein-CO-NH2. These can be added with a molecule-building program such as Pymol [22]. These additional groups are either treated as separated residues (as is case for GROMACS [63]) or as patches (as is the case for CHARMM [106]).
3. In periodic systems (nearly all lipid bilayer simulations will be periodic), it is important to make sure that the parts of the protein that are not in the bilayer are adequately solvated to ensure that the protein near one edge of the box does not "see" itself in the nearest periodic image. To avoid such problems, we have typically set up the system such that there is a 10 Å of water between the protein and its nearest box edge.
4. Some groups have made freely available their coordinates of pre-equilibrated lipid bilayers and these provide a useful start point. Some that are available at the time of writing are summarized in Table 1. Although some of these sites will also contain various parameter sets for download, Beckstein and colleagues [107]

have developed an online database for different lipid types and different force-fields. It is available at http://lipidbook.bioch.ox.ac.uk.

5. NPT refers to the thermodynamic ensemble used. In this case, constant number of particles (*N*), constant pressure (*P*) and constant temperature (*T*). This allows the volume of the system to change and hence the surface area of the lipid which can then be compared back to experiment as a measure of simulation quality.
6. Typically, the protein is relaxed in steps. For example there may be a period during which backbone atoms only are constrained, followed by just Cα atoms, following by no constraints during the production phase of the simulation.
7. An issue that requires constant revisiting is how often one writes simulation frames to the trajectory file. The problem is compounded by two factors: (a) the ever increasing size of system that can be reasonably addressed (currently routinely up to 200,000 atoms) and (b) the length of simulation time (of the order of tens of nanoseconds). A reasonable value for atomistic simulations is to write to disk every 5 ps, but again this will depend on what question you are trying to address.

Acknowledgements

We thank the Leverhulme Trust and Unilever for support and Dr Jorge Pikunic for the BtuB coordinates and useful discussions.

References

1. Wallin E, von Heijne G (1998) Genome-wide analysis of integral membrane proteins from eubacterial, archean, and eukaryotic organisms. Protein Sci 7:1029–1038
2. Terstappen GC, Reggiani A (2001) In silico research in drug discovery. Trends Pharmacol Sci 22:23–26
3. Lemieux MJ, Huang Y, Wang DN (2004) The structural basis of substrate translocation by the Escherichia coli glycerol-3-phosphate transporter: a member of the major facilitator superfamily. Curr Opin Struct Biol 14: 405–412
4. Guan L, Kaback HR (2006) Lessons from lactose permease. Annu Rev Biophys Biomol Struct 35:67–91
5. Gether U, Andersen PH, Larsson OM, Schousboe A (2006) Neurotransmitter transporters: molecular function of important drug targets. Trends Pharmacol Sci 27: 375–383
6. Ash WL, Zlomislic MR, Oloo EO, Tieleman DP (2004) Computer simulations of membrane proteins. Biochem Biophys Acta 1666:158–189
7. Sperotto MM, May S, Baumgaertner A (2006) Modelling of proteins in membranes. Chem Phys Lipids 141:2–29
8. Beckstein O, Biggin PC, Bond P, Bright JN, Domene C, Grottesi A et al (2003) Ion channel gating: insights via molecular simulations. FEBS Lett 555:85–90
9. Gumbart J, Wang Y, Aksimentiev A, Tajkhorshid E, Schulten K (2005) Molecular dynamics simulations of proteins in lipid bilayers. Curr Opin Struct Biol 15:423–431
10. Roux B (2005) Ion conduction and selectivity in K(+) channels. Annu Rev Biophys Biomol Struct 34:153–171
11. Bond PJ, Sansom MSP (2004) The simulation approach to bacterial outer membrane proteins. Mol Memb Biol 21:151–161

12. Beckstein O, Biggin PC, Sansom MSP (2001) A hydrophobic gating mechanism for nanopores. J Phys Chem B 105:12902–12905
13. Beckstein O, Sansom MSP (2006) A hydrophobic gate in an ion channel: the closed state of the nicotinic acetylcholine receptor. Phys Biol 3:147–159
14. Arinaminpathy Y, Biggin PC, Shrivastava IH, Sansom MSP (2003) A prokaryotic glutamate receptor: homology modelling and molecular dynamics simulations of GluR0. FEBS Lett 553:321–327
15. Ohkubo YZ, Pogorelov TV, Arcario MJ, Christensen GA, Tajkhorshid E (2012) Accelerating membrane insertion of peripheral proteins with a novel membrane mimetic model. Biophys J 102(9):2130–2139
16. Nielsen SO, Lopez CF, Ivanov I, Moore PB, Shelley JC, Klein ML (2004) Transmembrane peptide-induced lipid sorting and mechanism of Lα-to-inverted phase transition using coarse-grain molecular dynamics. Biophys J 87(4): 2107–2115
17. Anézo C, de Vries AH, Hoeltje H-D, Tieleman DP, Marrink SJ (2003) Methodological issues in lipid bilayer simulations. J Phys Chem B 107:9424–9433
18. Ulmschneider MB, Sansom MSP, Di Nola A (2005) Properties of integral membrane protein structures: derivation of an implicit membrane potential. Proteins 59:252–265
19. Basyn F, Charloteaux B, Thomas A, Brasseur R (2001) Prediction of membrane protein orientation in lipid bilayers: a theoretical approach. J Mol Graph Model 20:235–244
20. Tusnady GE, Dosztanyi Z, Simon I (2005) PDB_TM: selection and membrane localization of transmembrane proteins in the protein data bank. Nucleic Acids Res 33: D275–D278
21. Lomize MA, Lomize AL, Pogozheva ID, Mosberg HI (2006) OPM: orientations of protetins in membranes database. Bioinformatics 22:623–625
22. DeLano WL (2004) The PyMOL molecular graphics system. DeLano Scientific LLC, San Carlos, CA
23. Vriend G (1990) A molecular modelling and drug design program. J Mol Graph 8:52–56
24. Fiser A, Sali A (2003) Modeller: generation and refinement of homology-based protein structure models. Methods Enzymol 374:461–491
25. Eswar N, Webb B, Marti-Renom MA, Madhusudhan MS, Eramian D, Shen M-Y, et al. (2006) Comparative protein structure modeling using Modeller. Curr Protoc Bioinformatics, Chapter 5:Unit 5.6
26. Feller SE, MacKerell AD (2000) An improved empirical potential energy function for molecular simulations of phospholipids. J Phys Chem B 104(31):7510–7515
27. Klauda JB, Brooks BR, MacKerell AD Jr, Venable RM, Pastor RW (2005) An ab initio study on the torsional surface of alkanes and its effect on molecular simulations of alkanes and a DPPC bilayer. J Phys Chem B 109(11): 5300–5311
28. Chandrasekhar I, Kastenholz M, Lins RD, Oostenbrink C, Schuler LD, Tieleman DP et al (2003) A consistent potential energy parameter set for lipids: dipalmitoylphosphatidylcholine as a benchmark of the GROMOS96 45A3 force field. Eur Biophys J 32(1):67–77
29. Klauda JB, Venable RM, Freites JA, O'Connor JW, Tobias DJ, Mondragon-Ramirez C et al (2010) Update of the CHARMM all-atom additive force field for lipids: validation on six lipid types. J Phys Chem B 114(23): 7830–7843
30. Schuler LD, Daura X, van Gunsteren WF (2001) An improved GROMOS96 force field for aliphatic hydrocarbons in the condensed phase. J Comput Chem 22(11):1205–1218
31. Berger O, Edholm O, Jahnig F (1997) Molecular dynamics simulations of a fluid bilayer of dipalmitoylphosphatidycholine at full hydration, constant pressure and constant temperature. Biophys J 72:2002–2013
32. Jojart B, Martinek TA (2007) Performance of the general amber force field in modeling aqueous POPC membrane bilayers. J Comput Chem 28(12):2051–2058
33. Poger D, Van Gunsteren WF, Mark AE (2010) A new force field for simulating phosphatidylcholine bilayers. J Comput Chem 31(6): 1117–1125
34. Lyubartsev AP, Rabinovich AL (2011) Recent development in computer simulations of lipid bilayers. Soft Matter 7:25–39
35. Piggot TJ, Pineiro A, Khalid S (2012) Molecular dynamics simulations of phosphatidylcholine membranes: a comparative force field study. J Chem Theory Comput 8: 4593–4609
36. Li H, Robertson AD, Jensen JH (2005) Very fast empirical prediction and interpretation of protein pKa values. Proteins 61:704–721
37. Olsson MHM, Sondergaard CR, Rostkowski M, Jensen JH (2011) PROPKA3: consistent treatment of internal and surface residues in empirical pK(a) predictions. J Chem Theory Comput 7(2):525–537
38. Gordon JC, Myers JB, Folta T, Shoja V, Heath LS, Onufriev A (2005) H++: a server for estimating pKas and adding missing hydrogens to macromolecules. Nucleic Acids Res 33: W368–W371
39. Anandakrishnan R, Aguilar B, Onufriev AV (2012) H++ 3.0: automating pK prediction

and the preparation of biomolecular structures for atomistic molecular modeling and simulations. Nucleic Acids Res 40(Web Server issue), W537–W541.

40. Humphrey W, Dalke A, Schulten K (1996) VMD—visual molecular dynamics. J Mol Graph 14:33–38
41. Rostkowski M, Olsson MH, Sondergaard CR, Jensen JH (2011) Graphical analysis of pH-dependent properties of proteins predicted using PROPKA. BMC Struct Biol 11:6
42. Luzhkov VB, Åqvist J (2000) A computational study of ion binding and protonation states in the KcsA potassium channel. Biochim Biophys Acta 1481:360–370
43. Ranatunga KM, Shrivastava IH, Smith GR, Sansom MSP (2001) Side-chain ionization states in a potassium channel. Biophys J 80: 1210–1219
44. Bernèche S, Roux B (2002) The ionizastion state and the conformation of Glu-71 in the KcsA K(+) channel. Biophys J 82:772–780
45. Cordero-Morales JF, Cuello LG, Zhao Y, Jogini V, Cortes DM, Roux B et al (2006) Molecular determinants of gating at the potassium channel selectivity filter. Nat Struct Biol 13:319–322
46. Arinaminpathy Y, Sansom MSP, Biggin PC (2002) Molecular dynamics simulations of the ligand-binding domain of the ionotropic glutamate receptor GluR2. Biophys J 82:676–683
47. Arinaminpathy Y, Sansom MSP, Biggin PC (2006) Binding site flexibility: molecular simulation of partial and full agonists within a glutamate receptor. Mol Pharm 69:11–18
48. Kaye LS, Sansom MSP, Biggin PC (2006) Molecular dynamics simulations of an NMDA receptor. J Biol Chem 281:12736–12742
49. Bennett WF, Tieleman DP (2013) Computer simulations of lipid membrane domains. Biochim Biophys Acta 1828(8):1765–1776
50. Berkowitz ML (2009) Detailed molecular dynamics simulations of model biological membranes containing cholesterol. Biochim Biophys Acta Biomembr 1788(1):86–96
51. Feller SE (2000) Molecular dynamics simulations of lipid bilayers. Curr Opinion Coll Interface Sci 5:217–223
52. Mouritsen OG, Jorgensen K (1997) Small-scale lipid-membrane structure: simulation versus experiment. Curr Opin Struct Biol 7:518–527
53. Scott HL (2002) Modeling the lipid component of membranes. Curr Opin Struct Biol 12:495–502
54. Tieleman DP, Marrink SJ, Berendsen HJC (1997) A computer perspective of membranes: molecular dynamics studies of lipid bilayer systems. Biochim Biophys Acta 1331:235–270
55. Belohorcova K, Davis JH, Woolf TB, Roux B (1997) Structure and dynamics of an amphiphilic peptide in a lipid bilayer: a molecular dynamics study. Biophys J 73:3039–3055
56. Woolf TB, Roux B (1996) Structure, energetics, and dynamics of lipid-protein interactions—a molecular-dynamics study of the gramicidin-A channel in a DMPC bilayer. Proteins 24: 92–114
57. Faraldo-Gómez JD, Smith GR, Sansom MSP (2002) Setting up and optimization of membrane protein simulations. Eur Biophys J 31: 217–227
58. Staritzbichler R, Anselmi C, Forrest LR, Faraldo-Gomez JD (2011) GRIFFIN: A versatile methodology for optimization of protein-lipid interfaces for membrane protein simulations. J Chem Theory Comput 7: 1167–1176
59. Jo S, Kim T, Im W (2007) Automated builder and database of protein/membrane complexes for molecular dynamics simulations. PLoS ONE 2(9):e880
60. Cheng X, Jo S, Lee HS, Klauda JB, Im W (2013) CHARMM-gui micelle builder for pure/mixed micelle and protein/micelle complex systems. J Chem Inf Model 53(8):2171–2180
61. Wolf MG, Hoefling M, Aponte-Santamaría C, Grubmüller H, Groenhof G (2010) g_membed: Efficient insertion of a membrane protein into an equilibrated lipid bilayer with minimal perturbation. J Comput Chem 31(11):2169–2174
62. Schmidt TH, Kandt C (2012) LAMBADA and InflateGRO2: Efficient membrane alignment and insertion of membrane proteins for molecular dynamics simulations. J Chem Inf Model 52(10):2657–2669
63. Lindahl E, Hess B, van der Spoel D (2001) GROMACS 3.0: A package for molecular simulation and trajectory analysis. J Mol Model 7:306–317
64. Phillips JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E et al (2005) Scalable molecular dynamics with NAMD. J Comput Chem 26:1781–1802
65. Lomize MA, Pogozheva ID, Joo H, Mosberg HI, Lomize AL (2012) OPM database and PPM web server: resources for positioning of proteins in membranes. Nucleic Acids Res 40(Database issue):D370–D376
66. Christen M, Van Gunsteren WF (2006) Multigraining: An algorithm for simultaneous fine-grained and coarse-grained simulation of molecular systems. J Chem Phys 124: 154106.1–154106.7

67. Chang R, Ayton GS, Voth GA (2005) Multiscale coupling of mesoscopic and atomistic-level lipid bilayer simulations. J Chem Phys 122:244716
68. Shi Q, Izvekov S, Voth GA (2006) Mixed atomistic and coarse-grained molecular dynamics: simulation of a membrane-bound ion channel. J Phys Chem B 110: 15045–15048
69. Orsi M, Noro MG, Essex JW (2011) Dual-resolution molecular dynamics simulation of antimicrobials in biomembranes. J R Soc Interface 8(59):826–841
70. Rzepiela AJ, Louhivuori M, Peter C, Marrink SJ (2011) Hybrid simulations: combining atomistic and coarse-grained force fields using virtual sites. Phys Chem Chem Phys 13(22): 10437–10448
71. Wassenaar TA, Ingolfsson HI, Priess M, Marrink SJ, Schafer LV (2013) Mixing MARTINI: electrostatic coupling in hybrid atomistic-coarse-grained biomolecular simulations. J Phys Chem B 117(13): 3516–3530
72. Goetz R, Lipowsky R (1998) Computer simulations of bilayer membranes: self-assembly and interfacial tension. J Chem Phys 108(17): 7397–7409
73. Murtola T, Falck E, Patra M, Karttunen M, Vattulainen I (2004) Coarse-grained model for phospholipid/cholesterol bilayer. J Chem Phys 121:9156–9165
74. Shelley JC, Shelley MY, Reeder RC, Bandyopadhyay S, Moore PB, Klein ML (2001) Simulations of phospholipids using a coarse grain model. J Phys Chem B 105(40): 9785–9792
75. Smit B, Hilbers PAJ, Esselink K, Rupert LAM, van Os NM, Schlijper AG (1990) Computer simulations of a water/oil interface in the presence of micelles. Nature 348(6302):624–625
76. Stevens MJ, Hoh JH, Woolf TB (2003) Insights into the molecular mechanism of membrane fusion from simulation: evidence for the association of splayed tails. Phys Rev Lett 91(18):188102
77. Whitehead L, Edge CM, Essex JW (2001) Molecular dynamics simulation of the hydrocarbon region of a biomembrane using a reduced representation model. J Comput Chem 22:1622–1633
78. Tepper HL, Voth GA (2005) A coarse-grained model for double-helix molecules in solution: spontaneous helix formation and equilibrium properties. J Chem Phys 122:124906
79. Tozzini V (2005) Coarse-grained models for proteins. Curr Opin Struct Biol 15(2): 144–150
80. Lopez CF, Nielsen SO, Moore PB, Klein ML (2004) Understanding nature's design for a nanosyringe. Proc Natl Acad Sci U S A 101 (13):4431–4434
81. Venturoli M, Smit B, Sperotto MM (2005) Simulation studies of protein-induced bilayer deformations, and lipid-induced protein tilting, on a mesoscopic model for lipid bilayers with embedded proteins. Biophys J 88(3): 1778–1798
82. Marrink SJ, de Vries AH, Mark AE (2004) Coarse-grained model for semiquantitative lipid simulations. J Phys Chem 108:750–760
83. Bond PJ, Sansom MSP (2006) Insertion and assembly of membrane proteins via simulation. J Am Chem Soc 128:2697–2704
84. Shih AY, Arkhipov A, Freddolino PL, Schulten K (2006) Coarse grained protein-lipid model with application to lipoprotein particles. J Phys Chem B 110(8):3674–3684
85. Marrink SJ, Tieleman DP (2013) Perspective on the Martini model. Chem Soc Rev 42(16): 6801–6822
86. Yesylevskyy SO, Schafer LV, Sengupta D, Marrink SJ (2010) Polarizable water model for the coarse-grained MARTINI force field. PLoS Comput Biol 6(6):e1000810
87. Seo M, Rauscher S, Pomes R, Tieleman DP (2012) Improving internal peptide dynamics in the coarse-grained MARTINI model: toward large-scale simulations of amyloid- and Elastin-like peptides. J Chem Theory Comput 8(5): 1774–1785
88. Scott KA, Bond PJ, Ivetac A, Chetwynd AP, Khalid S, Sansom MSP (2008) Coarse-grained MD simulations of membrane protein-bilayer self-assembly. Structure 16(4):621–630
89. Sengupta D, Marrink SJ (2010) Lipid-mediated interactions tune the association of glycophorin A helix and its disruptive mutants in membranes. Phys Chem Chem Phys 12(40): 12987–12996
90. Periole X, Huber T, Marrink SJ, Sakmar TP (2007) G protein-coupled receptors self-assemble in dynamics simulations of model bilayers. J Am Chem Soc 129(33): 10126–10132
91. Periole X, Knepp AM, Sakmar TP, Marrink SJ, Huber T (2012) Structural determinants of the supramolecular organization of G protein-coupled receptors in bilayers. J Am Chem Soc 134(26):10959–10965
92. Deplazes E, Louhivuori M, Jayatilaka D, Marrink SJ, Corry B (2012) Structural investigation of MscL gating using experimental data and coarse grained MD simulations. PLoS Comput Biol 8(9):e1002683
93. Louhivuori M, Risselada HJ, van der Giessen E, Marrink SJ (2010) Release of content

through mechano-sensitive gates in pressurized liposomes. Proc Natl Acad Sci U S A 107(46): 19856–19860.

94. Samuli Ollila OH, Louhivuori M, Marrink SJ, Vattulainen I (2011) Protein shape change has a major effect on the gating energy of a mechanosensitive channel. Biophys J 100(7): 1651–1659
95. Holdbrook DA, Leung YM, Piggot TJ, Marius P, Williamson PT, Khalid S (2010) Stability and membrane orientation of the fukutin transmembrane domain: a combined multiscale molecular dynamics and circular dichroism study. Biochemistry 49(51): 10796–10802
96. Domanski J, Marrink SJ, Schafer LV (2012) Transmembrane helices can induce domain formation in crowded model membranes. Biochim Biophys Acta 1818(4):984–994
97. Goose JE, Sansom MS (2013) Reduced lateral mobility of lipids and proteins in crowded membranes. PLoS Comput Biol 9(4):e1003033
98. Javanainen M, Hammaren H, Monticelli L, Jeon JH, Miettinen MS, Martinez-Seara H et al (2013) Anomalous and normal diffusion of proteins and lipids in crowded lipid membranes. Faraday Discuss 161:397–417, discussion 419-59
99. Parton DL, Tek A, Baaden M, Sansom MS (2013) Formation of raft-like assemblies within clusters of influenza hemagglutinin observed by MD simulations. PLoS Comput Biol 9(4):e1003034
100. Chetwynd AP, Scott KA, Mokrab Y, Sansom MSP (2008) CGDB: A database of membrane protein/lipid interactions by coarse-grained molecular dynamics simulations. Mol Memb Biol 25(8):662–669
101. Stansfeld PJ, Sansom MSP (2011) From coarse grained to atomistic: a serial multiscale approach to membrane protein simulations. J Chem Theory Comput 7(4):1157–1166
102. Rzepiela AJ, Schafer LV, Goga N, Risselada HJ, De Vries AH, Marrink SJ (2010) Reconstruction of atomistic details from coarse-grained structures. J Comput Chem 31(6):1333–1343
103. Patra M, Karttunen M, Hyvönen MT, Falck E, Lindqvist P, Vattulainen I (2003) Molecular dynamics simulations of lipid bilayers: major artifacts due to truncating electrostatic interactions. Biophys J 84:3636–3645
104. Patra M, Karttunen M, Hyvönen MT, Falck E, Vattulainen I (2004) Lipid bilayers driven to a wrong lane in molecular dynamics simulations by subtle changes in long-range electrostatic interactions. J Phys Chem B 108: 4485–4494
105. de Vries AH, Chandraskhar I, van Gunsteren WF, Hunenberger PH (2005) Molecular dynamics simulations of phospholipid bilayers: influence of artificial periodicity, system size, and simulation time. J Phys Chem B 109:11643–11652
106. MacKerrell AD, Bashford D, Bellott M, Dunbrack RL, Evanseck JD, Field MJ et al (1998) All-atom empirical potential for molecular modeling and dynamics studies of proteins. J Phys Chem B 102:3586–3616
107. Domański J, Stansfeld P, Sansom MP, Beckstein O (2010) Lipidbook: a public repository for force-field parameters used in membrane simulations. J Membr Biol 236(3):255–258

Chapter 6

Membrane-Associated Proteins and Peptides

Marc F. Lensink

Abstract

This chapter discusses the practical aspects of setting up molecular dynamics simulations of membrane-associated proteins and peptides, and the analysis thereof. Topology files for selected lipids are provided and selected analysis tools presented. These include tools for the creation of lipid bilayers of mixed lipid content (**DOPE**) and easy extraction of lipid coordinates (**g_zcoor**, **g_xycoor**), the calculation of helical axes (**g_helixaxis**) and aromatic order parameters (**g_arom**), the determination of peptide- or protein-interacting lipids (**g_under**), and the investigation of lipid-specific interactions through the calculation of lipid-bridged residue–residue contacts (**g_prolip**).

Key words Molecular dynamics, Lipid bilayer, Membrane, Peptide–lipid interaction, Phospholipid, Cholesterol, GROMACS, Helix axis, DOPE, Lipid order parameter, Specific interaction

1 Introduction

The underrepresentation of membrane protein structures in the Protein Data Bank [1] is a direct result of the inherent difficulty of membrane protein crystallization [2], but it stands in sharp contrast with the relevance of membrane proteins to cellular functioning. Roughly 30 % of all genomic sequences encode for membrane proteins, in fact most major processes in the cell are initiated at the membrane surface. Due to the lack of atomic resolution structural data molecular modeling and simulation techniques are expected to play an increasingly relevant role in the study of membrane-related systems. Membrane protein simulations complicate matters with respect to soluble proteins at a number of levels. The generally larger size of membrane proteins makes for slower convergence, the presence of a lipid bilayer imposes a larger system box size, both adding up to longer simulation times. The longer simulation times produce larger trajectory files, which are more complicated and take longer to process. However, the bilayer plane typically aligns with one of the primary system axes, offering an easy frame of reference.

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_6, © Springer Science+Business Media New York 2015

This chapter presents some tips and tricks for the simulation of membrane-bound proteins and peptides. These include simulation setup and the creation of lipid bilayers of mixed content, but also selected analysis tools are presented, that for example allow the easy extraction of coordinates from the trajectory, the calculation of helical axes and aromatic order parameters, the determination of peptide- or protein-interacting lipids, and the investigation of specific protein–lipid interactions through the calculation of lipid-bridged residue–residue contacts. The tips and tricks presented in this chapter refer to the GROMACS [3] suite of programs (*see* **Note 1**), but their principles are generally applicable to other simulation packages. The presented simulation setup and analysis tips are based on two simulation studies: the association of a cationic peptide to a neutral and charged lipid bilayer [4], and the detection of protein–lipid binding in an integral membrane protein [5]. The analysis tools I wrote are either shell-script, or using the GROMACS C programming libraries. Most of these are package-independent since they only require a trajectory file, which is no more than a sequence of structures.

2 System Setup

The general setup of molecular dynamics simulations requires three input files: a structure file, describing the atomic positions of the molecules in the system; a topology file, in which the inter-atomic bonded and non-bonded connections are defined; and a parameter file, supplying the algorithm with the necessary runtime parameters. The first two of these I will briefly discuss. Programs or files in this section are printed in bold typeface.

2.1 The Coordinate File

The coordinate file contains the coordinates of all molecules in the system, in the order in which they appear in the topology file (or vice versa, if you want). Unless the simulation you want to run is based upon previous work, no complete coordinate file exists and you need to create one by combining a PDB file of your protein with a—preferentially equilibrated—lipid bilayer.

2.1.1 Situation A: Placing a Peptide on Top of the Lipid Bilayer

In this case the coordinates of peptide and lipids do not occupy the same space and a simple concatenation of input files suffices.

1. Prepare a PDB file with the coordinates of your peptide.
2. Rotate the structure to adopt the wanted orientation: parallel or perpendicular to the bilayer.
3. Have its geometric center coincide with that of the lipid bilayer and increase the z coordinates (*see* **Note 2**) by 5–6 nm (*see* **Note 3**).

4. Add the peptide coordinates to the (solvated) bilayer box (*see* **Note 4**).
5. Calculate the minimum box height needed and combine this *z* axis with the *x* and *y* axes of the original bilayer.
6. Remove counterions (if any) and solvate the box. In order to avoid water molecules being placed inside the hydrophobic bilayer core, increase the Van der Waals radius of the lipid acyl chain atoms to 6 Å (*see* **Note 5**).
7. Add counterions to make the system electrostatically neutral.

2.1.2 Situation B: Placing a Protein Inside the Lipid Bilayer

Using the procedure above will result in overlapping protein and lipid coordinates. Simply cutting away the excess lipids will not work since the formed vacuum is filled with water molecules, which lead to increased equilibration times. My preferred approach is using **inflategro**, a tool that expands the *x* and *y* coordinates of the bilayer, places the protein in the center and then iteratively reduces the lipid *x* and *y* coordinates towards their original values [6]. An alternative method is through **g_membed** [7]. The following procedure uses the principles of **inflategro**:

1. Prepare a PDB file with the coordinates of your protein.
2. Rotate and translate the structure to adopt the desired orientation (*see* **Note 6**).
3. Place the structure in the center of an equilibrated lipid bilayer, with all waters removed and of which the *x* and *y* coordinates are expanded by a factor of 4.
4. Reduce the system to its reference area per lipid by shrinking the lipid *x* and *y* coordinates iteratively by 2 %, deleting all lipids that have their phosphorus atom at a distance closer than 6 Å to any protein Cα atom. Let each iterative step be followed by 100 steps of steepest descent energy minimization while employing tight (10^5 kJ/nm^2) position restraints on the protein non-hydrogen atoms (*see* **Note** 7).
5. Expand the system box in the *z* direction to avoid the protein being too close to its own copy.
6. Solvate the box, employing increased Van der Waals radii for the lipid acyl chain atoms, and add counterions.

2.2 The Topology File

The common approach in setting up molecular dynamics simulations is to translate a complete coordinate file—containing the atomic coordinates of the bilayer, protein, water and counterions—into a topology using the residue building blocks. In spite of the maturation of molecular dynamics force fields for the simulation of proteins, lipid parameters are not as well integrated into these as one would like. A useful approach therefore is to separate the

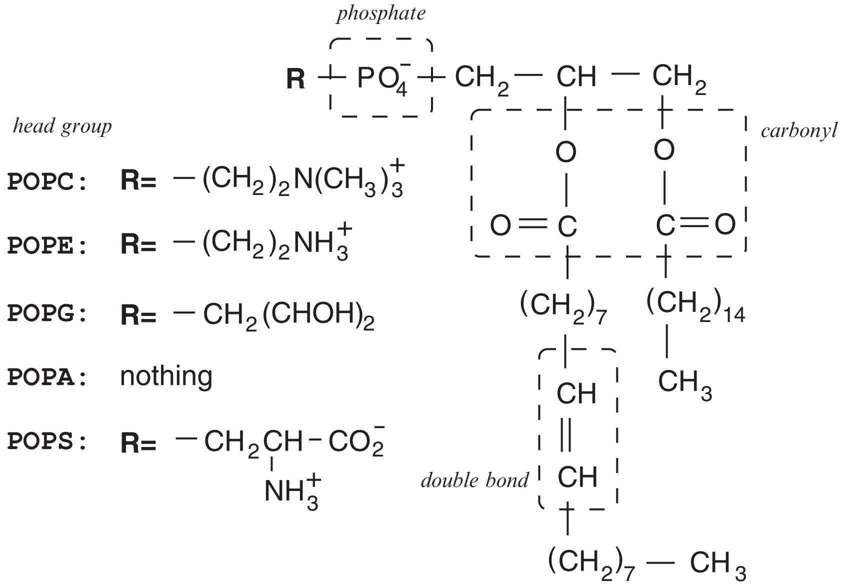

phosphate
head group
carbonyl
double bond
POPC: R= $-(CH_2)_2N(CH_3)_3^+$
POPE: R= $-(CH_2)_2NH_3^+$
POPG: R= $-CH_2(CHOH)_2$
POPA: nothing
POPS: R= $-CH_2CH(NH_3^+)-CO_2^-$

Fig. 1 Molecular structure of selected phospholipids, here with a *sn1* palmitoyl and *sn2* oleoyl tail. The different head groups determine the overall charge of the molecule: PG, PA and PS carry a negative charge, while PE and PC are neutral. The *dashed boxes* indicate commonly defined groups of atoms

creation of topology files for protein and lipids and use a container to combine them. The full procedure then becomes:

1. Extract the coordinates of your protein and process these with **pdb2gmx** to create a topology. In case you have a small peptide, consider capping the C- and/or N-terminus (*see* **Note 8**).
2. Cut everything from the topology file that is not referring to the molecule definition and save the result into a file called "**protein.itp**". This file you can then include at the appropriate position in your global container "**topol.top**".
3. Collect the topological descriptions for the other molecules in the system, i.e., lipids, water, and counterions, and combine the topology files (*see* **Note 9**).

2.3 The Lipid Bilayer

The easiest approach to create a lipid bilayer is to start from one that is publicly made available. A popular bilayer system is one that contains 128 POPC molecules (two leaflets of 64 molecules each). POPC contains a 15-carbon 1-palmitoyl chain and a 17-carbon 2-oleoyl chain, which has a double bond in the middle. The PC head group contains a negatively charged phosphate and a positively charged choline moiety, ensuring electrostatic neutrality and thus avoiding the necessity to include an additional 128 Na^+ counterions. The bilayer structure and topology files are freely available (*see* **Note 10**). Figure 1 shows the molecular structure of POPC and additional phospholipids. The head group function determines the chemistry and overall charge of the lipid.

2.4 Modification of Lipid Topology and Mixed-Lipid Bilayers

The creation of force field parameters is a field on its own. However, in the principle of parameter transfer, one copies parameters from well-calibrated protein residue parts onto a newly created lipid head group function. Charges can be obtained by fitting point charges to reproduce the electrostatic potential following ab initio density functional calculations. In this way one can quite easily produce POPG, POPE, POPS, and POPA topologies [4, 5] (*see* **Note 11**). Modification of the lipid topology files directly happens as follows:

1. Remove all atoms, bonds, pairs, angles, etc. involving the head group from the POPC lipid topology file.
2. Increase/decrease all references to atom number to account for the difference in number of head group atoms between POPC and the new lipid.
3. Add the new head group function to the section listing the atoms (*see* **Note 12**).
4. Add the parameters for the new head group function, copying from chemically equivalent groups in residues (*see* **Note 13**).
5. Add bonded terms (bond, angle, dihedral, and 1–4 pair interaction) for the atoms involving the head group and the connecting phosphate entity (*see* **Note 14**).
6. Perform a simulation of a single lipid in vacuo to check the stability of the structure in the newly created topology (*see* **Note 15**).

An alternative approach is to construct an appropriate rtp entry in the residue topology database and create the lipid topology automatically by processing the single-lipid structure with **pdb2gmx**, or by using public databases such as LipidBook [8] (*see* **Note 16**). However, whatever way you create your topologies, always have in mind that they need to be validated.

The resemblance of the various lipid types in terms of chemical structure may be exploited in order to create mixed bilayers, e.g., bilayers consisting of 80 % POPC and 20 % POPG. Since the acyl tail groups are the same, only the head group needs to be replaced. A step-by-step approach is as follows:

1. Perform a least-squares RMSD fit, fitting the phosphate groups of POPG and POPC onto each other, correctly aligning the oxygen atoms that point towards the head group and the acyl chains.
2. Extract the head group coordinates from the fit and the phosphate and tail coordinates from the reference.
3. Rename tail atoms to correspond to the names in the POPG topology.
4. Copy the coordinates of remaining lipids and add the new head group, phosphate, and acyl tail coordinates with POPG as residue name.

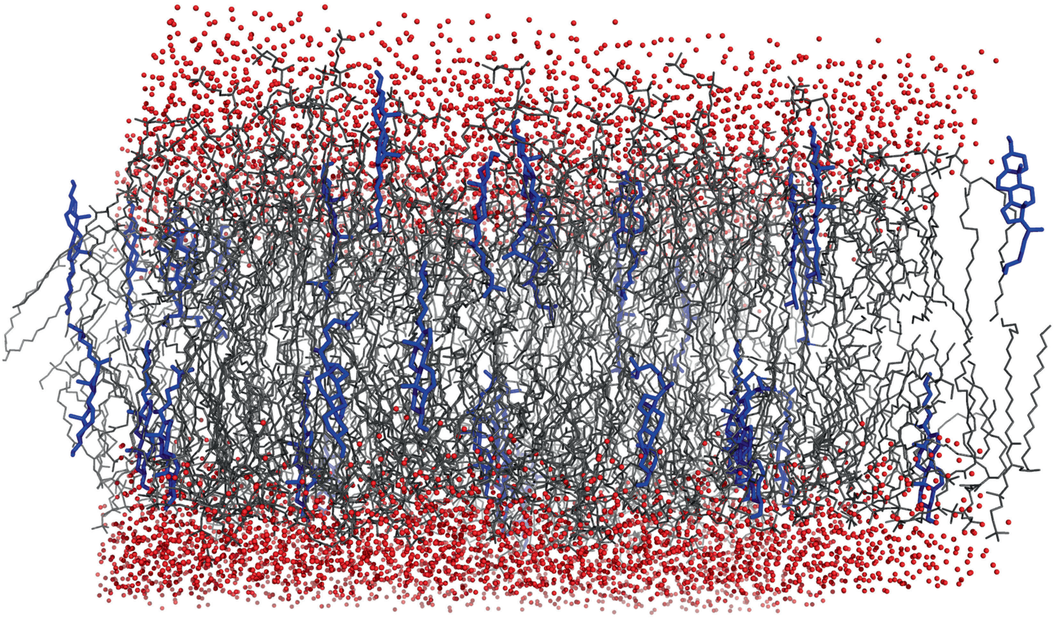


Fig. 2 A solvated bilayer containing 90 % DPPC and 10 % cholesterol molecules. DPPC displayed as *black wireframe*, cholesterol as *blue sticks*, and water molecules as *red spheres*. The configuration was created with **DOPE**, the figure prepared with **PyMol** (The PyMOL Molecular Graphics System, version 1.6.0, Schrödinger, LLC)

5. Repeat the procedure until the desired number of POPC lipids has been replaced.
6. Restore the original box and reorder the file (*see* **Note 9**).

In this way many lipid molecules can be incorporated as long as the new head group and/or tail atoms occupy more or less the same space (*see* **Note 17**). I have written a dedicated program for this purpose, called **DOPE** (*see* **Note 18**), which tries to align a replacement lipid with its target molecule, either best-fitting the molecular axes, or by comparing atom names and chemical types. The program will be described in a separate publication, but may be requested by contacting the author. Briefly, the program iteratively replaces a lipid from the input file by a molecule from a library of structures, aligning the molecule's molecular structure principal axes to the system axes. It subsequently checks overlap with other molecules in the system and accepts the new molecule when no clashes are found. If clashes do occur, the library molecule can be rotated about any of its axes and the new position is re-evaluated. Figure 2 shows a 9:1 DPPC–Cholesterol bilayer created by **DOPE**. The procedure started with an equilibrated 340-lipid POPC bilayer, deleting 2 lipid tail atoms to transform POPC into DPPC, and thereby keeping atomic positions. The system was re-equilibrated using molecular dynamics and in a second **DOPE** step 10 % DPPC molecules were replaced by cholesterol, using an a priori created library of cholesterol conformations.

3 Analysis

The dynamics of a protein is strongly affected by the presence of a lipid bilayer. Backbone hydrogen bond shielding [9] and a decreased dielectric constant in the membrane core [10] promote the formation of secondary structure, both α or β. In addition, the bilayer environment places a restraining force on the protein dynamics due to the decreased fluidity with respect to a soluble environment. Such external forces are characterized by slow-motion displacements of secondary structure elements, readily identified from RMSD plots after fitting to a common reference frame, typically the protein transmembrane domain. An additional frame of reference exists in the surface plane of the bilayer, ignoring eventual curvature effects. The orientation of a protein with respect to the membrane it is binding to is especially relevant in the case of peptide–membrane association.

3.1 Coordinate Frame in Bilayer Simulations, g_zcoor and g_xycoor

By choosing the appropriate reference axes for your simulation you can significantly facilitate subsequent analyses. In general, the *z* axis is made to coincide with the normal to the bilayer surface. Any property involving "distance to bilayer" is then calculated in the *z* axis only, possibly in relation to the bilayer or head group center. The program **g_zcoor** does exactly this: extract the (center-of-mass averaged) *z* coordinate of a combination of molecules and/or atoms.

Orthogonally to this, the *x* and *y* axes describe diffusion in the bilayer plane or surface. The program **g_xycoor** extracts the *x* and *y* coordinates of every single atom in any number of combinations of molecules and/or atoms. Modification of these absolute coordinates into relative ones, e.g., relative to a membrane-inserted protein or membrane-bound peptide, shows the restricted diffusional motion of annular or bound lipid molecules. For both programs, **g_zcoor** and **g_xycoor**, **Note 11** applies.

3.2 Calculation of Helical Axis, g_helixaxis

Hydrophobic mismatch is defined as the difference in hydrophobic length of a protein's transmembrane domain and the thickness of the lipid bilayer it spans. In order to minimize the exposed hydrophobic surface, proteins or peptides may aggregate, or adopt a tilted orientation [11]. The axis of a helix is best determined using a rotational least-squares fitting procedure, mapping the Cα's of the helix onto itself, but one residue out of phase, i.e., residue i is mapped onto residue $i+1$. A quaternion-based method identifies the screw transform (translation along and rotation about the helix axis) that will superimpose the two helices [12]. In the case of a single α-helix containing peptide, the angle between the helix and the normal to the bilayer plane identifies the tilt of the helix in the bilayer. For proteins containing multiple helices in their TM domain, the various angles between those helices provide information as to the internal organization of those helices.

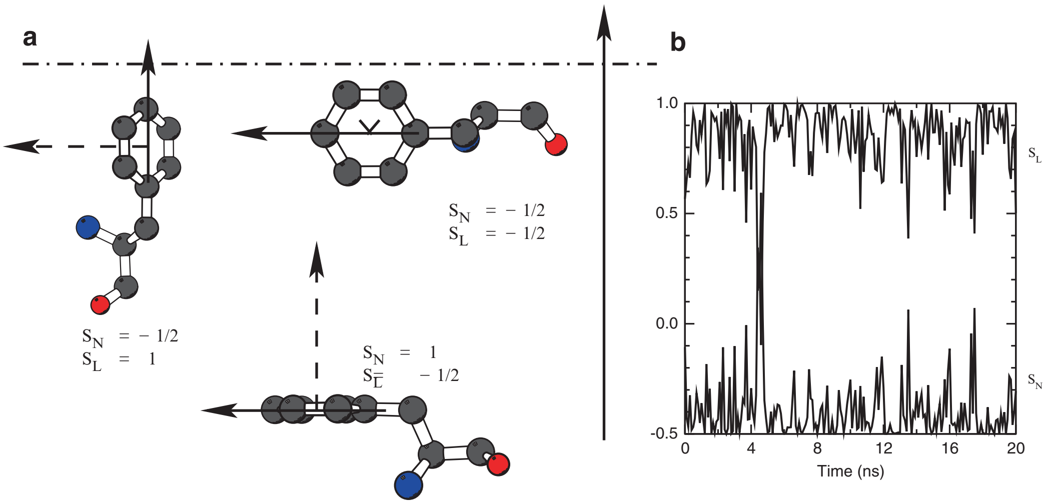


Fig. 3 Aromatic order parameters. (**a**) Visualization of aromatic order parameters. *Solid* and *dashed arrows* represent S_L and S_N, resp. When either *arrow* is aligned with the normal to the bilayer plane (*long arrow*), the respective order parameter equals 1. (**b**) Concomitant behavior of aromatic order parameters, for a single tryptophan residue during a 20 ns molecular dynamics simulation. Both order parameters cannot simultaneously be aligned to the *z* axis (equal 1), but they can be orthogonal to it (equal − ½). Notice the immediate decrease in S_L after an increase of S_N

The rotational least-squares method is fast, accurate and insensitive to noise, and thus able to deal well with imperfect helices. It has been implemented in a program that uses the gromacs development and analysis libraries: **g_helixaxis** (*see* **Note 11**). The program can read trajectory and single structure PDB files and outputs for each helix its angle with the *z*-axis as well as their inter-helical angles. Optionally, the initial point, and vector components and length of each helix are written, in a format following PDB standards and thus easily visualized using standard molecule viewers.

3.3 Orientation of Aromatic Residues, g_arom

Whereas the calculation of helical axes helps in determining the dynamics of secondary structure elements, at a more local level aromatic order parameters are used [13]. The aromatic order parameters S_N and S_L are calculated relative to the normal to the bilayer plane, through the formula $S = ½(3\cos^2\theta - 1)$. S_N relates to the normal to the aromatic ring, whereas S_L describes the vector from Cγ through the ring. θ is the angle between the respective vector and the bilayer normal. For $S = 1$ these vectors are aligned, whereas $S = -½$ means orthogonality. The different orientations and order parameter values are visualized in Fig. 3a. As these vectors cannot both simultaneously be aligned with the bilayer normal, the combinations $S_N = 1$ and $S_L = 1$ are mutually exclusive and an increase in the one induces a decrease in the other. They can, however, both equal –½, meaning that both vectors are orthogonal to the bilayer normal. The mutually exclusive behavior is illustrated in Fig. 3b.

Calculation of the aromatic order parameters has been implemented in an analysis tool that can read both trajectories and PDB files: **g_arom** (*see* **Note 11**). Detection of aromatic residues is automatic, but it is also possible to select the residues of interest. The vectors $C\gamma \rightarrow C\zeta$ for Phe and Tyr, and $C\gamma \rightarrow C\zeta_2$ for Trp, are used for the calculation of S_L, whereas the aromatic plane, defined by the atoms $C\gamma$, $C\varepsilon_1$, and $C\varepsilon_2$ for Phe and Tyr, and $C\gamma$, $C\zeta_2$, and $C\zeta_3$ for Trp, determines the normal to this plane, used in the calculation of S_N.

3.4 Lipids Interacting with a Protein or Peptide, g_under

The interaction between a peptide or protein and a lipid bilayer is not a static quantity that can be defined as the interaction between a certain number of residues and an arbitrary number of lipids. Although membrane-interacting residues may be likely to sustain this interaction once established, lateral lipid diffusion will take place, replacing individual lipids, much like a water molecule interacting through hydrogen bonding with a protein residue may exchange with bulk water.

Here I present a pragmatic way of defining lipids that interact with a peptide or protein, which is purely distance-based. This definition [4] is implemented in the program **g_under** (*see* **Note 11**). Essentially, lipids that come within a certain cutoff distance of the peptide are defined as interacting with this peptide. This distance can be calculated between any two peptide and lipid atoms, but also be restricted to use only backbone atoms. The distance criteria need not be the same in the *x*, *y* or *z* direction. In the case of a peptide hovering above a lipid bilayer, one can picture a cylinder with a radius 0.1 Å and height of 2 nm, hanging below a certain peptide atom. Any lipid atom entering this cylinder will define the lipid as interacting with the peptide (atom). This procedure includes lipids that have their acyl chains under, but the head group besides the peptide (*see* **Note 19**).

The resulting time-dependent evolution of peptide-interacting lipids can subsequently be used to calculate properties involving these lipids only, as opposed to the entire bilayer or bilayer leaflet. Similarly one can define a short cylinder with larger radius, or have its main axis align with the *x* or *y* axis, to investigate properties of lipids neighboring residues of integral membrane proteins.

3.5 Calculating Properties of Interacting Lipids

Most analysis programs calculate a quantity from the interaction between the atomic coordinates of one set of atoms vs. another. Doing so for every frame in the simulation one obtains the evolution of this quantity over time. It is usually—and also in the case of gromacs analysis tools—not possible to vary one of these sets of atoms, as one would need for example in the case of peptide-interacting lipids or when studying a shell of water molecules around an active site.

However, when the quantity to be calculated is cumulative, i.e., the quantity can be calculated a posteriori from each individual lipid molecule at the instantaneous time t, e.g., the average z coordinate of the phosphorus atoms of the protein- or peptide-interacting lipids, or the lipid order parameters, one simply needs to traverse the trajectory and extract—for the first example—the z coordinate for every single lipid. Then in a second step these can be combined with the list of peptide-interacting lipids to get the evolution of average z coordinate of all interacting lipids during the course of the simulation (*see* **Note 20**). Alternatively, one could cut the trajectory in pieces and scan these individually. This has the advantage that for a lipid that becomes interesting during only a fraction of the simulation not the entire trajectory needs to be scanned, but only a (small) part of it (*see* **Note 21**).

3.6 Bilayer Structure

Lipid deuterium order parameters describe the ordering of the lipid acyl chains with respect to the bilayer normal. They can be measured by NMR experiments, but also calculated from the lipid tail C–C dihedral angles [14]. They are expressed as a scalar value per lipid carbon atom that typically ranges between 0 for disordered and 0.5 for ordered lipid structure. The following example shows the calculation of lipid order parameters for lipids that are interacting with a peptide, following the previous section.

1. Calculate which lipids interact with the peptide for every frame in the trajectory.
2. For each lipid tail:
   a. Calculate the lipid order parameters for each individual lipid and lipid tail for every frame in the trajectory. These time frames should match the calculation of peptide-interacting lipids.
   b. For each time frame:
      - Extract the residue numbers of the peptide-interacting lipids.
      - Average the calculated order parameters for the given lipid tail for these lipids at the given time frame.
   c. Average these averaged order parameters over all time frames.

Figure 4 shows the difference in order parameters between interacting (open circles) and non-interacting (solid circles) lipids.

3.7 Lipid-Mediated Salt Bridges

Integral membrane proteins are immersed in a lipid environment and as such, their activity can be directly related to global membrane properties such as fluidity [15]. Anchoring of the proteins occurs through nonspecific protein–lipid interactions [16]. Many of these interactions are hydrophobic, e.g., lipid acyl chains that settle on a hydrophobic surface patch created by one or several

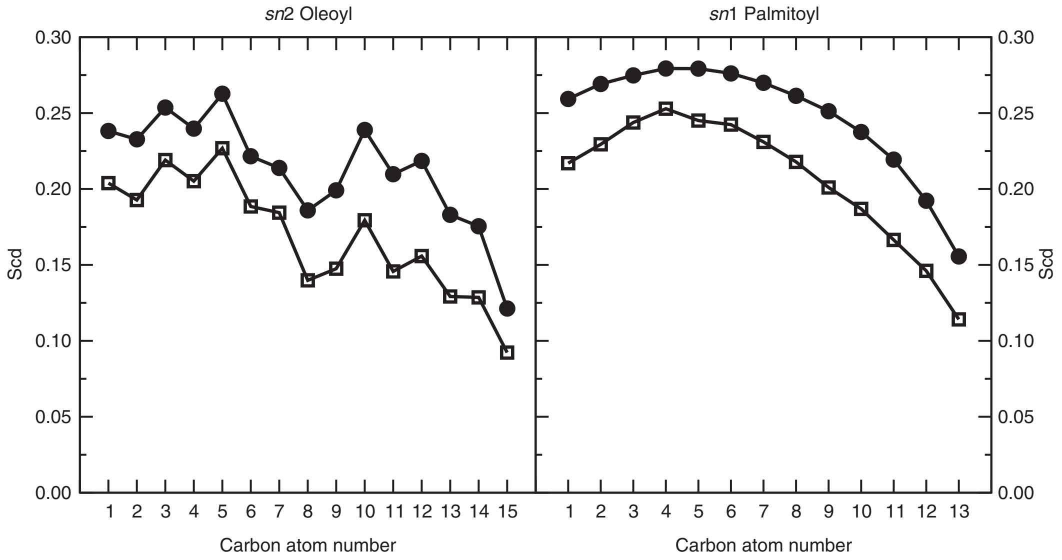


Fig. 4 Lipid deuterium order parameters calculated over a 50 ns molecular dynamics trajectory of a 16-residue peptide bound to a lipid bilayer. *Solid circles* denote order parameters calculated over all lipids, *open squares* are for peptide-interacting lipids only. Peptide-interacting lipids here account for about 12 % of the lipid bilayer (15 lipids), or 20–25 % of the bilayer leaflet

transmembrane helices. But in addition, the lipid carbonyl or phosphate groups can act as acceptor for hydrogen bonds emanating from the protein, or salt bridges may be formed between opposing charges, e.g., between phosphate and arginine or lysine. Hydrophobic interactions are the weakest of these and the strongest interactions are made by salt bridges, which are in fact a particularly strong form of a hydrogen bond. Lipids surrounding the protein and anchoring it in the bilayer are called annular lipids. Annular lipids show increased residence times but exchange with bulk lipids occurs on a regular basis. For sufficiently long simulations such effects can be quantified in the lipid diffusion. However, there is increasing evidence for the existence of non-annular lipid binding sites, where specifically bound lipids are necessary to achieve biological function [17, 18].

Molecular dynamics simulations may detect such strong interactions through the occurrence of lipid-mediated salt bridges, where a single lipid bridges both a negatively and positively charged at the same time. A striking example is a phosphatidylethanolamine (PE) lipid binding simultaneously two neighboring charged residues—D68 and K69—in lactose permease [5]. The binding to K69 is nonspecific—any phospholipid contains a phosphate group—but the binding of the ethanolamine moiety to D68 is PE-specific. In addition, D68 is a highly conserved residue in the Major Facilitator Superfamily, which groups together some 15,000 membrane transporter proteins [19]. The interaction is depicted in Fig. 5.

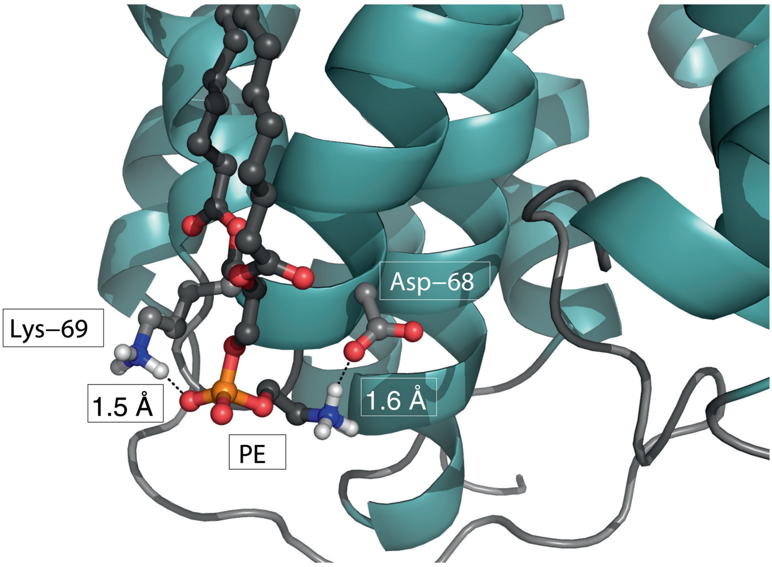


Fig. 5 Example of a lipid-mediated salt bridge. The figure shows a POPE lipid bound to both Asp-68 as well as Lys-69 of lactose permease (LacY). LacY is drawn in *cartoon* representation, the PE lipid and residues 68 and 69 in *ball-and-stick*. The bond lengths displayed are between the hydrogen bond donor and the hydrogen atom itself

Protein–lipid interactions can be investigated with the program **g_prolip** (*see* **Note 11**). The program takes a gromacs trajectory and topology file as input and then traverses this trajectory to look for lipid-bridged residue–residue contacts. More precisely, it detects when during the trajectory the same lipid is bound simultaneously to two different protein residues. By calculating the cumulative presence of a given lipid-mediated salt bridge (*see* **Note 22**) and combining this with the root mean square fluctuation (MSF) of the lipid donor and acceptor atoms (*see* **Note 23**) through the formula

$$F_{\text{atom}} = \Delta t_{\text{cumul}} \times \left(\text{MSF}_{\text{max}} - \text{MSF}_{\text{atom}} + \text{MSF}_{\text{min}}\right),$$

one gets the persistence factor F, which exists for both the donor (F_{donor}) as well as acceptor (F_{acceptor}) interaction. The persistence factor is an indication of the strength of interaction and is typically correlated with residue conservation [5].

3.8 Downloadable Files

The analysis programs

- **g_zcoor**, to plot average *z* coordinate,
- **g_xycoor**, to plot *x* and *y* coordinates,
- **g_helixaxis**, to calculate the axis of a helix,
- **g_arom**, to calculate aromatic order parameters,

- **g_under**, to calculate which lipids interact with a protein or peptide, and
- **g_prolip**, to calculated lipid-bridged residue–residue contacts,

and topology files for POPS, POPC, POPE, and POPG lipids are made available to the scientific community (*see* **Note 11**). Gromacs needs to be installed (*see* **Note 24**), as these programs dynamically link to the gromacs libraries, but to be able to use these programs the simulations need not necessarily be performed by gromacs.

4 Notes

1. http://www.gromacs.org/
2. We assume the *z* axis aligns with the normal to the bilayer plane.
3. A typical bilayer has a thickness of 4–4.5 nm. A 16-residue alpha-helical peptide has a length of about 2.5 nm. If we want to place the perpendicularly to the bilayer plane at a minimum distance of about 2 nm we need to overcome half the bilayer thickness, half the peptide length, and add the extra 2 nm, i.e., translate by at least 5.5 nm.
4. You can use a solvated bilayer box since the solvation procedure will remove overlapping waters.
5. This is the file **vdwradii.dat**, which can be copied from the gromacs topology directory to the working directory.
6. If your protein structure comes from the Protein Data Bank, it likely features in the Orientation of Proteins in Membranes database [20, 21]. The database contains membrane protein structures with a disk of dummy atoms located at the point in the lipid bilayer (at either side) where the hydrophilic to hydrophobic transfer energy derivative maximizes, i.e., roughly at the height of the phosphorus atoms in a phospholipid bilayer. The protein already contains the correct *x* and *y* orientation, so only a translation in the *z* axis is needed.
7. **Steps 3** and **4** can be taken care of by **inflategro**. After about eight iterations, the deflation can be increased to 5 % per step.
8. Capping is generally necessary to avoid artifacts from a terminal charge caused by the artificial chain breaking. Take especially care of capping if the simulations complement experiments where the peptide was capped at one or both ends. Capping is easiest performed using the residue topology database by adding a “residue” with the correct name at the terminus; hydrogens are then added automatically.

Some RMS fitting may be necessary, but the exact position of the cap atoms is not very important because the energy minimization will quickly relax them.

9. The force field definitions (bonded and non-bonded interactions) can be included in any order, but must appear before the molecule type definition. The final section defines the molecules that are present in the system, in this section the order of the molecules must reflect the order in which they are found in the coordinate file.
10. Many such files can be found at http://moose.bio.ucalgary.ca/
11. Files can be downloaded at http://cb.iri.univ-lille1.fr/Users/lensink/lipid/
12. Rename new lipid atoms to avoid overlap with existing ones. This step is not necessary if the new atoms have unique names.
13. If chemically equivalent groups are not available for the force field you are using, you will have to go through the whole process of deriving parameters—especially partial charges—from quantum mechanical calculations, following the procedure as described in the literature for your chosen force field.
14. Also here **Note 13** applies, but at this point the charges are already known. Other parameters are less critical, e.g., angles and dihedrals can be made to follow sp_2 or sp_3 hybridization and bond lengths taken from experimentally determined values (NMR or X-ray). Moreover, in most present-day simulations bond lengths are constrained.
15. Incorrect topologies will quickly explode or collapse. Check the final structure: if it looks okay, it probably is okay. Remove rotational center-of-mass motion to avoid accelerated spinning. Vacuum simulations should be sufficiently long (in the order of several nanoseconds) to allow the dissipation of energy in the limited number of degrees of freedom.
16. http://lipidbook.bioch.ox.ac.uk/
17. Overlap with water is not a problem since the conflicting water molecules can easily be removed, either manually or automatically via the solvation step.
18. Not to be confused with the lipid DOPE (di-oleoyl-phosphatidylethanolamine).
19. Specifically, first every atom that enters the cylinder is calculated and subsequently this group is expanded into full residues. The average distance of the resulting group of atoms to any other group of atoms can be calculated, with the possibility of excluding itself. More concretely, one could calculate the evolution of the distance between the average position of the phosphorus atoms of interacting and non-interacting lipids of one bilayer half during the course of the molecular dynamics simulation.

20. For a 128-lipid bilayer this still means that the trajectory has to be traversed 128 times. When only the peptide-interacting lipids are required, a first step would be the identification of these lipids to avoid unnecessary processing of the trajectory.
21. Scanning of a trajectory file containing all coordinates in the system, including water, may become prohibitively slow for extended simulation times. For many analyses not all coordinates are required and in those cases it is advised to create a copy of the trajectory file, but containing only those coordinates needed for the analysis. This step usually results in a trajectory that is small enough to avoid the necessity of cutting it in pieces.
22. This is defined as the combined fractional presence over the entire simulation and can be calculated through division of the number of frames the bridge is active by the total number of frames in the simulation.
23. This is the root mean square fluctuation of atomic positions, which basically gives information as to how mobile the atom is.
24. The programs compile against gromacs versions 4.5 and 4.6. Compilation against earlier and later versions may require minor adaptation of the code.

References

1. Berman H, Henrick K, Nakamura H, Markley JL (2007) The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data. Nucleic Acids Res 35(Database issue):D301–D303
2. Carpenter EP, Beis K, Cameron AD, Iwata S (2008) Overcoming the challenges of membrane protein crystallography. Curr Opin Struct Biol 18(5):581–586
3. Pronk S, Pall S, Schulz R, Larsson P, Bjelkmar P, Apostolov R et al (2013) GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. Bioinformatics 29(7):845–854
4. Lensink MF, Christiaens B, Vandekerckhove J, Prochiantz A, Rosseneu M (2005) Penetratin-membrane association: W48/R52/W56 shield the peptide from the aqueous phase. Biophys J 88(2):939–952
5. Lensink MF, Govaerts C, Ruysschaert JM (2010) Identification of specific lipid-binding sites in integral membrane proteins. J Biol Chem 285(14):10519–10526
6. Schmidt TH, Kandt C (2012) LAMBADA and InflateGRO2: efficient membrane alignment and insertion of membrane proteins for molecular dynamics simulations. J Chem Inf Model 52(10):2657–2669
7. Wolf MG, Hoefling M, Aponte-Santamaria C, Grubmuller H, Groenhof G (2010) g_membed: efficient insertion of a membrane protein into an equilibrated lipid bilayer with minimal perturbation. J Comput Chem 31(11):2169–2174
8. Domanski J, Stansfeld PJ, Sansom MS, Beckstein O (2010) Lipidbook: a public repository for force-field parameters used in membrane simulations. J Membr Biol 236(3):255–258
9. Garcia AE, Sanbonmatsu KY (2002) Alpha-helical stabilization by side chain shielding of backbone hydrogen bonds. Proc Natl Acad Sci U S A 99(5):2782–2787
10. Avbelj F, Luo P, Baldwin RL (2000) Energetics of the interaction between water and the helical peptide group and its role in determining helix propensities. Proc Natl Acad Sci U S A 97(20): 10786–10791
11. Zhang YP, Lewis RN, Hodges RS, McElhaney RN (1992) Interaction of a peptide model of a hydrophobic transmembrane alpha-helical segment of a membrane protein with phosphatidylcholine bilayers: differential scanning calorimetric and FTIR spectroscopic studies. Biochemistry 31(46):11579–11588
12. Christopher JA, Swanson R, Baldwin TO (1996) Algorithms for finding the axis of a helix:

fast rotational and parametric least-squares methods. Comput Chem 20(3):339–345

13. Tieleman DP, Forrest LR, Sansom MS, Berendsen HJ (1998) Lipid properties and the orientation of aromatic residues in OmpF, influenza M2, and alamethicin systems: molecular dynamics simulations. Biochemistry 37(50):17554–17561
14. Merz KM, Roux B (1996) Biological membranes: a molecular perspective from computation and experiment. Birkhauser, Boston, MA, xiii, 593 p., 1 leaf of plates
15. Vigh L, Escriba PV, Sonnleitner A, Sonnleitner M, Piotto S, Maresca B et al (2005) The significance of lipid composition for membrane activity: new concepts and ways of assessing function. Prog Lipid Res 44(5):303–344
16. Nyholm TK, Ozdirekcan S, Killian JA (2007) How protein transmembrane segments sense the lipid environment. Biochemistry 46(6): 1457–1465
17. Lee AG (2005) How lipids and proteins interact in a membrane: a molecular approach. Mol Biosyst 1(3):203–212
18. Paila YD, Tiwari S, Chattopadhyay A (2009) Are specific nonannular cholesterol binding sites present in G-protein coupled receptors? Biochim Biophys Acta 1788(2):295–302
19. Kaback HR (2005) Structure and mechanism of the lactose permease. C R Biol 328(6):557–567
20. Lomize MA, Lomize AL, Pogozheva ID, Mosberg HI (2006) OPM: orientations of proteins in membranes database. Bioinformatics 22(5):623–625
21. Lomize MA, Pogozheva ID, Joo H, Mosberg HI, Lomize AL (2012) OPM database and PPM web server: resources for positioning of proteins in membranes. Nucleic Acids Res 40(Database issue):D370–D376

Chapter 7

Coarse-Grained Force Fields for Molecular Simulations

Jonathan Barnoud and Luca Monticelli

Abstract

Molecular dynamics (MD) simulations at the atomic scale are a powerful tool to study the structure and dynamics of model biological systems. However, because of their high computational cost, the time and length scales of atomistic simulations are limited. Biologically important processes, such as protein folding, ion channel gating, signal transduction, and membrane remodeling, are difficult to investigate using atomistic simulations. Coarse-graining reduces the computational cost of calculations by reducing the number of degrees of freedom in the model, allowing simulations of larger systems for longer times. In the first part of this chapter we review briefly some of the coarse-grained models available for proteins, focusing on the specific scope of each model. Then we describe in more detail the MARTINI coarse-grained force field, and we illustrate how to set up and run a simulation of a membrane protein using the Gromacs software package. We explain step-by-step the preparation of the protein and the membrane, the insertion of the protein in the membrane, the equilibration of the system, the simulation itself, and the analysis of the trajectory.

Key words Coarse-graining, Molecular dynamics, Force field, MARTINI, Protein, Lipid membrane

1 Introduction

Atomistic molecular dynamics (MD) simulation provides structural and dynamic information on molecular systems on a sub-nanometer length scale, with femtosecond time resolution. It is a powerful tool to interpret experiments, to predict structure and dynamics of simple systems, and to get an insight into processes that are difficult to explore experimentally due to limited length or time resolution. Biologically relevant phenomena, like protein folding, ion channel gating, signal transduction, and membrane remodeling often occur on times scales of microseconds or greater [1]. These time scales are computationally expensive for atomistic MD simulations, which rarely extend beyond the microsecond. Sampling is often an issue: it is not always possible to run long enough or large enough simulations. Therefore, some phenomena are out of reach for state of the art atomistic MD.

Several strategies exist to increase the range of what can be sampled. One strategy is to make better use of modern hardware;

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215, DOI 10.1007/978-1-4939-1465-4_7, © Springer Science+Business Media New York 2015

for instance, the most recent versions of major MD packages support efficient use of graphical processing units (GPU). Another example of efficient use of hardware is Anton, a special purpose supercomputer built by the Shaw group; Anton's microchips were designed specifically to do computations that are frequent in MD calculations [2]. This new hardware can perform millisecond-long MD simulations [3]. Beside hardware improvements, the use of some algorithms can improve the sampling of energy landscapes by overcoming free energy barriers, for example Monte-Carlo simulations, simulated annealing, metadynamics algorithms [4], replica exchange [5], or biased MD algorithms like umbrella sampling [6]. Yet, these algorithms are not suitable for all problems and may require long runs to reach convergence.

Coarse-graining is another approach to tackle sampling issues and size limitations. It consists of reducing the complexity of the simulated system by grouping atoms together into virtual particles. Reducing the number of particles in the simulated system reduces the number of interactions to calculate, and smoothens the energy landscape; hence coarse-grained (CG) simulations are faster. The specific acceleration of CG simulations depends on the details of the models. The way CG models are built depends on the kind of scientific question they aim to address. When reducing the level of detail of the representation it is important not to oversimplify the description of the system, i.e., to preserve the features that drive the phenomenon of interest.

The variety of coarse-grained models is large. Solvent can be represented by explicit particles or by a potential that mimics its effect; the mapping of the system can vary from one particle per protein to several particles per amino acid; the functional forms can be the same as in all-atom simulations or entirely different, for example they can include tabulated potentials.

In general, the range of applicability of coarse-grained models is more limited than for atomistic models, i.e., CG models parameterized to reproduce certain phenomena and properties might fail to reproduce other phenomena and properties. This is due to intrinsic limitations in the transferability of the potentials between different chemical environments, thermodynamic conditions, etc. [7].

2 Theory

2.1 Coarse Grain Models for Proteins

A wide variety of CG models for proteins are currently available. Again, the precise coarse-graining methodology depends on the question to be answered, as the question defines the degrees of freedom that will be removed.

The first coarse-grained models for proteins were meant to tackle the problem of protein folding. In the pioneering work by Warshel and Levitt [8], published in 1975, each amino acid was represented by two spheres: one at the alpha carbon position, and one at the side chain centroid. The torsion angle between two consecutive residues was the only degree of freedom. Since then, several models have been developed to study protein folding and mechanics. These models can be very different from each other in their mapping, in the functional form and in the way their parameters are derived. For instance, the UNRES [9] force field uses anisotropic potentials to model interactions involving amino acid side chains. This model represents the main chain by one interaction site between two alpha carbons. The alpha carbons are used only to join the interactions sites and the side chains. The OPEP force field [10] and the protein force field by Bereau and Deserno [11] have similar mapping: both use a quasi-atomistic main chain and represent the side chain by a single bead. The difference between them lies mainly in the potential form and in target properties used in the parameterization. The PaLaCe force field [12] also uses a quasi-atomistic representation of the protein main chain. Yet, from the main chain, only the alpha carbon is used in the calculation of non-bonded interactions, while the other heavy atoms are used for bonded interactions, avoiding complicated angle and torsion potentials and allowing an explicit treatment of backbone hydrogen bonds. A coarser resolution is adopted for residue-specific non-bonded interactions, with just one or two interaction centers (except for Glycine) at the side-chain level. Besides their different mappings, those models differ in the way that interactions parameters are derived. The choice of the target properties reflects the type of problem to be tackled by each model: folding for OPEP and Deserno's model, structural fluctuations beyond the elastic regime for PaLaCe.

Other coarse-grained force fields do not tackle questions related to protein folding. For instance, Zacharias [13, 14] developed a CG model for protein docking. This model represents proteins as rigid bodies, with each residue modeled by up to three pseudo-atoms. This representation is used in energy minimizations from multiple starting points to find the protein association with the lowest energy [13]. Flexibility of the side chains is taken into account by using several rotamers for each residue; global flexibility can be modeled with energy minimization in low frequency normal modes. The model has been extended to docking of proteins with nucleic acids [14].

Mapping of CG models varies depending on the purpose of the model: when the purpose is to represent motions involving large molecular assemblies (tens of nanometers or more), then the mapping is typically coarser. An example is provided by clathrin self-assembly, studied by Den Otter and Matthews using a very coarse CG model [15, 16]. Clathrins are proteins that self-assemble to form a coat that shapes endocytic vesicles. Clathrin self-assembly involves tens of proteins and a large lipid membrane. Such simulations were made possible by modeling clathrin with only few particles representing the protein's general shape. In this case, small-scale structural details were sacrificed in favor of computational speed.

Multiple levels of coarse graining can be considered in a single simulation; this approach is often referred to as "multi-scale." Over the past decade, several research groups have been developing atomistic/CG multi-scale approaches. Such multi-scale models can consider different molecules at different levels of description in the same system; for example, a membrane protein can be described at the all-atom level while the membrane itself can be coarse-grained [17]. Another multi-scale approach considers different levels of representation within the same molecule; a protein can be described partly with all-atom and partly with CG representation [18, 19], or with several levels of coarse-graining at the same time. For instance, in their simulations of the interactions between actin and myosin, Taylor and Katsimitsoulia [20] used simultaneously three levels of CG representation. At the more detailed level, they represented secondary structure elements as cylinders; then they grouped the cylinders into domains and the domains into proteins. Lower level elements did not interact with each other unless they collided. Such approach allows considering interactions at the lower scale only when it is pertinent. The level of resolution can also be changed "on-the-fly," as proposed in the Adaptive Resolution Scheme (AdResS) by Delle Site and coworkers. The basic theoretical principles of this methodology are illustrated in a recent review [21].

2.2 The Martini Force Field

The Martini force field [22–25] is a popular coarse-grained model [26]. It has been developed to be fast, versatile, and to keep chemical specificity. Carefully tuned building blocks are assembled in a way that allows new molecules to be added to the model without redefining the force field. This makes the force field easily extensible.

Martini was originally developed for lipid and surfactant systems [22]. The force field was then extended to proteins [24, 25], carbohydrates [27], polymers [28, 29], carbon nanoparticles [30, 31] and other molecules [26]. It has been used to study phenomena like vesicle fusion [32], lipid phase transformations [33], or membrane tether formation [34]. Due to its speed, simulated systems can contain up to several million particles [34]. Martini has been built to be usable with Gromacs, but has been implemented in several other simulation engines [35, 36]. Therefore, it can benefit from the performance and the features of the most advanced simulation packages.

On average, MARTINI uses a 4:1 mapping: groups of four heavy atoms (and the associated hydrogen atoms) are represented by one interaction site (bead). Smaller beads are used to represent rings, with a 2:1 mapping. This 4:1 mapping has been chosen as a reasonable compromise between chemical specificity and computational efficiency. Similar mapping can be found in other coarse-grained models [37].

The beads are divided into four main types, depending on their polarity: beads can be polar, intermediate, apolar or charged. In addition, each main type is subdivided into subtypes to represent the hydrogen-bonding character (donor, acceptor, both donor and acceptor, neither donor nor acceptor) or the degree of polarity of the bead (five levels from low polarity to high polarity). This gives a total of 18 different particle types (plus their small counterparts used in ring structures). Each particle is characterized by its interactions with the other particles. These interactions are represented by a Lennard–Jones 12-6 potential with the van der Waals radius $\sigma = 0.47$ nm for regular beads and $\sigma = 0.43$ nm for small beads. The strength of the interaction, represented by the well depth ε_{ij}, ranges from 2.0 to 5.6 kJ/mol; it is scaled by 75 % for small beads. The potential is calculated with a cutoff at 1.2 nm and a shift function starting at 0.9 nm. Electrostatics is represented by a Coulomb potential screened with a relative dielectric constant $\varepsilon_{\text{rel}} = 15$. Like for the Lennard–Jones potential, the Coulomb potential is calculated with a cutoff of 1.2 nm and a shift function. The shift function for the Coulomb potential starts at 0 nm.

Non-bonded interactions were parameterized based on the properties of building block molecules, and particularly the free energy of transfer between water and organic solvents. Free energy of hydration and the free energy of vaporization were also used during the parameterization of the force field. Bonded interactions were parameterized to reproduce distributions of bond lengths, angles, and dihedral angles from atomistic simulations.

The representation of water is consistent with the rest of the model: four water molecules are represented by one Martini bead. A polarizable model of water is also available [38]. In this more complex model, each water particle consists of three beads: a central Lennard–Jones bead and two partially charged beads, with charges of opposite sign, bound to the central Lennard–Jones bead with a constraint. The two charged particles interact with all other beads only via a Coulomb potential (no Lennard–Jones interactions), and interact with each other only through an angle potential, which reduces the dipole to zero (at zero angle the two particles are overlapped) in the absence of an electric field. In the presence of an electric field (generated, for instance, by other charged beads), the water charges are separated and give rise to a dipole. Notice that use of the polarizable water model requires some changes in the Lennard–Jones interaction levels (*see* ref. 38 for details). This polarizable water model allows a better treatment of electrostatics but has a cost in terms of computer efficiency (it is a factor of 3 slower than the non-polarizable model).

MD simulations carried out with the Martini model are stable with integration time steps from 20 to 40 fs (although a time step of 20–25 fs is preferable). The large time step, in addition to the reduced number of degrees of freedom, and the short cutoffs used for non-bonded potentials, lead to a speed-up of 2–3 orders of magnitude compared to atomistic simulations. Time scales of tens of microseconds and longer are computationally affordable.

The Martini model for proteins was introduced in 2008 by Monticelli et al. [24] and updated in 2012 by de Jong et al. [25]. The model was built using the same philosophy as the rest of the Martini force field, so it is compatible with the other Martini molecules. The four-to-one mapping is generally used (two-to-one for ring moieties). One bead represents the main chain of each amino acid, and up to four beads represent the side chain, depending on the size of the amino acid. Particle types for each amino acid were chosen based on experimental measures of water–oil partitioning of side chain homologues. Hence, Martini reproduces accurately the amino acid hydrophobic scale and the partitioning of amino acids in a lipid bilayer, which makes the model especially suitable for membrane protein simulations.

The large radius of the beads imposes a restraint on the minimum distance attainable by beads with opposite charge; this restraint reduces very significantly the maximum strength of the attractive electrostatic interaction. Placing the charges off the center of the Lennard–Jones beads (on an extra particle that has no Lennard–Jones interactions) solves this issue. In version 2.2 of the force field, this technique is used for charged residues, making their mutual attraction more realistic [25].

As described above, the Martini force field reproduces in a realistic way amino acid hydrophobicity, and it features a fairly sophisticated treatment of side chain electrostatics. Nevertheless, Martini cannot be used to predict protein folding, and maintaining a stable protein fold requires restraining the secondary structure. This is mostly due to the very simplified treatment of the protein backbone. In addition, bonded interactions in Martini are tuned to reproduce distances and angles from the Protein Data Bank (PDB), and they are set based on the secondary structure. Once the secondary structure is chosen by the user (for the entire sequence of the protein), it is maintained throughout the simulation, and all bonded interactions can only fluctuate harmonically.

Possible applications of the MARTINI force field are the simulation of protein and lipid self-assembly, large-scale simulations of membrane proteins, and investigations of protein-lipid and protein-protein interactions. For instance, MARTINI can be used to characterize the tilt and orientation of transmembrane helical peptides and their dimerization [39, 40]. In some cases, having fixed the protein secondary structure, it is possible to simulate changes in the protein tertiary structure triggered by external forces. For example, Louhivuori et al. studied the relative motion of the transmembrane segments and gating in the MscL mechano-sensitive channel in a pressurized liposome [41].

Sometimes, restraining the secondary structure elements is not enough to maintain a correctly folded three-dimensional protein structure. In this case, additional restraints to a given protein conformation can be applied with the ELNEDYN elastic network [42]. This approach was shown to reproduce protein large-scale motions. The computational cost of such an elastic network grows with the number of springs involved. The SAHBNET elastic network is an attempt to reduce the number of springs by using information from the hydrogen bond network and the solvent accessible surface of the residues [43].

Constrained secondary structures are one limitation of the Martini force field. Other limitations have to be considered. For instance, the shape of the potential used for non-bonded interactions tend to over-structure fluids compared to atomistic simulations. Other limitations are common to most coarse-grained force fields, as they are due to the reduced number of degrees of freedom. The four-to-one mapping reduces both spatial and chemical resolution. A four bead long alkane can represent a hexadecane molecule as well as a pentadecane or a heptadecane. Because there are less degrees of freedom, entropy is underestimated, and changes in entropy might be represented unrealistically. As the model was parameterized based on free energies, the balance between entropy and enthalpy has to be considered with care, and it is often not correct. Dynamics is also a non-trivial issue. The free energy landscape is smoothened by the coarse-graining, hence sampling is faster and dynamics is also faster. Comparing diffusion of single molecules at the coarse-grained and atomistic resolution, a fourfold speed-up has been estimated on average. Yet, this speed-up factor is not the same for all molecules and depends on simulation conditions.

Despite the limitations above, there is a broad range of applications for the MARTINI force field. Principles and applications of the force field have been recently reviewed by Marrink and Tieleman [26].

In the following, we will describe a simple procedure to simulate a rhodopsin dimer embedded in a lipid bilayer using the Martini force field. None of the steps are specific to this particular protein, so the procedure can be applied to any membrane protein and any lipid membrane.

3 Materials

We will use the Gromacs software package to prepare, run, and analyze an MD simulation. We will use Gromacs version 4.5 [44] (or 4.6 but without GPU acceleration). *See* **Note 1** about how to use the Martini force field with Gromacs 4.6 and the GPU acceleration. Some additional third party scripts and programs need to be installed too. We will display plots with *xmgrace* (http://plasma-gate.weizmann.ac.il/Grace/). The *martinize* script can be found on the "tools" section of the Martini Web site: http://md.chem.rug.nl/cgmartini/index.php/tools2/proteins-and-bilayers. The *g_remove_water* script is available at www.github.com/hublot/g_remove_water, and the *replace_atoms* script at www.github.com/jbarnoud/replace_atoms. DSSP [45] can be downloaded from http://swift.cmbi.ru.nl/gv/dssp/. We assume the programs to be installed and accessible in the default search path, and the DSSP executable to be named *mkdssp*. Extensive documentation on how to install Gromacs can be found on www.gromacs.org.

We will use the Martini force field version 2.2 [25]. All force field files are available on the Martini Web site. We will use the particle description for the normal version of the force field (martini_v2.2.itp). **Note 2** details what to change in case one wants to use the polarizable water model. We will also need the description of lipids (martini_v2.0_lipids.itp), and the description of ions (martini_v2.0_ions.itp). Finally, some structure samples from the Martini Web site will be used: dopc_bilayer.gro and water.gro, that can be found in the "example applications" of the Martini Web site.

The instructions have been tested on GNU/Linux and MacOS X. They should work on Microsoft Windows as well, after a few minor adaptations.

4 Methods

We will describe how to perform a molecular dynamics simulation of a box containing a rhodopsin dimer embedded in a dioleoyl-phosphatidyl-choline (DOPC) bilayer, with explicit water and ions. The final box size will be about 13 nm × 13 nm × 12 nm in the X, Y, and Z dimension, respectively. The membrane will lie in the XY plane of that box. Besides defining the chemical content, starting an MD simulation also requires a description of initial coordinates and velocities of each particle. The initial velocities will be generated automatically to produce a Maxwell distribution. In Gromacs, a so-called "topology" file (TOP) contains all the information on the chemical content of the simulated system (types and number of molecules) and on the force field used to calculate their mutual interactions.

4.1 Preparation of the Protein

We will use a protein structure resolved by X-ray crystallography as a starting point to build a coarse-grained protein structure. The structure of rhodopsin can be downloaded from the Protein Data Bank (ID code 1L9H [46]). Rhodopsin was co-crystallized with several ligands that we will ignore. Also, the first residue of each chain is acetylated, and some residues are missing at the C-terminus; we will ignore those too. Obtaining a "clean" atomistic structure is beyond the scope of the present chapter.

Available on the Martini Web site, the *martinize.py* script does the conversion from atomistic protein structures to their Martini CG version. We will run the script on the PDB file downloaded from the databank; the script will ignore the ligand, so we do not need to edit the file:

```
martinize.py -f 1L9H.pdb -ff martini22 -dssp \
   mkdssp-o CG-1L9H.top -x CG-1L9H.pdb
```

Using the 2.2 version of the Martini force field, the script outputs the CG coordinates in CG-1L9H.pdb and the CG topology file in CG-1L9H.top. The topology file contains links to three files: martini.itp, Protein_A.itp, and Protein_B.itp. The first link is a placeholder for the force field file (that we will replace later with the appropriate file name for the Martini version we want to use), the two other files are generated by the *martinize* script and describe the particles and the connectivity for each protein chain. More precisely, ITP files list all the particles in each protein chain (with particle type and partial charge), and describes how they are bonded together, what are the distance constraints, the angles, and the dihedral angles. The coordinates are written using the PDB format. To convert the coordinate file from PDB format to GRO format, used by Gromacs, we type:

```
trjconv -f CG-1L9H.pdb -s CG-1L9H.pdb -o CG-1L9H.gro
# We select the group 0 (System)
```

The protein structure can be visualized with the VMD software [47]. VMD does not know how to connect coarse-grained beads, but there are several ways to display bonds in coarse-grained structures. One way is to display the protein backbone with the "dynamic bonds" representation. To do so, select the atoms named BB in the "Representation" window and choose "DynamicBonds" with a distance cutoff of 4 Å. This displays an approximation of the protein backbone connectivity; this approximation is typically sufficient to visualize the main features of the protein structure and the secondary structure elements. Another way to display the bonds requires the use of a script named *cg_bonds*, which can be found on the Martini Web site. The script *cg_secondary_structure* allows displaying the secondary structures with the "cartoons" representation of VMD. Figure 1 depicts the Martini CG representation of the protein, side by side with its atomistic structure.

4.2 Preparation of the Membrane

Now that the protein is ready, the next step is to add a lipid bilayer. This can be done with several methods. The main challenge is to get an equilibrated membrane patch by spending little computational resources. The easiest method is to start from a pre-equilibrated patch. Such patches can be found on the Martini Web site for some lipids; the instructions to change the lipid type are

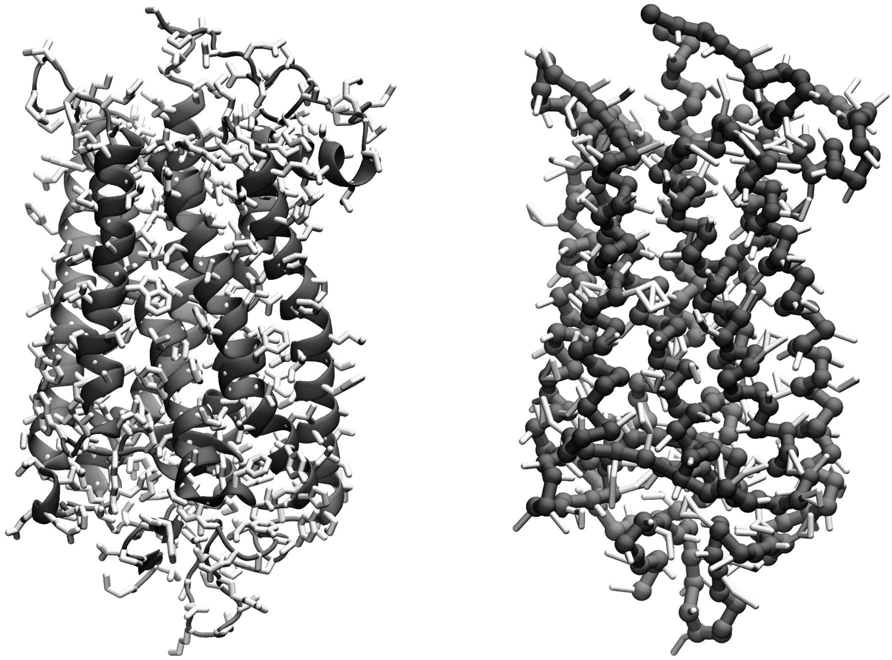


Fig. 1 Chain A of 1L9H at the atomistic (*left*) and coarse-grained (*right*) resolution. The protein backbone is represented in *dark grey* and the side chains in *light grey*. The coarse-grained structure has been drawn using the script *cg_bonds*

given in the Martini tutorial on the same Web site. A membrane can also be built by multiplying a single lipid and translating it on a plane. The second leaflet can be built by rotating the first one (not by mirroring it: the latter transformation would invert the stereochemistry of the lipids). Some automated tools exist, like the VMD [47] or CHARMM-GUI [48] membrane builders, but they do not handle the Martini model. Yet another way to build a membrane patch is to let it self-assemble from randomly placed lipids. This method is computationally expensive on atomistic systems, but easily usable with coarse-grained models. We will use the DOPC patch available on the Martini Web site that has been built by self-assembly. In this patch, leaflets contain different numbers of lipids, and the *X* and *Y* dimensions are different. In some cases, these features can be practically inconvenient as they make the analysis more difficult to interpret. Also, asymmetry of the leaflets can affect some membrane properties, including the lateral pressure profile of the membrane.

The membrane patch contains water and a bilayer of 128 DOPC molecules. The upper leaflet counts 65 lipids while the lower leaflet counts only 63 lipids. To fix the symmetry we will remove two lipids from the upper leaflet. With a text editor, remove lines 3–30 in the dopc_bilayer.gro file. Remove also all water beads; we will add them back later. Water beads have “W” as residue and atom name; *see* **Note 3** on how a GRO file is structured.

The number of particles present in the file is written in the second line. This number has to be exact as Gromacs tools use it. Since we removed atoms, we need to update the number of particles and change it from 3,292 to 1,764. Save the modified file as 126dopc.gro. Finally, to make the patch square, we will rescale the *Y* axis, so it will have the same length as the *X* axis:

```
editconf -scale 1 1.119142862 1 \
   -f 126dopc.gro -o 126dopc_square.gro
```

(Notice that this operation also rescales the coordinates of all particles in the system. We will get back to this below.)

Once rescaled, the box size is 7.02 nm in the *X* and *Y* dimensions. The radius of gyration of the protein is 2.7 nm; the maximum distance between two beads of the protein is about 8 nm, therefore the patch is too small. We can multiply the patch on the *XY* plane by running:

```
genconf -f 126dopc_square.gro \
   -o 504dopc.gro -nbox 2 2 1
```

The patch is now 14.05 nm in the *X* and *Y* dimensions and counts 504 lipids. The box is only 8.4 nm along the *Z*-axis. With such small size, the protein will interact with its periodic images along the *Z* direction, leading to possible artifacts in the simulations. Hence we need to increase the box size in the *Z* dimension. For convenience, we will also have the membrane centered in the box:

```
editconf -f 504dopc.gro -o 504dopc_rebox.gro \
   -box 14.0501014.05010 11 -c
```

By modifying the membrane size (membrane rescaling step, above), we perturbed bond lengths and angles in the lipids. This can result in very high energies and possibly explosion of the system. A simple solution is to perform an energy minimization with the steepest descent algorithm. To this end, we need to download the Martini particles definition and the Martini topology for lipids from the Martini Web site. The files are martini_v2.2.itp and martini_v2.0_lipids.itp. Then we need to write a topology file for the membrane (topol.top) and a parameter file for the energy minimization (param_em.mdp).

```
---- topol.top ----
#include "martini_v2.2.itp"
#include "martini_v2.0_lipids.itp"
[ system ]
Non-hydrated DOPC bilayer
[ molecules ]
```

```
DOPC   504
-------------------
---- param_em.mdp ----
title                        = Martini
integrator                   = steep
nsteps                       = 400
nstlist                      = 10
rlist                        = 1.4
coulombtype                  = Shift
rcoulomb_switch              = 0.0
rcoulomb                     = 1.2
epsilon_r                    = 15
vdw_type                     = Shift
rvdw_switch                  = 0.9
rvdw                         = 1.2
----------------------
```

With Gromacs, energy minimizations and MD simulations are performed in two steps. First a run input file has to be generated by the Gromacs preprocessor program, named *grompp*. *grompp* combines the initial structure (GRO), the topology file (TOP), and the parameter file (MDP; this contains all simulation parameters), into one run input file, named TPR file. Then the simulation engine, named *mdrun*, performs the energy minimization using the TPR file as the only input:

```
grompp -f param_em.mdp -c 504dopc_rebox.gro \
   -p topol.top -o 504dopc_em.tpr
mdrun -deffnm 504dopc_em -v
```

4.3 Protein Insertion in the Membrane

Inserting the protein in the lipid membrane requires two steps. First the protein has to be oriented with respect to the membrane; then it has to be embedded in the lipid bilayer. Orientation consists in rotating and translating the protein so that its transmembrane part overlaps with the membrane. This step can be done manually with visual tools like VMD. One can also refer to databases of already oriented proteins, such as the Orientation of Proteins in Membrane (OPM) [49, 50] database and the Protein Data Bank of Transmembrane Protein (PDBTM) [51–53]. These databases use programs available on the PPM [50] and the TMDET [54] Web servers, respectively. As an alternative, one can also use standalone programs, like Lambada [55], that can be integrated in an automatized workflow. The PPM and TMDET Web servers and Lambada do not handle Martini structures; yet, they can be applied to the atomistic structures before coarse-graining.

Because Martini is parameterized based on partitioning data, and thanks to the acceleration due to coarse-graining, proteins typically rotate and assume reasonable orientations spontaneously in CG simulations within a short time. Therefore, precise manual orientation of the protein is not necessary. Still, by approximately orienting the protein we can save some equilibration time.

We will place the protein at the center of a box of the same size as the box containing the lipid membrane, with its principal axis parallel to the membrane normal.

```
editconf -f CG-1L9H.gro -o orient.gro \
   -box 14.05010 14.05010 11 -princ -rotate 90 0 0
# We select the group 1 (Protein)
```

At this point it is necessary to concatenate the structure file of the oriented protein and the energy-minimized membrane:

```
cat orient.gro 504dopc_em.gro > concat.gro
```

Notice that the first line of a GRO file is the title of the system, the second line is the number of particles and the last line contains the dimensions of the box. These three lines need to appear only once in the concatenated file. The system title is chosen by the user; the box dimensions should be the same in the protein and the membrane structure files, and they should remain the same in the concatenated structure file; the number of particles has to be consistent with the new structure file (there should be now 8,519 particles in the system). We assume the protein to be first in the file, and we call the file concat.gro.

We now need to update the topology file (topol.top), to include the topology for each protein chain. We also change the system title and add the protein chains to the list of the molecules in the system (in the “[molecules]” section). The topol.top file now should look like:

```
---- topol.top ----
#include "martini_v2.2.itp"
#include "martini_v2.0_lipids.itp"
#include "Protein_A.itp"
#include "Protein_B.itp"
[ system ]
Non-hydrated DOPC bilayer and rhodopsin dimer
[ molecules ]
Protein_A   1
Protein_B   1
DOPC      504
-------------------
```

We have the proteins embedded in the lipid bilayer, but the system is not yet usable: some protein particles overlap with some lipid particles, which can cause numerical instability (overlap comes with very high forces) or artifacts (e.g., lipids sticking to the protein, when one of their beads is stuck inside a ring). The easiest way to address the issue of overlap is to remove the overlapping molecules, and then carry out a short MD simulation to equilibrate the lipids around the protein. This method is commonly used. Its main drawback is that often the overlap involves only a few particles in the lipid, while the rest of the molecule can be far from the protein. In this case, removing the entire lipid molecule will result in a hole around the protein, that will require some equilibration. Another method to fix the issue of overlapping lipids is to run an MD simulation with the protein replaced by a repulsive potential. The Gromacs software package includes g_*membed* [56], a tool that makes the protein "grow" progressively (by progressively scaling the protein coordinates), pushing the lipids away. Kandt and Tieleman [57] devised the *inflategro* method. Instead of "growing" the protein, this method expands the membrane, to reduce the number of overlapping lipids. Lipids that still overlap after the expansion are removed, then the membrane is contracted in successive steps until it reaches its initial size. At each contraction step, the energy of the system is minimized. The *inflategro* method has been updated in *inflategro2* [55]. Here the number of lipids to remove in each leaflet is calculated based on the initial area per lipid and the protein cross-sectional area. Only the lipids close to the protein are expended and contracted. Hence, the other lipids remain as they were before the procedure.

To reduce overlap between protein and lipid particles, we will use the *inflategro2* software, available at http://code.google.com/p/inflategro2/. The software involves an iterative procedure that calls *gromacs* to minimize the energy at each contraction step. To this end it requires two additional *gromacs* input files: a parameter file (MDP) and an index file (NDX). For the parameter file, we copy the file param_em.mdp created earlier to a new file named deflate.mdp. In this new file, we set the number of minimization step (nsteps) to 50 and add the following lines at the end:

```
energygrps  = Protein Dynamic Static
energygrp_excl = Protein Protein Static Static \
   Dynamic Static Protein Static
freezegrps  = Protein Static
freezedim  = Y Y Y Y Y Y
```

The index file defines groups of atoms. It can be generated by *make_ndx*. We run the following command:

```
make_ndx -f concat.gro -o index.ndx
```

The program generates some groups automatically, and lists them on the screen; then it prompts the user for further commands. As all the groups we need are generated automatically, so we can type “q” to save and quit. Now, we can run *inflategro2*.

```
inflategro2 -f concat.gro -n index.ndx \
   -p topol.top -m deflate.mdp
```

inflategro2 asks for a group to inflate. This group has to be the one that defines the membrane, so we choose the group called “DOPC” (group 13). As this group already contains the whole membrane we do not need to include other groups to the selection. Then, *inflategro2* prompts for a group that contains the protein.

The last generated structure will be shrink.20.gro. [We may encounter an issue for some simulation systems when using *infltategro2* on multicore computers. *See* **Note 4** on how to fix it.] *inflategro2* updates automatically topol.top; if we have to run the program again (for instance, if execution was interrupted), then we should first check that the number of lipids in topol.top is correct.

The protein is now embedded in the membrane and the lipids are reasonably well placed around it. A short MD simulation will equilibrate the system—i.e., orient the protein correctly and optimize the position of the lipids around the protein. The topology was updated by *inflategro2* to reflect the new number of lipids. The current state of the system is illustrated in Fig. 2.

4.4 Addition of the Solvent

The system is not yet hydrated. A solvent-free version of Martini is planned [26], but the current Martini model requires explicit solvent. So we will add explicit water to our system.

The program *genbox*, included in the Gromacs package, allows to fill the empty space in a box with a solvent; the solvent it taken from an equilibrated box; genbox also removes solvent molecules overlapping with the solute. An equilibrated box of water can be found on the Martini Web site. *genbox* does not know the size of Martini beads, so it needs further information to calculate when beads overlap. We can pass the approximate radius of a Martini water bead as an argument to *genbox*. To hydrate the system, we type:

```
genbox -cp shrink.20.gro -cs water.gro \
   -vdwd 0.23 -o hydrated.gro
```

Some water beads may be inserted in the bilayer. They would go out of the membrane during the equilibration run, but they may remain stuck for a while, which would make the equilibration longer. We can remove the water beads from the hydrophobic part of the membrane using the *g_remove_water* script, available at www.github.com/hublot/g_remove_water.

```
g_remove_water.py--ipid_atom GL1 \
   --ater_atom W -f hydrated.gro \
   -o hydrated2.gro
```

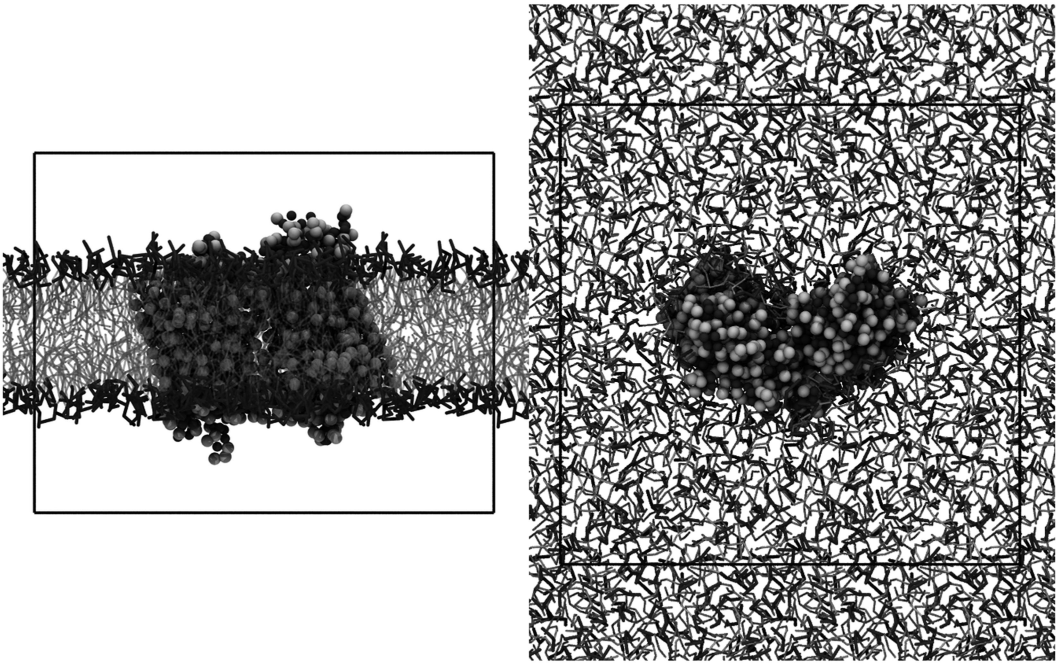


Fig. 2 Membrane protein system viewed from the side (*left*) and from the top (*right*). The protein is represented as *spheres*, with the backbone in *dark grey* and the side chains in *light grey*. The lipids are represented as *licorice*, with the polar head in *dark grey* and the tails in *light grey*. The *black rectangle* shows the border of the simulation box; outside of the box is the periodic image

We notice that the net charge of our system is −3*e*, because the protein is not neutral. To neutralize the system we will add sodium ions. We can replace three water particles with sodium ions by changing the atom name from "W" to "NA+," and the residue name from "W" to "ION" in the structure (GRO) file and in the topology (TOP) file. Notice that, in the GRO file, the alignment of the columns needs to be maintained.

It has been reported that water, in the Martini model, has a melting temperature higher than 0 °C, and it can freeze at room temperature under certain conditions. To address this issue, version 2.0 of the model introduced a new water particle type with a larger radius, which interacts with all non-water beads in exactly the same way as the original water bead, but has slightly different interaction with water beads: the sigma parameter of the Lennard–Jones interaction with water is increased to 0.57 nm; this way water packing is perturbed and freezing at room temperature is avoided. We will replace 5 % of the water particles by these "antifreeze" water particles. As the box should contain about 9,990 water particles—this number can change between two runs of the procedure, we will replace 500 random water particles with antifreeze water, by changing the atom name and the residue name from "W" to "WF." We will also reorder the atoms so that ions and antifreeze beads are grouped; then we name the modified file hydrated3.gro. The *replace_atoms* script automatizes such atom manipulations. Using this script, replacing water beads by ions and antifreeze water can be done with:

```
cat hydrated2.gro | replace_atoms.py -n 3 \
   -o W -r ION -a NA+ | replace_atoms.py \
   -n 500 -o W -r WF -a WF > hydrated3.gro
```

Now that we changed the content of the box, we need to update the topology file. topol.top should include the description of ions (also available on the Martini Web site). The 3 sodium ions, the 500 antifreeze water particles, and the 9,487 remaining water particles (this number can change) should be included in the list of molecules. The topol.top file should now look like:

```
---- topol.top ----
#include "martini_v2.2.itp"
#include "martini_v2.0_lipids.itp"
#include "Protein_A.itp"
#include "Protein_B.itp"
#include "martini_v2.0_ions.itp"
[ system ]
Hydrated DOPC bilayer and rhodopsin dimer
[ molecules ]
Protein_A    1
Protein_B    1
DOPC       467
NA+          3
WF         500
W         9487
-------------------
```

Recently, a program has been made available (on the MARTINI Web site) to build membrane system in an automatic manner. That program is called *insane*, and builds the membrane by repeating a lipid model on a grid with the right area per lipid. The program can also embed a protein in the membrane. Therefore it can replace many of the steps described above. Yet, the system that is built by *insane* is far from equilibrium, and one should carefully monitor the equilibration steps to avoid possible freezing and other artifacts. The *insane* program is available at http://md.chem.rug.nl/cgmartini/index.php/insane.

4.5 Minimize the Energy and Equilibrate

All the components of our systems are now in place, but the system most likely still has high energy, due mostly to non-optimal packing of the lipids and to unfavorable lipid-protein contacts. In addition, because we used a large distance criterion to minimize water overlap when we hydrated the box, the system density is probably too low. As a first step towards equilibration, we will run an energy minimization to reduce unfavorable contacts:

```
grompp -f param_em -c hydrated3.gro \
   -p topol.top -o em
mdrun -deffnm em -v
```

Then we will run a short MD simulation to equilibrate the system density. Like for the energy minimization, we need a parameter file (MDP). We will run the simulation for 20 ns with a timestep of 20 fs. This represents 1,000,000 integration steps. On a modern workstation, this should take about an hour. We will run the simulation at 310 K and 1 bar using the Berendsen weak coupling algorithm for temperature and pressure [58]. This algorithm may not result in a correct kinetic energy distribution, so we will use the Parrinello–Bussi thermostat [59] and the Parrinello–Rahman barostat [60] for the production simulation. The Parrinello–Bussi and Parrinello–Rahman algorithms tend to produce high fluctuations when temperature and pressure are too far from their target values, which makes equilibration longer. Write the param_eq.mdp file as follows:

```
---- param_eq.mdp ----
integrator                = md
dt                        = 0.02
nsteps                    = 1000000
nstcomm                   = 10
nstxout                   = 0
nstvout                   = 0
nstfout                   = 0
nstlog                    = 1000
nstenergy                 = 100
nstxtcout                 = 1000
xtc_precision             = 100
nstlist                   = 10
rlist                     = 1.4
coulombtype               = Shift
rcoulomb_switch           = 0.0
rcoulomb                  = 1.2
epsilon_r                 = 15
vdw_type                  = Shift
rvdw_switch               = 0.9
rvdw                      = 1.2
tcoupl                    = Berendsen
tc-grps                   = System
tau_t                     = 4.0
```

```
ref_t                     = 310
Pcoupl                    = berendsen
Pcoupltype                = semiisotropic
tau_p                     = 4.0
compressibility           = 1e-5 1e-5
ref_p                     = 1.0  1.0
gen_vel                   = yes
gen_temp                  = 310
constraints               = none
constraint_algorithm      = Lincs
lincs_order               = 4
lincs_warnangle           = 30
----------------------
```

Now we can run the equilibration:

```
grompp -f param_eq -c em.gro \
   -p topol.top -o eq
mdrun -deffnm eq -v
```

4.5.1 Is the System Equilibrated?

It is important to verify that the system is equilibrated before starting a production run. To this end, visual inspection is a useful, quick first step. The trajectory can be displayed with VMD [47]. During the first few steps of the trajectory, the protein tilts to find a suitable orientation in the membrane. The box changes size as water become denser and the area per lipid adjusts.

Visual inspection is not sufficient to determine if the system reached equilibrium. At the end of the equilibration run, box dimensions, potential energy and kinetic energy should have converged to stable values. Energies and box dimensions are stored in eq.edr. They can be extracted using *g_energy*, and visualized with *xmgrace*. Gromacs allows decomposing potential energy by groups of atoms, called energy groups. It is then possible to check if the non-bonded interaction between the protein and the lipids converged. Energy groups need to be specified in the simulation parameters before the run, though. See the Gromacs manual on “energy_grp” on how to use energy groups. Other properties like density profile or protein orientation can be checked too.

If the system did not reach equilibrium, the equilibration run should be extended.

4.6 The Production Run

After equilibration, we can run the actual simulation. The simulation parameter file for a production run is similar to the equilibration run, but some details need to be changed: the duration of the run and (possibly, but not necessarily) the temperature and pressure coupling algorithms. We will run the simulation for 1 μs, e.g., 50.000.000 steps, so the nsteps parameter has to be adapted. We will use the Parrinello–Bussi thermostat (v-rescale) and the Parrinello–Rahman barostat. We copy param_eq.mdp to param_md.mdp, and we edit the latter file, which has to contain the following parameters:

```
tcoupl                    = v-rescale
tau_t                     = 1.0
ref_t                     = 310
Pcoupl                    = parrinello-rahman
Pcoupltype                = semiisotropic
tau_p                     = 12.0 12.0
compressibility           = 3e-4  3e-4
ref_p                     = 1.0  1.0
```

We set gen_vel to "no" to start the simulation with the velocities generated during the equilibration as they are written in eq.gro. Then we run the simulation:

```
grompp -f param_md.mdp -c eq.gro \
   -p topol.top -o md
mdrun -deffnm md -v
```

4.7 Analysis

Once the simulation is done, one should again verify that system size and energy do not have any drifts. Again, this can be investigated with *g_energy*.

Analyzing a Martini trajectory is not different from analyzing any other MD trajectory. For example, one can look at the root mean square deviation with *g_rms* or the root mean square fluctuations with g_rmsf (*see* Fig. 3).

Some Gromacs-related tools will prompt for a group to process, and you may want to run some analyses on the protein main chain. In Martini, beads from the main chain are typically named "BB," so you can create group for them by using "a BB" as a command in *make_ndx*.

5 Notes

1. Gromacs 4.6 features an improved support for graphics processing units (GPU). This support requires the use of a new way to handle neighbor lists, the so-called "Verlet cutoff" scheme. Shift functions are replaced by exact cutoffs, changing the shape of the non-bonded potential. Reducing the cutoff to 1.1 nm instead of 1.2 nm seems to allow the use of this new algorithm without affecting the properties of common systems simulated with the Martini model. Overall, the change in the cutoff scheme results in a speedup by almost 100 %, but at this time the precise effects remain mostly untested.
2. Using the polarizable water model requires a few changes in the protocol described here. First of all, we need to replace the

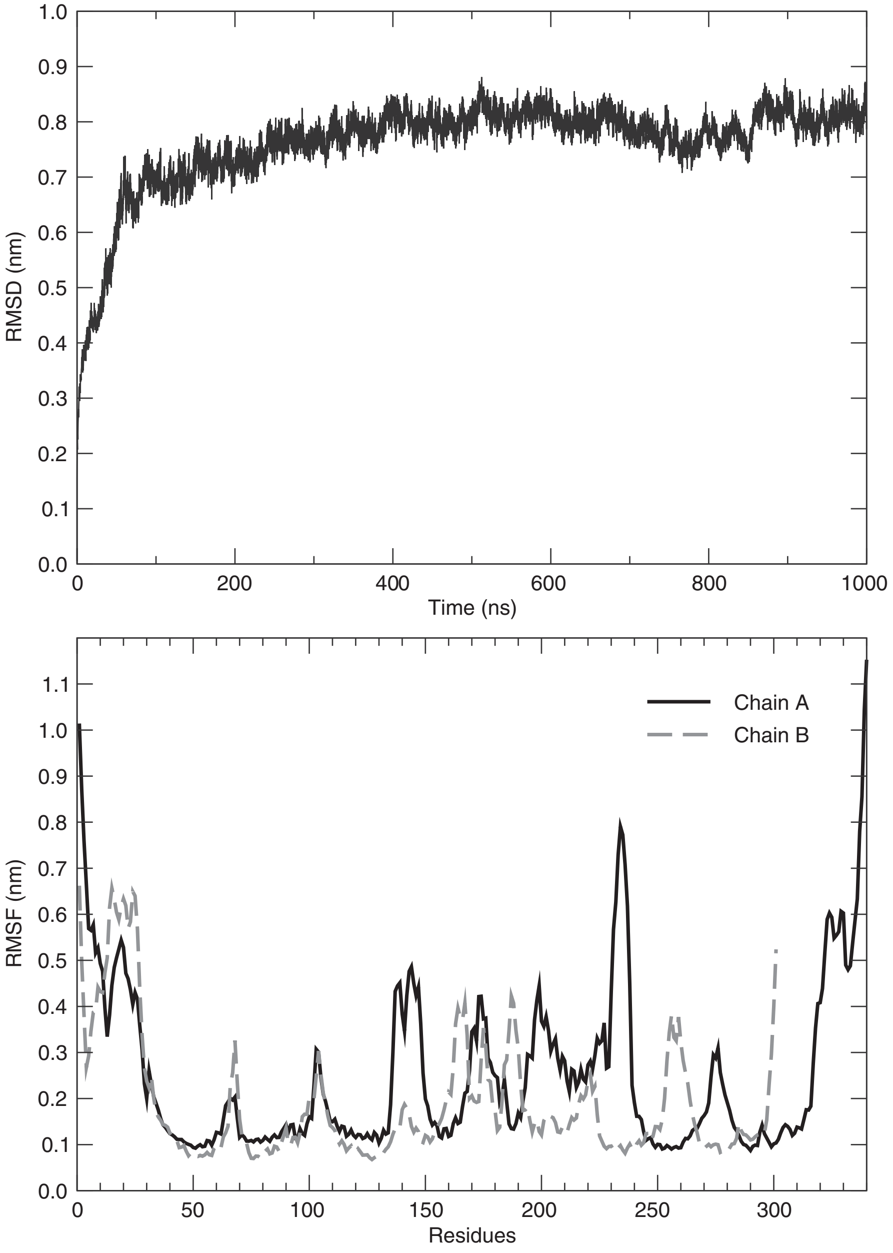


Fig. 3 Evolution of the root mean square deviation (RMSD) from the initial structure during the production run (*top panel*), and root mean square fluctuation (RMSF) of the backbone, averaged over the entire production run (*bottom panel*). The RMSD is calculated on the whole protein after a least mean square fit on the backbone

force field definition martini_v2.2.itp with martini_v2.2P.itp. For the hydration of the system, use the box of polarizable water (available on the Martini Web site). A script that converts normal Martini water into polarizable water is also available on the same Web site. In the MDP files, one should use a dielectric constant of 2.5 instead of 15 (epsilon_r=2.5). We can also use the particle mesh ewald (PME) algorithm for the calculation of electrostatic interactions (coulombtype=PME). In this case, we would use a real space cutoff of 1.2 nm (rcoulomb=1.2), a Fourier grid spacing of 0.12 nm (fourierspacing=0.12), and a fourth-order interpolation (pme-order=4).

3. GRO files have a very strict format. The first line is the title of the system, the second line is the number of atoms in the file, and the last line defines the periodic box. All other lines describe atoms. These atom lines are formatted by columns. The residue number, the residue name, the atom name, and the atom number are written, in this order, with five characters each. They are followed by the *X*, *Y*, and *Z* coordinates of the atom, written with eight characters each. Optionally are written the velocities for each dimension, also with eight characters each. There is no column separator. The periodic box is defined on the last line by at least three numbers representing the size of the box in the *X*, *Y*, and *Z* dimension, respectively. Optionally, six other values can follow. They are used to define non-rhombohedric boxes. The format of this last line is free and values are space separated.

4. *Inflategro2* works by successive energy minimizations. Since the systems contain large empty volumes, there may be issues with the assumptions of the algorithm used to parallelize the calculations. One may need to run the energy minimizations on a single core. This requires modifications to the *inflategro2* source code. In line 955 of *inflategro2*, replace:

   ```
   "mdrun -s %s -v -deffnm %s -c %s"
   ```

   by:

   ```
   "mdrun -s %s -v -deffnm %s -c %s -nt 1"
   ```

   Note that the option -nt 1 asks mdrun to use only one thread.

Acknowledgments

The authors thank Juliette Martin and Nicoletta Ceres for their useful comments on the manuscript.

References

1. Dror RO, Dirks RM, Grossman JP, Xu H, Shaw DE (2012) Biomolecular simulation: a computational microscope for molecular biology. Annu Rev Biophys 41:429–452
2. Shaw DE, Chao JC, Eastwood MP, Gagliardo J, Grossman JP, Ho CR et al (2008) Anton, a special-purpose machine for molecular dynamics simulation. Commun ACM 51:91
3. Lindorff-Larsen K, Piana S, Dror RO, Shaw DE (2011) How fast-folding proteins fold. Science 334:517–520
4. Bussi G, Laio A, Parrinello M (2006) Equilibrium free energies from non-equilibrium metadynamics. Phys Rev Lett 96:090601
5. Sugita Y, Okamoto Y (1999) Replica-exchange molecular dynamics method for protein folding. Chem Phys Lett 314:141–151
6. Torrie GM, Valleau JP (1977) Nonphysical sampling distributions in Monte Carlo free-energy estimation: umbrella sampling. J Comput Phys 23:187–199
7. Carbone P, Varzaneh HAK, Chen X, Müller-Plathe F (2008) Transferability of coarse-grained force fields: the polymer case. J Chem Phys 128:064904
8. Levitt M, Warshel A (1975) Computer simulation of protein folding. Nature 253:694–698
9. Liwo A, Oldziej S, Pincus MR, Wawak RJ, Rackovsky S, Scheraga HA (1997) A united-residue force field for off-lattice protein-structure simulations. I. Functional forms and parameters of long-range side-chain interaction potentials from protein crystal data. J Comput Chem 18:849–873
10. Maupetit J, Tuffery P, Derreumaux P (2007) A coarse-grained protein force field for folding and structure prediction. Proteins 69:394–408
11. Bereau T, Deserno M (2009) Generic coarse-grained model for protein folding and aggregation. J Chem Phys 130:235106
12. Pasi M, Lavery R, Ceres N (2013) PaLaCe: a coarse-grain protein model for studying mechanical properties. J Chem Theory Comput 9:785–793
13. Zacharias M (2003) Protein-protein docking with a reduced protein model accounting for side-chain flexibility. Protein Sci 12: 1271–1282
14. Setny P, Zacharias M (2011) A coarse-grained force field for Protein-RNA docking. Nucleic Acids Res 39:9118–9129
15. Den Otter WK, Renes MR, Briels WJ (2010) Asymmetry as the key to clathrin cage assembly. Biophys J 99:1231–1238
16. Matthews R, Likos CN (2013) Structures and pathways for clathrin self-assembly in the bulk and on membranes. Soft Matter 9:5794–5806. doi:10.1039/c3sm50737h
17. Shi Q, Izvekov S, Voth GA (2006) Mixed atomistic and coarse-grained molecular dynamics: simulation of a membrane-bound ion channel. J Phys Chem B 110:15045–15048
18. Rzepiela AJ, Louhivuori M, Peter C, Marrink SJ (2011) Hybrid simulations: combining atomistic and coarse-grained force fields using virtual sites. Phys Chem Chem Phys 13: 10437–10448
19. Zacharias M (2013) Combining coarse-grained nonbonded and atomistic bonded interactions for protein modeling. Proteins 81:81–92
20. Taylor WR, Katsimitsoulia Z (2010) A coarse-grained molecular model for actin-myosin simulation. J Mol Graph Model 29:266–279
21. Praprotnik M, Delle Site L (2013) Multiscale molecular modeling. Methods Mol Biol 924:567–583
22. Marrink SJ, de Vries AH, Mark AE (2004) Coarse grained model for semiquantitative lipid simulations. J Phys Chem B 108:750–760
23. Marrink SJ, Risselada HJ, Yefimov S, Tieleman DP, De Vries AH (2007) The MARTINI force field: coarse grained model for biomolecular simulations. J Phys Chem B 111:7812–7824
24. Monticelli L, Kandasamy SK, Periole X, Larson RG, Tieleman DP, Marrink S-J (2008) The MARTINI coarse-grained force field: extension to proteins. J Chem Theory Comput 4:819–834
25. De Jong DH, Singh G, Bennett WFD, Arnarez C, Wassenaar TA, Schäfer LV et al (2013) Improved parameters for the martini coarse-grained protein force field. J Chem Theory Comput 9:687–697
26. Marrink SJ, Tieleman DP (2013) Perspective on the Martini model. Chem Soc Rev 42:6801–6822. doi:10.1039/c3cs60093a
27. López CA, Rzepiela AJ, de Vries AH, Dijkhuizen L, Hünenberger PH, Marrink SJ (2009) Martini coarse-grained force field: extension to carbohydrates. J Chem Theory Comput 5:3195–3210
28. Rossi G, Fuchs PFJ, Barnoud J, Monticelli L (2012) A coarse-grained MARTINI model of polyethylene glycol and of polyoxyethylene alkyl ether surfactants. J Phys Chem B 116: 14353–14362
29. Rossi G, Monticelli L, Puisto SR, Vattulainen I, Ala-Nissila T (2011) Coarse-graining polymers

with the MARTINI force-field: polystyrene as a benchmark case. Soft Matter 7:698

30. Wong-Ekkabut J, Baoukina S, Triampo W, Tang I-M, Tieleman DP, Monticelli L (2008) Computer simulation study of fullerene translocation through lipid membranes. Nat Nanotechnol 3:363–368
31. Monticelli L (2012) On atomistic and coarse-grained models for C60 fullerene. J Chem Theory Comput 8:1370–1378
32. Marrink SJ, Mark AE (2003) The mechanism of vesicle fusion as revealed by molecular dynamics simulations. J Am Chem Soc 125: 11144–11145
33. Risselada HJ, Marrink SJ (2008) The molecular face of lipid rafts in model membranes. Proc Natl Acad Sci U S A 105:17367–17372
34. Baoukina S, Marrink SJ, Tieleman DP (2012) Molecular structure of membrane tethers. Biophys J 102:1866–1871
35. Baron R, Trzesniak D, de Vries AH, Elsener A, Marrink SJ, van Gunsteren WF (2007) Comparison of thermodynamic properties of coarse-grained and atomic-level simulation models. ChemPhysChem 8:452–461
36. Shih AY, Arkhipov A, Freddolino PL, Schulten K (2006) Coarse grained protein-lipid model with application to lipoprotein particles. J Phys Chem B 110:3674–3684
37. Shinoda W, DeVane R, Klein ML (2010) Zwitterionic lipid assemblies: molecular dynamics studies of monolayers, bilayers, and vesicles using a new coarse grain force field. J Phys Chem B 114:6836–6849
38. Yesylevskyy SO, Schäfer LV, Sengupta D, Marrink SJ (2010) Polarizable water model for the coarse-grained MARTINI force field. PLoS Comput Biol 6:e1000810
39. Monticelli L, Tieleman DP, Fuchs PFJ (2010) Interpretation of 2H-NMR experiments on the orientation of the transmembrane helix WALP23 by computer simulations. Biophys J 99:1455–1464
40. Castillo N, Monticelli L, Barnoud J, Tieleman DP (2013) Free energy of WALP23 dimer association in DMPC, DPPC, and DOPC bilayers. Chem Phys Lipids 169:95–105
41. Deplazes E, Louhivuori M, Jayatilaka D, Marrink SJ, Corry B (2012) Structural Investigation of MscL gating using experimental data and coarse grained MD simulations. PLoS Comput Biol 8:e1002683
42. Periole X, Cavalli M, Marrink S-J, Ceruso MA (2009) Combining an elastic network with a coarse-grained molecular force field: structure, dynamics, and intermolecular recognition. J Chem Theory Comput 5:2531–2543
43. Dony N, Crowet JM, Joris B, Brasseur R, Lins L (2013) SAHBNET, an accessible surface-based elastic network: an application to membrane protein. Int J Mol Sci 14: 11510–11526
44. Pronk S, Páll S, Schulz R, Larsson P, Bjelkmar P, Apostolov R et al (2013) GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. Bioinformatics 29:845–854
45. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22:2577–2637
46. Okada T, Fujiyoshi Y, Silow M, Navarro J, Landau EM, Shichida Y (2002) Functional role of internal water molecules in rhodopsin revealed by X-ray crystallography. Proc Natl Acad Sci U S A 99:5982–5987
47. Humphrey W, Dalke A, Schulten K (1996) VMD – visual molecular dynamics. J Mol Graph 14:33–38
48. Jo S, Lim JB, Klauda JB, Im W (2009) CHARMM-GUI Membrane Builder for mixed bilayers and its application to yeast membranes. Biophys J 97:50–58
49. Lomize MA, Lomize AL, Pogozheva ID, Mosberg HI (2006) OPM: orientations of proteins in membranes database. Bioinformatics 22:623–625
50. Lomize MA, Pogozheva ID, Joo H, Mosberg HI, Lomize AL (2012) OPM database and PPM web server: resources for positioning of proteins in membranes. Nucleic Acids Res 40:D370–D376
51. Tusnády GE, Dosztányi Z, Simon I (2004) Transmembrane proteins in the Protein Data Bank: identification and classification. Bioinformatics 20:2964–2972
52. Tusnády GE, Dosztányi Z, Simon I (2005) PDB_TM: selection and membrane localization of transmembrane proteins in the protein data bank. Nucleic Acids Res 33: D275–D278
53. Kozma D, Simon I, Tusnády GE (2013) PDBTM: protein data bank of transmembrane proteins after 8 years. Nucleic Acids Res 41:D524–D529
54. Tusnády GE, Dosztányi Z, Simon I (2005) TMDET: web server for detecting transmembrane regions of proteins by using their 3D coordinates. Bioinformatics 21:1276–1277
55. Schmidt TH, Kandt C (2012) LAMBADA and InflateGRO2: efficient membrane alignment and insertion of membrane proteins for molecular dynamics simulations. J Chem Inf Model 52:2657–2669

56. Wolf MG, Hoefling M, Aponte-Santamaría C, Grubmüller H, Groenhof G (2010) g_membed: efficient insertion of a membrane protein into an equilibrated lipid bilayer with minimal perturbation. J Comput Chem 31: 2169–2174

57. Kandt C, Ash WL, Tieleman DP (2007) Setting up and running molecular dynamics simulations of membrane proteins. Methods 41:475–488

58. Berendsen HJC, Postma JPM, van Gunsteren WF, DiNola A, Haak JR (1984) Molecular dynamics with coupling to an external bath. J Chem Phys 81:3684

59. Bussi G, Donadio D, Parrinello M (2007) Canonical sampling through velocity rescaling. J Chem Phys 126:014101

60. Parrinello M, Rahman A (1981) Polymorphic transitions in single crystals: a new molecular dynamics method. J Appl Phys 52:7182

Chapter 8

Tackling Sampling Challenges in Biomolecular Simulations

Alessandro Barducci, Jim Pfaendtner, and Massimiliano Bonomi

Abstract

Molecular dynamics (MD) simulations are a powerful tool to give an atomistic insight into the structure and dynamics of proteins. However, the time scales accessible in standard simulations, which often do not match those in which interesting biological processes occur, limit their predictive capabilities. Many advanced sampling techniques have been proposed over the years to overcome this limitation. This chapter focuses on metadynamics, a method based on the introduction of a time-dependent bias potential to accelerate sampling and recover equilibrium properties of a few descriptors that are able to capture the complexity of a process at a coarse-grained level. The theory of metadynamics and its combination with other popular sampling techniques such as the replica exchange method is briefly presented. Practical applications of these techniques to the study of the Trp-Cage miniprotein folding are also illustrated. The examples contain a guide for performing these calculations with PLUMED, a plugin to perform enhanced sampling simulations in combination with many popular MD codes.

Key words Enhanced sampling, Metadynamics, PLUMED, Replica exchange methods, Molecular dynamics, Collective variables, Free energy

1 Introduction

Uniquely providing insights into the structure and dynamics of complex biomolecular systems at the atomistic level, MD simulations can play a fundamental role in molecular biology. Unfortunately, the large numbers of particles that are needed for an accurate model of biomolecules and the complexity of their free-energy landscape make simulations computationally expensive and prevent exhaustive sampling by standard MD in all but the simplest cases. Recently, the development of dedicated hardware [1] and distributed computing protocols [2] has in part alleviated these issues. Nevertheless, the time scales accessible to MD are still significantly shorter than those typical of several interesting biomolecular processes as well as of many experimental techniques.

To extend the time scales of MD simulations, several advanced sampling methods have been proposed over the years [3, 4]. A comprehensive review of such methods is beyond the scope of this chapter.

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_8, © Springer Science+Business Media New York 2015

Here we focus on metadynamics [5] (MetaD), which is able at the same time to enhance sampling and reconstruct the free-energy surface (FES) as a function of a few selected degrees of freedom. Since its introduction in 2002, MetaD has been successfully applied to a variety of different problems in chemistry, biology, and solid-state physics [6]. Furthermore, MetaD can be seamlessly integrated with other advanced sampling algorithms, such as replica exchange method [7, 8] (REM), with immense synergistic benefits when studying complex biomolecular processes in large systems.

In this chapter, we present a concise introduction to the theory of MetaD and combined methods and a few practical applications to the study of Trp-Cage miniprotein [9] folding. We also provide all the scripts needed to run these examples with PLUMED [10], a plugin that enables enhanced sampling calculations with several popular MD codes. The chapter is organized as follows. In Subheading 2, we illustrate the basic theory of: (1) MetaD in its well-tempered variant [11] (Subheading 2.1), (2) REM, with a particular focus on parallel tempering (PT) (Subheading 2.2), (3) possible combinations of these methods (Subheading 2.3). In Subheading 3, we list the software used to carry out the simulations described in this chapter. In Subheading 4, we provide a step-by-step guide to the following simulations of the Trp-Cage miniprotein: MetaD (Subheading 4.2), PT-MetaD (Subheading 4.3), and PT-MetaD in the Well-Tempered Ensemble [12] (WTE) (Subheading 4.4). Finally, Subheading 5 contains a series of notes on detecting and solving practical issues that can rise when using the sampling techniques presented in this chapter.

2 Theory

2.1 Metadynamics

In MetaD, an external history-dependent bias potential is constructed in the space of a few selected degrees of freedom, generally called collective variables (CVs). CVs are functions S of the microscopic coordinates R of the system:

$$S(\boldsymbol{R}) = (S_1(\boldsymbol{R}), \ldots, S_d(\boldsymbol{R})), \tag{1}$$

which are able to provide a coarse-grained description of the process under study. In particular, CVs must distinguish the relevant states of the system and include all the kinetically relevant degrees of freedom. The MetaD bias potential (V_G) can be written as a sum of Gaussians deposited along the system trajectory in the CVs space. In the well-tempered approach [11], V_G has the following functional form at time t:

$$V_\mathrm{G}(S,t) = \int_0^t \mathrm{d}t' \ \omega(t') \cdot \exp\left(-\sum_{i=1}^{d} \frac{\left(S_i(\boldsymbol{R}) - S_i(\boldsymbol{R}(t'))\right)^2}{2\sigma_i^2} \right), \tag{2}$$

where σ_i is the width of the Gaussian for the *i*th CV. The time-dependent energy rate $\omega(t)$ is defined as:

$$\omega(t) = \omega_0 \cdot \exp\left(-\frac{V_{\mathrm{G}}(S,t)}{k_{\mathrm{B}}\Delta T}\right), \tag{3}$$

where ω_0 is an initial deposition rate, k_{B} is the Boltzmann constant, and ΔT is an input parameter with the dimension of a temperature.

It has been proven [11] that, in the long time limit, V_{G} converges to:

$$V_{\mathrm{G}}(S,t \to \infty) = -\frac{\Delta T}{T + \Delta T} \cdot F(S) + C, \tag{4}$$

where T is the temperature of the system and C is an irrelevant additive constant. $F(S)$ is the free energy as a function of the CVs:

$$F(S) = -k_{\mathrm{B}}T\log\left[\int \mathrm{d}\boldsymbol{R}\,\delta(S - S(\boldsymbol{R})) \cdot \exp\left(-\frac{U(\boldsymbol{R})}{k_{\mathrm{B}}T}\right)\right], \tag{5}$$

where $U(\mathrm{R})$ is the potential energy function. Equation 4 is often expressed in terms of the so-called *bias-factor* $\gamma = (T + \Delta T)/T$ as:

$$V_{\mathrm{G}}(S,t \to \infty) = \propto -\left(1 - \gamma^{-1}\right) \cdot F(S) + C. \tag{6}$$

In the long-time limit, the CVs probability density $P(S, t)$ can be written as:

$$P(S,t \to \infty) \propto \exp\left(-\frac{F(S)}{k_{\mathrm{B}}(T + \Delta T)}\right), \tag{7}$$

which corresponds to sampling the free-energy surface (FES) at the fictitious temperature of $T + \Delta T$. The extent of FES exploration can thus be regulated by tuning ΔT.

To understand the effect of V_{G}, let us consider a system whose dynamics can be captured by a one-dimensional free energy, with local minima separated by barriers much higher than thermal fluctuations (Fig. 1). In a standard MD simulation, the system would remain trapped in one of the local minima and it would not be possible to sample all the relevant regions in a reasonable simulation time. In a well-tempered MetaD simulation, Gaussians are progressively deposited along the trajectory resulting in the growth of the bias potential, which ultimately facilitates barrier crossing. If the parameters are properly chosen, the system will eventually sample all the relevant free-energy minima and V_{G} will converge. At that point, the free-energy profile can be estimated using Eq. 4.

One notable case of well-tempered MetaD corresponds to the choice of the potential energy of the system as CV [12]. In this situation, the MetaD bias leads to the sampling of a well-defined

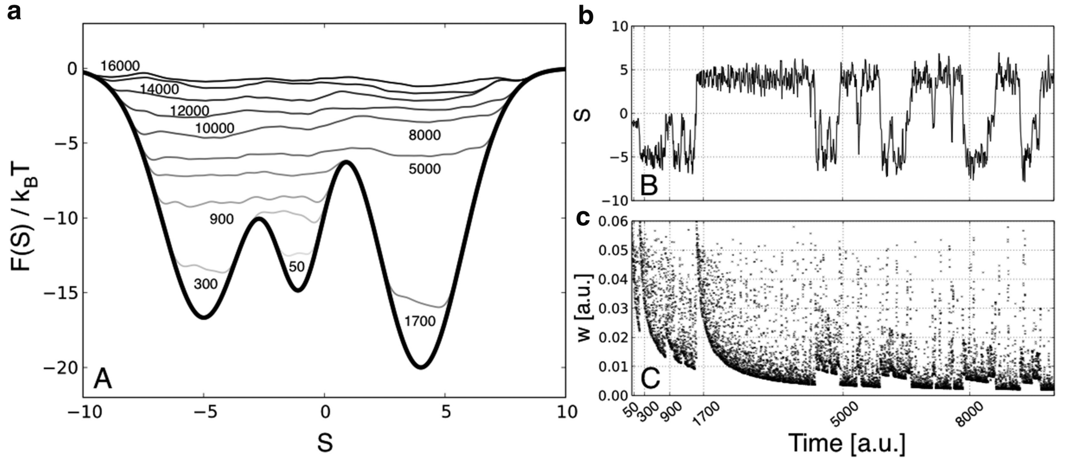


Fig. 1 Well-tempered MetaD simulation in a one-dimensional model system. (**a**) The free-energy profile (*thick black line*) is characterized by three local minima separated by energy barriers higher than $k_B T$. The sum of the underlying free energy and the bias potential is shown at different times of the simulation (*grey lines*). The bias potential is rescaled by the bias factor, following Eq. 6. (**b**, **c**) Time series of the CV S (**b**) and of the Gaussian height w (**c**) in the first 10,000 steps of simulation. The system is prepared in $S = -1$. As the bias potential grows, the Gaussian height w decreases, following the well-tempered recipe of Eq. 3. Around $t = 200$, the system escapes from the initial basin into the first minimum on the right. At this point the deposition rate suddenly increases (**c**), as expected in well-tempered MetaD when visiting a previously unexplored region of the CV space. After this basin is completely filled, the system starts diffusing between the first and second minima ($800 < t < 1{,}500$). When a sufficient amount of bias is accumulated, the system is pushed to visit the third basin on the right and the deposition rate increases again. Finally, around $t = 4{,}500$ the underlying free energy is almost completely compensated by the bias potential. At this point, the system starts diffusing smoothly in the CV space, while the Gaussian height is progressively decaying to zero

distribution called Well-Tempered Ensemble (WTE). In this ensemble, the average energy remains close to the canonical value but its fluctuations are enhanced in a tunable way, thus improving sampling.

In well-tempered MetaD, the bias deposition rate decreases with time as $1/t$. The dynamics of all the microscopic variables thus becomes progressively closer to thermodynamic equilibrium as the simulation proceeds. This feature allows to easily recover the equilibrium Boltzmann distribution of degrees of freedom other than the CVs, which typically is altered by the introduction of V_G. Indeed, a simple reweighting scheme has been proposed [13] in order to obtain "on-the-fly" estimates of expectation values of any variable during a well-tempered MetaD simulation. This algorithm significantly extends the capabilities of MetaD by allowing a quantitative contact between biased simulations and experiments [14].

2.2 Replica Exchange Method

In REM [7, 8], sampling is accelerated by properly modifying the original Hamiltonian of the system. This goal is achieved by simulating in parallel N non-interacting replicas of the system, each

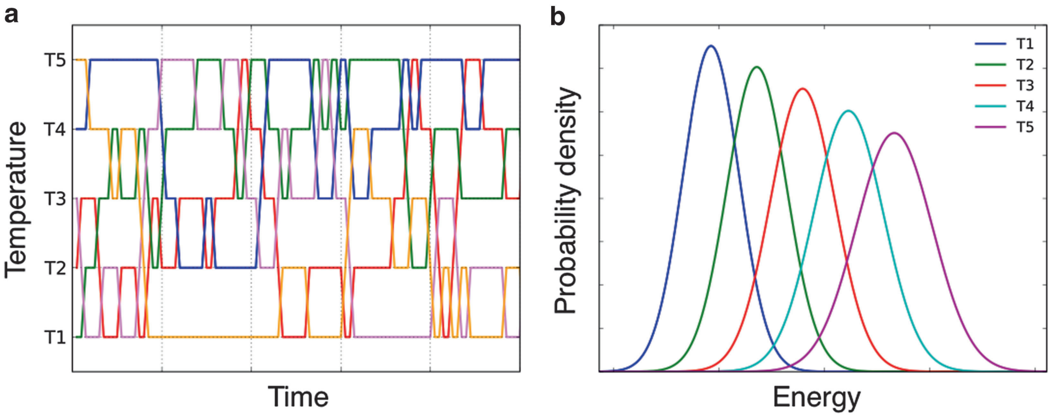


Fig. 2 Schematic representation of the PT scheme. (**a**) $N = 5$ independent copies of the system are simulated at different temperatures. Periodically, an exchange between replicas at different temperatures (typically neighbors) is proposed and accepted based on the Metropolis criteria defined in Eqs. 8 and 9. The time needed to span the entire temperature range, i.e., the round-trip time, is often used as measure of PT efficiency. (**b**) Quasi-Gaussian potential energy distributions at different temperatures, as typically observed in simulations of proteins in explicit solvent. The PT acceptance probability ultimately depends on the overlap of the potential energy distributions at two temperatures. The number of replicas needed to guarantee a similar overlap at a fixed temperature range scales as the square root of the number of degrees of freedom, thus making PT computationally prohibitive in large systems

evolving according to a different Hamiltonian. At fixed intervals, an exchange of configurations between two replicas is attempted while respecting detailed balance (Fig. 2a). One popular case of REM is parallel tempering (PT), in which replicas are simulated using the same potential energy function, but different temperatures. By accessing high temperatures, replicas are prevented from being trapped in local minima. In PT, exchanges are usually attempted between adjacent temperatures with the following acceptance probability:

$$p(j \to k) = \min\left\{1, \exp\left(\Delta_{j,k}^{\mathrm{PT}}\right)\right\}, \qquad (8)$$

with

$$\Delta_{j,k}^{\mathrm{PT}} = \left(\frac{1}{k_{\mathrm{B}}T_j} - \frac{1}{k_{\mathrm{B}}T_k}\right) \cdot \left(U\left(\boldsymbol{R}_j\right) - U\left(\boldsymbol{R}_k\right)\right), \qquad (9)$$

where R_j and R_k are the configurations at temperature T_j and T_k, respectively. Equation 9 indicates that the acceptance probability is ultimately determined by the overlap between the energy distributions of two replicas (Fig. 2b).

One advantage of PT is that there is no need to select a priori an arbitrary set of CVs, once the temperatures are chosen. However, the efficiency of the algorithm depends on the benefits provided by sampling at high-temperature. Therefore, an efficient diffusion in temperature space is required and configurational sampling is still limited by entropic barriers. Finally, PT scales poorly with system size. In fact, a sufficient overlap between the potential energy distributions of neighboring temperatures is required in order to obtain a significant diffusion. Therefore, the number of temperatures needed to cover a given temperature range scales as the square root of the number of degrees of freedom, making this approach prohibitively expensive for large systems.

2.3 Combined Approaches

MetaD and REM have been successfully applied to characterize a variety of systems [6, 15]. However, two issues have limited the efficiency of these two sampling algorithms in simulating large biomolecular systems. First, finding a small set of CVs that captures the slow dynamics of complex conformational changes might be a daunting task in MetaD. Second, the size of these systems makes multi-replica approaches computationally demanding, especially when using an explicit solvent model. These limitations can be alleviated by properly combining different sampling approaches.

In this spirit, one can apply the MetaD bias on some selected CVs within a multi-replica PT scheme. In the resulting PT-MetaD algorithm [16], *N* replicas performed in parallel a MetaD simulation at different temperatures, using the same set of configurational CVs. The REM acceptance probability is modified in order to account for the presence of the MetaD bias potential:

$$\begin{aligned}\Delta_{j,k}^{\mathrm{PT-MetaD}} &= \Delta_{j,k}^{\mathrm{PT}} + \frac{1}{k_{\mathrm{B}}T_j}\cdot\left[V_{\mathrm{G}}^{(j)}\left(S\left(\boldsymbol{R}_j\right),t\right) - V_{\mathrm{G}}^{(j)}\left(S\left(\boldsymbol{R}_k\right),t\right)\right] \\ &\quad + \frac{1}{k_{\mathrm{B}}T_k}\cdot\left[V_{\mathrm{G}}^{(k)}\left(S\left(\boldsymbol{R}_k\right),t\right) - V_{\mathrm{G}}^{(k)}\left(S\left(\boldsymbol{R}_j\right),t\right)\right], \qquad (10)\end{aligned}$$

where $V_{\mathrm{G}}^{(j)}$ and $V_{\mathrm{G}}^{(k)}$ are the bias potentials acting on the *j*-th and *k*-th replicas, respectively.

PT-MetaD is particularly effective because it compensates for some of the weaknesses of each method individually taken. The negative effect of neglecting a slow degree of freedom in the choice of the MetaD CVs is alleviated by PT, which allows the system to cross moderately high free-energy barriers on all degrees of freedom. On the other hand, the MetaD bias potential allows crossing higher barriers on a few selected CVs, in such a way that the sampling efficiency of PT-MetaD is greater than that of PT alone.

Nevertheless, PT-MetaD still suffers from the poor scaling of computational resources with system size. This issue may be circumvented by including the potential energy of the system among the set of MetaD CVs, as in the WTE approach [12]. This leads to the so-called PT-MetaD-WTE scheme [17], in which replica diffusion in temperature space is enhanced by the increased energy fluctuations at all temperatures.

3 Materials

Simulations of the Trp-Cage miniprotein have been carried out [17] using GROMACS version 4.5.3 [18] and the PLUMED plugin version 1.2.2 [10]. However, for didactical purposes the scripts reported here have been updated to PLUMED version 2 [19]. Figures have been prepared with UCSF Chimera [20] and Matplotlib [21]. All simulations should be run in parallel on a cluster machine. The reader should refer to GROMACS and PLUMED user manuals for detailed instructions about how to compile and execute the codes.

4 Methods

In this section we show a few applications of the enhanced sampling techniques introduced above to study the Trp-Cage miniprotein folding, using an atomistic description of both solute and solvent degrees of freedom. Trp-cage is a 20-residue protein whose structure has been determined by NMR [9] (PDB code 1L2Y), and its folding process has been extensively studied by several experimental [9, 22–25] and computational [26–31] techniques.

This section is organized as follows. In Subheading 4.1, we describe the steps needed to prepare and equilibrate the system. In Subheading 4.2, we illustrate a simple MetaD simulation that uses 2 CVs. In Subheadings 4.3 and 4.4, we combine MetaD with a multi-replica approach (PT) and with WTE, respectively. In Subheading 4.5, we present a quantitative analysis of the convergence of the simulations along with an estimate of the error in the reconstructed FES.

4.1 System Preparation

The initial structure of Trp-Cage is taken from Protein Data Bank (PDB code 1L2Y). The GROMACS tools are used to create the topology using the AMBER99-SB [32] force field and solvate the protein in a truncated octahedral box containing 3,717 TIP3P [33] water molecules. The initial conformation is first energy-minimized by 5,000 steps of steepest descent and then equilibrated at 300 K and 1 atm by a 1 ns simulation in the isothermal-isobaric ensemble (NPT). The final volume of the cell is 4.9 nm^3. To generate an initial unfolded structure, we heat the system at 600 K for 1 ns in the isothermal ensemble (NVT).

4.2 MetaD

Our goal is to study the folding landscape of Trp-Cage. Although this is a small protein, its folding is a complex molecular process which involves many degrees of freedom and a wide range of spatial and time scales. However, in a MetaD simulation it is very inefficient to simultaneously bias a large number of degrees of freedom due to the

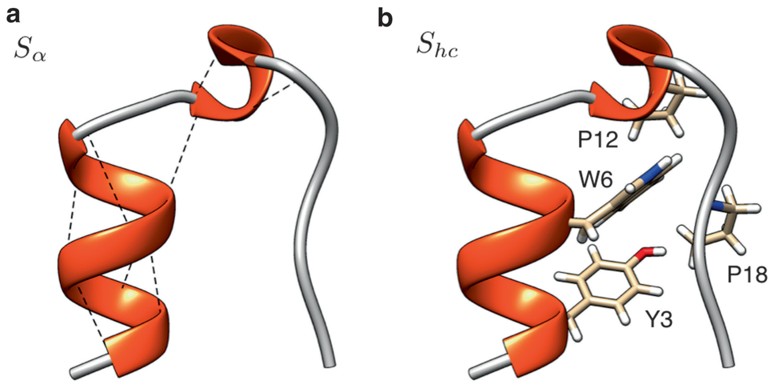


Fig. 3 Definition of the MetaD CVs used to study the folding of the Trp-Cage miniprotein. (**a**) $S\alpha$ counts the number of hydrogen bonds (*black dotted lines*) formed in and between the α-helical regions (*orange cartoon*) of the NMR native structure (**b**). S_{hc} counts the number of contacts in the hydrophobic core, defined here by residues Y3, W6, P12, and P18 (*ball and stick*)

exponential increase of the CVs space. Therefore, most applications of MetaD have been carried out using typically 2 or 3 CVs that were devised to capture the relevant modes of the process under study.

Here, we use 2 CVs that coarse-grain the most important driving forces in protein folding: the formation of the secondary structure and of the hydrophobic core (Fig. 3). In order to define the CVs, we take advantage of the following switching function, provided by the PLUMED plugin, that can be parameterized to quantify the formation of a “contact” between the two atoms *i* and *j*:

$$s_{ij} = \frac{1-\left(\frac{r_{ij}}{r_0}\right)^n}{1-\left(\frac{r_{ij}}{r_0}\right)^m}, \tag{11}$$

where r_{ij} is the distance between the two atoms, r_0 is a characteristic contact distance, and the pair n, m defines the steepness of the CV (*see* **Note 1**). The first CV ($S\alpha$) describes the number of backbone-backbone α-helical hydrogen bonds formed (Fig. 3a):

$$S_\alpha = \sum_{i=1}^{N_\text{H}}\sum_{j=1}^{N_\text{O}} s_{ij}, \tag{12}$$

where $r_0 = 0.25$ nm, $n = 8$, $m = 12$, and the sums are over the hydrogen and oxygen atoms that form an α-helical hydrogen bond in the native state. The second CV (S_hc) describes the number of contacts in the hydrophobic core (Fig. 3b):

$$S_\text{hc} = \sum_{\substack{i>j,\\ i,j\in\text{core}}} s_{ij}, \tag{13}$$

where $r_0 = 0.50$ nm, $n = 8$, and $m = 12$, and the sum is over representative side-chain Carbon atoms of the residues belonging to the hydrophobic core (Y3, W6, P12, and P18).

The MetaD bias potential is constructed using an initial deposition rate equal to 2.5 kJ/mol every 0.5 ps. Each Gaussian has a width equal to 0.4 for both CVs. The bias factor γ is set to 8 (*see* **Note 2**). The following input file can be used to run the MetaD simulation with PLUMED 2:

```
# Groups definition
h:    GROUP ATOMS=85,107,125,145,166,181,215,238
o:    GROUP ATOMS=4,20,39,60,79,96,173,180
core: GROUP ATOMS=46,106,189,270

# CVs definition
s_a:  COORDINATION GROUPA=h GROUPB=o PAIR  NN=8 MM=12 R_0=0.25
s_hc: COORDINATION GROUPA=core GROUPB=core NN=8 MM=12 R_0=0.50

# MetaD parameters
METAD…
  ARG=s_a,s_hc SIGMA=0.4,0.4 HEIGHT=2.5 PACE=250
  TEMP=300 BIASFACTOR=8 FILE=HILLS
…METAD
```

The groups are defined in terms of the atom indices in 1L2Y.pdb. The units of measurement are: kJ/mol for energy, nm for distances, K for temperature, and number of MD steps for time (here the time step is set to 2 fs). The MetaD simulation should be run following the standard GROMACS instructions, by adding the flag that activates PLUMED to the command line:

```
mdrun –plumed plumed.dat
```

where plumed.dat is the name of the PLUMED input file defined above.

We analyze the results of the MetaD simulation by examining the evolution of the system in CV space along with the deposition rate of the bias potential (Fig. 4). The system starts from a configuration with an unstructured helix ($S_\alpha \sim 2.6$) and a collapsed hydrophobic core ($S_{hc} \sim 8.0$). In a few nanoseconds, Trp-cage is pushed to break all the hydrogen bonds and to explore a region of configurational space with a completely broken helix ($S_\alpha \sim 0$). Afterwards, the systems seems to be stuck in this region for ~100 ns although it samples a broad range in the S_{hc} space. A significant amount of bias is accumulated in this part of the simulation as confirmed by its deposition rate that rapidly decreases following Eq. 3. After 100 ns, the system suddenly forms hydrogen bonds and escapes this region towards more native-like conformation with

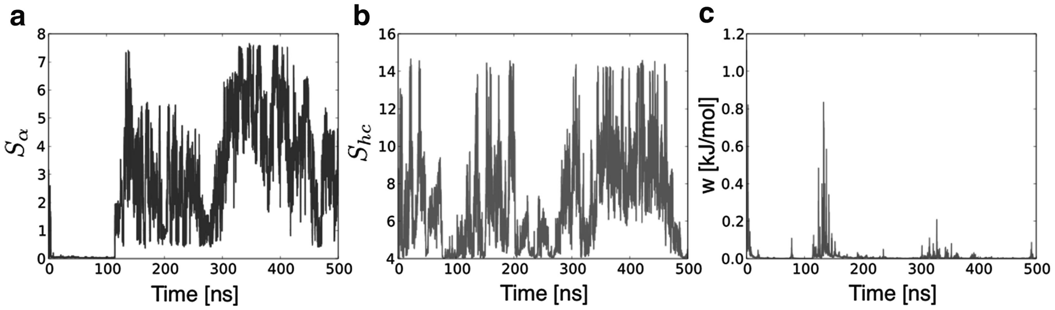


Fig. 4 MetaD simulations of the Trp-Cage miniprotein. Time series of the CVs (**a**) S_α and (**b**) S_{hc}, along with the Gaussian height (**c**). The 2 CVs seems to fail in describing all the relevant slow modes of the system, since we do not observe a smooth exploration of the CVs space in the time scale of this simulation

high number of α-helical hydrogen bonds and hydrophobic contacts. At this stage, the deposition of the bias potential suddenly increases again. During the rest of the simulation, the trajectory visits conformations with variable number of α-helical hydrogen bonds and hydrophobic contacts, but never returns to the region of completely unstructured configurations that has been explored earlier ($S_\alpha < 0.5$).

The difficulty of the bias potential in pushing Trp-Cage back and forth from one conformational region to the other is a symptom that our 2 CVs do not capture all the relevant modes of the system. Differently, we would observe a smoother exploration of the CV space and a quasi-diffusive dynamics upon convergence of the bias potential. This outcome is not totally unexpected, since the configurational ensemble of Trp-Cage is extremely wide and characterized by several metastable states that these CVs seem not to properly describe. At this point one can adopt several different strategies:

1. Complementing the existing set of CVs with additional CVs;
2. Devising more effective CVs. A possible choice include CVs that are able to describe the collective behavior of the folding process, such as the Path Collective Variables [34] or CVs based on dimensionality reduction [35, 36];
3. Biasing separately different CVs in a bias-exchange MetaD approach [37];
4. Combining MetaD with other sampling algorithms.

In the following subsection, we show how to combine MetaD in S_α and S_{hc} with PT and the resulting benefit to sampling efficiency.

4.3 PT-MetaD

To setup a PT-MetaD simulation, we need to choose the replicas temperature distribution (*see* **Note 3**) and equilibrate each replica prior to perform the MetaD simulation (*see* **Notes 4** and **5**).

We thus perform short 500 ps simulations in the NVT ensemble, so that at the end of equilibration the Cα root-mean-square deviation (RMSD) from the lowest-energy NMR structure ranges from 4 to 9 Å across the replicas.

We use the same set of CVs ($S\alpha$-S_{hc}) introduced in Subheading 4.2. One PLUMED input file per replica as the one described above should be prepared (plumed.dat.0, plumed.dat.1,…), in which the correct temperature is specified in the line of the METAD keyword. We have to follow the GROMACS instructions to activate PT, in this case using 100 replicas and exchanging every 100 steps [38]:

```
        mdrun -multi 100 -replex 100 -plumed plumed.dat
```

The first thing to check in a PT-MetaD simulation is the correctness of the REM setup, in particular the temperature distribution. This can be done by analyzing both the average acceptance rate between neighboring temperatures (reported in the GROMACS log file) and the overall trajectory of each replica in temperature space (Fig. 5a, *see* **Notes 6** and 7).

As for the MetaD simulation, we examine the evolution of the system in the CV space along with the deposition rate of the bias potential at 300 K (Fig. 6). At variance with single replica MetaD, this is not a continuous trajectory, due to the exchanges with other temperatures (*see* **Note 6**). It is clear from this analysis that the statistics accumulated at 300 K spans the entire range of CVs space throughout the simulation (Fig. 6a,b). This is confirmed by the smooth average decrease of the bias deposition rate over the simulation time (Fig. 6c).

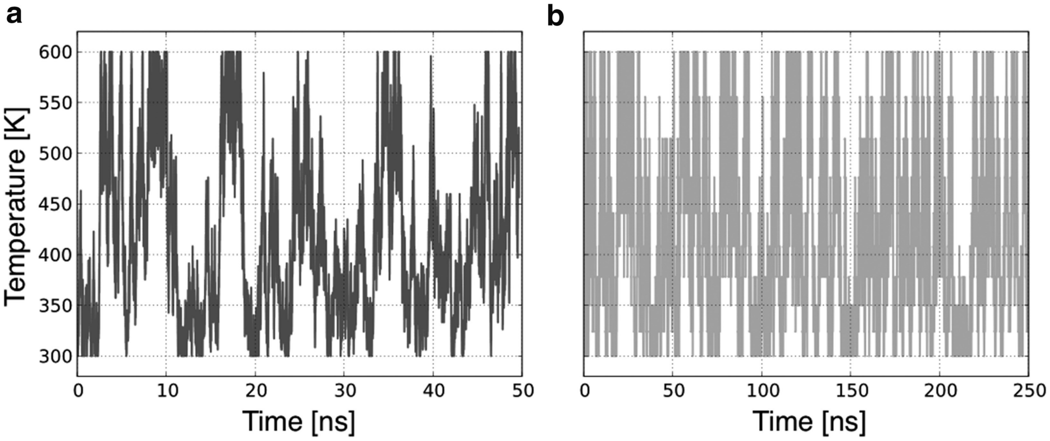


Fig. 5 Temperature diffusion of a representative replica in PT-Metad (**a**) and PT-MetaD-WTE (**b**). Time is measured per replica. Despite having only 10 intermediate temperatures instead of 100 to cover the same temperature range (300–600 K), the diffusion of PT-MetaD-WTE appears to be smooth thanks to the static bias on energy. The average round-trip times of PT-Metad and PT-MetaD-WTE are 6.3 ns and 4.0 ns, respectively

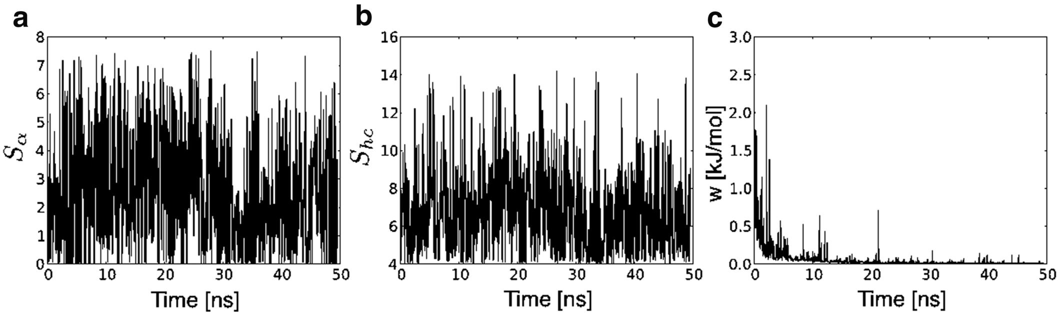


Fig. 6 PT-MetaD simulation of the Trp-Cage miniprotein. (Discontinuous) time series of (**a**) $S\alpha$ and (**b**) S_{hc} at 300 K, along with the Gaussian height (**c**). Time is measured per replica. Thanks to the exchange with other temperatures, an exhaustive sampling of the CVs space is achieved

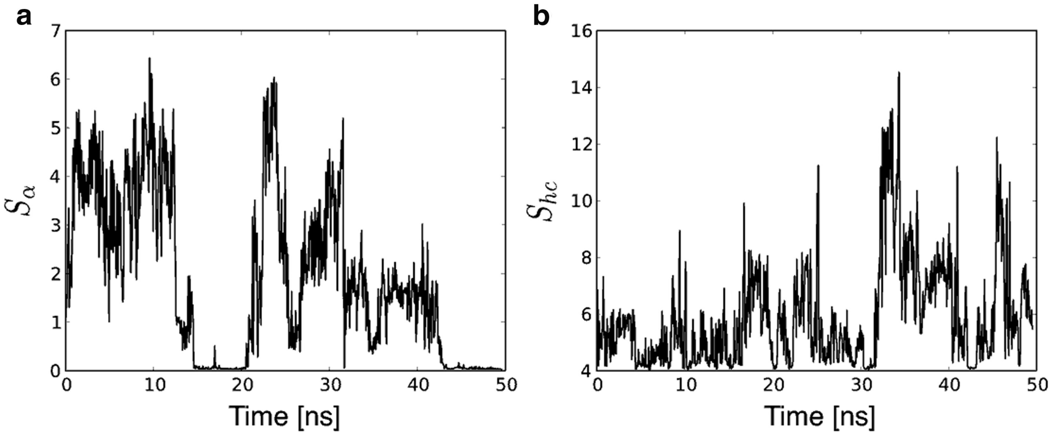


Fig. 7 PT-MetaD simulation of the Trp-Cage miniprotein. (Continuous) time series of (**a**) $S\alpha$ and (**b**) S_{hc} for a representative replica diffusing in temperature. Time is measured per replica. It is crucial to reconstruct the continuous trajectories of the replicas to assess whether the excursions in temperature do lead to an exhaustive sampling of the CVs space

Even if the behavior is reassuring of the correctness of the simulation setup, further analysis is required to declare convergence. Indeed, due to the REM protocol, 100 trajectories contribute to the statistics reported in Fig. 6. Since all the replicas are prepared in different regions of the CV space, the exhaustive sampling observed in Fig. 6 could result from the exchanges between replicas, which individually still suffer from sampling problems. Therefore we have to check the trajectories of each individual replica across temperatures to assess the diffusion in the CVs space (Fig. 7 and *see* **Note 6**). This step is crucial to verify whether diffusion in temperature space can effectively help the system crossing barriers in degrees of freedom not included in the CVs.

4.4 PT-MetaD-WTE

To reduce the number of replica needed to cover a given temperature range, we couple PT-MetaD with WTE. Enlarging the energy fluctuations by means of WTE leads to an increase overlap between potential energy distributions, thus allowing the use of a larger spacing between temperatures. Here we setup a simulation using only ten replicas and choosing the WTE parameters in order to achieve diffusion in temperature comparable to the PT-MetaD simulation described above (*see* **Note 7**).

In typical solvated biomolecular systems, the PT-MetaD-WTE protocol is best carried out in two steps (*see* **Note 8**). First, the bias in the potential energy space is converged at each temperature, so that the WTE is sampled at the end of this preliminary step. For this simulation, we use an initial deposition rate equal to 2.0 kJ/mol every 0.25 ps. Each Gaussian has a width equal to 500 kJ/mol. The bias factor γ is set to 24. The following input file can be used to run the WTE simulation with PLUMED 2:

```
# CV definition
ene: ENERGY

# MetaD parameters
METAD...
  ARG=ene SIGMA=500 HEIGHT=2.0 PACE=125
  TEMP=300 BIASFACTOR=24 FILE=HILLS
  GRID_MIN=-175000 GRID_MAX=-75000 GRID_BIN=500
  GRID_WSTRIDE=500000 GRID_WFILE=BIAS
...METAD
```

In Fig. 8a we show the trajectory in energy space along a preliminary 0.5 ns simulation in the NVT ensemble, followed by a

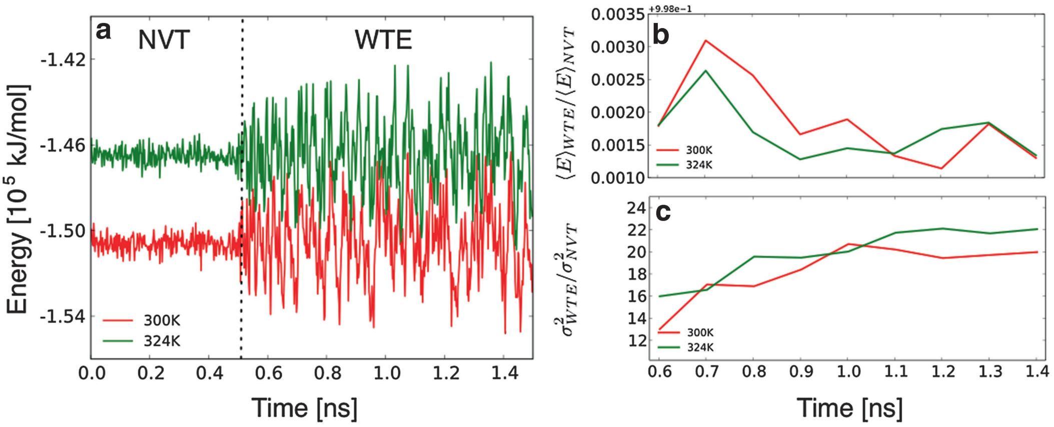


Fig. 8 Sampling WTE at two representative temperatures (300 and 324 K). (**a**) Time series of the potential energy during a 0.5 ns preliminary NVT simulation, followed by a 1 ns MetaD simulation used to converge the bias on the potential energy. While in the NVT part of the trajectory, the potential energy distributions at the two temperatures are well separated, in the WTE ensemble a significant overlap is obtained. (**b**, **c**) Time series of the ratio of the average potential energy (**b**) and fluctuations (**c**) to the canonical values. At the end of the simulation, these ratios are close to the theoretical WTE values ($\gamma = 24$)

1 ns MetaD simulation to converge the WTE bias at two representative temperatures. At the end of the latter simulation, the potential energy averages and fluctuations are close to the theoretical WTE values (Fig 8b and *see* **Note 9**).

In a second step, we run a PT-MetaD-WTE simulation using a history-dependent two-dimensional MetaD bias on $S\alpha$ and S_{hc} and a static bias on the energy. The latter has been stored on a grid and written to file (BIAS) at the end of the preliminary WTE run described above (*see* **Note 10**). The following input file can be used to run the PT-MetaD-WTE simulation with PLUMED 2:

```
# Groups definition
h:    GROUP ATOMS=85,107,125,145,166,181,215,238
o:    GROUP ATOMS=4,20,39,60,79,96,173,180
core: GROUP ATOMS=46,106,189,270

# CVs definition
s_a:  COORDINATION GROUPA=h GROUPB=o PAIR  NN=8 MM=12 R_0=0.25
s_hc: COORDINATION GROUPA=core GROUPB=core NN=8 MM=12 R_0=0.50
ene:  ENERGY

# MetaD parameters
METAD...
  ARG=s_a,s_hc SIGMA=0.4,0.4 HEIGHT=2.5 PACE=250
  TEMP=300 BIASFACTOR=8 FILE=HILLS
...METAD

# WTE bias
EXTERNAL ARG=ene FILENAME=BIAS
```

As done for the PT-MetaD simulation, we analyze the overall trajectory of each replica in temperature space (Fig. 5b and *see* **Note 6**). Despite having fewer replicas, diffusion in temperature is similar to that observed in PT-MetaD, thanks to the bias on energy. To make a more quantitative comparison, we calculate the average round-trip time τ, i.e., the time needed for a replica to move from the lowest to the highest temperature and back. This analysis shows that the two setups result into a very similar behavior ($\tau_{\mathrm{PT-MetaD}} = 6.2$ ns, $\tau_{\mathrm{PT-MetaD-WTE}} = 4.0$ ns). Finally, the evolution in the $S\alpha$-S_{hc} space can be checked using the analysis described in the previous section.

4.5 Analysis of the FES: Convergence and Error Estimate

The FES as a function of the MetaD CVs can be calculated by integrating the Gaussians deposited along the simulation after proper rescaling (*see* **Note 11**). For the PT-MetaD-WTE simulations, an additional step needs to be performed, i.e., the removal of the effect of the static bias on energy. This can be easily done by applying a Torrie–Valleau correction [39] to the statistics accumulated in the WTE ensemble.

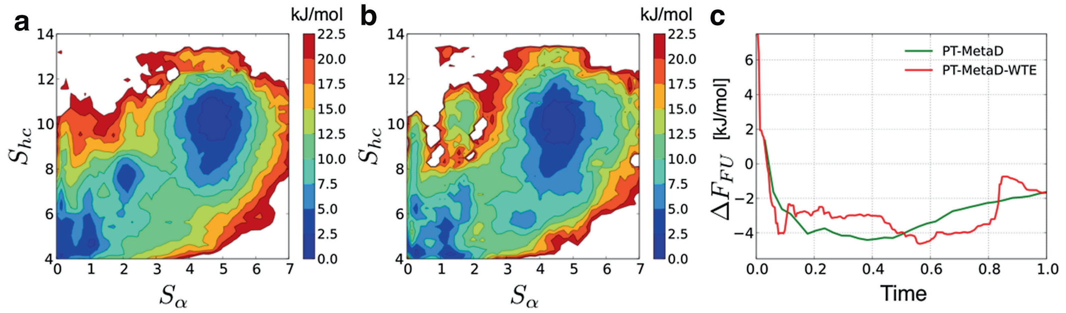


Fig. 9 Trp-Cage miniprotein FES from the PT-MetaD (**a**) and PT-MetaD-WTE (**b**) simulations. (**c**) Convergence can be assessed by monitoring the free-energy differences between relevant regions of the CVs space as the simulation progresses. Time is measured as the fraction of the total aggregated time (simulation time per replica multiplied by the number of replicas), i.e., 5.0 μs and 2.5 μs for PT-MetaD and PT-MetaD-WTE, respectively

In Fig. 9a,b, we report the FES obtained from the PT-MetaD and PT-MetaD-WTE simulations. By calculating the estimate of the FES as a function of time, we can assess more quantitatively the convergence of our simulation. Since monitoring the evolution of a multi-dimensional profile may represent a complex task, one typically adopts a coarse-grained representation of the FES. This can be done by defining relevant regions in the CVs space and calculating free-energy differences between pairs of them as a function of the simulation time (*see* **Note 12**). In our example, we define a folded (*F*) and an unfolded region (*U*) based on the value of *S*α alone (*U*: *S*α < 3; *F*: *S*α ≥ 3). ΔF_{FU} are calculated from the PT-MetaD and PT-MetaD-WTE simulations and plotted in Fig. 9c as a function of the simulation time.

The damped fluctuations in Fig. 9c provide an intuitive estimate of the precision of the free-energy reconstruction as a function of the simulation time (*see* **Note 13**). If an exact reference profile were available, one could estimate the accuracy by measuring the deviation of the calculated FES from the reference (*see* **Note 14**).

5 Notes

1. While in analysis the simplest definition of contact is a step function of the interatomic distance, in MetaD we need to use a continuous and differentiable function in order to calculate the additional forces due to the bias potential.
2. In order to perform a well-tempered MetaD simulation, one has to set the following parameters: the Gaussian width σ_i (one per CV), the initial deposition rate ω_0, and the well-tempered bias factor γ. The Gaussian width should be comparable to the shape of basins in the underlying FES. This can be estimated a

priori by performing short unbiased MD simulations and computing CV fluctuations. Typically, the Gaussian width should not be greater than 1/3 of the fluctuations. Recently, a more advanced approach has been proposed to automatically tune this parameter [40]. The initial deposition rate does not affect the long time behavior [11]. However, a small initial deposition rate would result in a longer filling time, while a too high rate might be problematic in the transient regime if the CVs are not properly chosen. Typically, in simulation of proteins an initial deposition rate of at most $1k_{\mathrm{B}}T$ per ps is used. The bias factor affects the probability distribution of the CV in the long time limit. Therefore, the optimal bias factor should be large enough to cross all the relevant barriers in the process under study, and small enough to limit sampling to the relevant regions of the CV space. As discussed in ref. 11, overestimation of the optimal value is to be preferred to underestimation.

3. In PT-MetaD simulations, one should carefully choose the value of the minimum and maximum temperature and how the replicas are distributed in this interval. The lowest temperature usually corresponds to the temperature of interest (typically 300 K). The highest temperature should guarantee a fast sampling of all the degrees of freedom other than the CVs. Replicas should be chosen so that a sufficient overlap between potential energy distributions of neighboring temperatures is achieved, thus guaranteeing a good acceptance probability for the exchanges. The proper distribution depends on the system specific heat and its dependence on temperature. When simulating proteins in explicit solvent in the NVT ensemble, the maximum temperature is typically set to 600–700 K and the appropriate temperature distribution is given in ref. 41.
4. Equilibration of all the replicas prior to applying the MetaD bias is of great importance in PT-MetaD. It is indeed not convenient to initiate the simulation from identical conformations since this would result in a long transient due to the large accumulation of bias in the initial region of the CVs space. Equilibration can be achieved either through PT run or multiple NVT simulations.
5. Some of the algorithms routinely used to perform simulations at constant temperature cannot reproduce the correct energy fluctuations of the NVT ensemble. Since the exchange process of the REM protocol is strongly dependent on the potential energy distributions, one must implement a correct thermostat, such as Nose–Hoover [42], Langevin, or Bussi–Donadio–Parrinello [43], to avoid possible artifacts [44].
6. The trr and xtc files produced by GROMACS contain configurations sampled at constant temperature; therefore, due to the PT exchanges between replicas, these are not continuous

trajectories. To reconstruct the continuous trajectory of each replica across the temperature space, one can use the GROMACS tools demux.pl and trjcat (see GROMACS manual for their respective use). Once these trajectories are reconstructed, one can use the PLUMED tool driver to recalculate the continuous CVs trajectory of each replica.

7. The γ parameter modulates the increase of the potential energy fluctuations in WTE and thus the overlap between potential energy distributions of neighboring temperatures. For a given system, one can tune γ in order to (1) obtain a more efficient diffusion in temperature once the number of temperatures is fixed, (2) reduce the number of replicas needed to span the temperature range of interest, without affecting the exchange efficiency. A wide range of different choices is thus possible, depending on the computational resources available to the user. According to ref. 17, a key quantity in PT-MetaD-WTE is the average replica round-trip time in temperature space, which should be minimized to achieve optimal efficiency in the FES reconstruction. However, choosing a too high γ is discouraged since it has several drawbacks: (1) slow convergence of the bias in energy space, (2) exploration of unlikely regions of the phase space, such as conformations in which the water is frozen, (3) inefficiency of reweighting procedures when the simulated ensemble is very different from the original one [45]. As an additional check to verify the WTE part of the simulation was performed correctly, we also strongly suggest carefully monitoring the early stages of a PT-MetaD-WTE simulation in order to ensure the expected exchanges are obtained.

8. Typically, in solvated biomolecular systems the potential energy and its fluctuations are dominated by the solvent degrees of freedom. Thus, the potential energy distribution is quasi-Gaussian and it is almost decoupled from the CVs that are used to describe the protein conformation. If one focuses on protein properties, it is convenient to use two different bias potentials: a one-dimensional static bias on the solvent-dominated potential energy, and a N-dimensional MetaD bias acting on the protein CVs. This strategy is to be preferred to the introduction of a (N+1)-dimensional MetaD bias acting simultaneously on the potential energy and protein CVs.

9. Two important things should be kept in mind when performing simulations in the WTE ensemble (both PT-WTE and PT-MetaD-WTE). First, during the course of the different steps, one should not change those MD parameters that affect the absolute value of the potential energy, such as the scheme to calculate the electrostatic potential or the cutoffs for electrostatic and van der Walls interactions. Second, the energy must be calculated at every time step to apply correct forces to

the dynamics. In GROMACS, this can be done by setting in the input file nstcalcenergy = 1.

10. The bias on the potential energy needed to sample the WTE can be stored on file in a grid format for later use. The boundaries of the grid should be specified in the PLUMED input file and must include the range of potential energy sampled at all temperatures. In order to properly evaluate the bias potential, the dimension of the grid side should be equal to at most half of the Gaussian width.

11. As discussed in the Subheading 2, we can estimate the FES from the Gaussians deposited along a MetaD simulation. In practice, this can be done using the PLUMED sum_hills tool, which automatically integrates the Gaussians and applies the correct scaling factor (Eq. 4).

12. When assessing convergence, it is convenient to define regions in the FES corresponding to local minima and to calculate free-energy differences between pair of them. In order to do that, we need to define the free energy of a basin (A) as:

$$F_{\mathrm{A}} = -k_{\mathrm{B}}T \cdot \log\left(\int_{S \in A} \mathrm{d}S \exp\left(-\frac{F(S)}{k_{\mathrm{B}}T} \right) \right). \tag{14}$$

13. The statistical error in a well-tempered MetaD simulation is proportional to $1/\sqrt{t}$ [11]. In practical applications, the precision in FES reconstruction has been estimated by the following approaches:
    (a) Using the standard deviation of the FES obtained by multiple independent MetaD simulations [46];
    (b) Using the standard deviation of the FES reconstructed at different times in a single simulation [47]. However, each point should be weighted proportionally to *t*, following the relation between error and simulation time mentioned above.

14. Different metrics have been used to compare an estimate of the free energy $F(S)$ to a reference profile $F_{\mathrm{REF}}(S)$ (*see* for example [48]). Among these:
    (a) The RMSD:

$$d_{\mathrm{RMSD}}\left(F(S), F_{\mathrm{REF}}(S)\right) = \sqrt{\frac{1}{\Omega_{\Omega}} \int \mathrm{d}S \left(F(S) - F_{\mathrm{REF}}(S)\right)^2}, \tag{15}$$

where Ω is the volume of the CVs space explored. This volume is often chosen to include only relevant regions of the CVs space, i.e., points within a few $k_{\mathrm{B}}T$ from the global minimum.

(b) The Kullback–Leibler divergence [49]:

$$d_{\mathrm{KL}}\left(F(S),F_{\mathrm{REF}}(S)\right)=\int_{\Omega}\mathrm{d}S\,\exp\left(-\frac{F_{\mathrm{REF}}(S)}{k_{\mathrm{B}}T}\right)\cdot\frac{F(S)-F_{\mathrm{REF}}(S)}{k_{\mathrm{B}}T},\quad(16)$$

which assigns a weight to each point based on its reference probability density.

Since free energies are always defined modulo an irrelevant additive constant, one should optimally align the two profiles in order to minimize the distance between them (using Eqs. 15, 16, or other metrics). When using RMSD as metrics, optimal alignment can be achieved by calculating the average free energy in the volume of interest Ω and subtract it to the profile, so that each free energy is offset to have its average value at zero.

Acknowledgements

AB thanks the Swiss National Science Foundation for financial support under the Ambizione grant PZ00P2_136856. J. P. acknowledges the support of NSF award CMMI-1032368. The simulations of Trp-Cage miniprotein were made possible in part by the National Science Foundation through TeraGrid resources provided by NICS. These simulations were also facilitated through the use of computational, storage, and networking infrastructure provided by the Hyak supercomputer system, supported in part by the University of Washington eScience Institute.

References

1. Shaw DE, Maragakis P, Lindorff-Larsen K et al (2010) Atomic-level characterization of the structural dynamics of proteins. Science 330:341–346
2. Beberg AL, Ensign DL, Jayachandran G, Khaliq S, Pande VS (2009) Folding@home: lessons from eight years of volunteer distributed computing, IEEE International Symposium on, Parallel & Distributed Processing, 2009. IPDPS 2009, 23-29 May 2009, Rome, pp. 1624–1631
3. Chipot C, Pohorille A (2007) Free energy calculations: theory and applications in chemistry and biology. Springer, Berlin
4. Dellago C, Bolhuis PG (2009) Transition path sampling and other advanced simulation techniques for rare events. Adv Polym Sci 221:167–233
5. Laio A, Parrinello M (2002) Escaping free-energy minima. Proc Natl Acad Sci U S A 99:12562–12566
6. Barducci A, Bonomi M, Parrinello M (2011) Metadynamics. Wir Comput Mol Sci 1:826–843
7. Sugita Y, Okamoto Y (1999) Replica-exchange molecular dynamics method for protein folding. Chem Phys Lett 314:141–151
8. Hansmann UHE (1997) Parallel tempering algorithm for conformational studies of biological molecules. Chem Phys Lett 281: 140–150
9. Neidigh JW, Fesinmeyer RM, Andersen NH (2002) Designing a 20-residue protein. Nat Struct Biol 9:425–430
10. Bonomi M, Branduardi D, Bussi G et al (2009) PLUMED: a portable plugin for free-energy

calculations with molecular dynamics. Comput Phys Commun 180:1961–1972

11. Barducci A, Bussi G, Parrinello M (2008) Well-tempered metadynamics: a smoothly converging and tunable free-energy method. Phys Rev Lett 100:020603
12. Bonomi M, Parrinello M (2010) Enhanced sampling in the well-tempered ensemble. Phys Rev Lett 104:190601
13. Bonomi M, Barducci A, Parrinello M (2009) Reconstructing the equilibrium Boltzmann distribution from well-tempered metadynamics. J Comput Chem 30:1615–1621
14. Barducci A, Bonomi M, Parrinello M (2010) Linking well-tempered metadynamics simulations with experiments. Biophys J 98:L44–L46
15. Earl DJ, Deem MW (2005) Parallel tempering: theory, applications, and new perspectives. Phys Chem Chem Phys 7:3910–3916
16. Bussi G, Gervasio FL, Laio A, Parrinello M (2006) Free-energy landscape for beta hairpin folding from combined parallel tempering and metadynamics. J Am Chem Soc 128:13435–13441
17. Deighan M, Bonomi M, Pfaendtner J (2012) Efficient simulation of explicitly solvated proteins in the well-tempered ensemble. J Chem Theory Comput 8:2189–2192
18. Hess B, Kutzner C, van der Spoel D, Lindahl E (2008) GROMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation. J Chem Theory Comput 4:435–447
19. Tribello GA, Bonomi M, Branduardi D, Camilloni C, Bussi G (2014) PLUMED 2: new feathers for an old bird, Comput Phys Commun 185:604–613
20. Pettersen EF, Goddard TD, Huang CC et al (2004) UCSF chimera - A visualization system for exploratory research and analysis. J Comput Chem 25:1605–1612
21. Hunter JD (2007) Matplotlib: a 2D graphics environment. Comput Sci Eng 9:90–95
22. Qiu LL, Pabit SA, Roitberg AE, Hagen SJ (2002) Smaller and faster: the 20-residue Trp-cage protein folds in 4 mu s. J Am Chem Soc 124:12952–12953
23. Streicher WW, Makhatadze GI (2007) Unfolding thermodynamics of Trp-cage, a 20 residue miniprotein, studied by differential scanning calorimetry and circular dichroism spectroscopy. Biochemistry US 46:2876–2880
24. Neuweiler H, Doose S, Sauer M (2005) A microscopic view of miniprotein folding: enhanced folding efficiency through formation of an intermediate. Proc Natl Acad Sci U S A 102:16650–16655
25. Ahmed Z, Beta IA, Mikhonin AV, Asher SA (2005) UV-resonance Raman thermal unfolding study of Trp-cage shows that it is not a simple two-state miniprotein. J Am Chem Soc 127:10943–10950
26. Zhou RH (2003) Trp-cage: folding free energy landscape in explicit water. Proc Natl Acad Sci U S A 100:13280–13285
27. Ota M, Ikeguchi M, Kidera A (2004) Phylogeny of protein-folding trajectories reveals a unique pathway to native structure. Proc Natl Acad Sci U S A 101:17658–17663
28. Juraszek J, Bolhuis PG (2006) Sampling the multiple folding mechanisms of Trp-cage in explicit solvent. Proc Natl Acad Sci U S A 103:15859–15864
29. Paschek D, Nymeyer H, Garcia AE (2007) Replica exchange simulation of reversible folding/unfolding of the Trp-cage miniprotein in explicit solvent: on the structure and possible role of internal water. J Struct Biol 157:524–533
30. Marinelli F, Pietrucci F, Laio A, Piana S (2009) A kinetic model of trp-cage folding from multiple biased molecular dynamics simulations. PLoS Comput Biol 5:e1000452
31. Lindorff-Larsen K, Piana S, Dror RO, Shaw DE (2011) How fast-folding proteins fold. Science 334:517–520
32. Hornak V, Abel R, Okur A et al (2006) Comparison of multiple amber force fields and development of improved protein backbone parameters. Proteins 65:712–725
33. Jorgensen WL, Chandrasekhar J, Madura JD, Impey RW, Klein ML (1983) Comparison of simple potential functions for simulating liquid water. J Chem Phys 79:926–935
34. Branduardi D, Gervasio FL, Parrinello M (2007) From A to B in free energy space. J Chem Phys 126:054103
35. Ceriotti M, Tribello GA, Parrinello M (2011) Simplifying the representation of complex free-energy landscapes using sketch-map. Proc Natl Acad Sci U S A 108:13023–13028
36. Spiwok V, Kralova B (2011) Metadynamics in the conformational space nonlinearly dimensionally reduced by Isomap. J Chem Phys 135:224504
37. Piana S, Laio A (2007) A bias-exchange approach to protein folding. J Phys Chem B 111:4553–4559
38. Sindhikara DJ, Emerson DJ, Roitberg AE (2010) Exchange often and properly in replica exchange molecular dynamics. J Chem Theory Comput 6:2804–2808
39. Torrie GM, Valleau JP (1977) Non-physical sampling distributions in monte-carlo free-energy

estimation - umbrella sampling. J Comput Phys 23:187–199

40. Branduardi D, Bussi G, Parrinello M (2012) Metadynamics with adaptive gaussians. J Chem Theory Comput 8:2247–2254

41. Prakash MK, Barducci A, Parrinello M (2011) Replica temperatures for uniform exchange and efficient roundtrip times in explicit solvent parallel tempering simulations. J Chem Theory Comput 7:2025–2027

42. Nose S (1984) A unified formulation of the constant temperature molecular-dynamics methods. J Chem Phys 81:511–519

43. Bussi G, Donadio D, Parrinello M (2007) Canonical sampling through velocity rescaling. J Chem Phys 126:014101

44. Rosta E, Buchete NV, Hummer G (2009) Thermostat artifacts in replica exchange molecular dynamics simulations. J Chem Theory Comput 5:1393–1399

45. Ceriotti M, Brain GAR, Riordan O, Manolopoulos DE (2012) The inefficiency of re-weighted sampling and the curse of system size in high-order path integration. P Roy Soc a-Math Phys 468:2–17

46. Angioletti-Uberti S, Ceriotti M, Lee PD, Finnis MW (2010) Solid-liquid interface free energy through metadynamics simulations. Phys Rev B 81:125416

47. Berteotti A, Barducci A, Parrinello M (2011) Effect of urea on the beta-hairpin conformational ensemble and protein denaturation mechanism. J Am Chem Soc 133: 17200–17206

48. Sutto L, D’Abramo M, Gervasio FL (2010) Comparing the efficiency of biased and unbiased molecular dynamics in reconstructing the free energy landscape of met-enkephalin. J Chem Theory Comput 6:3640–3646

49. Kullback S, Leibler RA (1951) On Information and Sufficiency. Ann Math Stat 22:142–143

Chapter 9

Calculation of Binding Free Energies

Vytautas Gapsys, Servaas Michielssens, Jan Henning Peters, Bert L. de Groot, and Hadas Leonov

Abstract

Molecular dynamics simulations enable access to free energy differences governing the driving force underlying all biological processes. In the current chapter we describe *alchemical* methods allowing the calculation of relative free energy differences. We concentrate on the binding free energies that can be obtained using non-equilibrium approaches based on the Crooks Fluctuation Theorem. Together with the theoretical background, the chapter covers practical aspects of hybrid topology generation, simulation setup, and free energy estimation. An important aspect of the validation of a simulation setup is illustrated by means of calculating free energy differences along a full thermodynamic cycle. We provide a number of examples, including protein–ligand and protein–protein binding as well as ligand solvation free energy calculations.

Key words Free energy, Molecular dynamics, Alchemical transitions, Protein–ligand binding, Protein–protein interaction, Non-equilibrium methods, Hybrid topology, Crooks Fluctuation Theorem

1 Introduction

Whether or not a process happens spontaneously is determined by its free energy. This is because, without an external source of energy, systems evolve to their lowest free energy state. Likewise, the rate at which that state is reached depends on free energy barriers along the pathways to that minimum. Hence, free energies are of central importance as they determine, e.g., binding affinities (spontaneous binding or not) or protein folding (the folded state is usually the free energy minimum). In addition, as barriers are linked to rates via rate theory, free energy barriers determine binding, folding, permeation, and reaction kinetics. Therefore, free energies are among the most critical thermodynamic quantities to accurately be derived by computational techniques, not only because they play such a fundamental role, but also because they can be directly and quantitatively compared to experimental data.

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_9, © Springer Science+Business Media New York 2015

Thermodynamically, a distinction is made between the so-called Helmholtz and Gibbs free energies. The Helmholtz free energy is defined as:

$$F = U - TS \tag{1}$$

with U the internal energy, T the temperature, and S the entropy of the system. The Gibbs free energy is defined as:

$$G = H - TS \tag{2}$$

with H the enthalpy of the system, defined as $H = U + pV$, with p the pressure and V the volume of the system.

Importantly, note that both definitions contain a term that is dependent on the internal (or potential) energy of the system, and another that depends on the entropy. This has the important implication that free energies, and hence, affinities or stabilities, are affected by both changes in interatomic interactions as well as entropic changes. Particularly, changes in solvent entropy can play a substantial or even dominant role as, e.g., in the hydrophobic effect, where solvent molecules are set free upon the formation of a hydrophobic protein or membrane core.

The difference between the Helmholtz and Gibbs free energy lies in the pV term. The Helmholtz free energy is applicable at constant volume, whereas the Gibbs free energy is used at constant pressure, where the pV term quantifies the work associated with a change in volume. Under physiological conditions and for a typical experimental setup we usually have isobaric conditions, hence the Gibbs free energy gradient acts as a driving force for a system. While we will return to the concepts of the Helmholtz and Gibbs free energies in Subheading 2.1, for the practical examples in this chapter we will focus on the Gibbs free energy.

In addition to being the quantity that is minimized at equilibrium, the free energy is also the *maximum* amount of energy that can be obtained from a spontaneous process at constant temperature, or conversely, the *minimum* amount of energy required to drive an uphill process. More precisely, the free energy is the energy obtained from (or required to drive) a process slow enough such that it is in equilibrium with its surroundings at all times. If enforced faster, friction results in non-equilibrium work values that on average are larger than the associated free energy change. The excess energy is dissipated as heat. Traditional free energy methods therefore require transitions slow enough such that the systems under investigation can be considered to be in equilibrium at all times. Remarkably, however, with the Jarzynski equality and the Crooks Fluctuation Theorem (CFT) given, free energies can also be derived from non-equilibrium work distributions, as described further below.

An important implication of the free energy follows from its role in statistical mechanics, stating that the probability to be in state *x* is directly related to its free energy:

$$p(x) \propto e^{-G(x)/k_B T} \tag{3}$$

where k_B is the Boltzmann constant. The term $e^{-G(x)/k_B T}$ is called the Boltzmann factor. This relation is particularly useful to estimate free energy differences between two states *A* and *B*:

$$\frac{p(A)}{p(B)} = e^{-(G(A)-G(B))/k_B T} = e^{-\Delta G/k_B T} \tag{4}$$

which also presents the most straightforward way to estimate free energies from simulation: provided that a sufficiently converged ensemble is available, usually requiring several reversible transitions between *A* and *B*, the free energy difference ΔG can directly be estimated from this relation by evaluating the populations in *A* and *B*.

Alternative simulation approaches to derive free energies are frequently tailored towards overcoming or avoiding free energy barriers associated, e.g., with binding, unbinding, conformational transitions, or (re)folding. Examples include, umbrella sampling [1] and thermodynamic integration [2]. In umbrella sampling, a biasing potential is employed to enforce transitions across high energy states along a predefined reaction coordinate. The effect of the biasing potential can be corrected for afterwards, yielding the free energy profile for a one-dimensional reaction coordinate or the free energy landscape for a multi-dimensional reaction coordinate. In thermodynamic integration and other *alchemical* approaches, the sampling of cumbersome binding and unbinding events is prevented by instead carrying out a transition in chemical space where the Hamiltonian describing system *A* is transformed into *B*. Making use of the fact that the free energy is a state variable, the relative free energy change can be estimated from the difference of the associated transition free energy for example in the context of a protein binding pocket and the same transformation in solution, for the calculation of relative binding free energies.

A remarkable characteristic of the free energy is that in many cases, its estimate converges faster than either the individual enthalpy and entropy contributions. This is due to the fact that the largest contributions to both arise from a solvent–solvent term that occurs in both but that exactly cancels out in the free energy estimate [3].

This chapter mainly focuses on the use of non-equilibrium methods for the calculation of binding free energies. For equilibrium approaches, we refer to excellent reviews in the literature [4–6].

2 Theory

2.1 Definition of Free Energy

The free energy surface of a system completely describes its thermodynamic and kinetic properties. An intuitive interpretation of the free energy is provided by its relation to the probability p_A of finding the system in a phase space volume A:

$$p_A = \frac{e^{-\beta F_A}}{e^{-\beta F}} \tag{5}$$

where F_A is the free energy of a phase space volume A, F is the Helmholtz free energy evaluated over the whole phase space, and $\beta = 1/k_B T$, with the Boltzmann constant k_B and temperature T. In most practical cases the *differences* between the state populations are of interest in contrast to the absolute free energy values. By considering two states, A and B, the free energy difference ΔF_{AB} can be expressed as

$$\Delta F_{AB} = F_B - F_A = -\frac{1}{\beta}\ln\frac{p_B}{p_A} = -\frac{1}{\beta}\ln\frac{Q_B}{Q_A} \tag{6}$$

Q denotes the canonical partition function: $Q = Q(N, V, T)$, with N number of particles and V volume of a container. The free energy is related to the partition function via $F = -1/\beta \ln Q(N, V, T)$.

From the latter expression (Eq. 6) several approaches to obtaining ΔF can be deduced. Firstly, a direct counting of events sampling phase space volumes A and B immediately enables access to the probabilities p_A and p_B. For example, in case of molecular binding, this method could be employed by simulating two molecules of interest (ligands A and B) and directly counting their respective probabilities of being in bound/unbound forms. While the approach is simple, its computational cost for large biologically relevant systems is usually beyond reach for the current state-of-the-art atomistic simulations.

Another method requires evaluation of the partition functions. However, while for an ideal gas example, the partition function has an analytical expression, for particles with complex interactions a numerical integration over the coordinates and momenta is required. A canonical partition function is defined as follows:

$$Q(N,V,T) = \frac{1}{h^{3N}N!}\int \ldots \int e^{-\beta H(\mathbf{p}_1 \ldots \mathbf{p}_N, \mathbf{q}_1 \ldots \mathbf{q}_N)} d\mathbf{p}_1 \ldots d\mathbf{p}_N d\mathbf{q}_1 \ldots d\mathbf{q}_N \tag{7}$$

where $H(\mathbf{p}, \mathbf{q})$ is a Hamiltonian of a system, $\mathbf{q}$ and $\mathbf{p}$ denote coordinates and momenta, respectively, h is Planck's constant. For a multi-particle system, integration over all the degrees of freedom is not computationally feasible. Hence, a simulation is often used to sample the accessible phase space volume. In a simulation, high energy microstates will be visited only rarely (or will not be visited at all), hence, rendering this approach unsuitable for the estimation of absolute free energies. For the free energy differences, however, the inaccessible phase space regions will be discarded for both states, *A* and *B*, thus allowing for an accurate assessment of ΔF due to cancellation of errors.

Up to now, we considered a canonical ensemble with the associated canonical partition function *Q* and Helmholtz free energy *F*. In practice, however, experimental measurements are usually performed at isothermal-isobaric conditions generating an *NPT* ensemble. In such a case, a partition function is defined as

$$Q(N,P,T)=\frac{1}{h^{3N}N!}\int\ldots\int e^{-\beta(H(\mathbf{p}_1\ldots\mathbf{p}_N,\mathbf{q}_1\ldots\mathbf{q}_N)+PV)}dVd\mathbf{p}_1\ldots d\mathbf{p}_N d\mathbf{q}_1\ldots d\mathbf{q}_N \qquad (8)$$

where *P* is the pressure. The Gibbs free energy is defined as $G=-1/\beta\ln Q(N,P,T)$. For the remainder of this chapter we assume constant pressure and therefore focus on the *NPT* ensemble with the associated Gibbs free energy. By considering the fact that a Hamiltonian of a system consists of a kinetic and potential energy components $H(\mathbf{p},\mathbf{q})=K(\mathbf{p})+U(\mathbf{q})$, the partition function can be separated into kinetic and configurational partition functions. As the $K(\mathbf{p})$ component depends only on the particle momenta, the kinetic partition function for states *A* and *B* does not change as long as there is no perturbation of mass between the states. Therefore, for the sake of simplicity, often only the configurational partition function is used. However, in this chapter, we will use a more general approach by considering the full Hamiltonian of a system.

In the following section, we will briefly introduce the most popular approaches for free energy estimation from equilibrium simulations. Further, we will concentrate on the conceptually different approaches relying on non-equilibrium transitions between states.

2.2 Free Energy Estimates from Equilibrium Simulations: Perturbation Method

The free energy perturbation (FEP) method introduced by Zwanzig [7] (Pohorille et al. [8] attribute the first derivation to Landau [9]) can be derived from Eq. 6:

$$\Delta G_{AB}=G_B-G_A=-\frac{1}{\beta}\ln\frac{Q_B}{Q_A}=-\frac{1}{\beta}\ln\left\langle e^{-\beta(H_B(\mathbf{p},\mathbf{q})-H_A(\mathbf{p},\mathbf{q}))}\right\rangle_A \qquad (9)$$

where the angular brackets $\langle\ldots\rangle$ denote an ensemble average. The approach relies on equilibrium sampling at state *A* and subsequent evaluation of the produced configurations with the Hamiltonian of state *B*. The Hamiltonian is controlled by an external parameter often denoted as λ: $H_\lambda(\mathbf{p},\mathbf{q})=H(\mathbf{p},\mathbf{q},\lambda)$. The accuracy of the FEP method is strongly dependent on the phase space overlap of the states *A* and *B*. In case the overlap is small, the configurations generated at state A will be identified as high energy microstates when evaluated with the Hamiltonian $H_B(\mathbf{p}, \mathbf{q})$, in turn contributing little to the exponential average. Hence, the approach is known to converge slowly and is only tractable for the ΔG estimation between the states exhibiting large overlap in the phase space. More information concerning the accuracy, convergence, and usage of FEP can be found in [8, 10–12].

Another method to estimate ΔG from end state equilibrium sampling was proposed by Bennet and is referred to as Bennet's Acceptance Ratio (BAR) [13]. Bennet's estimate is expressed by:

$$\Delta G_{AB} = \frac{1}{\beta} \ln \frac{\langle f(H_A(\mathbf{p},\mathbf{q}) - H_B(\mathbf{p},\mathbf{q}) + C) \rangle_B}{\langle f(H_B(\mathbf{p},\mathbf{q}) - H_A(\mathbf{p},\mathbf{q}) - C) \rangle_A} + C \tag{10}$$

here f represents Fermi function $f(x) = 1/(1 + \exp(\beta x))$ and $C = \frac{1}{\beta} \ln \left(\frac{Q_A}{Q_B} \frac{n_B}{n_A} \right)$, n_A and n_B are the numbers of configurations generated in the states A and B, respectively. Equation (10) can be solved numerically by finding such C, that

$$\sum_B f(H_A(\mathbf{p},\mathbf{q}) - H_B(\mathbf{p},\mathbf{q}) + C) = \sum_A f(H_B(\mathbf{p},\mathbf{q}) - H_A(\mathbf{p},\mathbf{q}) - C) \tag{11}$$

Having determined C yields the free energy difference: $\Delta G = -\frac{1}{\beta} \ln \frac{n_B}{n_A} + C$. BAR is a minimal variance free energy estimate. Shirts et al. [14] also derived Bennet's formula using maximum likelihood formulation, thus, demonstrating that BAR provides the most likely free energy estimate for the observed work distributions.

Both methods introduced so far (FEP and BAR) were defined for free energy calculations by sampling physical end states of a system. Simulations, however, allow accessing unphysical (*alchemical*) pathways as well, by coupling the Hamiltonians of the two states $H_\lambda = (1 - \lambda)H_A + \lambda H_B$. Different λ dependent coupling functions have been investigated [15, 16]. For the λ values 0 and 1, the system is at the physical states A and B, respectively, whereas for the values $0 < \lambda < 1$ the system is in a mixed unphysical state. The path between the states A and B can be divided into discrete states (also called stratification). Performing equilibrium simulations at the intermediate states ensures a larger phase space overlap between the ensembles adjacent to one another along the λ coordinate. FEP or BAR estimators can be employed to obtain free energy differences between the intermediate states which can later be summed up to yield ΔG_{AB} between the physical end states. The weighted histogram analysis (WHAM) [17] and multistate Bennet Acceptance Ratio (MBAR) [18] methods were developed to estimate free energy difference by considering all intermediate states simultaneously. Ways to achieve an optimal distribution of the discrete states to ensure maximal phase space overlap along the path have also been investigated [19, 20].

2.3 Free Energy Estimates from Equilibrium Simulations: Thermodynamic Integration

A conceptually different method from the ones described above, termed thermodynamic integration (TI) [2], obtains the free energy difference by integrating the average force exerted on the system along the λ variable during an *alchemical* transition:

$$\Delta G_{AB} = \int_0^1 \left\langle \frac{\partial H}{\partial \lambda} \right\rangle_\lambda d\lambda \tag{12}$$

Several approaches have been developed to use thermodynamic integration in simulations. The slow growth TI requires performing transition between the states very slowly, such that the system remains close to equilibrium at all times. In this case, the free energy value is equal to the work done during the transition. Another assumption states that an instantaneous $\partial H/\partial\lambda$ value at any λ_i is equal to an ensemble average at that λ_i. In practice, due to limited sampling the system is kept at a quasi-equilibrium state, which in turn results in inaccurate free energy estimates as some work is dissipated. Convergence issues related to the slow growth TI are well known and have been thoroughly investigated [21–23].

A discrete thermodynamic integration method (DTI) is based on dividing the path along the λ coordinate into discrete steps, similarly as for the FEP and BAR approaches discussed in the previous section. An equilibrium simulation is started at every λ_i state and an average $\left\langle \frac{\partial H}{\partial \lambda} \right\rangle_{\lambda_i}$ is evaluated. A numerical integration of the averages directly yields a ΔG value. The accuracy of DTI depends on the sampling time and distribution of the discrete states. As it was the case for the multi-state FEP and BAR methods, DTI requires phase space overlap between the neighboring ensembles along the λ coordinate. The numerical integration scheme was also demonstrated to influence accuracy of the free energy estimates [20, 24, 25].

2.4 Theory of Non-equilibrium Free Energy Calculations

The methods described in the current section rely on the work measurements over non-equilibrium transitions, and hence are sometimes termed NEW. In 1997 Jarzynski derived an identity relating an exponential average of work during non-equilibrium transitions to the free energy difference of the canonical ensemble [26] (it was later shown by Cuendet [27], that the equality also holds for an *NPT* ensemble):

$$e^{-\beta\Delta G_{AB}} = \left\langle e^{-\beta W} \right\rangle \tag{13}$$

The Jarzynski equality requires the transitions to be started from an equilibrium ensemble. The work values can be obtained from a relation similar to that of thermodynamic integration:

$$W = \int_0^1 \frac{\partial H}{\partial \lambda} d\lambda \tag{14}$$

In contrast to TI, in Eq. 14 instantaneous $\partial H/\partial \lambda$ values are integrated. The resulting work contains both a contribution from the free energy difference and the dissipated work along the transition path. The FEP approach (Eq. 9) can be viewed as a special case of the Jarzynski equality, where a transition from A to B is performed instantaneously.

Similarly to FEP, of which the convergence strongly depends on the phase space overlap between the end states, the accuracy of the free energy differences calculated using Jarzynski's formula depends on rare events, where little work is dissipated. Fast transitions, driving a system far from equilibrium, would yield large work values, contributing little to the exponential average, hence slowing the convergence of the ΔG estimation.

Several free energy estimators have been developed based on the Jarzynski equality. In case the transitions are performed in the near equilibrium regime, a Gaussian approximation for the work distribution $P(W)$ is valid due to the central limit theorem [28]. Cumulant expansion of a Gaussian distribution allows expressing the free energy difference as

$$\Delta \hat{G} = \langle W \rangle_n - \frac{\beta \hat{\sigma}_W^2}{2} \tag{15}$$

where $\langle W \rangle_n$ is the mean and $\hat{\sigma}_W^2$ is the variance of a $P(W)$ distribution. The variance of a free energy estimate is given by $\hat{\sigma}_W^2 / n + \beta^2 \hat{\sigma}_W^4 / (2n-2)$, with n denoting number of measured work values. The estimator ought to be used only if the assumptions for the Gaussian approximation of a work value distribution are fulfilled, i.e. many work values are obtained and the transitions keep the system near equilibrium [29]. Otherwise, the free energy difference can be estimated from the Jarzynski equality directly. Such an estimator, however, has been shown to be biased given a limited number of observed work values [29]. The Jarzynski estimator is defined as

$$\Delta \hat{G} = -\frac{1}{\beta} \ln \left\langle e^{-\beta W} \right\rangle_n \tag{16}$$

A more general relation, enabling the combination of the work value distributions from forward and backward transitions to obtain the Helmholtz free energy difference was derived by Crooks [30, 31].

The relation is also known as the Crooks Fluctuation Theorem (CFT). Chelli demonstrated the validity of the Crooks equation for an *NPT* ensemble [32].

$$\frac{P_f(W)}{P_r(-W)} = e^{\beta(W-\Delta G)} \tag{17}$$

where $P_f(W)$ and $P_r(-W)$ correspond to the work distributions during forward and reverse processes, respectively. The Jarzynski identity can be derived from the Crooks equality, as demonstrated by Crooks [30]. CFT requires that the transitions between the states were started from equilibrium ensembles. However, similar to the Jarzynski identity, there is no requirement for a system to reach an equilibrium at the final state of a transition. A number of estimators have been developed to extract the free energy difference from the work distributions using Crooks equality [14, 33–35].

For the cases where an overlap between the work distributions is large, the free energy difference can be estimated directly from the CFT:

$$\ln\frac{P_f(W)}{P_r(-W)} = \beta W - \beta\Delta\hat{G} \tag{18}$$

Plotting the left hand of Eq. 18 against the work values yields a line with the slope β. The line intercepts the work axis at a value equal to $\Delta\hat{G}$. While the approach is easy to implement, there are several caveats regarding such a direct estimation. Firstly, a sufficient overlap between the work histograms is achieved only for near-equilibrium transitions where little work is dissipated. Secondly, only the work values from the overlap region will contribute to the free energy estimate, whereas the rest of the measurements will not be used.

To alleviate the latter problems, the work histograms can be approximated by an analytical distribution. Following from the CFT, the intersection point of the two distributions corresponding to the forward and reverse transitions marks a work value equal to ΔG. Nanda et al. [33] proposed using a universal probability density function [36] allowing to account for the asymmetry of the distributions. Goette and Grubmüller showed that in practice a Gaussian approximation also yields accurate free energy estimates [35]. They derived a Crooks Gaussian Intersection (CGI) estimator which is expressed as

$$\Delta\hat{G} = \frac{\dfrac{\langle W_f\rangle_{n_f}}{\hat{\sigma}_f^2} - \dfrac{-\langle W_r\rangle_{n_r}}{\hat{\sigma}_r^2} \pm \sqrt{\dfrac{1}{\hat{\sigma}_f^2\,\hat{\sigma}_r^2}(\langle W_f\rangle_{n_f} + \langle W_r\rangle_{n_r})^2 + 2\left(\dfrac{1}{\hat{\sigma}_f^2} - \dfrac{1}{\hat{\sigma}_r^2}\right)\ln\dfrac{\hat{\sigma}_r}{\hat{\sigma}_f}}}{\dfrac{1}{\hat{\sigma}_f^2} - \dfrac{1}{\hat{\sigma}_r^2}} \tag{19}$$

where $\langle W_f \rangle_{n_f}$, $\langle W_r \rangle_{n_r}$ are work averages and $\hat{\sigma}_f^2$, $\hat{\sigma}_r^2$ are variances of the work distributions for the forward and reverse transitions, respectively. The estimator predicts two intersection points, unless the distributions overlap completely, which would only be the case for equilibrium transitions. The intersection point in between of the $\langle W_f \rangle_{n_f}$ and $-\langle W_r \rangle_{n_r}$ is of relevance. The statistical error for the estimator can be calculated by means of bootstrapping. As the estimator depends on a Gaussian approximation, the validity of this assumption can be assessed using a statistical test, e.g. Kolmogorov–Smirnov [37].

The already introduced BAR estimator (Eq. 10) can also be used for non-equilibrium simulations. Shirts et al. provide the following expression for BAR [14]

$$\sum_{i=1}^{n_f} \frac{1}{1+\exp(\ln \frac{n_f}{n_r} + \beta(W_i - \Delta\hat{G}))} = \sum_{j=1}^{n_r} \frac{1}{1+\exp(\ln \frac{n_r}{n_f} - \beta(W_j - \Delta\hat{G}))} \tag{20}$$

where n_f and n_r are the numbers of transitions in forward and reverse directions, respectively. A numerical solution of Eq. 20 provides a maximum likelihood estimator of the free energy difference. The estimator is asymptotically unbiased and its variance converges to the inverse of a Fisher information when the number of work observations goes to infinity. An analytical expression for the variance [14] is given by

$$\hat{\sigma}^2_{\Delta\hat{G}} = \frac{1}{\beta^2 n_{f+r}} \left(\left\langle \frac{1}{2+2\cosh(\ln \frac{n_f}{n_r} + \beta(W_i - \Delta\hat{G}))} \right\rangle_{n_{f+r}}^{-1} - \left(\frac{n_{f+r}}{n_f} + \frac{n_{f+r}}{n_r} \right) \right) \tag{21}$$

where n_{f+r} corresponds to the total number of forward and reverse transitions and the angular brackets denote averaging over the transitions in both directions. The maximum likelihood estimator was later generalized by Maragakis et al. who introduced a Bayesian free energy estimator based on CFT [34].

2.5 Concepts of Single and Dual Topology

As discussed in the previous sections, *alchemical* free energy calculations explore unphysical pathways by combining Hamiltonians of physical states with an external parameter λ. Having two (or more) separate Hamiltonians implies the necessity to define multiple topologies for the system at every end state. To be able to couple Hamiltonians, a mapping between the topologies needs to be established. Two approaches of constructing the topologies for *alchemical* free energy calculations have been introduced [38, 39].

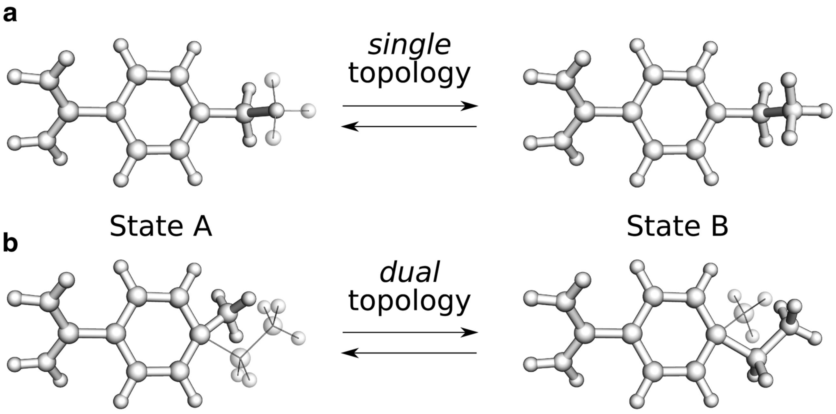


Fig. 1 Two topology generation approaches illustrated by an example of a methyl- to ethyl-benzamidinium mapping. The dummy atoms are represented as *transparent spheres*. (**a**) In the *single* topology approach a methyl group is morphed into an ethyl. (**b**) For the *dual* topology mapping, the whole group is annihilated upon a transition and a new group is created

In a *single* topology approach every atom of state *A* is mapped to an atom of state *B*. In case the states have different number of atoms, dummy particles are introduced. The total number of atoms in a system corresponds to the atom number of the larger state. The bonded interactions of dummy atoms are usually left intact during a transition, resulting in an ideal gas *molecule* state (i.e., the bonded interactions are not switched off) in one of the end-states [40].

In a *dual* topology approach, atoms that are different for the two states are defined separately, i.e. they are present in the system simultaneously. The atoms that are different for the states *A* and *B* do not interact with each other and are controlled by a λ parameter. It was found that the ideal gas molecule approach (the same as for the *single* topology) ought to be used to avoid convergence problems [40].

The ΔG values for a transition calculated using different topology mappings will give rise to different free energy estimates, since the actual end states will differ for the *single* and *dual* topologies. The $\Delta\Delta G$ values, however, do not depend on the choice of the topology mapping as long as the same procedure is used across the thermodynamic cycle [40, 41]. In principle, both topology mapping approaches can be combined if required. For example, the topology for a part of a molecule may be designed to follow the *single* topology approach, while the other part may be represented by a *dual* topology (Fig. 1).

2.6 Soft-Core Potential

Transitions exploiting unphysical pathways across a thermodynamic cycle may require particle creation or annihilation. This is always the case for a *dual* topology approach, whereas for a *single* topology particle creation/annihilation is required only when the number of atoms at the end states differs. In a classical molecular mechanics force field description, non-bonded terms described by the Coulomb and Lennard-Jones potentials contain a singularity at inter-particle distances $r=0$. For the end state simulations, reaching a singularity point is prohibited by strong Pauli repulsion term (r^{-12}) of the Lennard-Jones potential. For the unphysical transitions, however, very short inter-atomic distances may be encountered when approaching the end states. Another caveat along the *alchemical* paths comes from simultaneous switching of the electrostatic and van der Waals interactions. When the system approaches an end state, the created/annihilated atoms have weak van der Waals repulsion, whereas electrostatic attraction may be strong enough to bring particles to distances close to zero. To avoid the latter problem, decoupling electrostatic and van der Waals interactions involved in transitions were suggested [42]. In this approach, firstly, the Coulomb interactions are switched off. In a subsequent separately performed transition the van der Waals interactions are modified and finally the electrostatic interactions are restored. The numerical instability problem, however, still remains for the part of the procedure involving the Lennard–Jones interaction modification. To avoid numerical instabilities using the classical non-bonded interaction potentials, the integration time step needs to be decreased with the change of the λ parameter [43].

Another approach is to "soften" the non-bonded interactions along an *alchemical* transition [43, 44]. The "soft-core" potential for the Coulomb and van der Waals interactions can be written as follows:

$$\begin{aligned} V_{ij}(r_{ij}) = & \frac{q_i q_j}{4\pi\varepsilon_0\varepsilon_r(\alpha_Q(1-\lambda)+r_{ij}^p)^{1/p}} \\ & +4\lambda\varepsilon_{ij}\left(\frac{1}{(\alpha_{LJ}(1-\lambda)+(r_{ij}/\sigma_{ij})^s)^{12/s})}-\frac{1}{(\alpha_{LJ}(1-\lambda)+(r_{ij}/\sigma_{ij})^s)^{6/s})}\right) \end{aligned} \tag{22}$$

where p and s are integer parameters, r_{ij} is an interatomic distance, q_i and q_j denote partial charges, ε_{ij} and σ_{ij} represent Lennard–Jones parameters, ε_0 and ε_r are the dielectric constant in vacuum and relative dielectric constant, respectively.

"Softening" of the non-bonded interactions alleviates the problems of singularity points and numerical instability. The "soft-core" potential can be used for both, the Coulomb and van der Waals interactions, hence, the switching of all the non-bonded interactions can be performed simultaneously. While the "soft-core" potential defined in Eq. 22 is routinely used in the equilibrium free energy calculations, its application for a number of non-equilibrium simulation setups revealed potential caveats [45]. As the potential becomes flat for the very short inter-particle distances, overlapping particles exert no repulsive force on each other. This situation may create unwanted additional minima in the potential, where particles are kept in a close proximity throughout a transition, eventually leading to a strong repulsion when reaching an end state. The strong repulsions result in large work dissipation decreasing accuracy of the free energy estimation. A different "soft-core" the approach was proposed to solve the problems of singularity points, numerical instabilities and additional minima [45]. In the latter approach the "softening" of the non-bonded interactions is applied at the force level by modifying the non-bonded interactions such that a finite, but non-zero, force is reached at short inter-particle distances.

3 Topology Generation

Topology generation is the particular aspect of an *alchemical* free energy setup that makes it different from a regular molecular dynamics simulation. The topology must describe both states of a molecule undergoing a transition. Therefore, regardless of which topology approach one chooses, *single* or *dual*, a mapping between the atoms of the two states needs to be established.

At the first step, a mapping algorithm, when provided with the structures of two molecules, should list atoms that need to be morphed into one another or be turned into dummies. To automate such a process several approaches are available. A graph theory based connectivity analysis of the molecules can be used to find a subset of connected atoms for morphing, e.g. a maximum common subgraph algorithm. The atoms not falling within the identified subset would be marked to become dummies in one of the states. The drawback of a graph based approach is the fact that while the atoms mapped for morphing may be close to each other in a graph representation, in Cartesian coordinates the distance between them may be large, resulting in potential convergence issues in the simulations.

A different approach to atom mapping is based on a Euclidean distance criterion. For the two superimposed molecules distances between all the atom pairs need to be calculated. By defining a threshold value (e.g., 0.5 Å) pairs of atoms with distances below the threshold are selected for morphing. The threshold parameter can be adjusted depending on a specific situation. However, one needs to be careful and avoid introducing unreasonable mappings of spatially distant atoms. Creating fragments in a molecule connected via dummies should be avoided, since bonded interactions of the dummies would restrain the degrees of freedom that need not to be restricted. Similarly, breaking ring systems when morphing atoms may lead to stability and convergence issues. Therefore, it is better to follow a *dual* topology approach and create/annihilate intact rings.

Superpositioning of the molecules plays an important role in the distance based topology generation. The atoms to be used for superpositioning can either be defined in advance based on the knowledge of the molecular structures at hand or an unsupervised alignment and superpositioning method can be applied, as, e.g., implemented in Open3DALIGN algorithm [46].

In the second step, a merged topology file should be created. The merged file needs to contain the atoms that are to be morphed during a transition, as well as the dummies of both states. The structure file has to be adjusted accordingly to contain all the atoms described by the topology. In a GROMACS [47] topology file, the atomic details, as well as bonded and non bonded parameters for every interaction can be defined for the two $\lambda=0$ and $\lambda=1$ states separately. In that case, the generation of a merged topology requires assigning the parameters for the two states where a change occurs, and carefully adjusting atom numbering due to the possible introduction of dummies. The Python based package pmx (formerly known as Pymacs [48]) contains a set of tools for the distance based atom mapping and merged topology generation.

In case the merged topologies for a set of molecules are intended to be used frequently, it is convenient to generate a library containing the rules for atom mapping. For example, calculating amino acid mutation effects is often of interest when assessing protein thermostability, protein–protein or protein–ligand binding. The aforementioned pmx package provides a pre-calculated mapping for all amino acid mutations, as well as the required tools for the merged topology and mutated structure generation.

For ligands, however, establishing the mapping between the atoms beforehand is not possible, hence, the topology generation process needs to be carried out from the beginning for every molecule pair of interest. In fact, it is often the case that the 3D structures and topologies of individual ligands need to be generated from a simplified molecule representation (a 2D structure, a SMILES, or SMARTS string). As a starting point here, an empirical 3D structure generator can be employed, e.g., OpenBabel [49], ChemAxon's Marvin package tool MolConverter, CORINA [50]. A decision on the protonation state of a ligand must be made at this step as well. If the protonation state is expected to play an important role for the binding of a particular ligand, the different protonation states can be treated as separate molecules and the topologies for each of them could be generated. Subsequently, the calculated free energy differences can be used to estimate the relative pKa shifts for the titratable sites. If the molecular 3D structure was generated from scratch, it is suggested to optimize the geometry with a quantum chemical package of choice. Several methods for the atomic partial charge assignment are available: one such method requires the calculation of an electrostatic potential (ESP) followed by a procedure of fitting the ESP surface onto atoms [51, 52].

Alternatively, the computationally less demanding semiempirical AM1-BCC [53] partial charges have been shown to be of comparable quality in the solvation free energy estimation [54]. Generation of the bonded parameters for a molecule depends on the force field of choice. Nowadays the main biomolecular force fields have been generalized to include small organic compounds. In addition, automated atom typing and bonded parameter assignment procedures are readily available: the Antechamber [55] module allows topology generation for the Generalized Amber Force Field (GAFF) [56]; CHARMM General Force Field (CGenFF) [57] topologies can be created with the ParamChem [58, 59] utility; GROMOS force field compatible topologies can be generated with an Automated Topology Builder (ATB) [60]; ligand topologies for the OPLS force field can be generated using MKTOP [61] software.

4 Thermodynamic Cycles

As previously mentioned, the absolute free energy of a system is difficult to determine, but fortunately most problems can be formulated in terms of relative free energies. Free energy differences are both more approachable and contain important information about the system. The change in free energy due to binding ($\Delta G_{\mathrm{binding}}$) can be determined experimentally (e.g., by means of calorimetry). Although absolute binding affinities can be calculated using, for example, umbrella sampling, such calculations are usually cumbersome, as the whole binding/unbinding process needs to be taken into account, while its path is unknown in most cases. Therefore, investigating the effect of a change of the system (e.g., an amino acid mutation or a ligand modification) on binding is usually more feasible. For this, the double free energy difference ($\Delta\Delta G_{\mathrm{mutation,binding}}$) is calculated.

The *alchemical* methods calculate the work needed to “move” a system from one state to another through unphysical pathways. As the free energy of a system is a function of its state, the free energy difference found between the states is independent of the path taken between them. Sufficient sampling is of critical importance in all free energy methods, and the computational difficulty of reaching convergence dramatically increases with the magnitude of the perturbation.

Binding free energies usually involve a large perturbation—in one state, both binding partners are free in solution, in the other they are in a complex. The phase space overlap between these two states is often small resulting in a slow convergence. An *alchemical* transition from state A (a wild-type protein or a ligand) to the state B (a mutated or modified molecule), while physically impossible, requires a much smaller perturbation. As we are interested in the difference in the binding free energies between the states A and B,

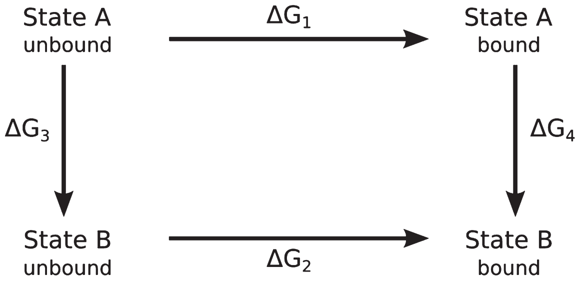


Fig. 2 Schematic representation of a thermodynamic cycle. To determine the change in binding affinity upon a protein mutation or ligand modification, the difference between the binding free energy of the ligand or protein in state *A* (ΔG_1) and that of the state *B* (ΔG_2) need to be determined. However, the values (ΔG_3) and (ΔG_4) are more accessible to the free energy perturbation methods. Using the conservation of energy in this closed cycle, we can derive $\Delta\Delta G = \Delta G_2 - \Delta G_1 = \Delta G_4 - \Delta G_3$

we can make use of the fact that the free energy of a state does not depend on the path taken to reach it. Hence, the free energy differences along a closed cycle of reactions (like the one depicted in Fig. 2) will always add up to zero. This feature allows calculation of the double differences in free energy ($\Delta\Delta G$) of binding, thermostability, partitioning in different solvents, etc. between two states of a system

$$\Delta\Delta G = \Delta G_2 - \Delta G_1 = \Delta G_4 - \Delta G_3 \tag{23}$$

The resulting $\Delta\Delta G$ is often more accurate than the ΔG values used to calculate it, as systematic errors (like those caused by the presence of dummy particles and limited sampling) cancel out when the free energy differences are subtracted. Note that this way, relative binding affinities are obtained without studying the actual binding/unbinding event.

The two branches of a thermodynamic cycle can be combined into a single transition by performing both transitions in the same box. This is important to bear in mind, when an *alchemical* modification involves a charge change between the states. In such a case, the simulation box in one of the end states carries a non-zero charge, hence, treatment of the electrostatic interactions by means of the Ewald summation introduces artifacts. Meanwhile methods that would not introduce such artifacts, e.g. a simple cut-off based Coulomb interaction calculation, are generally less accurate. Hence, the best solution is to design a specific simulation setup to keep the system neutral at all times during a transition. One such simulation setup is to combine both branches of a thermodynamic cycle in one simulation box (Fig. 3). In such a *double-system/single-box* setup,

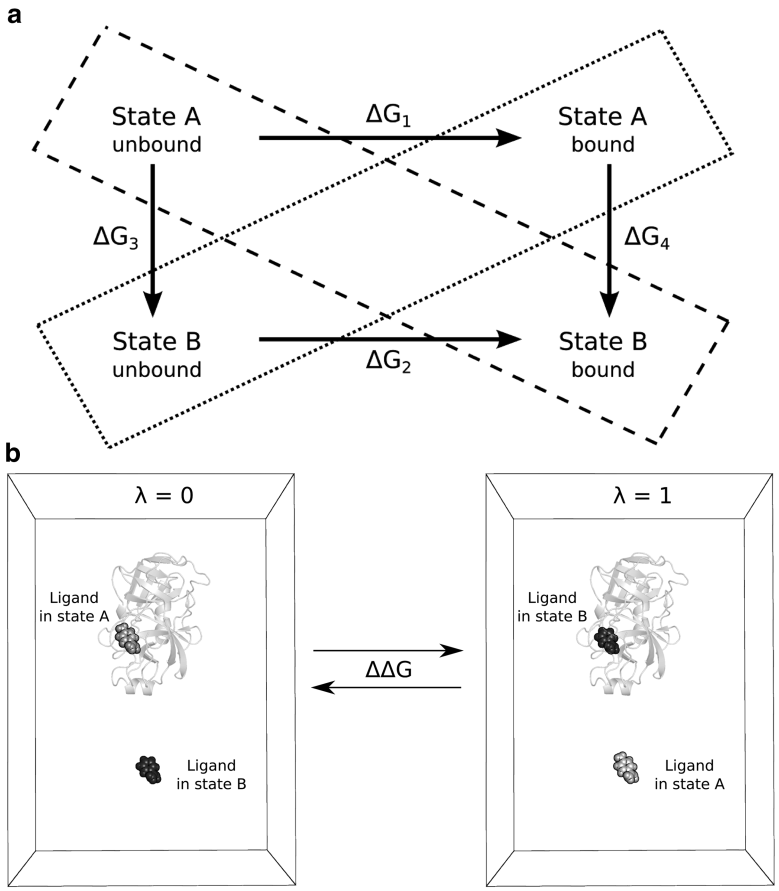


Fig. 3 *Double-system/single-box* setup. (**a**) Two branches of a thermodynamic cycle are placed in one simulation box. The different boxes in the scheme are indicated by the *broken* and *dotted lines*. (**b**) An example of a simulation setup for the ligand binding free energy calculation. During a transition the enzyme bound ligand is being morphed from the state *A* to *B*, whereas the solvated molecule undergoes a reverse transition from *B* to *A* state. The net charge of the simulation box remains zero. The free energy estimate corresponds to a double free energy difference: $\Delta\Delta G = \Delta G_4 - \Delta G_3$

a state *A* molecule bound to an enzyme is positioned in the same box with the solvated state *B* molecule. The other end state contains the same compounds, but in inverted roles: state *B* molecule is in contact with an enzyme, while molecule *A* is solvated. In this way, molecules *A* and *B* are present in the simulation throughout a transition and no net charge change occurs.

It is important to position the solvated (free) and protein bound molecules at a sufficiently large distance from each other to prevent direct interaction, because the branches of the thermodynamic cycle, that are now in a single box, are assumed to be independent. In practice, placing the structures ~ 3 nm apart is sufficient to obtain accurate free energy estimates. To prevent an interaction between the solvated molecule and the protein due to motions during the simulation, a position restraint on a single atom of the molecule free in solution can be imposed. Once a system is set up this way, the estimated free energy corresponds to the $\Delta\Delta G$ of binding.

Even if a charge changing mutation/modification is set up to remain in a neutral simulation box during an alchemical transition, some unwanted electrostatic artifacts may persist due to the finite size and periodicity effects [62].

5 Validation Using Closed Thermodynamic Cycles

To achieve accurate and valuable free energy estimates from simulations, it is important to aim at the best possible setup within the approximations made. If and when available, known experimental results can be used to validate a particular setup but even then, there can be multiple reasons for a discrepancy between experiments and simulations. For instance, it is often difficult to completely match experimental conditions with simulation conditions, e.g. trace amounts of a certain ion in the buffer could have a large effect, but are often neglected in simulations. Also, it can never be excluded that there is an error in the experimental results.

From the simulation side the major factors affecting the accuracy of the results are conformational sampling and the Hamiltonian/force field. One can validate the setup independent of external factors such as the force field or known experimental results, by utilizing closed thermodynamic cycles where all transition branches are accessible and are explicitly computed. The result of summing up all branches should then be zero. Any deviation from zero gives access to the accuracy of the approach used. The simplest type of a closed cycle is using the *double-system/single-box* setup (*see* Subheading 4), where in one box A2B and B2A are present. A2B can be the transition of one amino acid to another in the case of protein binding, or can be the change from one ligand to another in the case of ligand binding.

The *double system in a single box* approach is used in the following example to obtain a closed cycle. A transition from leucine to alanine (L2A) is combined with the reverse transition from alanine to leucine (A2L) in a single system. Ten independent equilibrium simulations of 10 ns are launched and convergence is verified as a function of the number of independent trajectories involved in computing the free energy. Using more independent simulations reduces the error bar. Note that using only one 10 ns trajectory can produce a result that is close to zero, but the error bar is large. In this case, an additional 10 ns of simulation can increase the

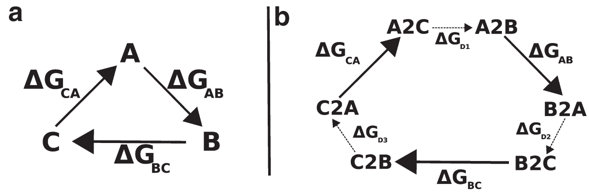


Fig. 4 (**a**) Closed thermodynamic cycle involving three states. (**b**) Including the effects of the dummy atoms a closed thermodynamic cycle with three states is transformed to one with six states

deviation from zero, but thereby indicates that 10 ns of sampling is not sufficient for the simulation to converge. Combining all ten simulations results in a small error bar (with zero within).

Thus, using short simulation times may often result in seemingly good results for closed thermodynamic cycles, since in that case all simulations stay close to the starting structure and the conformational space is insufficiently explored. Assuming that sufficient exploration of the conformational space is required, using longer simulations would be preferred. However, using longer simulations also holds the risk that they will get trapped in artificial minima caused by force field artifacts. To avoid this we recommend using multiple short simulations as done in this example. This approach ensures better sampling and reduces the risk of getting trapped in artificial minima. Furthermore, a more rigorous and straightforward error estimation could be performed using this approach, provided that a sufficient number of transitions is achieved for each independent simulation. In such a case the free energy can be calculated for each trajectory separately and the error can be evaluated for the deviation among the independent ΔG estimates.

In the example above we used a convenient *double-system/single-box* setup. However, in case of a closed cycle is constructed by considering the branches of a thermodynamic cycle separately, one additional caveat in terms of topology construction needs to be taken into account. Both *single* and *dual* topology procedures mostly involve dummy atoms, and those need to be considered in a closed cycle. A simple thermodynamic cycle, as represented in Fig. 4a, containing three vertices, might end up in a cycle with six vertices (Fig. 4b), and edges containing only dummy transitions that are not easily accessible [40, 41]. In a typical free energy simulation one is often interested in the difference between two ΔG values, where the contribution of the dummy atoms cancels out, e.g. in protein thermostability calculations the effect of the dummy atoms is the same in the reference (unfolded) state as in the folded state, or for ligand binding affinities the effect is the same for the ligand in solvent as the ligand bound to a protein. Therefore, one possibility for the construction of a valid closed thermodynamic

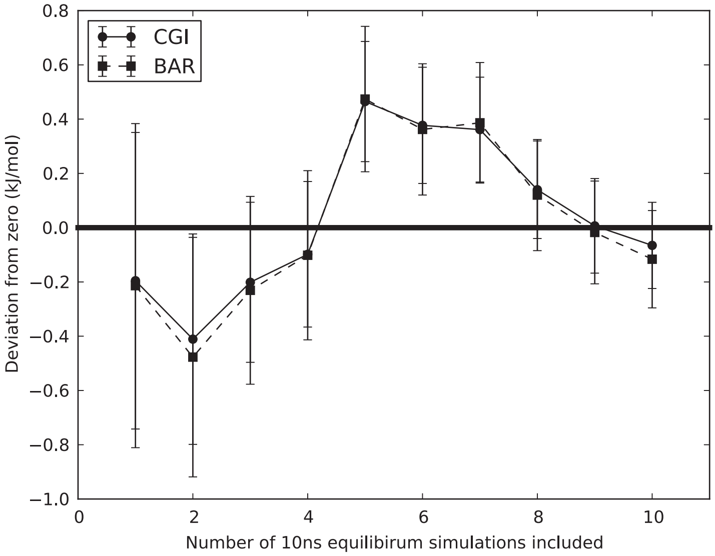


Fig. 5 Results for a double system in a single box, having both L2A and A2L structures in one box. The number of equilibrium simulations of 10 ns used can be read from the *x*-axis. Each point on the graph consists of 100 non-equilibrium trajectories of 50 ps

cycle with three vertices, might be by having $\Delta\Delta G$ values at the edges instead of ΔG. As a simpler alternative, the *double-system/single-box* setup as shown in the example above could be used (Fig. 5).

6 Non-equilibrium Free Energy Calculation Setup

In this section we will outline the main steps of the *alchemical* non-equilibrium free energy calculation setup. The described protocol is compatible with the ΔG estimation based on both the Jarzynski equality and CFT. In the first step of the procedure, equilibrium ensembles of the system in both end states need to be generated. Afterwards, short transition simulations are performed starting from the structures selected from the equilibrium ensembles. The work performed by the system is calculated by numerically integrating the $\partial H/\partial\lambda$ curves for every transition. Finally, the free energy difference between the two end states is estimated from the accumulated work values.

To create equilibrium ensembles for both end states of a transition, simulations of the system with the hybrid topology (Subheading 3) can be performed by keeping the transition controlling variable λ constant at one of the two end states. This has an advantage that structures from the equilibrium simulations can directly be used as starting structures for the transitions. On the other hand, if equilibrium ensembles generated without a hybrid topology are already available, they can be re-used, e.g. when the same residue of a protein needs to be mutated to more than one target, or if an alternative method is used to generate the equilibrium ensemble which is not compatible with the use of hybrid topologies. In these cases, the hybrid topology can be introduced into structures after generating the equilibrium ensembles with a regular (non-hybrid) topology, but it is advisable to perform a short simulation with the hybrid topology prior to a transition, to allow the system to relax.

For the non-equilibrium transitions, a number of structures (from 100 as in published works [35, 48] up to 450 as in the ATP example described later in this chapter) are picked from both equilibrium ensembles. For each of these structures, a short (usually on the order of 50–200 ps) simulation is performed, during which the topology is changed from one state to the other. For more details on the simulation parameters *see* **Note 1**.

The generalized force $\partial H/\partial\lambda$ associated with this transition is calculated and written out by the simulation program.

The work value for each transition can be obtained by numerical integration of the $\partial H/\partial\lambda$ curves. Then, two histograms are created from all work values associated with transitions in a single direction (Fig. 6). The CGI, BAR, or another method (Subheading 2.4) can be employed to extract the free energy difference from the work distributions.

The computational time necessary to calculate free energy differences using non-equilibrium approach depends on several factors, including first of all the size of the system and the significance of the change, but also the desired precision. It might be preferable to get fast approximate results in a screening process, while more computational effort would be spent to obtain a precise value for a specific mutation. The quality of the simulation protocol can be validated using closed thermodynamic cycles (Subheading 5), but a closer look at the transition work distributions may also indicate how the results can be improved.

As the CFT is based on the assumption that the transition runs start from an equilibrium ensemble, improving the sampling of these ensembles usually yields the greatest improvement to the quality of the result. This is best achieved by running several parallel simulations than a single long one as molecular dynamics simulations tend to “get stuck” in local energy minima.

The difference in work values between the forward and backward transitions is caused by the fact that the simulations are performed in non-equilibrium conditions. The work distributions for forward and backward transitions will be closer to each other the slower the transitions are. Hence, increasing the length of these simulations would improve the result if the work distributions are not properly overlapping.

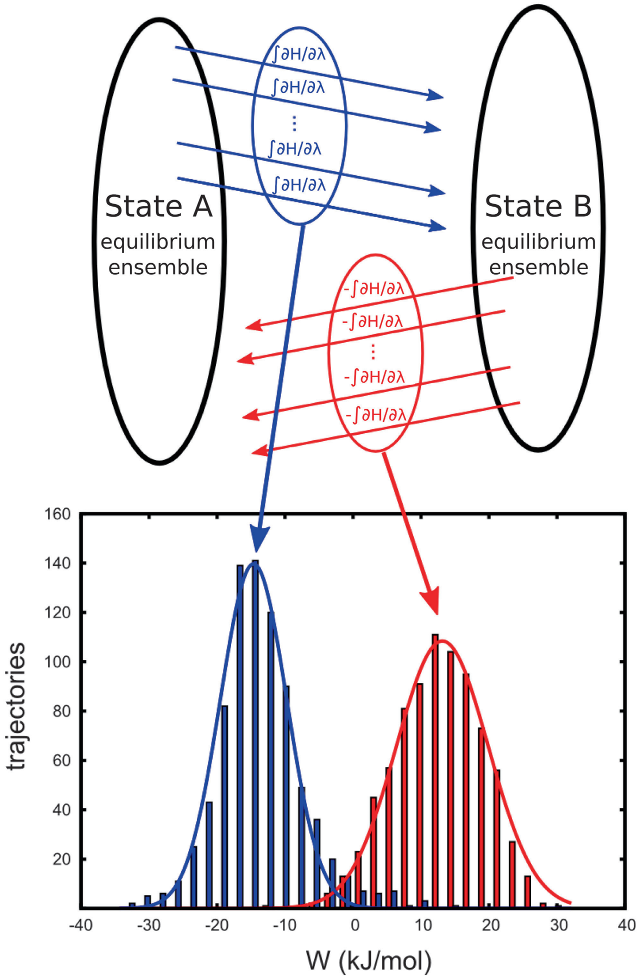


Fig. 6 *Non-equilibrium free energy calculation setup.* Two equilibrium ensembles are generated (e.g. by MD simulations) for the system in state *A* and *B*. From these ensembles, snapshots are selected and used to set up short non-equilibrium transition simulations, in which the state *A* is turned into *B* and vice versa

7 Trypsin Inhibitors

Binding affinity estimation for small organic compounds is of high importance in the search for potential drug candidates. Hence, we will analyze in more detail a study of *alchemical* $\Delta\Delta G$ calculations for a set of trypsin inhibitors. Talhout et al. [63] performed isothermal calorimetry (ITC) measurements of the binding free energies for a number of *p*-*n*-alkylbenzamidinium molecules. We will use this set of ligands to illustrate the workflow of the *alchemical* ligand binding free energy calculations, and the ITC measurements will serve us as a reference to assess the quality of our estimates. More information on the computational studies on this set of trypsin inhibitors can be found in the publications [45, 63].

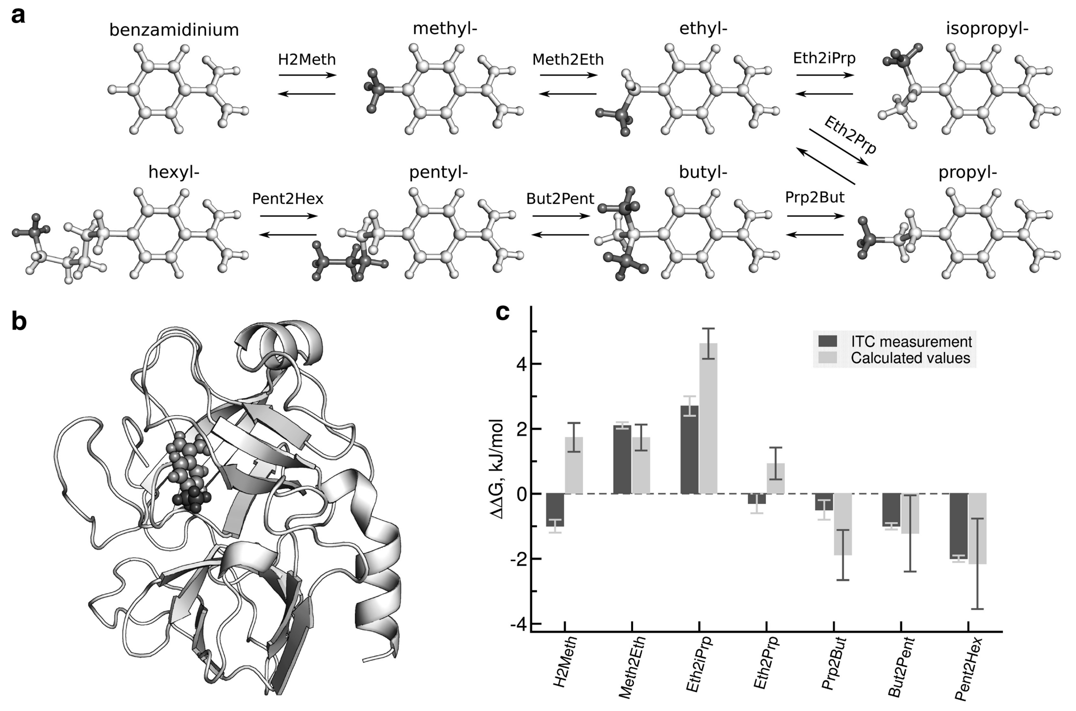


Fig. 7 Trypsin inhibitor analysis. (**a**) A set of alkylbenzamidinium molecules. Molecule ordering corresponds to the pairs of ligands for which the free energy differences were calculated. (**b**) Structure 3PTB with a co-crystallized benzamidine served as a starting structure for the MD simulations. (**c**) $\Delta\Delta G$ values from the ITC measurements [63] and calculated estimates

7.1 Topology and Starting Structure

Before starting the ligand topology generation, the molecules were superimposed on the atoms comprising a common scaffold. In the current example, carbons of a rigid benzene ring were well suited for the superposition. Subsequently, the order for the *alchemical* morphs was established: one of the possibilities is shown in Fig. 7a. Growing an alkyl chain on a benzamidinium several carbons at a time ensured a sufficient phase space overlap between the end state ligands. The ligand topologies were generated using the GAFF methodology [56]. A *single* topology approach was used to establish a mapping between the molecules. Atoms that ought to be transformed into one another were identified using a distance criterion after structural superpositioning.

The starting structure for the simulations (Fig. 7b) was obtained from a crystal structure in the Protein Data Bank (id 3PTB [64]). Since the crystal structure contained a co-crystallized benzamidine molecule, all the ligands of interest were superimposed onto the experimentally determined structure. Such a construction of a starting structure rests on an assumption that the analyzed compounds share a similar binding pose with the benzamidine moiety. In case a co-crystallized ligand is not available or the binding pose is not well defined, additional analysis is needed. For example, a molecular docking combined with a long time scale molecular dynamics simulation can be used to establish a starting structure with a reliable ligand pose for the *alchemical* free energy calculations.

7.2 Simulation Setup

The thermodynamic cycle for the binding $\Delta\Delta G$ estimation was constructed by considering the transitions between the two ligand states separately: in water and in contact with trypsin. Such a simple cycle construction was possible, since no charge change was involved in any of the transitions.

Firstly, 10 ns equilibrium molecular dynamics simulations were performed at the end states for the ligands in water and in the bound form. While this time scale does not cover large conformational motions or significant changes in the binding pose, it provides an equilibrium sampling in a free energy minimum close to the starting structure. For the cases, where in a simulation time the system travels far from its initial point, longer sampling times may be required to cover the relevant phase space volume. The non-equilibrium transitions were spawned from the snapshots extracted from the equilibrium trajectories. Here, 100 simulations (50 ps each) were performed going in both directions: A to B and B to A. *See* also **Notes 2** and **3** for more technical details on the equilibrium sampling and non-equilibrium transitions, respectively.

7.3 Estimation of the Free Energy Differences

Integration over the $\partial H/\partial\lambda$ curves to obtain the work values and subsequent application of the CGI method enabled calculation of the ΔG values for the parts of the thermodynamic cycle: $\Delta G_{\text{water}}^{A\rightarrow B}$ and $\Delta G_{\text{trypsin}}^{A\rightarrow B}$ (*see* **Note 4**). The final $\Delta\Delta G$ estimates are shown in Fig. 7c. A positive value in the graph indicates that the ligand in state B is a weaker binder in comparison with the state A molecule.

For all the ligand pairs the difference between the calculated and experimentally measured free energy values was smaller than 1 kcal/mol. In most cases the $\Delta\Delta G$ values were estimated very accurately, and only the transition from benzamidine to methylbenzamidine showed a discrepancy of ~ 1 kT. Larger errors for the estimated free energy differences can be observed for the longer alkyl chains. This effect is a nice illustration of the increased uncertainty in the calculated results due to the larger dissipated work values. *Alchemically* grown longer chains face stronger hindrance and steric clashes, leading to an increase in the amount of work dissipated along the path. The errors could be reduced by performing the transitions slower (e.g., 100 ps for a transition).

In the current example the results were presented as the double differences in free energy. In practice, however, an estimate of the ΔG values may be of interest. Conversion from the $\Delta\Delta G$ to the single free energy differences can be attained by setting one of the ΔG estimates to an experimental value (e.g., benzamidinium ITC

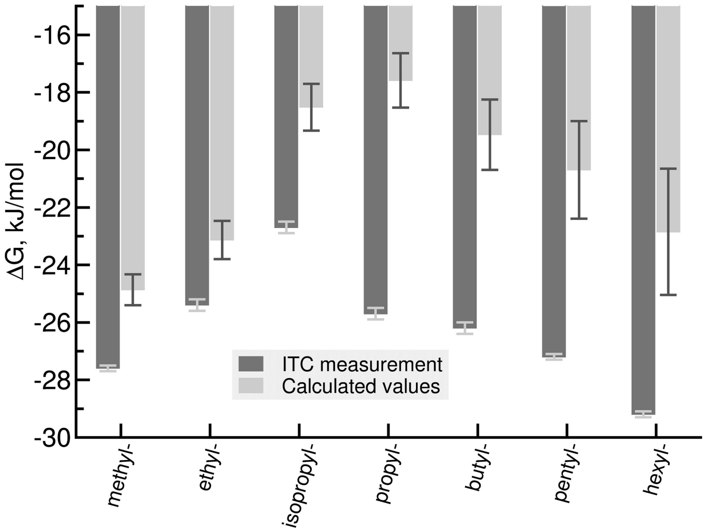


Fig. 8 ΔG values calculated by propagating the double free energy differences ($\Delta\Delta G$) along a chain of ligands (*see* Fig. 7a) The results are less accurate than the $\Delta\Delta G$ values due to the error accumulation

measurement) and propagating the differences together with the associated errors along the chain of compounds (Fig. 8). It is clear that the estimates appear far more inaccurate due to the accumulation of errors. This result illustrates the importance of an appropriate simulation setup addressing a question of interest. For the case where ΔG estimates are to be compared to the experimental measurements, a single reference structure could be used to create the pairs for transitions. Although the perturbations in such a scenario would be larger than when considering a chain of ligands, the problem of error accumulation could be avoided. In addition, such a setup would allow for internal consistency checks via closed thermodynamic cycles.

8 Protein–Protein Binding: Binding of α-Chymotrypsin with Its Inhibitor Turkey Ovomucoid Third Domain

In this example the change in binding free energy upon mutation in the complex of the Kazal-type Ovomucoid from Turkey (OMTKY3) and α-chymotrypsin (CHT) is computed. Leucine 18 of OMTKY3 is mutated to A,I,W,F,Y,V,M,S, and T residues. The results are compared to both experimental data [65] and other theoretical predictions [66]. Note that this example is well suited as a practical hands-on test for the principles learned in this chapter. The structure is available in the Protein Data Bank (id 1CHO

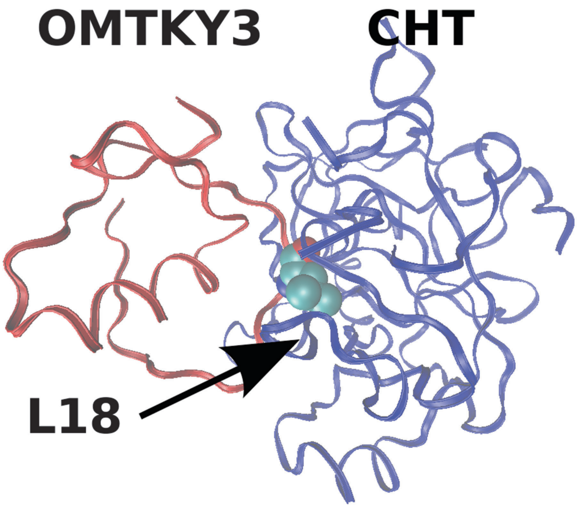


Fig. 9 The complex of Kazal-type Ovomucoid from Turkey (OMTKY3) and α-chymotrypsin (CHT). Leucine at position 18 is mutated. The structure is taken from the pdb-database (PDB id 1CHO)

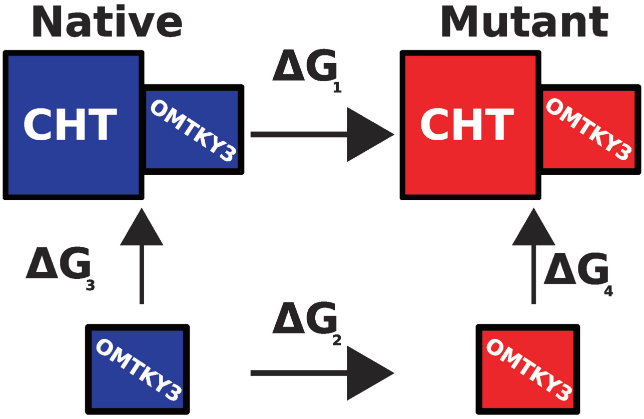


Fig. 10 Thermodynamic cycle used for protein–protein binding

[67]), and calculations can be set up using freely available software (`pmx` [48] and Gromacs [47]).

The CHT:OMTKY3 complex is shown in Fig. 9, the position (L18) that is mutated is at the interface of the two binding partners. The thermodynamic cycle in Fig. 10 was used to compute the binding free energy. In this cycle the reference is OMTKY3. The shift in binding free energy upon mutation ($\Delta\Delta G$) is computed from the difference between ΔG_1 and ΔG_2, since computing the difference between ΔG_3 and ΔG_4 directly would be computationally too expensive.

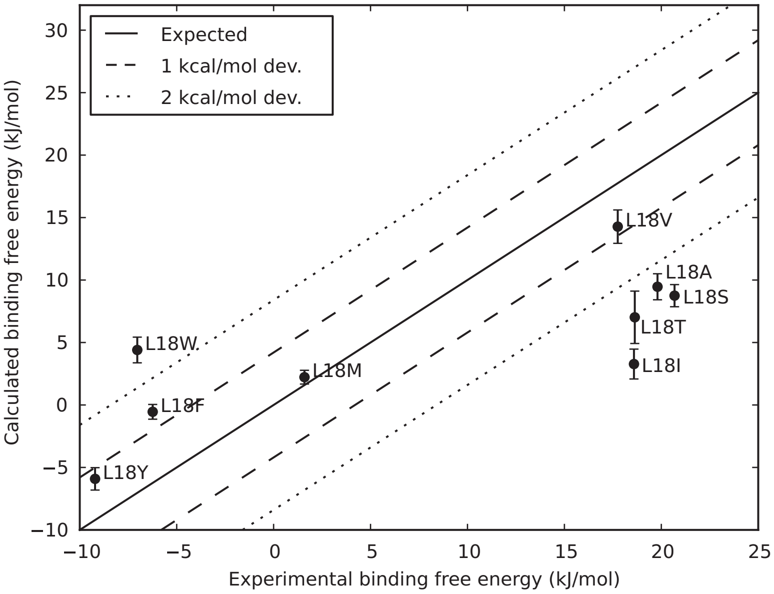


Fig. 11 Results of mutations in OMTKY3:CHT complex. The average unsigned error is 7.9 kJ/mol and correlation coefficient is 0.83

Setting up these simulations consists of four phases. First the topology is prepared, next the equilibrium simulations are performed, followed by non-equilibrium simulations and finally the results are analyzed to extract the free energy. This procedure is performed for both the complex and for free OMTKY3 in solvent. The topology is initially prepared using the Gromacs [47] `pdb2gmx` tool and the Amber99sb [68] force field. Equilibrium simulations of 20 ns are performed for the native protein and all mutants. From those equilibrium simulations, 100 snapshots are selected and hybrid structures and topologies are created using `pmx` [48] (*see* **Note 5**). Those are followed by non-equilibrium transitions of 100 ps each. The free energies are extracted using the BAR method [14] (*see* **Note 4**). BAR was used in this case, since the CGI method requires a Gaussian distribution of the work values, which is not always the case for all mutations in this complex. The results of the calculations are shown in Fig. 11. The correlation coefficient with the experiment is 0.83 and the average unsigned error is 7.9 kJ/mol. Particularly, the mutants with the largest destabilization are predicted poorly. The same mutations were predicted by Benedix et al. [66], where a slightly better result was obtained, with a correlation coefficient of 0.9 and an average unsigned error of 7.9 kJ/mol.

When comparing to experimental results, their origin has to be taken into account. The experimental results [65] used here are obtained by enzyme assays, where the binding affinity is not measured directly, but instead the inhibition of proteolysis is measured. The inhibition of proteolysis might be related to the binding affinity of OMTKY3, but other factors might play a role, e.g., a certain mutation could trigger a conformational change in CHT which inhibits proteolysis, without having a stronger binding affinity than another mutant.

9 ATP and Magnesium Complex in Solution

In the example given here, we show how seemingly simple free energy calculations involving a highly charged ligand (ATP and a bound magnesium) can result in substantial inaccuracies if not treated correctly. Many biological functions require the presence of an ATP molecule which is often only active with a bound magnesium ion. The binding free energy of an ATP · Mg complex can be calculated via a thermodynamic cycle where the complex is annihilated once from the binding pocket of the protein and once in an unbound (free) state which is represented by an annihilation of the ATP · Mg in water. The second transformation is in fact the solvation free energy. Here, we present some of the pitfalls that might occur merely in the calculation of the solvation free energy of ATP · Mg, as it presents a challenging system by itself and may be found in much more complex systems as well (e.g., in protein complexes).

9.1 System Setup: Charged Systems

The annihilation of an ATP^{-4} molecule, even when attached to a Mg^{+2} ion, incorporates a change in the overall system charge which would influence the resulting free energy. One possibility to neutralize the system is to add 2 counter Na^{+} ions that would turn into dummies during the transition as well. However, this effectively turns the free energy of ATP · Mg solvation, into a combined solvation free energy of [ATP · Mg] + 2Na. Moreover, an issue encountered in equilibrium free energy calculations of this system is that the interaction of the disappearing Na^{+} ions with the disappearing $ATP \cdot Mg^{+2}$ is not sufficiently sampled in the simulated time scales.

A second and more computationally expensive possibility that avoids the coupling of the counter ions to the transition is the *double-system/single-box* setup (Subheading 4). For this example, one ATP · Mg complex is annihilated and another is solvated in the same box, while the structures are positioned ~ 4 nm apart in order to avoid any interaction between them. The Mg^{+2} ion itself cannot migrate between the ATP molecules when its interactions are turned off, because it is kept close to its respective ATP via a distance restraint. Topology generation is relatively simple, where *A*

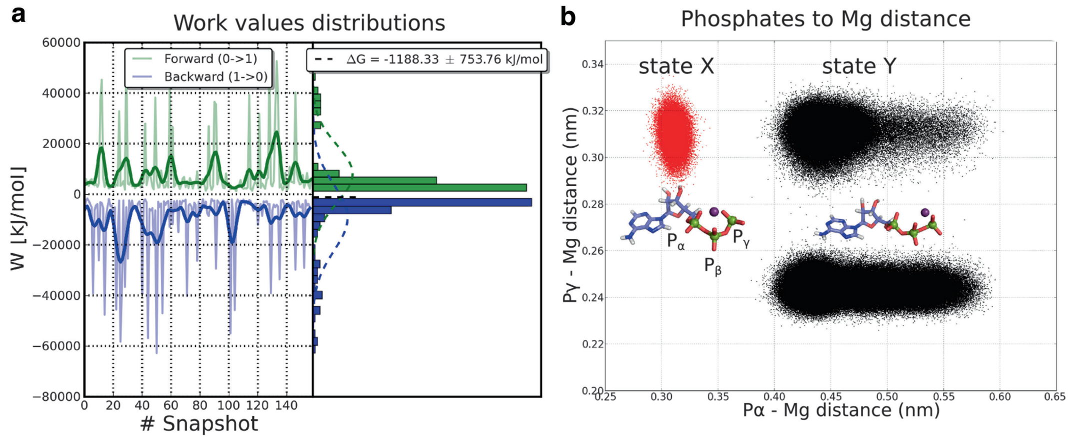


Fig. 12 (**a**) Non-equilibrium free energy calculation of ATP solvation and desolvation which should serve as a closed cycle. The left section of the plot shows the work values as a function of the transition number. Since the transitions are started from consecutive frames it could indicate whether the equilibrium trajectory is drifting. The right region of the plot combines the work values into histograms. (**b**) Distribution of ATP and Mg orientations measured according to distance from P_α and P_γ. P_β is not shown since its distance does not vary and fluctuates around 0.3 nm

and *B* states are constructed such that one molecule is present at $\lambda = 0$ while the second is annihilated (turned to dummies), and vice versa for $\lambda = 1$. This type of setup allows either for directly computing the binding free energy in one simulation box (if the ligand is annihilated while bound to a protein on one side of a box, and solvated on the other side), or to compute a closed cycle as a consistency check. In this example we performed the latter.

9.2 Calculating a Closed Cycle

Having constructed the system as described above, 50 ns equilibrium ensembles were obtained for $\lambda = 0$, $\lambda = 1$ and, afterwards, fast non-equilibrium transitions were performed 200 ps each. The results are shown in Fig. 12a, indicating that the distributions of the resulting work values do not overlap nor fit into a gaussian function, whereas the estimated ΔG is far from zero.

How is it possible that a transition and its opposite do not result in a zero free energy change? Sampling and symmetry are key issues here. The two ATP · Mg complexes were found not to be entirely symmetric in the conformations covered by their equilibrium ensembles. Specifically, it is the Mg^{+2} ion that adopts different orientations relative to the phosphates of the ATP. In Fig. 12b one can differentiate between two states, which are mutually exclusive for the two ATP molecules in the box: state *X*, where the Mg^{+2} is triply coordinated by all three phosphates that adopt a similar distance of 0.3 nm to it, and state *Υ*, where Mg^{+2} is doubly coordinated and found slightly further away from the P_α (~0.45 to 0.55 nm). The lower cluster in state *Υ*, where the distance to P_γ

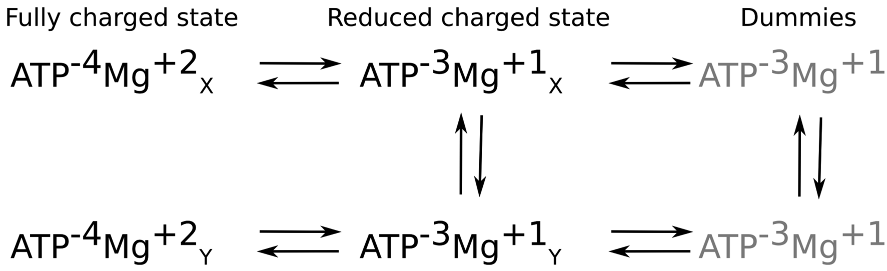


Fig. 13 An extended closed cycle of an ATP · Mg complex annihilated and solvated in solution. The *horizontal arrows* are performed in both directions simultaneously for a closed cycle consistency check

shortens, arises from the initial structure, but it gradually equilibrates into the top cluster of state Υ within 1–20 ns and never re-visits the former cluster. Since these conformations do not interchange, they can be seen as two distinct chemical species. Thus, turning one chemical species on, and another off does not represent two opposite reactions and will not converge to $\Delta G = 0$.

Spontaneous transitions between these conformations do not occur in MD simulations with a length of 100 ns, and they are kept separated by a steep barrier. However, the barrier is solely maintained by the highly attractive Coulomb interactions between the phosphates and the magnesium. Since the free energy is a state function, the path from the fully solvated to annihilated (uninteracting dummies) state could be chosen to pass through a reduced charge state, where the charge on the magnesium ion is scaled to +1, while the charge on the phosphate groups is scaled in reverse to maintain the neutrality of the system. This effectively creates two transitions that need to be calculated: one from a fully appeared and fully charged state to a reduced charge state, and then another one into dummies.

To enable the convergence of the closed cycle of annihilating and solvating an ATP · Mg complex in solution, the calculation is decomposed further into the two ATP orientations, such that transitions (annihilation or solvation) will be performed for each orientation (X and Υ) of the ATP[1] as depicted by the horizontal arrows in Fig. 13. There is no need to compute the vertical transitions, but we would like to note that both vertical transitions maintain a $\Delta G \sim 0$. In the reduced charge state, conformations X and Υ interchange multiple times, while they are equally populated. As for the dummy state, the difference between X and Υ is merely a difference of orientation in space, which is imposed by a distance restraint.

The distribution of work values for each of the four horizontal transitions shown in Fig. 13 is plotted in Fig. 14. These transitions are as before, performed in both directions to close a thermodynamic cycle. Four hundred and fifty transitions were performed in

[1] Distance restraints on the $P\alpha$, $P\gamma$ and the Mg^{+2} will keep the atoms in their respective orientation in one of the two top clusters shown in Fig. 12b.

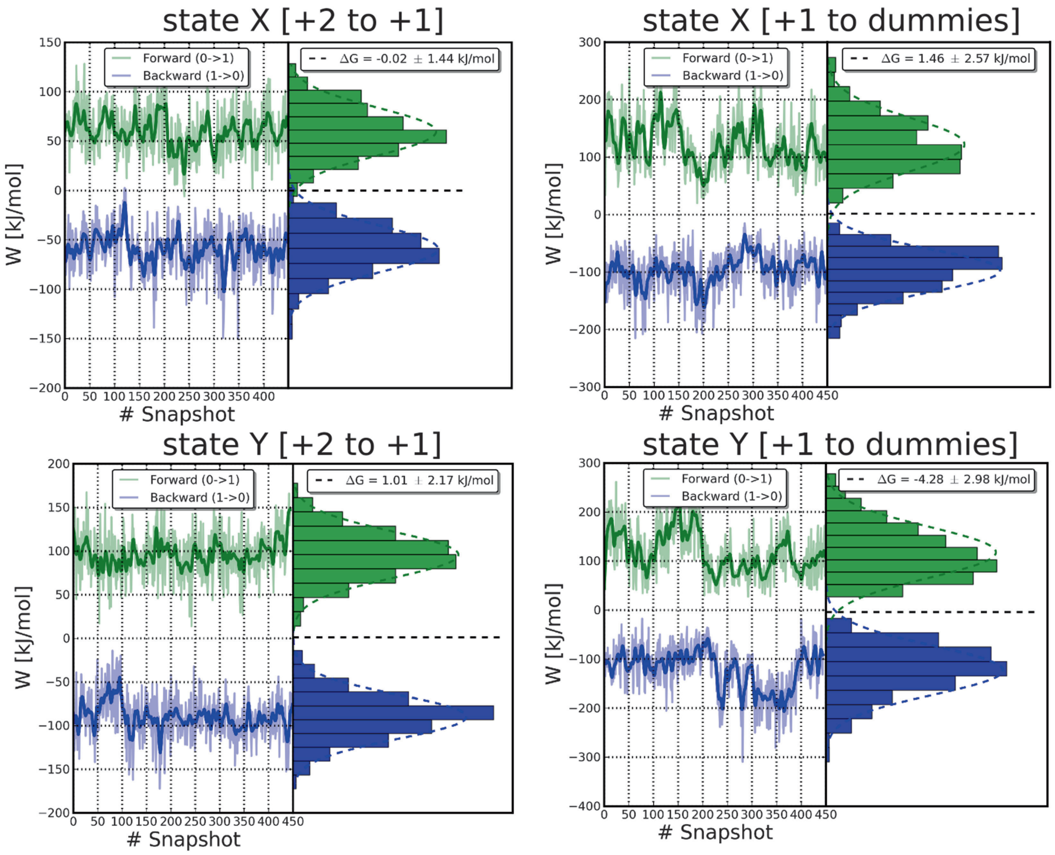


Fig. 14 Results from non-equilibrium transitions as depicted by the *horizontal arrows* in Fig. 13. The snapshot number represents the transitions

each direction, their length was increased to 2 ns. Indeed, this time the resulting ΔG values are mostly within 1.5 kJ/mol away from zero[2], except for the last transition (reduced charge state into dummies in state Υ). This state is much more flexible than its fully charged version, and might need more time to converge, i.e. via longer equilibrium simulations, however, electrostatic artifacts could remain due to the nature of the mutation (turning off a charge) and the finite size of the system. In principle, determining whether the generated equilibrium ensemble is indeed at equilibrium is difficult. However, sometimes examining the series of work values taken from consecutive initial snapshots from the equilibrium ensemble may indicate whether there is a drift and whether more equilibration time is needed. For example, if we would only have a quarter of the equilibrium trajectory of state $X_{+2 \text{ to } +1}$ and the corresponding transitions (first 120 transitions in Fig. 14), the drift in work values might have hinted at a trajectory drift, but further equilibration and transitions from later snapshots show that those work values are reproduced again and again.

[2] A shorter transition time of 200 ps also gave a result that was fairly close to zero, but longer times were used to reduce the error.

Table 1
Parameters and suggested values for the non-equilibrium free energy calculations

Parameter	Value
`init-lambda`	0 or 1
`delta-lambda` (equilibration)	0
`delta-lambda` (transition)	±1/`nsteps`
`nstdhdl`	1
`sc-coul`	yes
`sc-alpha`	0.3
`sc-sigma`	0.25
`sc-power`	1

Having arrived at a consistent zero free energy change for a closed cycle of ATP in solution, one can be more confident in the accuracy of ATP · Mg annihilation in protein complexes. For practical purposes, a similar conformation decomposition can be done when computing the binding free energy of an ATP · Mg complex in a protein. As indicated earlier, in such a case, the box will contain ATP · Mg and a protein with ATP · Mg in its binding pocket which will be simultaneously turned on and off. This time, rather than receiving a zero from each transition (and an overall zero from ATP · Mg annihilation and resolvation in both orientations), the combined transition from a fully charged state to a dummy state would yield the binding free energy of ATP · Mg_X and ATP · Mg_Y. It can then be combined using the following formula [69]:

$$\Delta G = -\beta^{-1} \ln[\exp(-\beta\Delta G_X) + \exp(-\beta\Delta G_Y)] \qquad (24)$$

10 Notes

1. In the Gromacs simulation package the free energy code is activated by setting the flag `free-energy=yes` in a molecular dynamics parameter (mdp) file. Triggering this option automatically enables the $\partial H / \partial \lambda$ output to an external file. The initial state of a system is set by defining `init-lambda` to be equal to 0 or 1 for the states *A* and *B*, respectively. Setting the two aforementioned parameters is sufficient to perform an equilibrium sampling simulation at one of the end states. For the transition runs, an increment in λ needs to be specified by setting the parameter `delta-lambda` to a non-zero value. `delta-lambda` has to be estimated such that an end state is

reached within the defined number of integration steps. For example, for a transition from `init-lambda=1` to the end state $\lambda=0$ in 50 ps with an integration time step of 2 fs, 25,000 integration steps are required. Hence, `delta-lambda` has to be set to $-4e^{-5}$. For the non-equilibrium free energy calculation it is important to collect all the available $\partial H/\partial \lambda$ values, therefore, the frequency of output should not be reduced, i.e. `nstdhdl=1`. We recommend using the soft-core potential energy function for both the van der Waals and electrostatic interactions. Table 1 summarizes the essential parameters for the non-equilibrium free energy calculations as defined in Gromacs. The suggested values should be adjusted to a particular problem at hand.

2. The equilibration simulations at the end states ($\lambda=0$ and $\lambda=1$) are performed following the standard unbiased molecular dynamics simulation setup. If the hybrid topology for a system is generated, the free energy code must be switched on and the λ state has to be defined (*see* **Note 1**). It is also important to consider which ensemble, canonical or *NPT*, is sampled, since on this choice depends which free energy will be calculated—Helmholtz or Gibbs. Dispersion correction for the energy and pressure has a significant effect on the accuracy of estimated free energies [70, 71] (simulation parameter in Gromacs `DispCorr=EnerPres`).
3. When setting up non-equilibrium free energy simulations, three parameters concerning the simulation length need to be decided upon. Firstly, the length of an equilibrium sampling simulation has to be defined. The time dedicated for this step very much depends on a particular system at hand, and on the time scale of conformational changes: an equilibration may vary from a nanosecond to microsecond range. Secondly, the number of spawned non-equilibrium transitions needs to be chosen. The starting structures, with the associated velocities, need to cover the sampled equilibrium ensemble. In practice, 100 transitions in each direction are often sufficient to reach a converged free energy estimate. The third parameter, the time spent for a single transition, depends on the scale of perturbation between the states *A* and *B*. Extending the time from 50 ps up to several nanoseconds per simulation will improve convergence, as for the slower transitions less work will be dissipated. A convenient method to assess the convergence is by monitoring the extent of an overlap between the work distributions for the forward and backward transitions: for the larger overlap free energy estimate is more accurate. While the second and third parameters are responsible for the convergence and statistical error of the free energy estimate, it is the equilibration time which mostly contributes to the accuracy of the $\Delta\Delta G$ estimate.

4. The work required for a transition is obtained by numerically integrating the $\partial H/\partial\lambda$ curves. It is important to bear in mind that the integration ranges for the forward transition are defined to be 0 and 1, not to confuse them with the actual simulation time (which is, e.g., reported in the Gromacs $\partial H/\partial\lambda$ output). For the reverse transition integration from 1 to 0 has to be performed and afterwards the sign of the work value needs to be inverted (*see* Eq. 17 in Subheading 2.4). This effectively allows integration from 0 to 1 without the subsequent sign inversion for the reverse transition. The `pmx` [48] package contains the script (`analyze_crooks.py`) performing the integration using Gromacs $\partial H/\partial\lambda$ output directly for an arbitrary number of forward and reverse trajectories. Estimate of the free difference using the CGI approach (Eq. 19) or BAR (Eq. 20) can also be obtained with `analyze_crooks.py` in the `pmx` package.

5. The hybrid topology for an amino acid mutation can be created employing the pre-generated mutation database available in the `pmx` package. Firstly, the hybrid structural file containing atoms of both states needs to be generated (script `mutate.py`). In the second step, the standard topology generation is performed using the newly created structure file (`pdb2gmx` in Gromacs). As the hybrid structure (e.g., A2L amino acid denoting alanine in state *A* and leucine in state *B*) is not present in the standard biomolecular force fields, one needs to use the specifically modified force field files provided in the `pmx`. Finally, the parameters for the *B* state of the system need to be added to the topology file. This can be accomplished with the `make_bstate.py` script from the `pmx` package.

References

1. Torrie GM, Valleau JP (1977) Nonphysical sampling distributions in Monte Carlo free-energy estimation: umbrella sampling. J Comput Phys 23(2):187–199
2. Kirkwood JG (1935) Statistical mechanics of fluid mixtures. J Chem Phys 3:300–313
3. Ben-Naim A (1992) Statistical thermodynamics for chemists and biochemists. Plenum Press, New York
4. Chipot C, Pohorille A (eds) (2007) Free energy calculations. Theory and applications in chemistry and biology. Springer series in chemical physics, vol 86
5. Christ CD, Mark AE, van Gunsteren WF (2010) Basic ingredients of free energy calculations: a review. J Comput Chem 31(8):1569–1582. ISSN 1096-987X. doi:10.1002/jcc.21450. http://dx.doi.org/10.1002/jcc.21450
6. Michael S, David M (2013) An introduction to best practices in free energy calculations. In: Biomolecular simulations: methods and protocols. Methods in molecular biology, vol 924. Humana Press, New York, pp 271–311
7. Zwanzig RW (1954) High-temperature equation of state by a perturbation method. I. Nonpolar gases. J Chem Phys 22: 1420–1426
8. Pohorille A, Jarzynski C, Chipot C (2010) Good practices in free-energy calculations. J Phys Chem B 114(32):10235–10253
9. Landau LD (1938) Statistical physics. The Clarendon Press, Oxford
10. Lu N, Kofke DA (2001) Accuracy of free-energy perturbation calculations in molecular simulation. I. Modeling. J Chem Phys 114: 7303–7311

11. Lu N, Kofke DA (2001) Accuracy of free-energy perturbation calculations in molecular simulation. II. Heuristics. J Chem Phys 115: 6866–6875

12. Shirts MR, Pande VS (2005) Comparison of efficiency and bias of free energies computed by exponential averaging, the Bennett acceptance ratio, and thermodynamic integration. J Chem Phys 122(14):144107-1–144107-16

13. Bennett CH (1976) Efficient estimation of free energy differences from Monte Carlo data. J Comput Phys 22(2):245–268

14. Shirts MR, Bair E, Hooker G, Pande VS (2003) Equilibrium free energies from nonequilibrium measurements using maximum-likelihood methods. Phys Rev Lett 91(14):140601

15. Mezei M (1992) Polynomial path for the calculation of liquid state free energies from computer simulations tested on liquid water. J Comput Chem 13(5):651–656

16. Steinbrecher T, Mobley DL, Case DA (2007) Nonlinear scaling schemes for Lennard-Jones interactions in free energy calculations. J Chem Phys 127:214108

17. Kumar S, Rosenberg JM, Bouzida D, Swendsen RH, Kollman PA (1992) The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. J Comput Chem 13(8):1011–1021

18. Shirts MR, Chodera JD (2008) Statistically optimal analysis of samples from multiple equilibrium states. J Chem Phys 129:124105

19. Shenfeld DK, Xu H, Eastwood MP, Dror RO, Shaw DE (2009) Minimizing thermodynamic length to select intermediate states for free-energy calculations and replica-exchange simulations. Phys Rev E 80(4):046705

20. Buelens FP, Grubmüller H (2012) Linear-scaling soft-core scheme for alchemical free energy calculations. J Comput Chem 33(1): 25–33

21. Mazor M, Pettitt BM (1991) Convergence of the chemical potential in aqueous simulations. Mol Simul 6(1–3):1–4

22. Mitchell MJ, McCammon JA (1991) Free energy difference calculations by thermodynamic integration: difficulties in obtaining a precise value. J Comput Chem 12(2):271–275

23. Straatsma TP, McCammon JA (1991) Multiconfiguration thermodynamic integration. J Chem Phys 95:1175

24. Jorge M, Garrido NM, Queimada AJ, Economou IG, Macedo EA (2010) Effect of the integration method on the accuracy and computational efficiency of free energy calculations using thermodynamic integration. J Chem Theory Comput 6(4):1018–1027

25. Bruckner S, Boresch S (2011) Efficiency of alchemical free energy simulations. II. Improvements for thermodynamic integration. J Comput Chem 32(7):1320–1333

26. Jarzynski C (1997) Nonequilibrium equality for free energy differences. Phys Rev Lett 78(14):2690–2693

27. Cuendet MA (2006) The Jarzynski identity derived from general hamiltonian or non-hamiltonian dynamics reproducing NVT or NPT ensembles. J Chem Phys 125:144109

28. Hummer G (2001) Fast-growth thermodynamic integration: error and efficiency analysis. J Chem Phys 114:7330–7337

29. Gore J, Ritort F, Bustamante C (2003) Bias and error in estimates of equilibrium free-energy differences from nonequilibrium measurements. Proc Natl Acad Sci USA 100(22): 12564–12569

30. Crooks GE (1998) Nonequilibrium measurements of free energy differences for microscopically reversible Markovian systems. J Stat Phys 90(5–6):1481–1487

31. Crooks GE (1999) Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences. Phys Rev E 60(3):2721–2726

32. Chelli R, Marsili S, Barducci A, Procacci P (2007) Recovering the Crooks equation for dynamical systems in the isothermal-isobaric ensemble: a strategy based on the equations of motion. J Chem Phys 126:044502

33. Nanda H, Lu N, Woolf TB (2005) Using non-Gaussian density functional fits to improve relative free energy calculations. J Chem Phys 122(13):134110-1–134110-8

34. Maragakis P, Ritort F, Bustamante C, Karplus M, Crooks GE (2008) Bayesian estimates of free energies from nonequilibrium work data in the presence of instrument noise. J Chem Phys 129:024102

35. Goette M, Grubmüller H (2009) Accuracy and convergence of free energy differences calculated from nonequilibrium switching processes. J Comput Chem 30(3):447–456

36. Bramwell ST, Christensen K, Fortin J-Y, Holdsworth PCW, Jensen HJ, Lise S, López JM, Nicodemi M, Pinton J-F, Sellitto M (2000) Universal fluctuations in correlated systems. Phys Rev Lett 84(17):3744–3747

37. Massey FJ Jr (1951) The Kolmogorov-Smirnov test for goodness of fit. J Am Stat Assoc 46(253):68–78

38. Pearlman DA, Kollman PA (1991) The overlooked bond-stretching contribution in free energy perturbation calculations. J Chem Phys 94:4532–4545

39. Pearlman DA (1994) A comparison of alternative approaches to free energy calculations. J Phys Chem 98(5):1487–1493
40. Boresch S, Karplus M (1999) The role of bonded terms in free energy simulations. 2. Calculation of their influence on free energy differences of solvation. J Phys Chem A 103(1):119–136
41. Boresch S, Karplus M (1999) The role of bonded terms in free energy simulations: 1. Theoretical analysis. J Phys Chem A 103(1): 103–118
42. Bash PA, Singh UC, Langridge R, Kollman PA (1987) Free energy calculations by computer simulation. Science 236(4801):564–568
43. Beutler TC, Mark AE, van Schaik RC, Gerber PR, van Gunsteren WF (1994) Avoiding singularities and numerical instabilities in free energy calculations based on molecular simulations. Chem Phys Lett 222(6):529–539. ISSN 0009-2614
44. Zacharias M, Straatsma TP, McCammon JA (1994) Separation-shifted scaling, a new scaling method for Lennard-Jones interactions in thermodynamic integration. J Chem Phys 100: 9025–9031
45. Gapsys V, Seeliger D, de Groot BL (2012) New soft-core potential function for molecular dynamics based alchemical free energy calculations. J Chem Theory Comput 8(7): 2373–2382
46. Tosco P, Balle T, Shiri F (2011) Open 3DALIGN: an open-source software aimed at unsupervised ligand alignment. J Comput Aided Mol Des 25(8):777–783
47. Hess B, Kutzner C, van der Spoel D, Lindahl E (2008) GROMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation. J Chem Theory Comput 4(3): 435–447
48. Seeliger D, De Groot BL (2010) Protein thermostability calculations using alchemical free energy simulations. Biophys J 98(10):2309–2316. ISSN 0006-3495
49. O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR (2011) Open Babel: an open chemical toolbox. J Cheminf 3(1):1–14
50. Sadowski J, Gasteiger J, Klebe G (1994) Comparison of automatic three-dimensional model builders using 639 X-ray structures. J Chem Inf Comput Sci 34(4):1000–1008
51. Singh UC, Kollman PA (1984) An approach to computing electrostatic charges for molecules. J Comput Chem 5(2):129–145
52. Bayly CI, Cieplak P, Cornell W, Kollman PA (1993) A well-behaved electrostatic potential based method using charge restraints for deriving atomic charges: the RESP model. J Phys Chem 97(40):10269–10280
53. Jakalian A, Bush BL, Jack DB, Bayly CI (2000) Fast, efficient generation of high-quality atomic charges. AM1-BCC model: I. Method. J Comput Chem 21(2):132–146
54. Mobley DL, Dumont É, Chodera JD, Dill KA (2007) Comparison of charge models for fixed-charge force fields: small-molecule hydration free energies in explicit solvent. J Phys Chem B 111(9):2242–2254
55. Wang J, Wang W, Kollman PA, Case DA (2006) Automatic atom type and bond type perception in molecular mechanical calculations. J Mol Graph Model 25(2):247–260
56. Wang J, Wolf RM, Caldwell JW, Kollman PA, Case DA (2004) Development and testing of a general amber force field. J Comput Chem 25(9):1157–1174
57. Vanommeslaeghe K, Hatcher E, Acharya C, Kundu S, Zhong S, Shim J, Darian E, Guvench O, Lopes P, Vorobyov I, MacKerell AD Jr (2010) CHARMM general force field: a force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. J Comput Chem 31(4):671–690
58. Vanommeslaeghe K, MacKerell AD Jr (2012) Automation of the CHARMM General Force Field (CGenFF) I: bond perception and atom typing. J Chem Inf Model 52(12): 3144–3154
59. Vanommeslaeghe K, Raman EP, MacKerell AD Jr (2012) Automation of the CHARMM General Force Field (CGenFF) II: assignment of bonded parameters and partial atomic charges. J Chem Inf Model 52(12):3155–3168
60. Malde AK, Zuo L, Breeze M, Stroet M, Poger D, Nair PC, Oostenbrink C, Mark AE (2011) An automated force field topology builder (ATB) and repository: version 1.0. J Chem Theory Comput 7(12):4026–4037
61. Ribeiro AAST, Horta BAC, de Alencastro RB (2008) MKTOP: a program for automatic construction of molecular topologies. J Braz Chem Soc 19(7):1433–1435
62. Rocklin GJ, Mobley DL, Dill KA, Hünenberger PH (2013) Calculating the binding free energies of charged species based on explicit-solvent simulations employing lattice-sum methods: an accurate correction scheme for electrostatic finite-size effects. J Chem Phys 139(18):184103.
63. Talhout R, Villa A, Mark AE, Engberts JBFN (2003) Understanding binding affinity: a combined isothermal titration calorimetry/molecular dynamics study of the binding of a

series of hydrophobically modified benzamidinium chloride inhibitors to trypsin. J Am Chem Soc 125(35):10570–10579

64. Marquart M, Walter J, Deisenhofer J, Bode W, Huber R (1983) The geometry of the reactive site and of the peptide groups in trypsin, trypsinogen and its complexes with inhibitors. Acta Crystallogr Sect B Struct Sci 39(4):480–490
65. Lu W, Apostol I, Qasim MA, Warne N, Wynn R, Zhang WL, Anderson S, Chiang YW, Ogin E, Rothberg I, Ryan K, Laskowski M (1997) Binding of amino acid side-chains to S_1 cavities of serine proteinases. J Mol Biol 266(2): 441–461
66. Benedix A, Becker CM, de Groot BL, Caflisch A, Böckmann RA (2009) Predicting free energy changes using structural ensembles. Nat Methods 6(1):3–4
67. Fujinaga M, Sielecki AR, Read RJ, Ardelt W, Laskowski M, James MNG (1987) Crystal and molecular structures of the complex of α-chymotrypsin with its inhibitor turkey ovomucoid third domain at 1.8 Å resolution. J Mol Biol 195(2):397–418
68. Hornak V, Abel R, Okur A, Strockbine B, Roitberg A, Simmerling C (2006) Comparison of multiple Amber force fields and development of improved protein backbone parameters. Proteins Struct Funct Bioinform 65(3): 712–725
69. Mobley DL, Chodera JD, Dill KA (2006) On the use of orientational restraints and symmetry corrections in alchemical free energy calculations. J Chem Phys 125(8):084902. doi: 10.1063/1.2221683. http://link.aip.org/link/?JCP/125/084902/1
70. Shirts MR, Pitera JW, Swope WC, Pande VS (2003) Extremely precise free energy calculations of amino acid side chain analogs: comparison of common molecular mechanics force fields for proteins. J Chem Phys 119(11): 5740–5761
71. Shirts MR, Mobley DL, Chodera JD, Pande VS (2007) Accurate and efficient corrections for missing dispersion interactions in molecular simulations. J Phys Chem B 111(45): 13052–13063

Part II

Conformational Change

Chapter 10

The Use of Experimental Structures to Model Protein Dynamics

Ataur R. Katebi, Kannan Sankar, Kejue Jia, and Robert L. Jernigan

Abstract

The number of solved protein structures submitted in the Protein Data Bank (PDB) has increased dramatically in recent years. For some specific proteins, this number is very high—for example, there are over 550 solved structures for HIV-1 protease, one protein that is essential for the life cycle of human immunodeficiency virus (HIV) which causes acquired immunodeficiency syndrome (AIDS) in humans. The large number of structures for the same protein and its variants include a sample of different conformational states of the protein. A rich set of structures solved experimentally for the same protein has information buried within the dataset that can explain the functional dynamics and structural mechanism of the protein. To extract the dynamics information and functional mechanism from the experimental structures, this chapter focuses on two methods—Principal Component Analysis (PCA) and Elastic Network Models (ENM). PCA is a widely used statistical dimensionality reduction technique to classify and visualize high-dimensional data. On the other hand, ENMs are well-established simple biophysical method for modeling the functionally important global motions of proteins. This chapter covers the basics of these two. Moreover, an improved ENM version that utilizes the variations found within a given set of structures for a protein is described. As a practical example, we have extracted the functional dynamics and mechanism of HIV-1 protease dimeric structure by using a set of 329 PDB structures of this protein. We have described, step by step, how to select a set of protein structures, how to extract the needed information from the PDB files for PCA, how to extract the dynamics information using PCA, how to calculate ENM modes, how to measure the congruency between the dynamics computed from the principal components (PCs) and the ENM modes, and how to compute entropies using the PCs. We provide the computer programs or references to software tools to accomplish each step and show how to use these programs and tools. We also include computer programs to generate movies based on PCs and ENM modes and describe how to visualize them.

Key words HIV-1 protease, Principal component analysis, Elastic network model, Protein dynamics, Acquired immunodeficiency syndrome, Protein data bank

1 Introduction

There are large numbers of structures in the protein data bank (PDB [1]) for many categories of enzymes. Shown in Fig. 1 are the most abundant enzyme structures ordered by enzyme commission (EC) numbers. Some other examples for individual EC categories,

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215, DOI 10.1007/978-1-4939-1465-4_10, © Springer Science+Business Media New York 2015

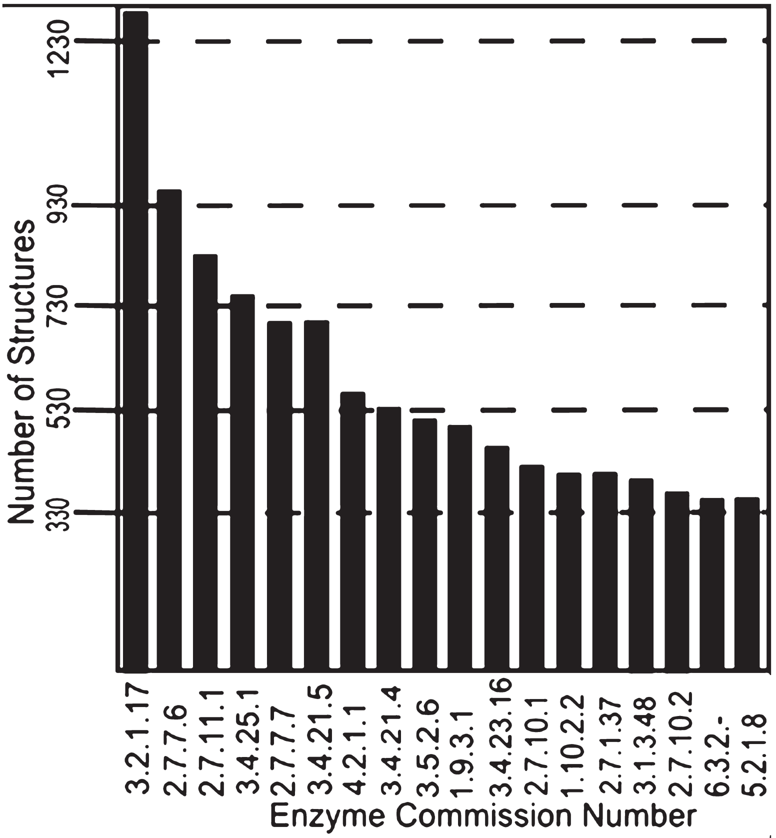


Fig. 1 Numbers of related protein structures available for extracting protein functional dynamics—snapshot of the PDB statistics for the largest categories of enzymes (08/30/2013). In total, there are over 17,000 enzyme structures, and a significant number of structures for many diverse enzyme types. The most common structure on the left of this histogram with 1,285 structures is EC 3.2.1.17 that includes lysozymes, and at the *right* side is 5.2.1.8 acetylcholinesterases with 337 different structures (taken from enzyme classification data provided by PDB: http://www.pdb.org/pdb/statistics/histogram.do?mdcat = entity&mditem = pdbx_ec&name = Enzyme%20Classification) [1])

with the numbers of their related structures in parentheses are: 3.4.21: Serine endopeptidases (2,459), 3.4.23: Aspartic endopeptidases (1,146), 3.4.24: Metalloendopeptidases (727), 3.4.22: Cysteine endopeptidases (720), 3.4.11: Aminopeptidases (292), 3.4.19: Omega peptidases (244), 3.4.17: Metallocarboxypeptidases (144), 3.4.14: Dipeptidyl-peptidases (120), 3.4.25: Threonine endopeptidases (109), 2.7.7, Nucleotidyltransferases (107), 3.4.21: Serine endopeptidases (105), 3.4.16: Serine-type carboxypeptidases (97), 2.7.7: Nucleotidyltransferases (106), 3.4.23: Aspartic endopeptidases (77), and 3.4.19: Omega peptidases (58). In addition, there are many structures of non-enzyme proteins—structural proteins, immunoglobulin Fab's, viral proteins, and many others. The PDB has many additional ways to search for functionally related structures that are invaluable for finding structures with similar dynamics. You can search by biological process such as gene ontology (GO), cellular component, molecular function, and transporter classification. In addition there are many receptors with multiple reported structures. Overall, there is abundant data to investigate functional protein dynamics of many classes of proteins directly from experimental structures.

Important conformational changes can readily be extracted from a set of PDB structures for a protein and these are found to relate directly to function. Experimental structures can be a rich source of information. It is well established that functionally related structures must have similar structures and similar dynamics—building on the broad experience of many researchers. There have been several efforts at extracting dynamics from specific sets of experimental structures. One approach is principal component analysis (PCA) [2–4], a statistical method based on covariance analysis. PCA can transform the original space of correlated variables into a greatly reduced space of independent variables (i.e., the principal components or PCs). By performing PCA, most of a system's variance will usually be captured in a quite small subset of the PCs. PCA has been applied often to analyze trajectory data from MD simulations to find the essential dynamics [5, 6]. Teodoro et al. applied PCA to the dataset composed of many conformations for HIV-1 protease [7, 8]. They found that PCA transformed the original high-dimensional representation of protein motions into a low-dimensional one that provides the dominant protein motions. This is a huge reduction in dimensionality from hundreds of thousands to fewer than 50 degrees of freedom. Howe [9] used PCA to classify the structures in NMR ensembles automatically, according to correlated structural variations, and the results have shown that two different representations of the protein structure, the Cα coordinate matrix and the Cα–Cα distance matrix, gave equivalent results and permitted the identification of structural differences between conformations. More recent efforts include our own previous efforts in analyzing the HIV-1 protease set [10], those of the Bahar group [11], and our efforts in developing the MAVEN program [12], as well as related efforts by the Bahar group with their ProDy [13], and Grant with his Bio-3D [14]. Any of these can provide a similar set of starting tools.

On the other hand, the Elastic Network Models (ENM) have proven themselves to be highly useful in representing the global motions for a wide variety of diverse protein structures [15–19]. For modeling and simulating the dynamics of proteins, ENMs can be applied on multiple scales [20–23]. All atom ENM models give a finer description of protein dynamics. The most common coarse-graining involves a single-site per residue representation, in which the sites are identified by the Cα atoms and connected by uniform springs. The dynamics of such interconnected model can be described by the Gaussian Network Model (GNM) [17] or the Anisotropic Network Model (ANM) [15]. GNM has been very successful in yielding information on the magnitudes of the fluctuations of the protein structures but provides no directional information or the 3-D nature of motion of the protein is considered in the model. However, in reality protein fluctuations are generally directional and anisotropic [24, 25]. ANM considers the anisotropy of the protein structure in modeling its dynamics and thus ANM computed collective motions are more relevant to biological function and mechanism of the protein molecule.

In this chapter, we give an example of how to use computational methods to extract protein dynamics from a large set of experimental structures of HIV-1 protease. Behind this is the implicit assumption that there is a significant amount of information about protein dynamics, mechanisms and allostery buried within the structures in the PDB. We will show how to utilize PCA to extract dynamics from the abundantly available HIV-1 protease structures and how to compute the agreement between PCA-based protein motion and the ANM modeled motion, and describe how these could be used in simulations with a new structure-based elastic network model.

2 Theory

2.1 Principal Component Analysis (PCA)

PCA is a multivariate technique to analyze a dataset where the observations are described quantitatively by a set of inter-correlated variables. The goals of PCA are to (1) extract the most important information from the data; (2) remove noise and compress the data set by keeping only the important information; (3) simplify the description of the data set; and (4) analyze the structure of the observations and the variables. This method generates a set of new orthogonal variables called principal components (PCs). Each PC is a linear combination of the original variables. Hence, PCA can be considered as a mapping of the data points from the original variable space to the PC space. PCs are rank ordered in such a way that PC1 represents the maximum variance among all possible choices for the first axis. Similarly, PC2 represents the second highest variance contribution, and so forth through all the modes. Usually only a few PCs are sufficient to understand the internal structure of the data [26].

For extracting functional dynamics from the PDB experimental structures, PCA is performed on the structure datasets. The input is the set of coordinates of all of the structures in the set [7, 8]. From these data, the average position of each point in the structure is computed as $\langle r_i \rangle$ and the covariances for pairs of points i and j are computed according to

$$c_{ij} = \left\langle \left(r_i - r_i \right) \left(r_j - r_j \right) \right\rangle \tag{1}$$

where brackets ⟨⟩ indicate averages over the entire set of structures. The covariance matrix C can be decomposed as

$$C = P\Delta P^{\mathrm{T}}, \tag{2}$$

where the eigenvectors P represent the principal components (PCs) and the eigenvalues are the elements of the diagonal matrix Δ. The eigenvalues are sorted in order. Each eigenvalue is directly proportional to the amount of the variance it captures.

2.2 Elastic Network Model (ENM)

Anisotropic Network Model (ANM) is an elastic network model used to compute the directions of the normal modes from a single structure [15]. In ANM, the potential energy V is a function of the displacement vector D of each point in the structure

$$V = \frac{\gamma}{2} DHD^{\mathrm{T}}, \tag{3}$$

where γ is the spring constant for all closely interacting points in a structure, and H is the Hessian matrix containing the second derivatives of the energy, with respect to each of the coordinates $r = \langle x, y, z \rangle$. For a structure with n residues, the Hessian matrix H contains $n \times n$ super-elements of size 3×3. The Hessian matrix H can be decomposed [7, 8, 15] as

$$H = M\Lambda M^{\mathrm{T}}, \tag{4}$$

where Λ is a diagonal matrix comprising the eigenvalues with the eigenvectors forming the columns of the matrix M. This decomposition generates $3n - 6$ normal modes (the first six modes account for the rigid body translations and rotations of the system and must be factored out, meaning that we actually perform singular value decomposition to extract the normal modes) reflecting the vibrational fluctuations. We like to further mention that for ANM coarse graining, it is shown that a cutoff distance of any value from 10 to 13 Å is appropriate for placing the springs and such an ANM model represents the realistic protein dynamics. In this chapter, we use a cutoff distance of 13 Å.

2.3 Structure-Based New ANM

The internal distance changes in a set of structures can provide information that can be used directly to derive new structure-based elastic network models. We have extracted spring constants between all residue pairs in a set of structures by simply relating these to the inverse of the variance of internal distance changes between pairs of residues, as the spring stiffness (normalized between 0 and 1). We have applied a cutoff of 13 Å to limit the range of interactions. However the difference between the conventional ANM described in the previous section and this modified ANM is that here the values for the spring constants are obtained directly from the structure set rather than using a uniform value or distance dependent values, as is customary with ENM.

2.4 Comparing Directions of Motions Using Overlaps

The alignment between the directions of motion, for example between a given PC and a given normal mode, is measured by their overlap, which was defined as the dot product of the two vector directions by Tama and Sanejouand [27]

$$O_{ij} = \frac{\left|P_i \cdot M_j\right|}{\left\|P_i\right\|\left\|M_j\right\|}, \tag{5}$$

where P_i is the *i*th PC for model P and M_j is the *j*th PC or normal mode for model M. A perfect match yields an overlap value of 1. They also defined the cumulative overlap (CO) between the first *k* vectors of *M* and P_i as

$$\mathrm{CO}(k) = \left(\sum_{j=1}^{k} O_{ij}^2\right)^{\frac{1}{2}} \tag{6}$$

which measures how well the first *k* PCs for model M together can capture the motion of a single PC for model P.

2.5 Coarse-Grained Global Entropies Calculated from Principal Component Analysis

As covariance matrix can be decomposed as in Eq. 2 of Subheading 2.1, an approximation of the entropy from the PCs can be obtained as well [10, 28]:

$$\Delta S = \mathrm{Const}\sum_{i=1}^{N}\lambda_i\left(\mathrm{PC}_i\mathrm{PC}_i^{\mathrm{T}}\right) \tag{7}$$

where PC_i is the *i*th PC, and λ_i is the *i*th eigenvalue, *N* is the total number of eigenvalues.

Andricioaei et al. also reported a similar result for entropy calculation from the covariance matrices of the atomic fluctuations as shown in equation 7 of their paper [29]. It should be noted that this expression is different from that for normal modes of the elastic network models, which because of the averaging normally involved the inverse of the eigenvalues.

3 Materials

There are a huge number of available HIV-1 protease structures in the PDB (564 X-ray and three NMR structures as of 07/26/2013), which provides a remarkably rich set of different conformational

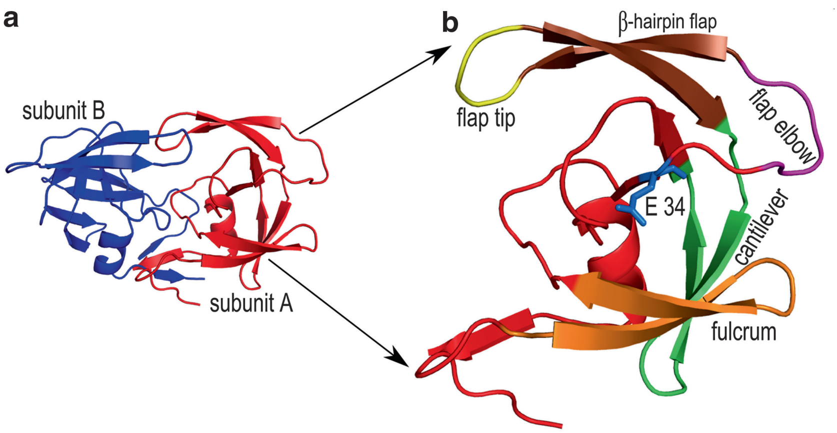


Fig. 2 Description of HIV-1 protease homo-dimer and its critical structural components that facilitate the functional dynamics (**a**) HIV-1 protease has two symmetric subunits—subunit A (*red*) and subunit B (*blue*). (**b**) Each subunit has several structural components that are important for its coordinated motions. *Fulcrum* (*orange*, residues 9–21) is a comparatively less mobile region that swings up and down similar to the flap elbow. *E-34* (*blue*)—Hinge residue which is responsible for transmitting the motion from the fulcrum to the flap region. *Flap elbow* (*magenta*, residues 37–42)—Hinge residue E-34 drives the motion of this region to transfer the dynamics further away from the fulcrum to the upper flap region. This loop can make top-down and bottom-up swings. When the flap elbow swings from top to bottom, the flap domain opens up, and when it swings upward the flap domain closes. The *Flap domain* (residues 43–58) consists of flap tip (*yellow*, residues 49–52) and β-hairpin flaps (*dark orange*, residues 43–48 and 53–58). Opening and closing of the flap domains enable the protein to bind ligands and release its products after proteolysis. *Cantilever* (*green*, residues 59–75) functions as a base for the flap domain. The C-terminal β-hairpin flap is held by the N-terminal end of the cantilever and this arrangement is important to control the swinging of the flap [30, 31]

states, which can be viewed as direct structural information on the protein's dynamics.

The approach described here computes the essential or most important protein motions from multiple structures of the same protein, in contrast to using just the two structures such as the "open" and "closed" conformations, which have often been used to define the endpoints of conformational transitions. To demonstrate this approach, we use HIV-1 protease as an example. Its abundant experimentally determined structures are complemented by the relatively small size of the protein. In the next section, first, we will give a description of the structural components that are important to drive the motion of the HIV-1 protease structure. Then, we will describe the dataset of HIV-1 structures that we have used to perform our computations.

3.1 HIV-1 Protease Architecture

HIV-1 protease functions as a homo-dimer as shown in Fig. 2a. The dimer has a single active site and 99 residues per monomer. Each monomer has three domains: a terminal domain (residues 1–4 and 95–99 of each chain), which is important for the dimerization and stabilization; a core domain (residues 10–32 and 63–85 of each chain), for dimer stabilization and catalytic site stability; and a flap domain that includes two solvent accessible loops (residues 33–43 of each chain) followed by two flexible flaps (residues 44–62 of each chain) important for ligand binding interactions. The conserved Asp25-Thr26-Gly27 active site triad is located at the interface between parts of the core domains. The active site of HIV-1 protease is formed at the homo-dimer interface. Each monomeric unit has important structural components as identified in Fig. 2b that are important for its functional dynamics. The principal advantage of this structural arrangement is that the hinge residue *E 34* causes the up-down swinging motion of the *flap elbow* (residues 37–42), which transmits the motion generated in the *fulcrum* (residues 9–21) to drive the dynamics of the *flap domain* (residues 42–58), whose conformation switches between open and closed states to facilitate substrate trapping in the catalytic pocket and product release following hydrolysis [30, 31].

3.2 HIV-1 Protease Structure Set (X-Ray-329)

We have used 329 PDB structures of HIV-1 protease for the computations to extract protein dynamics from experimental structures. The PDB Ids of the data set are here (*see* **Notes 1** and **2**):

1A30	1A8G	1A8K	1A94	1A9M	1AAQ	1AID	1AJV	1AJX	1AXA	1B6J	1B6K	1B6L
1B6M	1B6P	1BDL	1BDQ	1BDR	1BV7	1BV9	1BWA	1BWB	1C6X	1C6Y	1C6Z	1C70
1D4S	1D4Y	1DAZ	1DIF	1DMP	1DW6	1EBK	1EBW	1EBY	1EBZ	1EC0	1EC1	1EC2
1EC3	1F7A	1FEJ	1FF0	1FFF	1FFI	1FG6	1FG8	1FGC	1FQX	1G2K	1G35	1GNM
1GNN	1GNO	1HBV	1HIH	1HIV	1HOS	1HPO	1HPS	1HPV	1HPX	1HSG	1HSH	1HTE
1HTF	1HTG	1HVH	1HVI	1HVJ	1HVK	1HVL	1HVR	1HVS	1HWR	1HXW	1IIQ	1IZH
1IZI	1K1U	1K2B	1K2C	1K6C	1K6P	1K6T	1K6V	1KJ4	1KJ7	1KJF	1KJG	1KJH
1LZQ	1M0B	1MER	1MES	1MET	1MEU	1MRW	1MRX	1MSM	1MSN	1MT7	1MT8	1MT9
1MTB	1MTR	1MUI	1N49	1NH0	1NPA	1NPV	1NPW	1ODW	1ODX	1PRO	1QBR	1QBS
1QBT	1QBU	1RL8	1RPI	1RQ9	1RV7	1SDT	1SDU	1SDV	1SGU	1SH9	1SP5	1T3R
1T7I	1T7J	1T7K	1TCX	1TW7	1U8G	1VIJ	1VIK	1XL2	1XL5	1YT9	1YTG	1YTH
1Z8C	1ZBG	1ZLF	1ZPK	1ZSF	1ZSR	2A1E	2A4F	2AID	2AOF	2AQU	2AVM	2AVO
2AVS	2AVV	2AZC	2B7Z	2BB9	2BBB	2BPV	2BPW	2BPX	2BPY	2BPZ	2BQV	2CEJ
2CEM	2CEN	2F3K	2F80	2F81	2F8G	2FDD	2FDE	2FGU	2FGV	2FNS	2FNT	2FXD
2FXE	2HB3	2HC0	2HS1	2HS2	2I4D	2I4U	2I4V	2I4W	2I4X	2IDW	2IEN	2IEO
2J9J	2J9K	2JE4	2NMZ	2NNK	2NNP	2O4K	2O4L	2O4P	2O4S	2P3A	2P3B	2P3C
2P3D	2PK5	2PK6	2PQZ	2PWC	2PWR	2PYM	2PYN	2Q3K	2Q63	2Q64	2QAK	2QCI
2QD6	2QD7	2QD8	2QHC	2QHY	2QHZ	2QI0	2QI1	2QI3	2QI4	2QI5	2QI6	2QI7
2QMP	2QNN	2QNP	2QNQ	2R38	2R3T	2R3W	2R43	2R5P	2R5Q	2RKF	2UPJ	2UXZ
2UY0	2Z4O	3A2O	3AID	3BGB	3BGC	3BVA	3BVB	3CKT	3CYW	3CYX	3D1X	3D3T

(continued)

(continued)

3FX5	3GGA	3GGV	3GGX	3GI4	3GI5	3GI6	3I7E	3KF0	3KFN	3KFR	3KFS	3LZS
3LZU	3MWS	3NDU	3NDX	3NU3	3NU4	3NU5	3NU6	3NU9	3NUJ	3NUO	3O9F	3O9G
3O9H	3O9I	3OK9	3OTS	3OXC	3PWM	3PWR	3QAA	3R4B	3S43	3S53	3S54	3S56
3S85	3SO9	3T11	3U7S	3UCB	3UF3	3UHL	4DQB	4DQC	4DQE	4DQG	4DQH	4EJ8
4EJK	4EJL	4FAE	4FL8	4FLG	4FM6	4HVP	4I8W	4I8Z	4J54	4J55	4J5J	4PHV
7HVP	7UPJ	8HVP	9HVP									

The following method section gives a step by step description of how to retrieve these PDB files from the protein databank and then how to extract the dynamics from these structures.

4 Methods

To successfully complete the procedures described in this section, one needs the following software/programs:

- Perl 5—Several perl scripts are included here. Perl programming language [32] can be downloaded free at www.perl.org.
- Python—A python script is used to calculate the internal distances between residue pairs for the set of 329 protein structures. A Python environment can be downloaded at http://www.python.org/.
- Matlab—Several Matlab scripts are included here that can be executed in a Matlab programming environment [33]. Matlab product site is http://www.mathworks.com/products/matlab/.
- MAVENs—This software was developed in the Jernigan lab [12]. In our Matlab code, we have invoked several MAVEN functions:
  - ANM.m—This is a function from MAVEN [12] used in experimentalDynamics.m to compute ENM normal modes from a given PDB structure.
  - modeAnimator.m—This is a function from MAVEN used in experimentalDynamics.m to visualize the ENM modes and PCs by creating movies.
  - readPDB.m, writePDB.m—These two Matlab functions from MAVEN are used to read and write PDB files, respectively.
  - CompareVectors.m—This function from MAVEN is used in experimentalDynamics.m to compare the directions of PCs and ENM modes.
  - plot_compareVectors.m—This function from MAVEN plots the results obtained from the above CompareVectors.m.
  - mat2vec.m—This function converts a matrix to a vector.

MAVEN is available for download at http://maven.sourceforge.net.

- MUSTANG—Multiple structural alignment will be done using this program [34]. This program can be installed only on a Linux operating system. MUSTANG can be downloaded at http://www.csse.monash.edu.au/~karun/Site/mustang.html.
- PyMOL—This software has a free version for academic use [35]. This can be used to visualize the structures and their dynamics. PyMOL can be downloaded at http://pymol.org/.

The following Table 1 summarizes the steps that are discussed in this section—starting from processing raw PDB structures to computing PCs and ANM modes, and comparing their dynamics.

4.1 Extracting Cartesian Coordinates from Raw PDB Files

In this section, we will describe how to prepare the dataset X-ray-329 for PCA. The 329 PDB Ids are listed in the pdbIds.txt file. Download these files from the protein data bank (http://www.pdb.org/pdb/download/download.do) (Download options: download Type—PDB File Format, Compression Type—uncompressed) and save them in a sub folder named *data-raw* under the parent folder *experimentalDynamics*. The downloaded PDB files have a lot of extra information that we will not be used.

The records of ATOM type for residue 8 and modified residue 67 of PDB file 2p3a are shown in Schema 1. The important fields are labeled. Each residue of a protein is recorded in this way. When a residue in the protein is modified with a non-amino acid type molecule, HETATM keyword is used to identify that record. The TER key word is used as an end of chain marker. The PDB file has other detailed information and have different record identifiers. We will retain the ATOM type records for the Cα atoms of each residue or modified residue for our calculation. When more than one alternate location is recorded, we arbitrarily retain the first alternate location for that ATOM.

The following three subsections describe how to copy the Cartesian coordinates from each PDB file and align these structures.

4.1.1 Preparing a Data Set for MUSTANG from Raw PDB Files

Download and save the following perl scripts in the same folder—*experimentalDynamics*. Run these perl scripts in the same sequence as they are listed below:

- copyBackboneAtoms.pl—This program copies the backbone ATOM and HETATM from a set of PDB files.
  - perl copyBackboneAtoms.pl
  - Output files after running this program will be saved in data-backbone subfolder of the *experimentalDynamics* parent folder.

Table 1
Summary of the steps for extracting biomolecular dynamics

Program/file name	Function
Subheading 4.1 Extracting Cartesian coordinates from raw PDB files Subheading 4.1.1 Data set preparation for MUSTANG from raw PDB files	
copyBackboneAtoms.pl	Copies backbone ATOM and HETATM from a set of PDB files.
retainFirstAltLocation.pl	Retains the first alternate location for each ATOM and HETATM when multiple locations for that ATOM/HETATM exist. It operates on a set of PDB files.
replaceHETATM.pl	Replaces the keyword HETATM with the keyword ATOM in a set of PDB files.
retainCA.pl	Copies the CA atoms from a set of PDB files with no TER keyword between chains to comply with the MUSTANG input file format.
Subheading 4.1.2 Multiple structural alignment using MUSTANG	
Subheading 4.1.3 Data set preparation for PCA from MUSTANG output	
copyChainsToPDBs.pl	Copies the chains from alignAll.pdb to individual PDB files.
pdbIds.txt	This file list the PDB ids for 329 PDB structures used here.
Subheading 4.2 Principal Component Analysis (PCA) Subheading 4.2.2 Comparing and visualizing PCs and ANM modes Subheading 4.3 Comparing PCs and structure-based ANM Subheading 4.4 Computing Entropy using PCs	
experimentalDynamics.m	This Matlab program (1) computes principal components from aligned structures, (2) computes ENM modes, (3) computes the overlap between PCs and ENM modes, (4) computes entropies from PCs.
readAlignedPDBcoordinates.m	This Matlab function reads the coordinates of aligned PDB structures and returns the coordinates of those structures.
internal.py	This program, written in Python, calculates the internal distances of Mustang aligned structures.
calc_Entropy_PC.m	This Matlab function computes entropy from computed PCs.

The above files, the files used from MAVEN, other accessory files and dataset can be downloaded at http://ribosome.bb.iastate.edu/4papers/2013/ataur/experimentalDynamics/

  - Running this program will retain the backbone atoms for each ATOM and HETATM record. A sample output for residue 8 and modified residue 67 is shown in Schema 2.

- retainFirstAltLocation.pl—This program retains the first alternate location for each ATOM when multiple alternative locations for that ATOM exist. It operates on a set of PDB files.
  - perl retainFirstAtlLocation.pl
  - Output files after running this program will be saved in *data-backbone-singleAltLocation* subfolder of the *experimentalDynamics* parent folder.

Record Id Atom No Alt Loc Ind Res No Cart. Coordinates

```
ATOM     72  N   AARG A   8      26.288  -8.483  -5.941  0.60 23.05           N
ATOM     73  N   BARG A   8      26.517  -8.547  -6.064  0.40 23.23           N
ATOM     74  CA  AARG A   8      25.875  -8.023  -4.614  0.60 21.71           C
ATOM     75  CA  BARG A   8      26.053  -8.180  -4.733  0.40 22.23           C
ATOM     76  C   AARG A   8      26.929  -8.501  -3.624  0.60 21.10           C
ATOM     77  C   BARG A   8      27.135  -8.490  -3.723  0.40 21.09           C
ATOM     78  O   AARG A   8      27.328  -9.676  -3.635  0.60 20.83           O
ATOM     79  O   BARG A   8      27.754  -9.554  -3.789  0.40 21.01           O
ATOM     80  CB  AARG A   8      24.484  -8.512  -4.224  0.60 22.35           C
ATOM     81  CB  BARG A   8      24.802  -8.972  -4.362  0.40 22.44           C
ATOM     82  CG  AARG A   8      23.395  -7.948  -5.115  0.60 23.39           C
ATOM     83  CG  BARG A   8      23.547  -8.611  -5.150  0.40 23.08           C
ATOM     84  CD  AARG A   8      22.022  -8.424  -4.762  0.60 24.19           C
ATOM     85  CD  BARG A   8      22.313  -9.292  -4.597  0.40 23.91           C
ATOM     86  NE  AARG A   8      21.030  -8.042  -5.770  0.60 26.15           N
ATOM     87  NE  BARG A   8      22.360 -10.733  -4.833  0.40 26.33           N
ATOM     88  CZ  AARG A   8      20.261  -8.897  -6.410  0.60 27.91           C
ATOM     89  CZ  BARG A   8      21.743 -11.654  -4.103  0.40 26.73           C
ATOM     90  NH1AARG A   8      20.376 -10.207  -6.178  0.60 28.97           N
ATOM     91  NH1BARG A   8      21.020 -11.326  -3.036  0.40 27.04           N
ATOM     92  NH2AARG A   8      19.386  -8.454  -7.293  0.60 29.86           N
ATOM     93  NH2BARG A   8      21.872 -12.918  -4.435  0.40 27.99           N
```

A. Records for Residue 8

```
HETATM  572  N   ACME A  67      31.550 -12.012   8.379  0.70 29.54           N
HETATM  573  N   BCME A  67      31.558 -11.938   8.292  0.30 29.11           N
HETATM  574  CA  ACME A  67      32.726 -12.421   7.660  0.70 31.90           C
HETATM  575  CA  BCME A  67      32.776 -12.485   7.653  0.30 30.89           C
HETATM  576  CB  ACME A  67      32.828 -11.366   6.558  0.70 32.28           C
HETATM  577  CB  BCME A  67      33.184 -11.741   6.377  0.30 30.93           C
HETATM  578  SG  ACME A  67      34.001 -11.803   5.322  0.70 38.15           S
HETATM  579  SG  BCME A  67      34.526 -12.577   5.566  0.30 33.46           S
HETATM  580  SD  ACME A  67      33.313 -13.228   4.106  0.70 38.82           S
HETATM  581  SD  BCME A  67      33.296 -13.133   3.990  0.00 19.59           S
HETATM  582  CE  ACME A  67      31.653 -13.713   4.229  0.70 33.41           C
HETATM  583  CE  BCME A  67      31.702 -13.776   4.257  0.00 18.95           C
HETATM  584  CZ  ACME A  67      31.498 -14.878   3.263  0.70 34.43           C
HETATM  585  CZ  BCME A  67      31.546 -14.934   3.274  0.00 21.88           C
HETATM  586  OH  ACME A  67      31.382 -14.483   1.906  0.70 35.37           O
HETATM  587  OH  BCME A  67      31.370 -14.537   1.922  0.00 21.78           O
HETATM  588  C   ACME A  67      33.921 -12.522   8.646  0.70 32.23           C
HETATM  589  C   BCME A  67      33.963 -12.613   8.623  0.30 31.53           C
HETATM  590  O   ACME A  67      35.001 -11.978   8.379  0.70 33.07           O
HETATM  591  O   BCME A  67      35.085 -12.194   8.323  0.30 31.96           O
```

B. Records for Residue 67

Schema 1 The records of ATOM type for residue 8 and modified residue 67 of the PDB file 2p3a

– After running this program, a PDB file will have residues with only the backbone atoms and only the first alternate location will be retained in case of multiple alternate locations. Output for residue 8 and modified residue 67 is shown in Schema 3.

Record id

Atom No

Alt Loc Ind

Res No

Cart. Coordinates

```
ATOM      72   N   AARG  A   8      26.288    -8.483    -5.941   0.60   23.05       N
ATOM      73   N   BARG  A   8      26.517    -8.547    -6.064   0.40   23.23       N
ATOM      74   CA  AARG  A   8      25.875    -8.023    -4.614   0.60   21.71       C
ATOM      75   CA  BARG  A   8      26.053    -8.180    -4.733   0.40   22.23       C
ATOM      76   C   AARG  A   8      26.929    -8.501    -3.624   0.60   21.10       C
ATOM      77   C   BARG  A   8      27.135    -8.490    -3.723   0.40   21.09       C
```

A. Records for Residue 8

```
HETATM   572   N   ACME  A  67      31.550   -12.012     8.379   0.70   29.54       N
HETATM   573   N   BCME  A  67      31.558   -11.938     8.292   0.30   29.11       N
HETATM   574   CA  ACME  A  67      32.726   -12.421     7.660   0.70   31.90       C
HETATM   575   CA  BCME  A  67      32.776   -12.485     7.653   0.30   30.89       C
HETATM   588   C   ACME  A  67      33.921   -12.522     8.646   0.70   32.23       C
HETATM   589   C   BCME  A  67      33.963   -12.613     8.623   0.30   31.53       C
```

B. Records for Residue 67

Schema 2 A sample output of "copyBackboneAtoms.pl" for residues 8 and 67

Record id

Atom No

Alt Loc Ind

Res No

Cart. Coordinates

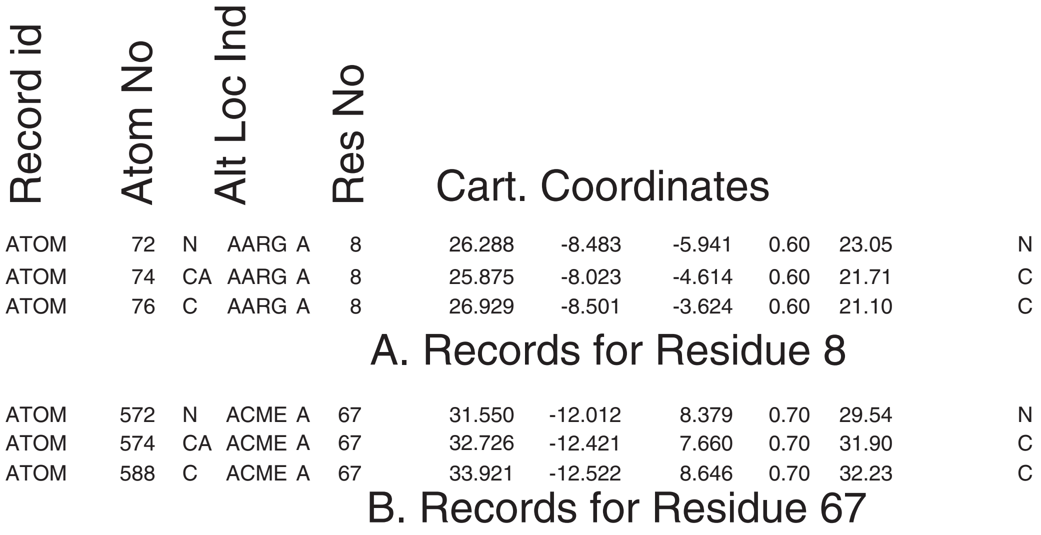
```
ATOM      72   N   AARG  A   8      26.288    -8.483    -5.941   0.60   23.05       N
ATOM      74   CA  AARG  A   8      25.875    -8.023    -4.614   0.60   21.71       C
ATOM      76   C   AARG  A   8      26.929    -8.501    -3.624   0.60   21.10       C
```

A. Records for Residue 8

```
HETATM   572   N   ACME  A  67      31.550   -12.012     8.379   0.70   29.54       N
HETATM   574   CA  ACME  A  67      32.726   -12.421     7.660   0.70   31.90       C
HETATM   588   C   ACME  A  67      33.921   -12.522     8.646   0.70   32.23       C
```

B. Records for Residue 67

Schema 3 The output of "retainFirstAtlLocation.pl" for residues 8 and 67

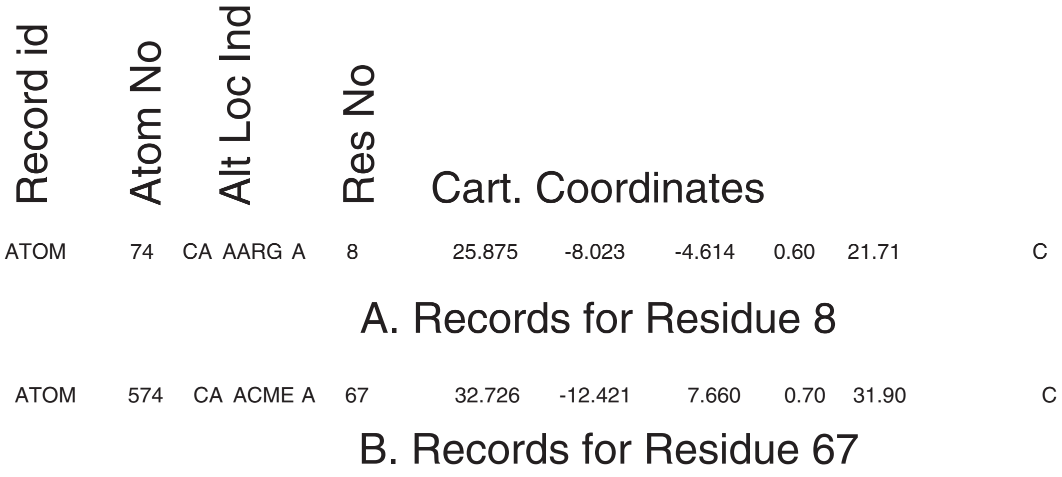

Record id
Atom No
Alt Loc Ind
Res No
Cart. Coordinates
A. Records for Residue 8
B. Records for Residue 67

```
ATOM    72  N   AARG A   8    26.288   -8.483   -5.941  0.60  23.05        N
ATOM    74  CA  AARG A   8    25.875   -8.023   -4.614  0.60  21.71        C
ATOM    76  C   AARG A   8    26.929   -8.501   -3.624  0.60  21.10        C

ATOM   572  N   ACME A  67    31.550  -12.012    8.379  0.70  29.54        N
ATOM   574  CA  ACME A  67    32.726  -12.421    7.660  0.70  31.90        C
ATOM   588  C   ACME A  67    33.921  -12.522    8.646  0.70  32.23        C
```

Schema 4 The output of "replaceHETATM.pl" for residues 8 and 67

- replaceHETATM.pl—This program replaces the keyword HETATM with the keyword ATOM in a set of PDB files.
  - perl replaceHETATM.pl
  - Output files after running this program will be saved in *data-backbone-singleAltLocation-NoHETATM* subfolder of the *experimentalDynamics* parent folder.
  - MUSTANG [34] removes all records with the keyword HETATM before multiple structural alignment. To prevent the removal of needed data, replaceHETATM.pl program replaces the keyword HETATM with ATOM so that MUSTANG will use the HETATM coordinates as required in the multiple structure alignment. Sample output for residue 8 and modified residue 67 is shown in Schema 4.
- retainCA.pl— This program copies the CA atoms from a set of PDB files.
  - perl retainCA.pl
  - Output files after running this program will be saved in *data-CA* subfolder of the *experimentalDynamics* parent folder.
  - This program retains the records of the Cα atoms only. Therefore, for each residue only the record for Cα will be copied. Also, the TER keyword to separate the chains will not be retained in the output file so that MUSTANG considers the whole structure (multiple chains) as one chain. A sample output for residue 8 and modified residue 67 is shown in Schema 5.



```
ATOM      74   CA  AARG  A   8        25.875    -8.023     -4.614    0.60   21.71            C
ATOM      574  CA  ACME  A   67       32.726    -12.421     7.660    0.70   31.90            C
```

Schema 5 The output of "retainCA.pl" for residues 8 and 67

4.1.2 Aligning PDB Structures Using MUSTANG

There are several successful multiple structural alignment programs such as MUSTANG [34] , TM-align [36], DaliLite [37], etc. A Wikipedia page has a list of multiple structure alignment software/programs (http://en.wikipedia.org/wiki/Structural_alignment_software, 10/15/2013). We have used MUSTANG for multiple structural alignments of the selected PDB structures. MUSTANG does not consider sequence information in its alignment algorithm. Rather, it performs a structural alignment by finding maximal similar substructures. Thus it can capture the conformational variations among the structures much better than the alignment algorithms that rely upon sequence similarity information. Moreover, in its alignment MUSTANG uses the Cα backbone atoms only. The running time MUSTANG 3.2.1 for the alignment of the selected dataset of 329 structures is approximately 5.00 h on a Linux machine with the following configuration—Linux version 2.6.18-348.4.1.el5 Intel(R) Xeon(R) CPU E5630 @2.53GHz. MUSTANG needs a Linux operating system. After installing MUSTANG under the *experimentalDynamics* folder on a Linux machine, save and copy the following file *description* in the same folder; and copy *data-CA* subfolder with the structures in the same folder as well:

- *data-CA*: This folder has all the backbone PDB files for multiple structural alignments.
- Description: This file has the path of the source directory where MUSTANG will find the input files for multiple structural alignment. After the path information, this file also has the list of the PDB file names that MUSTANG will read from the source directory. The list of the filenames in this file is in

the same order as the list of the PDB Ids in the pdbIds.txt file which has the 329 PDB Ids that are listed in Subheading 3.2. Update the line in *description* file that records the path of the source directory for the input files (path to the files in *data-CA* subfolder) that would be aligned.

Run the following command to execute MUSTANG:

– mustang-3.2.1 -f description -o alignAll -F fasta -r ON

This will create the following two files:

- alignAll.pdb: This file contains the aligned structures. Each chain corresponds to a specific PDB file and the header of this file lists the file names in the same order (*see* **Note 3**).
- alignAll.afasta: This contains the alignment of the amino acid sequences of the HIV-1 proteases based on the structural alignment.

4.1.3 Preparing Data for PCA from MUSTANG Output Files

- copyChainsToPDBs.pl—This perl script will copy each chain of alignAll.pdb file to the corresponding PDB file according to PDB Ids listed in the pdbIds.txt file.
  – perl copyChainsToPDBs.pl pdbIds.txt alignAll.pdb

This will create a subfolder *alignedPDBs* in the *experimentalDynamics* folder. This subfolder will have the 329 PDB files with the aligned Cα atoms of each structure. So when the Cartesian coordinates of each file will be placed in a matrix such that each row corresponds to the coordinates of one PDB Id, this matrix can be used for principal component analysis (*see* **Note 4**).

4.2 Use of Cartesian PCs to Extract Functional Dynamics from the Protein Structures

Matlab script experimentalDynamics.m reads the Cartesian coordinates of the structures from the MUSTANG aligned files and perform PCA on them.

4.2.1 Significance of Principal Components (PCs)

Figure 3 shows the distribution of the 329 PDBs Ids projected onto the space of the first few PCs from three separate views—PC1–PC2 (panel a), PC1–PC3 (panel b), and PC2–PC3 (panel c). In panels a and b, open and closed structures are clearly separated in two regions (open structures on the left side and closed structures on the right side) and the intermediate conformations (1aid, 3t11, 4ej8, etc.) spanning the middle region. The PC2–PC3 view in panel c, the structures are distributed based on conformational differences in the flap elbow region.

We used the MAVEN function *modeAnimator.m* to animate the motions of the structure along PC1, PC2, and PC3 vectors. The following code calculates the conformations along PC1 and can be found in *experimentalDynamics.m* matlab function:

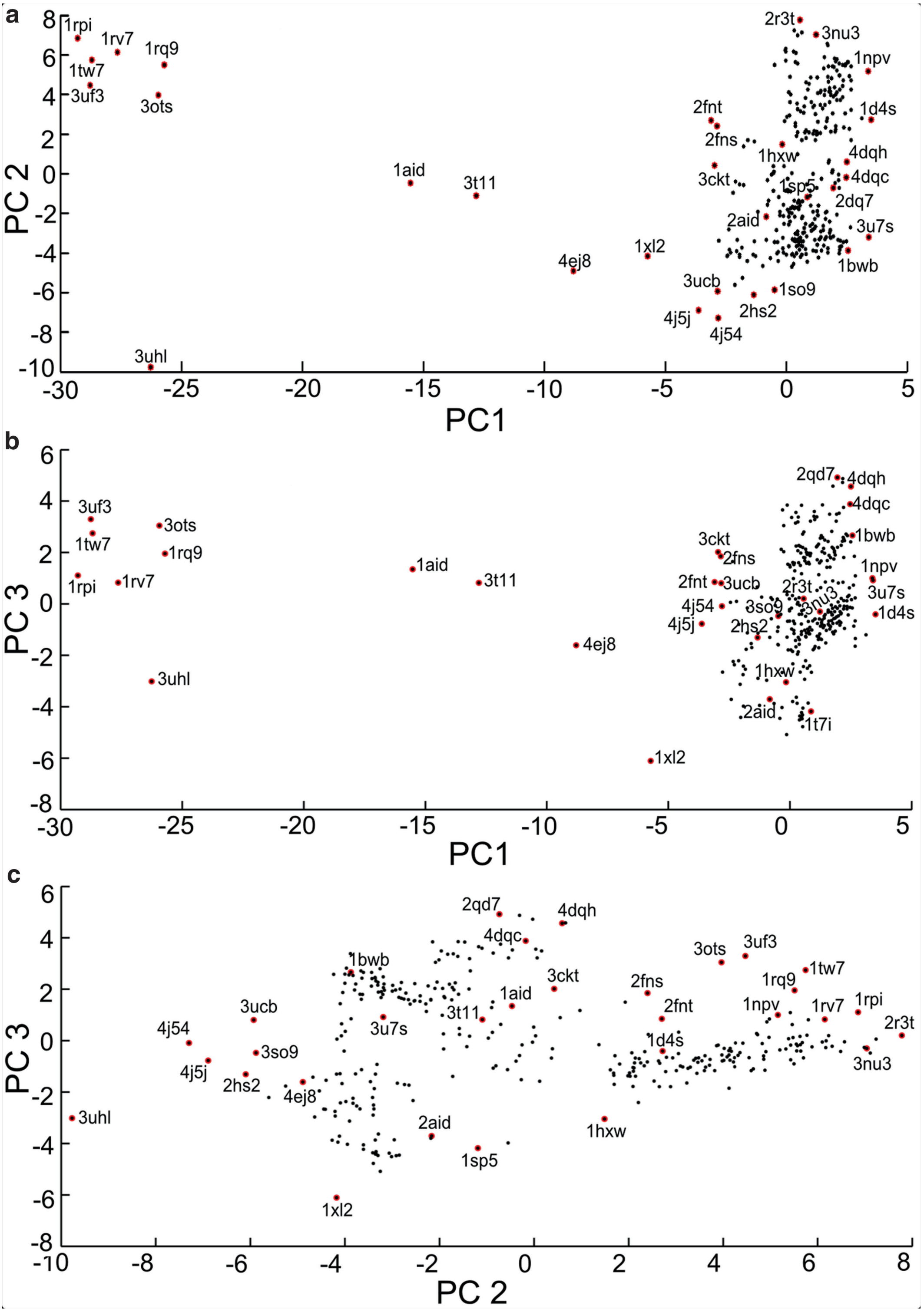


Fig. 3 Distributions of the 329 PDB structures by PCA. (**a**) Distribution of the structures on a PC1-PC2 plot. (**b**) Distribution of the structures on a PC1–PC3 plot. (**c**) Distribution of the structures on a PC2–PC3 plot. In plots **a** and **b**, open structures are located on the *left* side; closed structures are located on the *right* side; and the intermediate structures fall *in between*. Distribution of structures on PC2–PC3 plot (panel **c**) is based on primarily on the conformational differences along the flap elbow region. PC1, PC2, and PC3 capture 30 %, 20 %, and 7 % of the variances in the dataset, respectively

```
m = readPDB(ifname,1); %read the MUSTANG refer-
ence structures
c = sqrt(length(m.IND)/ sum(PC(:,1).^2));
%c controls the vector displacement amount
modeAnimator(m,PC(:,1),'',c,c/10,
ofname,'',0,'',1);
%use PC1 as the mode vector to simulate the
motion of the structure
```

The motion of the structure along PC1, PC2, and PC3 can be observed by opening the corresponding file using PyMOL visualization software. It is evident that, PC1 is closely related to the opening and closing (or expansion/contraction) of the flaps and the ligand binding cavity as shown in Fig. 4a. The two extreme ends of PC1 motion correspond closely to the closed (+) and the open (−) experimental structures (closed: PDB 1ebw, open: PDB 1rpi). The PC2 and PC3 correspond to twisting motions that are best seen in a perpendicular direction to those of PC1. PC2 is predominantly a twisting motion of the flap domains (panel C), whereas PC3 is predominantly a hinge motion of the core domains moving towards and away from the flaps (panel D).

experimentalDynamics.m also has code to visualize structures by using the ANM modes and the generated frames are saved in PDB file format that can be visualized using PyMOL software.

4.2.2 Comparing PC Based and ANM Computed Dynamics

Matlab program experimentalDynamics.m has the code to compute the ANM modes by using the MAVEN function ANM.m, and it then computes the overlap and the cumulative overlaps with the previously computed PCs by using another MAVEN function CompareDynamics.m. Figure 5, generated by MAVEN function plot_compareDynamics.m, shows the overlaps between the first ten PCs and the first ten ANM modes. The highest overlap is 60 % found between PC1 and ANM mode 3.

Table 2 shows the cumulative overlaps between PCs and the ANM modes. The cumulative overlap between each of the first and the second PCs and the first 20 modes is above 80 %. Interestingly, the cumulative overlap reaches 80 % between the second PC and the first six modes. This clearly indicates that given an appropriate experimental dataset the motions captured by the PCs conform quite closely with the ANM motions.

4.3 New Internal Distance Based ANM Motions

The use of structural information in ANM improves the modeling of the protein dynamics. Subheading 2.3 describes a way to derive spring constants from the structures. Here, we compute the inverse of the variance of the internal distances from the aligned structures in the MUSTANG aligned file *align.pdb* by using the Python program *internal.py*. The calculated inverse values are stored in hiv.329.var.sc file that could be downloaded at the link in the footnote of Table 1. MAVEN function ANM.m can be modified to use

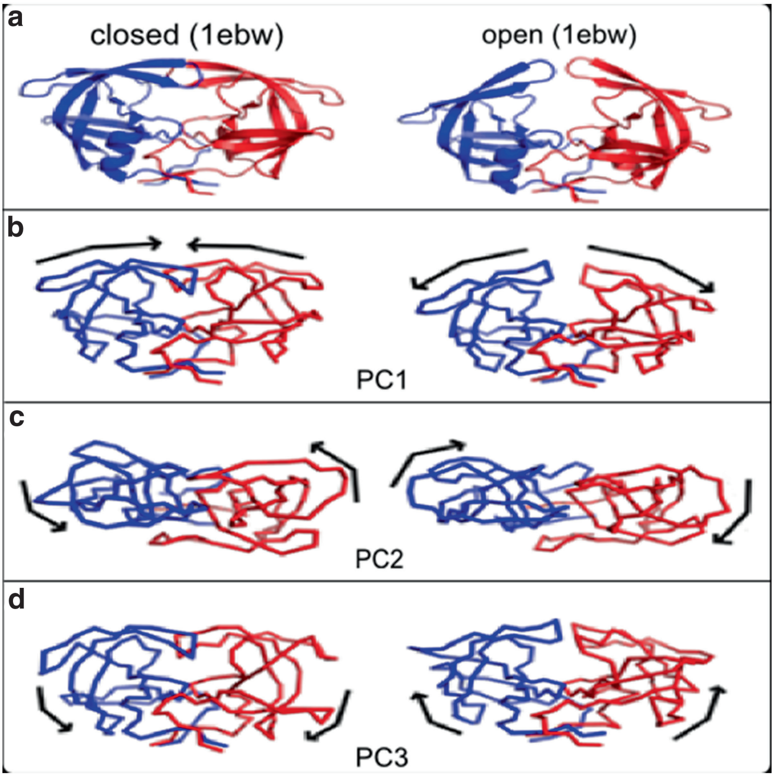


Fig. 4 Visualization of the first three PCs of HIV-1 protease on the structures. (**a**) Structures showing the closed form (*left*, PDB 1ebw) and open form (right, PDB 1rpi) of HIV-1 protease. The two subunits are shown in *red* and *blue* color and in *ribbon diagram*. (**b**) Snapshots of the structures displaced along the directions of PC1 shown in connected line segment. The direction of motions of the protein along each PC is shown with a *black arrow*. It can be seen that the opening-closing motion of the flaps can be easily identified from the extrema of PC1. Two extrema are shown for each motion in each row, together with *arrows* that indicate the directions for transition to the other structure. (**c**) PC2 images are shown looking down from the top of those in PC1 and PC3. PC2 is a twisting of the flap regions whereas (**d**) PC3 is a hinge motion between the core and flaps, with the core and flaps moving to and fro relative to one another

these values as the spring constants to compute the normal modes. Table 3 shows the overlaps of PCs based on these internal distances and the new ANM modes. The highest overlap is 79 % that occurs between PC1 and mode 2, which is much higher than the highest overlap (60 %) that occurred between PCs and conventional ANM modes (Fig. 5).

Table 4 shows the cumulative overlap between PCs and the new ANM modes. We can see that cumulative overlap between PC1 and the first three modes reaches 90 % which is quite high compared to the cumulative overlap between PC1 and the first three modes (62 % as shown in Table 2). However, the cumulative

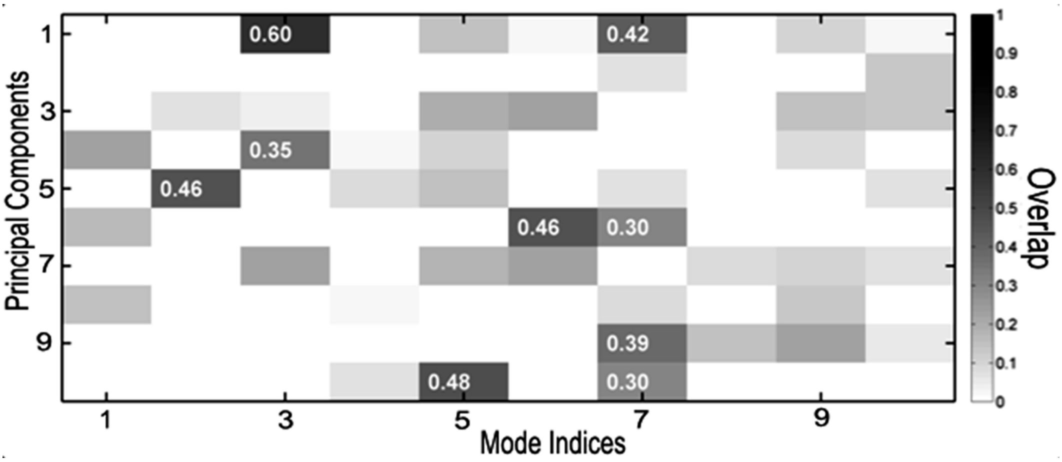


Fig. 5 Overlap between PCs and ANM modes. PC1 and mode 3 gives the highest overlap 60 %

Table 2
Cumulative overlap between the first three PCs and sets of the ANM modes

ANM modes/PCs	PC1	PC2	PC3
3 modes	0.62	0.71	0.44
6 modes	0.64	**0.80**	0.54
10 modes	0.77	**0.83**	0.59
20 modes	**0.80**	**0.85**	0.65

CO between a PC and ANM modes is shown in *bold type* if it is greater than 0.80

Table 3
Overlaps between PCs and the new ANM modes

PCs/newANM modes	Mode 1	Mode 2	Mode 3
PC1	0.09	0.79	0.40
PC2	0.34	0.01	0.24
PC3	0.34	0.01	0.10

Table 4
Cumulative overlaps between PCs and the new ANM modes

New ANM modes/PCs	PC1	PC2	PC3
3 modes	**0.90**	0.42	0.35
6 modes	**0.91**	0.44	0.41
20 modes	**0.95**	**0.89**	**0.84**

Values in *bold* indicate cumulative overlaps above 80 %

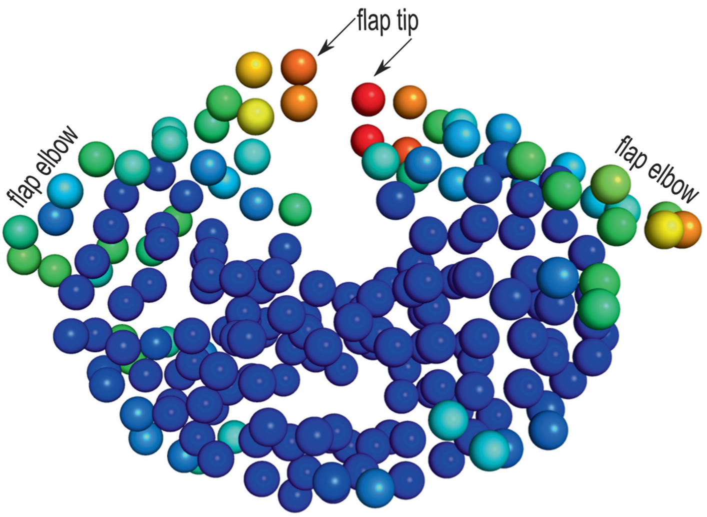


Fig. 6 Depiction of entropies of HIV-1 protease structure (PDB 1rpi) computed from PCs. Residues are colored spectrally according to the entropy values—coloring from *red* for the highest entropy to *blue* for the lowest entropy. Some of the residues along the flap and flap elbow regions on the subunit A (*right* subunit) have higher entropies than the same residues on subunit B (*left* subunit).

overlap between PC2 and the first three modified ANM 42 %; on the other hand this value between PC2 and the first three conventional ANM modes is 71 %, a much higher value. Therefore, in some cases cumulative overlap between a PC and the new ANM modes gets improved compared to the similar values between a PC and the conventional ANM modes. But when 20 new ANM modes are included, the values are constantly higher.

Taken together, this suggests that modified ANM can improve the performance of the ANM models.

4.4 Computing Entropy Using PCs

We compute the entropy of the HIV-1 protease system using Eq. 7 described in Subheading 2.5. By using calc_Entropy_PC.m matlab program, we compute the entropy from the principal components of the 329 aligned HIV-1 protease structures. The residues of HIV-1 protease are colored in Fig. 6 according to the entropy values. It is clear from the figure that the entropies are asymmetrically distributed in the two HIV-1 protease subunits. Subunit A (right subunit) has higher entropies along the flap and flab elbow regions.

5 Conclusion

This chapter gives the background of two important methods—PCA and ENM. By following the steps with the set of 329 HIV-1 PDB structures, one can get a hands-on experience on how to apply PCA to extract dynamics and mechanism information by capturing the conformational variability buried in different PDB structures of the same protein. One can also learn how to model the functionally important global motions of the protein using the widely accepted ANM model and compare the dynamics and mechanism found from experimental structures by PCA and from the ANM model. The higher overlaps between PCs and modified ENM modes indicate that a rich dataset of protein structures can play an important role in understanding functional dynamics and mechanism of the protein.

Moreover, the PC's represent the variability apparent within the sets of structures, and hence these are used as a direct measure of the conformational entropy of the protein structure.

This approach can also be extended to other highly diverse protein structure sets. The PDB database continues to grow rapidly—in 2008 there were ~43,000 protein structures and now in 2013 there are more than 90,000 structures [1]. In the future if new technologies for X-ray structure determination are developed that are much more efficient and very rapid, then there will be truly abundant structures of related proteins, including aberrant protein structures from patients. Among the various structures there are many single proteins with multiple X-ray structures determined under different conditions, as well as NMR structures. Generally proteins are robust and not easily disturbed by different environments or mutations; and the preponderance of evidence suggests that proteins have a limited range of conformations that are essential for their function. Therefore, the approach described here can generally be used to extract dynamics of any protein with significant numbers of available experimental structures.

6 Notes

1. *Selecting a set of structures*: There are 564 HIV-1 X-ray structures in PDB (07/26/2013). Among them, 329 PDB structures are selected so that the MUSTANG structural alignment does not produce any gaps in the corresponding aligned sequences. If a different set of structures is selected that produces gaps after multiple structural alignment, the residues in a structure that fall along the gaps on the alignment need to be removed before the PCA calculation.
2. *Construction of the selected dataset*: It is important to select a dataset that represents the whole conformational landscape of a protein structure. In panels A and B of Fig. 3, the open and closed structures are clustered on the left and the right side, respectively, and the intermediate conformations (1aid, 3t11, 4ej8, etc) span the middle region. Though the number of

closed structures is much higher than the number of open and intermediate structures, this dataset is a good selection as it has representation from whole conformational landscape.

3. *Caution in the use of MUSTANG*: MUSTANG output file align.pdb is found to break lines in some structures. Therefore, once the align.pdb is generated from MUSTANG, it needs to be normally scanned to detect and fix such broken lines.

4. *PCA on all the backbone ATOMs*: *data-backbone-singleAltLocation-NoHETATM* subfolder in the *experimentalDynamics* folder has the structures with all backbone atoms. These structures can, as well, be used for MUSTANG alignment and then subsequent PCA and other related operations.

   PCA can also be done on all atoms of each structure. In that case, first, the structures need to process to keep the same atoms for each residue in all structures and then use MUSTANG to align the structures. Afterwards, the Cartesian coordinates of all structures need to be extracted and perform PCA on them. For this, "noOfAtoms" variable in *experimentalDynamics.m* need to be initialized accordingly.

Acknowledgments

We gratefully acknowledge the support provided by NIH Grant R01GM072014 and NSF Grant MCB-1021785.

We used several Matlab functions from MAVEN by Zimmerman et al. [12] as mentioned in Subheading 4. MAVEN is also useful to compute PCs, PC-plot, ANM modes, overlap between PCs and ANM modes.

References

1. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The protein data bank. Nucleic Acids Res 28(1):235–242, PMCID:PMC102472
2. Hotelling H (1993) Analysis of a complex of statistical variables into principal components. J Educ Psychol 24:417–441
3. Manly B (1986) Multivariate statistics—a primer. Chapman & Hall, Boca Raton
4. Pearson K (1901) On lines and planes of closest fit to systems of points in space. Philos Mag 2(6):559–572
5. Amadei A, Linssen AB, Berendsen HJ (1993) Essential dynamics of proteins. Proteins 17:412–425
6. Amadei A, Linssen AB, de Groot BL, van Aalten DM, Berendsen HJ (1996) An efficient method for sampling the essential subspace of proteins. J Biomol Struct Dyn 13:615–625
7. Teodoro ML, Philips GN Jr, Kavraki LE (2002) A dimensionality reduction approach to modeling protein flexibility. J Comput Biol 10:299–308
8. Teodoro ML, Philips GN Jr, Kavraki LE (2003) Understanding protein flexibility through dimensionality reduction. J Comput Biol 10:617–634
9. Howe PW (2001) Principal components analysis of protein structure ensembles calculated using NMR data. J Biomol NMR 20:61–70
10. Yang L, Song G, Carriquiry A, Jernigan RL (2008) Close correspondence between the motions from principal component analysis of multiple HIV-1 protease structures and elastic

network modes. Structure 16:321–330, PMCID:PMC2350220

11. Yang LW, Eyal E, Bahar I, Kitao A (2009) Principal component analysis of native ensembles of biomolecular structures (PCA_NEST): insights into functional dynamics. Bioinformatics 25:606–614, PMCID:PMC2647834
12. Zimmermann MT, Kloczkowski A, Jernigan RL (2011) MAVENs: motion analysis and visualization of elastic networks and structural ensembles. BMC Bioinformatics 12:264, PMCID:PMC3213244
13. Bakan A, Meireles LM, Bahar I (2011) ProDy: protein dynamics inferred from theory and experiments. Bioinformatics 27:1575–1577, PMCID:PMC3102222
14. Grant BJ, Rodrigues AP, ElSawy KM, McCammon JA, Caves LS (2006) Bio3d: an R package for the comparative analysis of protein structures. Bioinformatics 22:2695–2696
15. Atilgan AR, Durell SR, Jernigan RL, Demirel MC, Keskin O, Bahar I (2001) Anisotropy of fluctuation dynamics of proteins with an elastic network model. Biophys J 80:505–515, PMCID:PMC1301252
16. Bahar I, Jernigan RL (1994) Cooperative structural transitions induced by non-homogeneous intramolecular interactions in compact globular proteins. Biophys J 66:467–481
17. Bahar I, Atilgan AR, Erman B (1997) Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. Fold Des 2:173–181
18. Bahar I, Erman B, Haliloglu T, Jernigan RL (1997) Efficient characterization of collective motions and interresidue correlations in proteins by low-resolution simulations. Biochemistry 36:13512–13523
19. Bahar I, Jernigan RL (1997) Inter-residue potentials in globular proteins and the dominance of highly specific hydrophilic interactions at close separation. J Mol Biol 266:195–214
20. Bahar I, Rader AJ (2005) Coarse-grained normal mode analysis in structural biology. Curr Opin Struct Biol 15:586–592, PMCID:PMC1482533
21. Chennubhotla C, Rader AJ, Yang LW, Bahar I (2005) Elastic network models for understanding biomolecular machinery: from enzymes to supramolecular assemblies. Phys Biol 2:S173–S180
22. Jernigan RL, Yang L, Song G, Doruker P (2008) Elastic network models of coarse-grained proteins are effective for studying the structural control exerted over their dynamics. In: Voth G (ed) Coarse-graining of condensed phase and biomolecular systems. Taylor and Francis, Boca Raton, pp 237–254
23. Bahar I (2010) On the functional significance of soft modes predicted by coarse-grained models for membrane proteins. J Gen Physiol 135:563–573, PMCID:PMC2888054
24. Ichiye T, Karplus M (1987) Anisotropy and anharmonicity of atomic fluctuations in proteins: analysis of a molecular dynamics simulation. Proteins 2:236–259
25. Kuriyan J, Petsko GA, Levy RM, Karplus M (1986) Effect of anisotropy and anharmonicity on protein crystallographic refinement. An evaluation by molecular dynamics. J Mol Biol 190:227–254
26. Abdi H, Williams LJ (2010) Principal component analysis. WIREs Comput Stat 2:433–459
27. Tama F, Sanejouand YH (2001) Conformational change of proteins arising from normal mode calculations. Protein Eng 14:1–6
28. Yang L, Song G, Jernigan RL (2007) How well can we understand large-scale protein motions using normal modes of elastic network models? Biophys J 93:920–929, PMCID:PMC1913142
29. Andricioaei I, Karplus M (2001) On the calculation of entropy from covariance matrices of the atomic fluctuations. J Chem Phys 115:6289–6292
30. Harte WE Jr, Swaminathan S, Mansuri MM, Martin JC, Rosenberg IE, Beveridge DL (1990) Domain communication in the dynamical structure of human immunodeficiency virus 1 protease. Proc Natl Acad Sci U S A 87:8864–8868, PMCID:PMC55060
31. Hornak V, Okur A, Rizzo RC, Simmerling C (2006) HIV-1 protease flaps spontaneously open and reclose in molecular dynamics simulations. Proc Natl Acad Sci U S A 103:915–920, PMCID:PMC1347991
32. Larry Wall (2011) Perl 5. Version 5.12.4
33. Matlab Version 7.11.0.584 (2010) The MathWorks Inc., Natick, Massachusetts
34. Konagurthu AS, Whisstock JC, Stuckey PJ, Lesk AM (2006) MUSTANG: a multiple structural alignment algorithm. Proteins 64:559–574
35. The PyMOL Molecular Graphics System Version 1.4. (2012) Schrödinger, LLC
36. Zhang Y, Skolnick J (2005) TM-align: a protein structure alignment algorithm based on the TM-score. Nucleic Acids Res 33:2302–2309, PMCID:PMC1084323
37. Holm L, Sander C (1996) Mapping the protein universe. Science 273:595–603

Chapter 11

Computing Ensembles of Transitions with Molecular Dynamics Simulations

Juan R. Perilla and Thomas B. Woolf

Abstract

A molecular understanding of conformational change is important for connecting structure and function. Without the ability to sample on the meaningful large-scale conformational changes, the ability to infer biological function and to understand the effect of mutations and changes in environment is not possible. Our Dynamic Importance Sampling method (DIMS), part of the CHARMM simulation package, is a method that enables sampling over ensembles of transition intermediates. This chapter outlines the context for the method and the usage within the program.

Key words Conformational transition, Sampling intermediates, Relative free energy, Statistical mechanics of proteins, Structure–function

1 Introduction

Starting from a seminal paper by McCammon et al. [1], the field of protein molecular dynamics has evolved rapidly. This reflects the interest of a broad section of the biophysics community in understanding how a detailed X-ray or NMR structure can be connected to measured function [2]. The force-fields that are integrated on the computer to enable the sampling of motions have also improved dramatically along with the hardware that is enabling more and more complex systems of growing size and timescales to be explored on a detailed level with the use of computers. Probably the most extensive tests of force-fields and simulation times have come from the DE Shaw group and their recent sets of microsecond to millisecond simulations of BPTI and of a WW-domain [3]. In this case the computations were able to reach conformational substates that were not obvious from the X-ray structures, but that are fully consistent with the available experimental information. The exciting news from their results is that the force-fields currently in use seem to be capable of generating insights on much longer time-scales than had ever before been attempted.

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215, DOI 10.1007/978-1-4939-1465-4_11, © Springer Science+Business Media New York 2015

The less exciting news is that reaching these time scales, even with supercomputer access, is challenging for larger systems and for complex conformational change. Thus there remains a need to develop methods to enhance sampling of rare events that are poorly sampled in a general molecular dynamics trajectory.

We suggest that understanding large conformational change is the next real frontier for the molecular dynamics community. The finding that fluctuations from the X-ray or NMR starting points agree with experiment is now pretty well agreed by most researchers. The DE Shaw results thus further confirm and extend the validity of the molecular dynamics method. However, this validation of the method still leaves many questions for how to best apply it to a given biophysical question. If we want to know how a channel gates or how a kinase switches states, then it is not obvious that the best solution is the DE Shaw one of simply running the calculations for a very, very long time. Phrased another way, the number of interesting biophysical questions far outnumbers the available computer cycles. So, we need to find ways to efficiently use the computer resources that we do have access to and to most confidently sample on the interesting biological changes. Phrased another way, to reach these important biologically inspired questions demands a new level of computation that will not be readily met by even the most advanced computer systems. In general the initial excitement about the molecular dynamics method as a *panacea* then led to disappointments and is now in a slowly growing re-enthusiasm as the computer speeds have reached a stage where more elaborate calculations with greater statistical accuracy can be performed. This is also reflected in the recent Nobel Prize awards for the development of the approach.

1.1 Why Transitions Are Important

A key to understanding protein function is a detailed molecular description of the different conformations reached by each protein [4]. Examples of these states are protein folding transitions, protein activation by the exposure of an otherwise buried catalytic site, the opening/closing of channels to allow the passage of ions though the cellular membrane, auto-inhibition, and conformational changes that induce dimerization of monomeric units in order to activate signaling pathways as in human epidermal growth factor receptor(*see* Fig. 1). The motions involved in these changes range from hinge, shear, or rotation of subunits to the arrangement of amino acid side chains [7].

The density of conformational states may be dependent on physiological conditions and on the presence/absence of ligands that stabilize a particular conformation. Thus this ability to sample different conformations reflects the ever changing environment that cells and proteins are exposed to. Understanding at the atomic level the conformational changes and, the motions involved in them, provides insights into the nontrivial knowledge of the

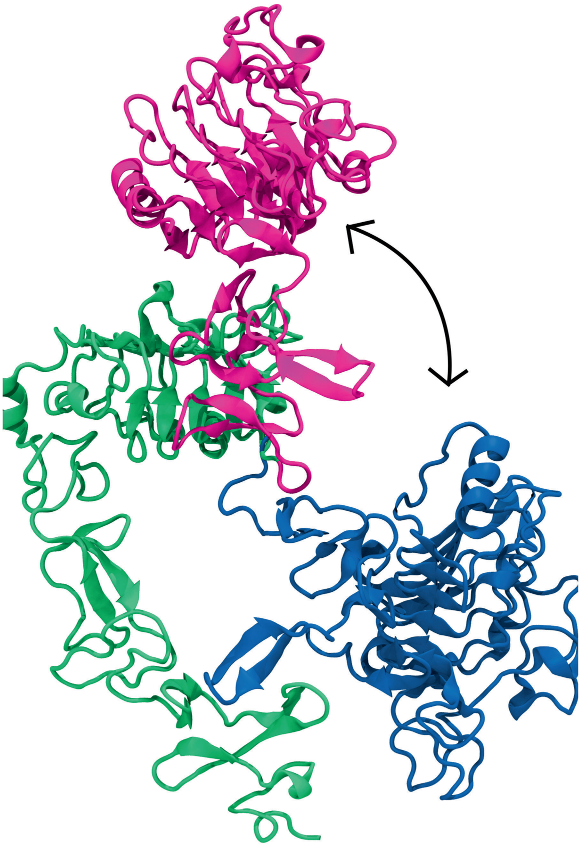


Fig. 1 Human epidermal growth factor receptor (HER3) exhibits a large conformational change that prevents exposure of the dimerization arm (*blue* and *pink*) by interactions between domains I and IV (*blue* and *green*) [5, 6]

biological function associated with them. In addition, by sampling fully on the transitions the ability to make a thermodynamic connection to experiment is put on a stronger footing. If the transitions in addition to the stable states are understood, then all the kinetics and the relative free energies between the states becomes a computed outcome to be compared with experiment.

1.2 Simulation Techniques

The basic idea of molecular dynamics is the integration of a coupled set of differential equations. By specifying the initial positions (*xyz*-space) through the use of a structure determined by X-ray or NMR methods, and by drawing an initial assignment of velocities to be consistent with a thermodynamic temperature, the equations of motion in classical space are well defined. The problems with the methods currently used are largely due to limitations in the time-scale of sampling and the relative accuracy of the potential functions. The potential function defines the forces that are used in the $\vec{F} = m\,\vec{a} = -\nabla\phi(\vec{x})$ that underlies the solution of the coupled set of equations. By pushing the boundaries of the largest systems that can be explored, by understanding the limitations of the force-fields, the community moves along towards a better set of confidence bars on when the simulations can be trusted.

In order to study the rare events associated with transitions, several methods have been proposed. Probably the earliest is the use of minimum energy, adiabatic, pathways. The work of Fisher [8, 9], and Elber et al. [10] may serve as examples of this research direction [11–16]. In their approach a starting point is first minimized to a low gradient RMS on the force and then minimum energy pathways are explored to move from one conformation to another. This has been extended with some attempts to understand the effects of temperature on the pathways [17]. Similar ideas have also been proposed for a type of low-energy Brownian walk to find a minimum energy path [18]. Modifications to this basic idea have been explored by many groups. For example work by Paci and Karplus has examined the use of experimental restraints to try and define the intermediates [19]. Perhaps the most currently well-known example is the transition path sampling methods developed by the Chandler group [20].

In transition path sampling, based on ideas developed by Pratt [21] and on work of Pratt and Chandler [22], an initial path estimate is made to connect the beginning and ending states. From this initial path a series of possible candidate improved paths are sampled in the Monte Carlo space generated by random moves from the initial path. The method has been explored and expanded from its initial foundations by the Bolhuis [23] and the Dellago groups [20]. In particular there has been considerable progress in identifying the class of perturbations from the original candidate path that will most likely lead to the starting and ending points rather than diverging.

Alternative pathway finding algorithms have worked with ideas developed by the Elber group [10]. In general these approaches use a transform of the problem to the space of finding spatial coordinates that fit the discretized space. Examples of these are the MaxFlux formalism [24], the string methods developed by the Vanden-Eijnden group [25, 26], the Onsager–Machlup formalism of Eastman and Doniach [27, 28] and the elastic rubber band approach first proposed by Jónsson et al [29–33]. The class of mathematical optimization that is used depends on solutions to diffusion equations and asks for the most optimal arrangement of conformational steps along a fixed set of time points. In this way the problem is transformed from one of dynamics to one of sampling on intermediate conformations that sample well on the underlying energy surface. A problem with this approach is that entropy contributions to the free energy of the conformational change are not well sampled. Recent work from the Elber group has developed a related, but new class of methods based on Milestoning [34]. A problem with this method is that a set of intermediate points needs to be defined for the transition to proceed: this relates it back to transition path sampling in that a small number of key intermediate points act as assumptions for where the transitions need to be occurring and act to systematically steer the sampling. This can be very good if the intermediates are well chosen and systematically bad if the intermediates are poorly selected.

Another class of methods has been based on biased selection from the dynamics. This has been implemented in different ways, and a survey of some of the approaches and their relative advantages and disadvantages was assessed by the Post group [35]. For example, in biased molecular dynamics [19, 36] a one-sided potential is added to a standard MD simulation. This bias favors moves towards a target structure and penalizes moves away and so attempts to gently perturb the dynamics towards a transition. This is somewhat similar to meta-dynamics [37] in that the potential is adjusted during the simulation and that it aims at creating a flat distribution of states between the initial and final points. A related concept is that of weighted ensemble Brownian (WEB) dynamics [38] where a set of equally weighted conformations is iterated forward in time to create a set of Brownian walkers that are eventually evenly distributed over the barrier crossing space. One advantage of the WEB dynamics is that the relative probability of a particular transition path is determined during the trajectory production. The Zuckerman group [39] has shown that this method can produce high quality transitions in their test systems.

Steered and targeted methods have been employed quite frequently for sampling on large conformational change. Both of these methods have problems with providing an unbiased estimate of change. Steered methods will have a large systematic error, depending on their choice of pulled coordinates and the rate of pulling. In principle, though rarely in practice, the non-equilibrium work based method can be used to correct for a calculation of the free energy change with steered molecular dynamics, but the distributions required to get convergence within the steered simulations have not been examined carefully. Plastic or targeted molecular dynamics will generate a single transition depending on the strength of the additional forces and so would need to be run with a systematic variation of force constants and sampled starting velocities to generate a range of transitions [40]. As the method is generally used it provides an estimate of conformational change and is more likely to create a systematic bias than a genuine sample of intermediate states, due to its use of root mean square (RMS) as a force to connect the starting with the ending states [41]. An extreme version of both the usefulness of the transitions and the difficulty in assessing their truthfulness is the Database of Macromolecular Movements web pages maintained by the Gerstein group. This has the advantage of presenting a large number of conformational changes for many protein systems, but the disadvantage of not being able to provide any strong metric for the confidence with which the conformational change has been presented [42].

1.3 Importance Sampling and SDE

Biased sampling with or without correction has become an important method in the molecular dynamics community. This can be seen as starting from the importance sampling for Monte Carlo methods that started with the Manhattan project in the 1940s. In that method the distribution of events that are collected is focused by a known *umbrella* or *bias* that forces more events to be sampled within the biasing region. Since the biasing function is known, the unbiased distribution can be recovered by dividing out. In effect this is simply multiplying and dividing by the same number (i.e., by one). This use of importance sampling and its improvement in accuracy for estimating ensemble averaged events was instrumental in modeling the diffusion of neutrons and in the whole design of the atomic bomb. As the method was developed its application to statistical mechanical problems seemed natural, it was widely employed for ideal gases and other homogeneous mixes. With the development of molecular dynamics, several biasing methods, some with correction and some without have been developed. For example, there is a strong community that uses the steered molecular dynamics methods of the Schulten group. These can enable a new force to be added into the simulation that pulls on particular atoms or groups of atoms and thus enhances sampling of events that are tied into the pulled direction. While it is easy to see the utility of being able to steer the simulation, it is difficult to connect the resulting trajectories into experimental measurements. Thus, other methods have worked strongly on how best to combine information from multiple biased *windows* to come up with an unbiased estimate.

Probably the most well known of these approaches is the weighted histogram analysis method (WHAM) developed by Swendson and coworkers [43]. In this approach the question is phrased as how to optimally estimate the unbiased free energy surface, given samples from multiple biased umbrellas distributed along a one or two dimensional reaction coordinate surface. The extension of this idea by Shirts and colleagues (MBAR) suggests that the probability arguments for optimally combining information will continue to be an important and fruitful topic for extending the sampling of protein motions and their free energy surfaces [44, 45].

Dynamic Importance Sampling (DIMS) uses another type of importance sampling, with correction, the difference being in that the corrections are applied to the drawn events as they happen, by comparing their relative likelihood from the biasing distribution to that inferred likelihood from the unbiased distribution [46–51]. The method was developed based on ideas from stochastic differential equations and their simulations with variance reduction methods [52]. This enables the DIMS method to sample on large conformational change without having to define a low dimensional reaction coordinate for the conformational change.

An intriguing suggestion is that the apo forms of a protein will have a free energy surface that is related to the holo forms [53, 54]. That is important, because the computational methodology that is pursued by DIMS requires that the same set of atoms be present at the beginning and the end of the simulation. Thus, by focusing the computational power on the apo states and the protein conformational changes, without the ligand, much can be learned about the behavior of the protein. Nonetheless, the nature of the changes from the apo to the holo state is not clearly defined and thus the framework of the population shift model should be seen as a starting point for a more detailed analysis of the role of the ligand in conformational change. In particular, the nature of the transition state in the absence and the presence of the ligand is an interesting question.

2 Transition States and Order Parameters

Chandler has provided the most commonly cited definition for the transition state. This can be understood as a region where a set of random walkers will be equally likely to fall back towards the starting point or to move along to the ending point. Finding and describing this surface for a protein system is still in its infancy, with the concept being reasonable, but no well-defined algorithm that all agree will work for all systems to define and to prove that the true transition state has been found. This means that to prove that an intermediate is at the transition state, or even part of the pathway for a large conformational change will, in principle, require an even larger set of calculations to prove that the committer probabilities are consistent with the expectations for the likelihood of a transition moving in each direction along all candidate pathways. Thus there is a need for methods that can improve on the sampling of intermediate states and that can provide connections to the unbiased estimate of the probability of reaching those intermediate states from the starting point.

The first application of these ideas, developed using the framework of statistical mechanics, but not yet applied in a simulation setting, was to the isomerization of butane. This simple model system is ideal at the level of transitions, since the one-dimensional reaction coordinate is easy to define and the sampling, even at the time of the Chandler paper, could be exhaustive. Chandler's work showed that a $6kT$ difference over the barrier from the minimum could be sampled and the transition state as the population dividing surface between these stable states could be analyzed. This opened the computational community up to the idea of how to define a transition state and how to determine rates over a free energy surface.

Recent work has shown that a poorly determined order parameter may lead to systematically poor results [55]. This is important for the work applying the effective transfer entropy formalism [6, 56], where multiple order parameters can be defined by the application of nonlinear time series analysis. In addition this relates to efforts to determine the kinetics from the transitions where both sampling and possible systematic errors will need to be considered [57]. This is illustrated in a simple conceptual surface where q is an order parameter along a complex multidimensional terrain. In one situation the transition state is fully sampled along the order parameter. In contrast, for another situation, the order parameter creates a systematic error in sampling along the transition surface, suggesting that the barrier is found about half-way along the order parameter, rather than realizing that the bottleneck is closer to the starting state in the projection of the dimensions. This problem can only be considered even more difficult for a multidimensional system with an even larger number of degrees of freedom that need to be averaged out in order to get good sampling.

3 Methods for Analysis of Functional Protein Modes

Principal component analysis (PCA) is a commonly used method to extend the molecular dynamics fluctuations sampled in a trajectory into a cohesive story for the most important sampled motions [58]. For instance, by performing PCA on 100 ns simulations of human growth factor (*see* Fig. 2) it is possible to identify collective motions that could be easily related to the conformational changes seen in the crystal structures (Fig. 1). Thus, PCA has become the *go-to* tool for this type of analysis, since there has been little else to use for this type of question and the formal analysis of the method has not been pushed to the same level as the force-fields and the long-time sampling [59]. But, as the time-scales for sampling has improved the ability to question whether the principal component analysis is valid and under what conditions has started to become a *front burner* issue. This is an important concern, because if the modes that are inferred from the trajectories can truly be used for longer-time analysis and extension, then the modes represent an important extension, by analysis, of the original trajectories. The current research, however, suggests that the method concentrates too much information on fluctuations from a single stable state, so that if there are complex motions between multiple stable states, there will be nonlinear mixing that will make the modes defined by the method less informative on what is happening on the molecular scale.

To infer more from the trajectories there are several alternatives. One is that a complete description of the relative free energy surface can be determined with sufficient sampling. This is an extension of ideas from Stillinger about the connections between adiabatic and *at temperature* simulations. In the complete, and most satisfying, form of the description, all the quasi-stable states would be clearly defined and the relative routes (pathways) between them would be cleanly described. In a sense this is the underlying concept behind the Markov analysis that is being pioneered by the Vande and IBM groups [60]. In the Markov analysis the long-time molecular dynamics trajectory is divided into basins of attraction and transitions between those basins. Ideally there would be more than sufficient sampling so that the effective modes defined by these stable

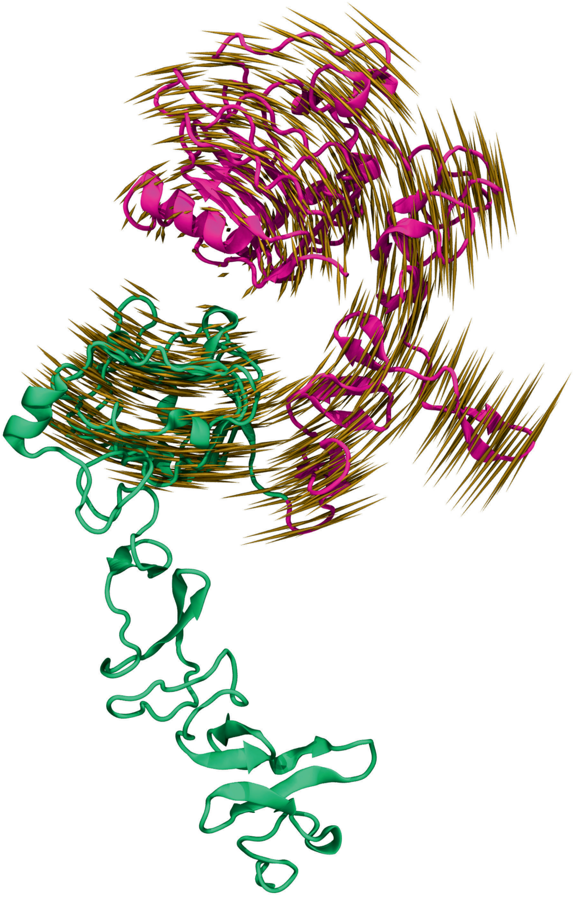


Fig. 2 Most dominant mode from PCA analysis for the human epidermal growth factor receptor (HER3). The direction of the motion is similar to the conformational change observed in crystal structures (Fig. 1)

states could be defined afterwards by understanding the relative contributions of each stable state. That would then, in principle, provide a mechanism for a comparison of the Markov model and the principal component model.

Some work has been proposed to follow the low frequency modes as a way to sample on the longer time scale events [46, 61–63]. In particular work on the low frequency modes for the K-channel has had mixed results [64], suggesting that the modes seen in the lowest energy starting state may not immediately lead to the appropriate transition pathways [65]. This has built on the ability to compute normal modes for larger and larger systems [66], an approach that we use to improve our own sampling of candidate motions for dynamic importance sampling [46]. For example, the Brooks group has shown that the method of normal mode following can be applied to a large range of sample calculations [67]. In a somewhat related way, elastic network models using simple harmonic springs between heavy atom groups have been used as a coarse-grained approach to enable some aspects of conformational change to be sampled [68–70]. These models have, however, been shown to be limited in their ability to sample on the all-atom pathways seen in more complex transitions, so they may have utility in providing a very fast look at a candidate transition, but will not enable a confident sampling on a range of intermediate states.

There has been intriguing work from the Thorpe group on the use of conformational restraints and accounting for the main degrees of freedom to try and determine pathways between states [71]. Even if the methods are approximate, the suggestion that some pathways can be determined by a rough accounting of domain motions and flexible links is potentially important.

An alternative to the complete enumeration is to ask if the model can be reduced by nonlinear methods [72–77]. This is the framework we explored for the Nitrogen receptor protein C [56] and the larger and more complex human epidermal growth factor receptor shown in Fig. 1 [6]. The approach consists of asking if a nonlinear reduction to a reduced description can help to predict the most important fluctuations from the starting point. Some precedents for this type of question are in the work of the Wriggers group and the use of mutual information from the Gilson and Grubmuller groups. Mathematically the precedents to this idea are from Granger causality analysis and transfer entropy ideas [78–81]. It is worth mentioning, that previous to our work, the only use of the effective transfer entropy in the setting of biomolecular simulations is from the van der Vaart group [82]. In their approaches they mainly focused on the use of the method to pick out key residues in the bound and free states of a DNA binding protein.

4 Implementation of Dynamic Importance Sampling (DIMS) in CHARMM

DIMS (*see* **Notes**) is implemented and available in CHARMM since the c35a1 release. It must be enabled before compilation by adding the DIMS flag in the “install.com” file. In order to support Normal Mode biasing the VIBBLOCK flag must be added as well. In addition, the soft-ratcheting implementation of DIMS offers support for the parallel version of CHARMM.

The basic work flow with DIMS is simple:

1. Load structures. The target must be stored in the DIMS coordinates set using COOR COPY DIMS.
2. Setup DIMS using the DIMS command.
3. Run dynamics.

4.1 Loading the Structures

Loading structures in CHARMM is performed via the READ COOR command. When using DIMS, the first structure that must be loaded is the target structure. Then it must be copied to the DIMS set using the COOR COPY TARG command. Once the DIMS set has been initialized the starting structure can be loaded.

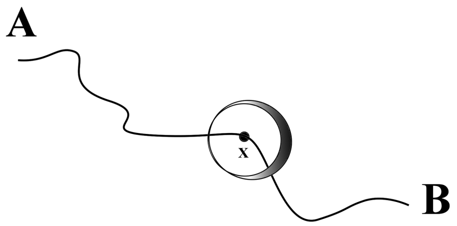


Fig. 3 If a trial molecular dynamics step is towards the target *B* then the motion is accepted. A motion away from the target is only accepted with a certain probability, and this probability decreases as the trial move is away from the target. The algorithm is similar to a Brownian ratchet system with the general direction of the random walk being towards the target [83]

```
! target configuration

 OPEN READ UNIT 1 CARD NAME target.crd

 READ COOR CARD UNIT 1

 CLOSE UNIT 1


 COOR COPY DIMS          ! target must be copied to DIMS set


 ! starting configuration

 OPEN READ UNIT 1 CARD NAME start.crd

 READ COOR CARD UNIT 1

 CLOSE UNIT 1
```

4.2 Setting Up DIMS

DIMS offers two types of biasing mechanism: soft-ratcheting and NM following. In the first case, DIMS uses a predefined progress variable. At every timestep (*see* Fig. 3), once a trial structure *x* has been generated by following Langevin's equation, DIMS determines whether the motion is towards or away from the target state. In case that the movement is towards the target *B* then the motion is accepted. If the trial move *x* moves away from the target, then it is only accepted with a certain probability controlled by the rejection constant φ [46]. In order to enable the soft-ratcheting mechanism, a DIMS line must be added to the CHARMM input file:

```
DIMS DCAR  1e-5          -   ! set up DCAR-DIMS

   orient 1000            -   ! re-orient at every 1000 step

   SELE mydims END     -   ! DIMSatom selection (biased atoms)

   SELE orient END        -   ! RMS-fit atom selection

   COFF 1.0 halt               ! stop biasing when target is 1.0 A from target
```

This example gently moves the system toward the target without a restriction on the total time. The rejection constant in this example is set to $\varphi = 1 \times 10^{-5}$, selection of the rejection constant is system dependent, and must be explored before production runs. If the barrier height is not high enough under some conditions this algorithm will not converge. When the barriers to conformation change are small this approach will converge with a better DIMS or Onsager–Machlup (OM) score. As DIMS uses by default RMSD as the progress parameter, it is generally required to align the two structures in order to eliminate rotations and translations. In the previous example the structures are aligned every 1,000 steps using the second atom selection for the alignment.

As previously mentioned, DIMS can also use a bias based on the Normal Modes from the initial structure, and recalculate them as the simulation progresses. Calculation of the modes is performed by using the block normal mode method available in CHARMM. A typical usage of the NM-biasing method is:

```
DIMS DBNM -
DSCALe 0.1 SKIP 500 BSKIP 50 NBIAs 27                          - ! dims options
SERL GENR SCAL 0.5882 TMEM 420 MEMO 20 MEMA 400 NMOD 30 - ! BNM options
COFF 2.0 HARD                                             - ! NM Hard Cutoff
ORIEnt 20                                                 - ! Orient every 1000th
SELE mydims END                                           - ! DIMS selection
COMB 3 NBES 15                  - ! Store 15 best modes and then group them
                                - ! by 1, 2 and 3 for each try.
NWINDow 12                      - ! Avoid normal modes previously used within
                                - ! a window of 12 modes
MTRA @I NMUNit 10               - !  @i trajectories per file, write output to unit 10
DSUNIT                          - !  Use unit 11 to store dims score
```

Normal modes are computationally expensive calculations as they require calculation and diagonalization of the Hessian matrix. Therefore, DIMS only recalculates the normal modes every *s* steps (500 in the example) and uses them for a given number of steps. Modes are scaled by a constant factor determined DSCAL. Similar to the soft-ratcheting, when RMSD is enabled, alignment of the structures is recommended. Additionally, in this example, self-avoidance is enabled for a window of 12 steps from the current move. Although all modes are evaluated, only the best 15 (NBES) are kept, and linear combinations of up to three modes are evaluated at every time-step using a previously defined progress variable (e.g., RMSD).

5 Notes

Trajectories generated by DIMS are completely independent [46], therefore ensembles of transitions can be generated by running multiple instances of CHARMM. Additionally, DIMS can also compute the Onsager–Machlup action functional for each trajectory as the simulation progresses, this calculation can be activated by adding the flag OMSC to the DYNA command [46]. The current implementation of DIMS is not limited to just RMSD as progress variable, it can also use interatomic distances, angles, and dihedrals, as well as, combinations of these in order to generate collective variables. This flexibility allows, for instance, the use of native contacts of the target structure as the progress variable.

References

1. McCammon JA, Gellin B, Karplus M (1977) Dynamics of folded proteins. Nature 267:585–590
2. Schlick T, Collepardo-Guevara R, Halvorsen LA, Jung S, Xiao X (2011) Biomolecular modeling and simulation: a field coming of age. Q Rev Biophys 1–38
3. Shaw DE et al (2010) Atomic-level characterization of the structural dynamics of proteins. Science 330:341–346
4. Creighton TE (1993) Proteins: structures and molecular properties. Macmillan, New York
5. Ferguson KM et al (2003) EGF activates its receptor by removing interactions that autoinhibit ectodomain dimerization. Mol Cell 11:507–517
6. Perilla JR, Leahy DJ, Woolf TB (2013) Molecular dynamics simulations of transitions for ECD epidermal growth factor receptors show key differences between human and drosophila forms of the receptors. Proteins 81:1113–1126
7. Gerstein M, Lesk AM, Chothia C (1994) Structural mechanisms for domain movements in proteins. Biochemistry 33:6739–6749
8. Fischer S (1992) Conjugate peak refinement: an algorithm for finding reaction paths and accurate transition states in systems with many degrees of freedom. Chem Phys Lett 194: 252–261
9. Gruia AD, Bondar A-N, Smith JC, Fischer S (2005) Mechanism of a molecular valve in the halorhodopsin chloride pump. Structure 13:617–627
10. Elber R, Karplus M (1987) A method for determining reaction paths in large molecules: application to myoglobin. Chem Phys Lett 139:375–380
11. Olender R, Elber R (1997) Yet another look at the steepest descent path. J Mol Struct 398:63–71
12. Czerminski R, Elber R (1990) Self-avoiding walk between 2 fixed-points as a tool to calculate reaction paths in large molecular-systems. Int J Quant Chem 24:167–186
13. Ulitsky A, Elber R (1990) A new technique to calculate steepest descent paths in flexible polyatomic systems. J Chem Phys 92: 1510–1511
14. Czerminski R, Elber R (1989) Reaction-path study of conformational transitions and helix formation in a tetrapeptide. Proc Natl Acad Sci U S A 86:6963–6967
15. Czerminski R, Elber R (1990) Reaction-path study of conformational transitions in flexible systems—applications to peptides. J Chem Phys 92:5580–5601
16. Choi C, Elber R (1991) Reaction-path study of helix formation in tetrapeptides—effect of sidechains. J Chem Phys 94:751–760
17. Elber R, Shalloway D (2000) Temperature dependent reaction coordinates. J Chem Phys 112:5539–5545
18. Berkowitz M, Morgan J, Mccammon J (1983) Generalized Langevin dynamics simulations with arbitrary time-dependent memory kernels. J Chem Phys 78:3256–3261
19. Paci E, Vendruscolo M, Dobson CM, Karplus M (2002) Determination of a transition state at atomic resolution from protein engineering data. J Mol Biol 324:151–163
20. Dellago C, Bolhuis PG, Csajka FS, Chandler D (1998) Transition path sampling and the calculation of rate constants. J Chem Phys 108: 1964–1977

21. Pratt LR (1986) A statistical method for identifying transition states in high dimensional problems. J Chem Phys 85:5045–5048
22. Chandler D, Pratt LR (1976) Statistical mechanics of chemical equilibria and intramolecular structures of nonrigid molecules in condensed phases. J Chem Phys 65: 2925–2940
23. Bolhuis PG, Chandler D (2000) Transition path sampling of cavitation between molecular scale solvophobic surfaces. J Chem Phys 113: 8154–8160
24. Huo S, Straub JE (1997) The MaxFlux algorithm for calculating variationally optimized reaction paths for conformational transitions in many body systems at finite temperature. J Chem Phys 107:5000–5006
25. Ren W, Eijnden EV, Maragakis P, Weinan E (2005) Transition pathways in complex systems: application of the finite-temperature string method to the alanine dipeptide. J Chem Phys 123:134109
26. Maragliano L, Fischer A, Vanden-Eijnden E, Ciccotti G (2006) String method in collective variables: minimum free energy paths and isocommittor surfaces. J Chem Phys 125:24106
27. Eastman P, Gronbech-Jensen N, Doniach S (2001) Simulation of protein folding by reaction path annealing. J Chem Phys 114: 3823–3841
28. Onsager L, Machlup S (1953) Fluctuations and irreversible processes. Phys Rev 91:1505–1512
29. Jónsson H, Mills G, Jacobsen KW (1998) Classical and quantum dynamics in condensed phase simulations. In Berne BJ, Coker DF. Proceedings of the International School of Physics. LERICI, Villa Marigola. pp. 385–404
30. Crehuet R, Field MJ (2003) A temperature-dependent nudged-elastic-band algorithm. J Chem Phys 118:9563–9571
31. Peters B, Heyden A, Bell A, Chakraborty A (2004) A growing string method for determining transition states: comparison to the nudged elastic band and string methods. J Chem Phys 120:7877–7886
32. Trygubenko S, Wales D (2004) A doubly nudged elastic band method for finding transition states. J Chem Phys 120:2082–2094
33. Mathews D, Case D (2006) Nudged elastic band calculation of minimal energy paths for the conformational change of a GG non-canonical pair. J Mol Biol 357:1683–1693
34. Kuczera K, Jas GS, Elber R (2009) Kinetics of helix unfolding: molecular dynamics simulations with milestoning. J Phys Chem A 113: 7461–7473
35. Huang H, Ozkirimli E, Post CB (2009) Comparison of three perturbation molecular dynamics methods for modeling conformational transitions. J Chem Theory Comput 5: 1304–1314
36. Marchi M, Ballone P (1999) Adiabatic bias molecular dynamics: a method to navigate the conformational space of complex molecular systems. J Chem Phys 110:3697–3702
37. Laio A, Parrinello M (2002) Escaping free-energy minima. Proc Natl Acad Sci U S A 99:12562–12566
38. Huber G (1996) Weighted-ensemble Brownian dynamics simulations for protein association reactions. Biophys J 70:97–110
39. Zhang BW, Jasnow D, Zuckerman DM (2007) Efficient and verified simulation of a path ensemble for conformational change in a united-residue model of calmodulin. Proc Natl Acad Sci U S A 104:18043–18048
40. Maragakis P, Karplus M (2005) Large amplitude conformational change in proteins explored with a plastic network model: Adenylate kinase. J Mol Biol 352:807–822
41. van der Vaart A, Karplus M (2005) Simulation of conformational transitions by the restricted perturbation-targeted molecular dynamics method. J Chem Phys 122:114903
42. Echols N, Milburn D, Gerstein M (2003) MolMovDB: analysis and visualization of conformational change and structural flexibility. Nucleic Acids Res 31:478–482
43. Kumar S, Rosenberg JM, Bouzida D, Swendsen RH, Kollman PA (1992) The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. J Comput Chem 13:1011–1021
44. Shirts MR, Chodera JD (2008) Statistically optimal analysis of samples from multiple equilibrium states. J Chem Phys 129:124105
45. Pohorille A, Jarzynski C, Chipot C (2010) Good practices in free-energy calculations. J Phys Chem B 114:10235–10253
46. Perilla JR, Beckstein O, Denning E, Woolf T (2011) Computing ensembles of transitions from stable states: dynamic importance sampling. J Comput Chem 32:196–209
47. Zuckerman DM, Woolf TB (1999) Dynamic reaction paths and rates through importance-sampled stochastic dynamics. J Chem Phys 111:9475–9484
48. Jang H, Woolf TB (2006) Multiple pathways in conformational transitions of the alanine dipep-

tide: an application of dynamic importance sampling. J Comput Chem 27:1136–1141

49. Zuckerman DM, Woolf TB (2002) Rapid determination of multiple reaction pathways in molecular systems: the soft-ratcheting algorithm. (eprint). arXiv: physics/0209098
50. Zuckerman DM, Woolf TB (2000) Efficient dynamic importance sampling of rare events in one dimension. Phys Rev E 63:016702
51. Woolf T (1998) Path corrected functionals of stochastic trajectories: towards relative free energy and reaction coordinate calculations. Chem Phys Lett 289:433–441
52. Wagner W (1987) Unbiased Monte Carlo evaluation of certain functional integrals. J Comput Phys 71:21–33
53. Swift RV, Mccammon AJ (2009) Substrate induced population shifts and stochastic gating in the PBCV-1 mRNA capping enzyme. J Am Chem Soc 131
54. Bahar I, Chennubhotla C, Tobi D (2007) Intrinsic dynamics of enzymes in the unbound state and relation to allosteric regulation. Curr Opin Struct Biol 17:633–640
55. Dickson BM, Makarov DE, Henkelman G (2009) Pitfalls of choosing an order parameter for rare event calculations. J Chem Phys 131:074108
56. Perilla JR, Woolf TB (2012) Towards the prediction of order parameters from molecular dynamics simulations in proteins. J Chem Phys 136(164101):164101
57. Xin Y, Doshi U, Hamelberg D (2010) Examining the limits of time reweighting and Kramers' rate theory to obtain correct kinetics from accelerated molecular dynamics. J Chem Phys 132:224101
58. García A (1992) Large-amplitude nonlinear motions in proteins. Phys Rev Lett 68: 2696–2699
59. Ma J (2005) Usefulness and limitations of normal mode analysis in modeling dynamics of biomolecular complexes. Structure 13: 373–380
60. Singhal N, Snow C, Pande V (2004) Using path sampling to build better Markovian state models: predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. J Chem Phys 121:415–425
61. Kitao A, Go N (1999) Investigating protein dynamics in collective coordinate space. Curr Opin Struct Biol 9:164–169
62. Petrone P, Pande V (2006) Can conformational change be described by only a few normal modes? Biophys J 90:1583–1593
63. Lange O, Grubmüller H (2006) Can principal components yield a dimension reduced description of protein dynamics on long time scales? J Phys Chem B 110:22842–22852
64. Denning EJ, Woolf TB (2010) Cooperative nature of gating transitions in K(+) channels as seen from dynamic importance sampling calculations. Proteins 78:1105–1119
65. Miloshevsky GV, Jordan PC (2007) Open-state conformation of the KcsA K+ channel: Monte Carlo normal mode following simulations. Structure 15:1654–1662
66. Florence TF, Xavier G, Osni M, Yves-Henri S (2000) Building-block approach for determining low-frequency normal modes of macromolecules. Proteins 41:1–7
67. Zhenggo W, Brooks BR (2005) Normal-modes-based prediction of protein conformational changes guided by distance constraints. Biophys J 88:3109–3117
68. Zheng W, Doniach S (2003) A comparative study of motor-protein motions by using a simple elastic-network model. Proc Natl Acad Sci U S A 100:13253–13258
69. Kim MK, Chirikjian GS, Jernigan RL (2002) Elastic models of conformational transitions in macromolecules. J Mol Graph Model 21: 151–160
70. Kim MK, Jernigan RL, Chirikjian GS (2002) Efficient generation of feasible pathways for protein conformational transitions. Biophys J 83:1620–1630
71. Lei M, Zavodszky MI, Kuhn LA, Thorpe MF (2004) Sampling protein conformations and pathways. J Comput Chem 25:1133–1148
72. Schreiber T (1997) Detecting and analyzing nonstationarity in a time series using nonlinear cross predictions. Phys Rev Lett 78: 843–846
73. Schreiber T (2000) Measuring information transfer. Phys Rev Lett 85:461–464
74. Kaiser A, Schreiber T (2002) Information transfer in continuous processes. Physica D 166:43–62
75. Kantz H et al (1993) Nonlinear noise reduction: a case study on experimental data. Phys Rev E 48:1529–1538
76. Kantz H, Schreiber T (2004) Nonlinear time series analysis. Cambridge University Press, Cambridge
77. Stamati H, Clementi C, Kavraki LE (2010) Application of nonlinear dimensionality reduction to characterize the conformational landscape of small peptides. Proteins 78: 223–235

78. Barnett L, Barrett AB, Seth AK (2009) Granger causality and transfer entropy are equivalent for Gaussian variables. Phys Rev Lett 103:238701

79. Barrett AB, Barnett L, Seth AK (2010) Multivariate Granger causality and generalized variance. Phys Rev E 81:041907

80. Hirata Y, Aihara K (2010) Identifying hidden common causes from bivariate time series: a method using recurrence plots. Phys Rev E 81:016203

81. Jamsek J, Palus M, Stefanovska A (2010) Detecting couplings between interacting oscillators with time-varying basic frequencies: instantaneous wavelet bispectrum and information theoretic approach. Phys Rev E 81:036207

82. Kamberaj H, van der Vaart A (2009) Extracting the causality of correlated motions from molecular dynamics simulations. Biophys J 97: 1747–1755

83. Magnasco MO (1993) Forced thermal ratchets. Phys Rev Lett 71:1477–1481

Chapter 12

Accelerated Molecular Dynamics and Protein Conformational Change: A Theoretical and Practical Guide Using a Membrane Embedded Model Neurotransmitter Transporter

Patrick C. Gedeon, James R. Thomas, and Jeffry D. Madura

Abstract

Molecular dynamics simulation provides a powerful and accurate method to model protein conformational change, yet timescale limitations often prevent direct assessment of the kinetic properties of interest. A large number of molecular dynamic steps are necessary for rare events to occur, which allow a system to overcome energy barriers and conformationally transition from one potential energy minimum to another. For many proteins, the energy landscape is further complicated by a multitude of potential energy wells, each separated by high free-energy barriers and each potentially representative of a functionally important protein conformation. To overcome these obstacles, accelerated molecular dynamics utilizes a robust bias potential function to simulate the transition between different potential energy minima. This straightforward approach more efficiently samples conformational space in comparison to classical molecular dynamics simulation, does not require advanced knowledge of the potential energy landscape and converges to the proper canonical distribution. Here, we review the theory behind accelerated molecular dynamics and discuss the approach in the context of modeling protein conformational change. As a practical example, we provide a detailed, step-by-step explanation of how to perform an accelerated molecular dynamics simulation using a model neurotransmitter transporter embedded in a lipid cell membrane. Changes in protein conformation of relevance to the substrate transport cycle are then examined using principle component analysis.

Key words Biological transport, Membranes, Molecular dynamics simulation, Neurotransmitter transport proteins, Protein conformation

1 Introduction

Classic molecular dynamics (cMD) simulations are used to study the kinetic behavior of proteins. By implementing fundamental laws of motion, the technique is able, on an atom-by-atom basis, to accurately predict the dynamic behavior of proteins in various modeled environments. The technique effectively samples conformational space in a time-dependent manner, and if conducted for

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_12, © Springer Science+Business Media New York 2015

a sufficient number of time-steps, produces an accurate representation of a system's potential energy landscape. The resulting trajectories can then be analyzed, allowing for insight to a wide variety of dynamic processes, including protein conformational change.

This technique is invaluable given that proteins are complex structures, interacting dynamically with their environment in response to a variety of thermodynamic, ionic, and other factors. While experimentally determined structures provide valuable three-dimensional information and often a necessary starting point for modeling a given protein, the experimental process provides a single snapshot of a protein, frequently in a non-physiological environment. In order to obtain a detailed mechanistic understanding of protein function, as it is necessary for rational drug development for example, a detailed understanding of dynamic protein conformational change in an environment representative of defined physiological or pathological conditions and under parameters amenable to subtle molecular change is necessary.

While cMD simulations are highly beneficial in this regard, simulation time is limited to the nanosecond or microsecond timescale at best, despite implementation of state-of-the-art supercomputers with highly advanced processing ability. Although processing ability continues to improve, the limit in achievable timescale is further reduced as an increased understanding of biological systems results in the need to model more accurate and complex systems, containing multimers of large proteins in various contexts for example.

Given these limitations, cMD simulations often fall short of effectively exploring the full energy landscape. Exceedingly large numbers of computational steps are required for rare events to occur allowing systems to overcome energy barriers and transition from one potential energy minima to another. Furthermore, the energy landscape for proteins often consists of multiple potential energy wells separated by high free energy barriers. Still, by means of exceedingly long all-atom cMD simulations, often conducted on special-purpose machines using specialized force fields, investigators have achieved timescales far in excess of those previously accessible to computational study, gaining insight into a variety of complex dynamic protein behavior, including protein folding and conformational change within the folded state [1, 2].

In order to avail a more widely available and expeditious technique to simulate the infrequent transition between potential energy minima and therefore realistically capture protein dynamics, alternate computational approaches are necessary. One such approach, termed steered molecular dynamics (SMD), involves applying external forces to a system to explore its mechanical responsiveness [3]. While this approach has proven effective in a number of contexts [4–8], it relies on user defined forces most accurately applied given prior knowledge of the conformational state of interest.

In contrast, accelerated molecular dynamics (aMD) provides a straightforward and effective way to simulate infrequent events required for protein conformational change without previous knowledge of conformational states, potential energy wells, or barriers. The aMD method alters the amount of computational time a system spends in a given potential energy minima by adding a bias potential, $\Delta V(\mathbf{r})$, to the true potential in such a way that potential surfaces in vicinity of the minima are raised, while those closer to the barrier or saddle point are not affected. This allows for a reduction in computational time spent at a potential energy minima and an increase in the ability of a system to move over potential barriers. The effect of the bias is removed by correcting statistics sampled on the biased potential. The method results in preservation of the underlying shape of the potential energy surface and converges to the correct canonical probability distribution, allowing for an approach that efficiently and accurately explores conformational space [9].

Today, aMD simulations are routinely performed to assess time-dependent protein conformational change [10–15] and are fully integrated into commonly used software packages including NAMD [16, 17] and Amber [18, 19]. Performing aMD simulations is straightforward and consists of steps similar to those in cMD simulations, with the addition of defining aMD specific variables.

2 Theory

Expanding on a previous hyperdynamics method explored by Voter [20, 21], the aMD method allows for assessment of protein conformational change by reducing the computational time a simulated protein spends in a potential energy basin, allowing the system to transverse potential energy barriers more readily. This is accomplished by adding a bias potential to the true potential when the systems potential energy falls below a threshold level (Fig. 1). While Voter's method of implementing the bias potential requires the Hessian matrix to be diagonalized at each time step in order for identification of transition state regions, the aMD method is based on a simpler bias potential proposed by Steiner et al. [22] and implemented by Rahman and Tully [23]. In this "puddles" method, the bias potential is selected so that the produced modified potentials near the minima remain constant if the true potential of the system falls below a selected threshold level. Accordingly, diagonalization of the Hessian matrix is not required at each step, allowing for the simulation method to be applied to larger systems such as proteins.

Specifically, a nonnegative, continuous bias boost potential function $\Delta V(\mathbf{r})$ is defined such that when the true potential of a system, $V(\mathbf{r})$, falls below a specified boost energy, E, the simulation is carried out using the modified potential $V^{*}(\mathbf{r}) = V(\mathbf{r}) + \Delta V(\mathbf{r})$,

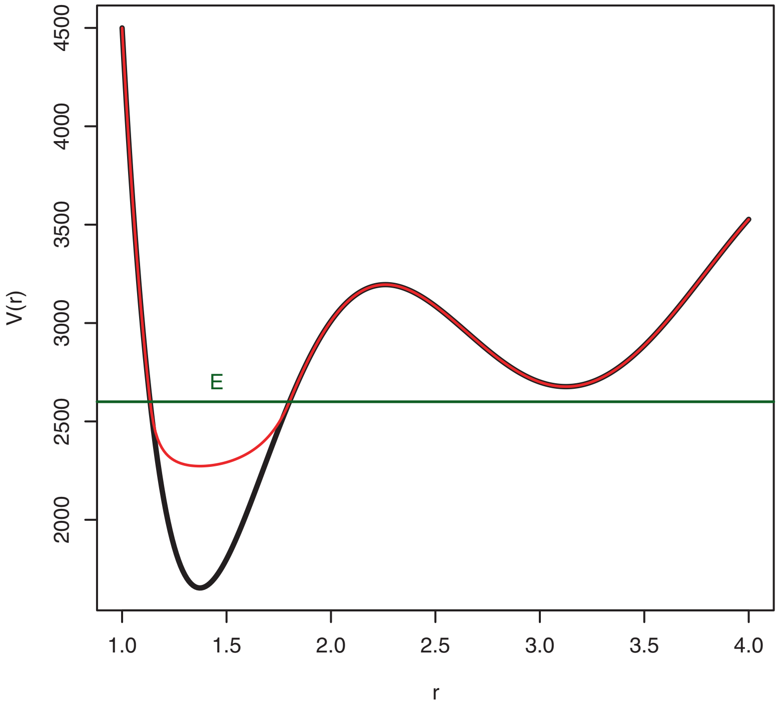


Fig. 1 Graphical representation of the biased potential, the threshold boost energy, E, and the normal potential

while when $V(\mathbf{r})$ is greater than or equal to E, the simulation is carried out using the true potential such that $V^*(\mathbf{r}) = V(\mathbf{r})$. The relationship between the true potential, modified potential, boost energy, and bias boost potential is

$$V^*(\mathbf{r}) = \begin{cases} V(\mathbf{r}), & V(\mathbf{r}) \geq E, \\ V(\mathbf{r}) + \Delta V(\mathbf{r}), & V(\mathbf{r}) < E \end{cases} \tag{1}$$

aMD provides a robust method that, in comparison to cMD simulation, accelerates the state to state transition of large molecules such as proteins. By using the bias potential to modify the true potential of the system, the transition of the system from one state to another takes place at an accelerated rate with a timescale Δt^* that is nonlinear, such that

$$\Delta t_i^* = \Delta t \mathrm{e}^{\beta \Delta V\left[\mathbf{r}(t_i)\right]}. \tag{2}$$

Accordingly, the clock is advanced at each step based on the strength of the bias potential, where Δt represents the actual time step on the unmodified potential. If the bias potential, $\Delta V(\mathbf{r}) = 0$, as is the case when $V(\mathbf{r})$ is greater than or equal to the boost energy E, the system is on true potential and $\Delta t^* = \Delta t$. Statistics can then be used to estimate the total simulation time based on the following equations:

$$t^* = \sum_i^N \Delta t_i^* = \Delta t \sum_i^N \mathrm{e}^{\beta \Delta V[\mathbf{r}(t_i)]}, \tag{3}$$

$$t^* = t\left\langle \mathrm{e}^{\beta \Delta V[\mathbf{r}(t_i)]} \right\rangle, \tag{4}$$

where $\left\langle \mathrm{e}^{\beta \Delta V[\mathbf{r}(t_i)]} \right\rangle$ is the boost factor, a measure of the simulation acceleration extent, and N is the total number of simulation steps.

Importantly, the aMD method converges to the canonical distribution, allowing for accurate determination of equilibrium and other thermodynamic properties. The phase space for the modified potential can be reweighted at each point by multiplying individual configurations by the bias strength at each configuration, resulting in a corrected ensemble average equivalent to that observed with the normal potential [24, 25]. This approach as well as other reweighting approaches allowing for accurate free energy calculation of the resulting trajectories continue to be explored.

In the original method described by Rahman and Tully, the boost potential $\Delta V(\mathbf{r})$ is defined as $E - V(\mathbf{r})$. With this method, when the true potential energy is below the threshold boost energy E, the modified potential energy, $V^*(\mathbf{r}) = E$. This produces flat regions or "puddles" at potential energy basins. While this method is computationally inexpensive, complications due to the discontinuity at points where the unmodified potential meets the modified potential result in the need for special computations which increase the computational burden overall. More importantly, at a high boost energy, the flat modified potential exists at a level higher than most transition state regions. As a result the system may undergo a "random walk" and is slow to converge.

Alternatively, the aMD method utilizes a "snow drift" approach which fills the minima, producing a more smooth landscape. The shape of the underlying potential energy surface is maintained even at a high boost energy E. The method results in a smooth transition where the unmodified potential energy above the boost energy meets the modified potential energy. In order to accomplish this, $\Delta V(\mathrm{r})$ is defined by the equation

$$\Delta V(\mathbf{r}) = \frac{\left(E - V(\mathbf{r})\right)^2}{\alpha + \left(E - V(\mathbf{r})\right)}, \tag{5}$$

The tuning parameter α determines the depth of the modified potential energy basin, such that when $\alpha = 0$, the energy basin is flat, just as in the method described by Rahman and Tully, while as α increases the depth of the modified potential energy basin decreases.

It is important that the values for E and α be selected carefully in the course of an aMD simulation, as these values determine how

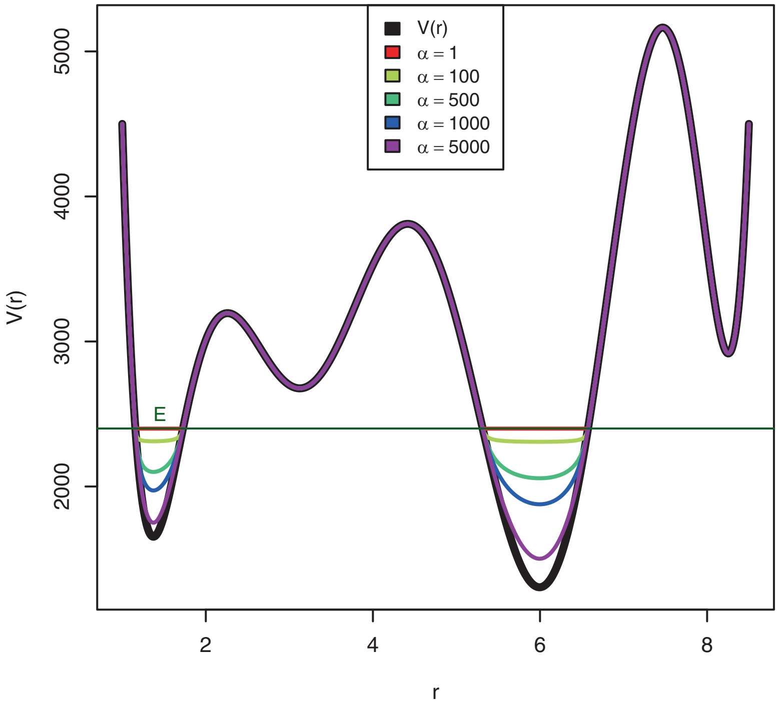


Fig. 2 Graphical representation of a hypothetical potential energy function and different bias potentials plotted at various α values and at a relatively low threshold boost energy, E

aggressive a given simulation is accelerated and how accurately the energy landscape is maintained respectively. The value of E should be selected such that it is larger than the minimum $V(\mathbf{r})$, V_{min}, at the beginning of the simulation. Consider for example a situation in which E is selected such that it is lower than a system's potential energy minimum. Where $V(\mathbf{r}) > E$, $V^*(\mathbf{r}) = V(\mathbf{r})$, the energetics of the system will be maintained just as in a typical molecular dynamic simulation. Alternatively, E can be selected such that it is greater than a system's potential energy minimum, $V^*(\mathbf{r}) = V(\mathbf{r}) + \Delta V(\mathbf{r})$. Since proteins tend to have several potential energy minima spaced closely together, it is advisable to calculate the average true potential energy of a protein over a short period of cMD simulation, and to use this value as the minimum potential energy from which E can be determined.

When the value of E is set low, the modified potential energy remains below the transition state regions and the probability of overcoming energy barriers and enhanced energy sampling is diminished (Fig. 2). In this situation the value of α is of lesser importance to protein conformational sampling and the aMD simulation as a whole, providing that α is large enough so that the modified potential energy basins are not flat ("puddles") since this would result in the calculated force becoming discontinuous where the modified potential energy equals the boost energy.

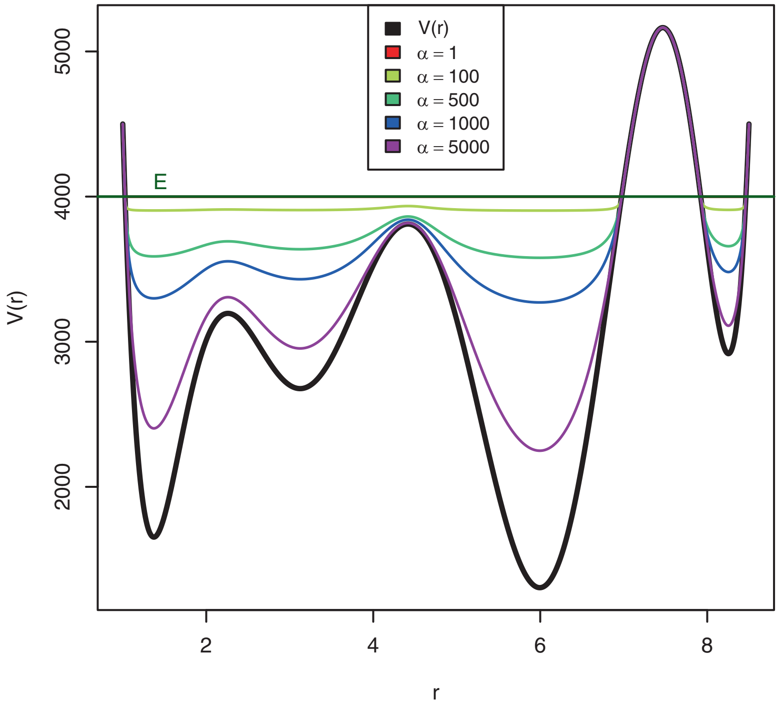


Fig. 3 Graphical representation of a hypothetical potential energy function and different bias potentials plotted at various α values and at a relative high threshold boost energy, E

Alternatively, when the value of E is set to be high, the value of α becomes much more significant (Fig. 3). If the value of E is set high and the value of α is set low, the modified potential becomes equivalent in most places and the system undergoes a random walk. In order to prevent this and to maintain the basic shape of the underlying unmodified potential energy, when the value of E is set high, the value of α must likewise be set high.

Taken together, as indicated above, the first step in determining the correct values for E and α is to determine the minimum potential energy of a given system, which is best determined by calculating the average potential energy of the true potential over a short period of cMD simulation, and then using this value as the minimum potential energy ($V_{\min}$). The value of E should then be chosen so that it is greater than the minimum potential energy. The magnitude by which E is greater than $V_{\min}$ will determine the aggressiveness of acceleration in the simulation. In practice, the value of E is usually selected so that it equals $V_{\min}$ plus 2.5–3.0 times the number of amino acids in the simulation. The value of α should then be set so that it is equal to $E - V_{\min}$. At this value the modified potential energy will preserve the landscape of the underlining unmodified potential energy wells and will merge with the original potential in a smooth fashion [9].

While aMD simulation allows rapid sampling of multiple conformational states, analyzing the resultant trajectories can be problematic due to the large amount of data produced. While the traditional root mean square deviation (RMSD) method can be used to distinguish between different conformational states, this method is not as effective as the system undergoes transitions between subtle yet significant conformational states yielding low RMSD values. Instead, principal component analysis (PCA) can be used as a more sensitive method to distinguish between different conformational states.

PCA reduces the dimensionality of large data sets by calculating a covariance matrix and it eigenvectors. Vectors with the highest eigenvalues become the most significant principal components. When principal components are plotted against each other, similar structures cluster. Each cluster then theoretically represents a different protein conformational state.

Since PCA can be calculated using coordinates from any subset of atoms within a given protein, atom selection can have a large effect on observed outcomes. A common protocol used to avoid sample noise from random fluctuations is to calculate the PCA only for backbone carbon atoms. Alternatively, specific residues or segments of a given protein can be selected based on experimental observations and isolated in a PCA analysis. For example, following aMD simulation of the leucine transporter, the combination of coordinate positions for the backbone carbon atoms of the transmembrane helical domains 1b and 6a was a better discriminator of conformations than the calculations of either the whole structure or any other helical domains alone or in combination [15].

3 Methods

Below we present a protocol for performing aMD simulation and assessing protein conformational change. As an example, we highlight our recent work modeling the bacterial leucine transporter (LeuT), a homologue of the eukaryotic Na^+/Cl^--dependent neurotransporters responsible for terminating synaptic transmission by driving the cellular uptake of neurotransmitters, including the biogenic amines. These proteins are the targets of numerous pharmacological compounds and their dysfunction is associated with disorders of the nervous system. Through the use of aMD simulation, we have gained insight to the function of this class of proteins [26, 15].

In order to provide the reader with a guide such that an equivalent technique can be extended to the study of any protein of interest, we will discuss (1) building a simulation environment suitable for aMD simulation and the study of protein conformational change, (2) preparatory energy minimizations, heating the system, cMD equilibration, and aMD production runs, and (3) analysis of protein conformational change using PCA.

3.1 Computer Software

A text editor is useful for many of the manipulations described below. Helpful functions include the ability to quickly copy and paste several lines of text, search and replace functionality with features such as wild-cards and line number restrictions and the ability to quickly move to a given line number or string of text. More complicated tasks are often facilitated by the use of a procedural programing language such that complex edits to a PDB file are straightforward. UCSF Chimera [27] is a program for visualization, analysis, and manipulation of molecular structures and when used in conjunction with MODELLER [28] is useful in this context for adding missing residues during the initial preparation of the system. The Visual Molecular Dynamics (VMD) package [29] offers robust molecular visualization as well as a variety of molecular building and manipulation options; it will be used here for visualization, the creation of lipid membrane starting coordinates, and embedding the protein in the lipid membrane. AMBER [18] offers highly scalable code designed for high-performance simulation of large molecular systems; it will be used in the following example for placing hydrogen atoms, solvation, energy minimization, heating the system, cMD equilibration, and aMD production runs. The Bio3D utility package in the R statistical computing software environment [30] will be used to perform PCA analysis.

3.2 Obtain the Starting Protein Coordinates

Prior to performing an aMD simulation, it is necessary to obtain starting coordinates for the protein of interest. These may come from crystal structures, nuclear magnetic resonance (NMR) structures, or various computational models. In the case of LeuT, the crystal structure coordinates [31] can be downloaded from the Protein Data Bank, entry 2A65 (www.rcsb.org; MMDB accession no. 34395). When working with a new PDB file, it is a good practice to open the PDB file with a text editor to carefully review the header information in order to gain an overall understanding of the types of residues present, both those that are part of the protein and otherwise, potential limitations of the model, and any other pertinent information.

3.3 Build Coordinates for a Lipid Membrane

For many membrane proteins, the surrounding lipid membrane structure plays a critical role in influencing protein dynamics. In order to realistically model and most accurately capture transporter dynamics we will embed the LeuT in a lipid bilayer. Here we will demonstrate this by using the VMD Membrane Plugin 1.1 to generate the starting coordinates for a lipid membrane and then use VMD's graphic user interface (GUI) front end to align the newly generated membrane with the existing crystal coordinates.

The VMD Membrane Plugin 1.1 uses pre-built patches of lipid bilayers to generate a rectangular membrane of the necessary size. The algorithm used to build the lipid bilayer patches creates two lipid layers, with each layer consisting of a two-dimensional hexagonal lattice of lipids. In order to allow for straightforward insertion of proteins, the lipid tails are nearly fully extended. The distance between the layers and the lattice period is set to correspond with experimental data corresponding to actual membrane thickness and surface density respectively. In order to introduce some disorder to the patches, each lipid is placed in the lipid plane in a random orientation and a truncated Gaussian spread is used in the perpendicular direction. Additional disorder is introduced by a 1 picosecond equilibration in vacuum, serving the additional function of eliminating steric collisions while still leaving most of the lipid tails extended. Finally, the VMD Membrane Plugin 1.1 properly hydrates the lipid head groups, simplifying downstream solvation and equilibration of the protein-membrane system. A more detailed description of the VMD Membrane Plugin 1.1 can be found at: http://www.ks.uiuc.edu/Research/vmd/plugins/membrane/.

In the following steps, we build the membrane coordinates in VMD:

1. Select *Extensions→Modeling→Membrane Builder* to launch the VMD Membrane Plugin.
2. Under *Lipid*, select *POPE*. This defines the lipid composition. Currently, POPE and POPC lipid structures are supported. The lipid composition should be selected to most closely match the lipid composition in vivo as guided by experimental data. Here we selected POPE as this is compatible with the phosphatidylethanolamine-predominant monoamine transporter protein milieu.
3. Under *Membrane X Length* and *Membrane Y Length* we specify the membrane dimensions in Ångströms. For this system, we enter *110* for both dimensions.
4. The *Output Prefix* and *Topology* can be left as the default settings. Click *Generate Membrane* to create the specified membrane.

3.4 Align the Membrane and Protein Coordinates

Positioning the membrane properly must be done carefully and with appropriate consideration for the underlying biology. One way to accomplish this is to position the membrane such that polar head groups and hydrophilic lipid tails make contact with specific residues of the transmembrane protein, as guided by experimental data. If this data is not available, an alternate method involves calculating the center of mass of the main center vestibule of the transmembrane protein and aligning this with the center of mass of the generated membrane. This process is explained in detail in the VMD *Membrane Proteins Tutorial* by Alek Aksimentiev et al. [33]. Here, we position the LeuT based on a combination of experimental data and an analysis of the polarity of the residues at the transporter-lipid interface as follows:

After generating the membrane and water coordinates as detailed in Subheading 3.3, without closing VMD, load the LeuT crystal structure file (PDB 2A65).

1. Load PDB 2A65 by selecting *File→ New Molecule*. Under *Load Files for:* ensure that *New Molecule* is selected. Click on *Browse* and navigate to the folder where the coordinates are saved. Selected the correct file and click *Open*. Under *Determine file type:* PDB should be selected automatically if the file was given the correct extension (.pdb). Select *Load*.
2. In the VMD Main window, the currently loaded molecules are listed in the Molecule List Browser. Two separate molecules should be listed in this window: one containing the generated membrane and water coordinates and a second containing the 2A65 coordinates.

In order to better visualize our protein, membrane, and water molecules, we will alter the graphical representation. Select *Graphics→ Representations* and adjust the representation as follows:

1. Under *Selected Molecule* select the molecule corresponding to the lipid membrane. Under selected atoms type *resname POPE and not hydrogen* and click on *Apply*.
2. Click on *Create Rep*. This creates a second representation for the molecule containing the lipid coordinates. Under *Selected Atoms* type *resname TIP3* and click *Apply*. This creates two separate representations for the coordinates generated by *Membrane Builder 1.1*, one representation for the lipid molecules (without hydrogen atoms) and a second representation for the water molecules. In the white box under *Create Rep*, where the two separate representations are listed, double click on the representation containing the water molecules. The text should turn red and the water molecules should disappear from the OpenGL Display. While the water molecules are important to the simulation, positioning the membrane correctly in relation to the protein is facilitated by hiding the water molecules.
3. Under *Selected Molecule* click on the dropdown menu and select the molecule containing the LeuT coordinates. Change the *Coloring Method* to *ResType* and *Drawing Method* to *Cartoon* and click *Apply*. This allows us to distinguish between nonpolar residues (white), polar residues (green), basic residues (blue), and acidic residues (red). This information will be helpful in properly positioning the lipid membrane.
4. In the VMD Main window, select *Display→ Orthographic*.

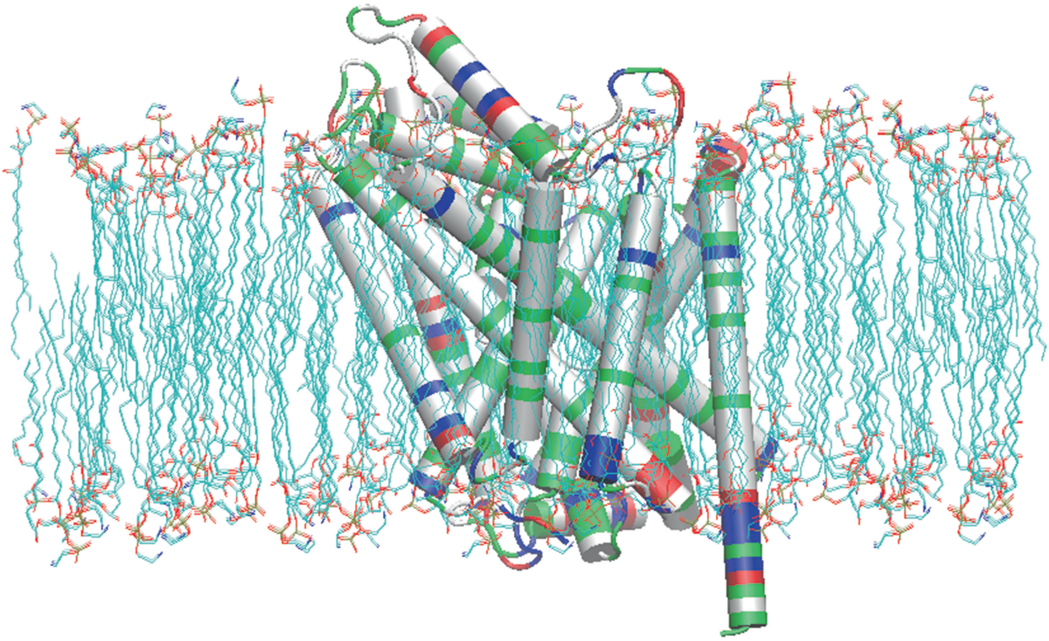


Fig. 4 LeuT positioned within the POPE lipid membrane. Nonpolar residues (white) at the membrane interface in contact with the hydrophobic lipid tails of the POPE membrane and polar residues (green) at the membrane interface either in contact with the polar head groups our out of the transverse plane of the lipid bilayer

We are now ready to move the crystal structure coordinates while holding the newly generated membrane coordinates fixed.

1. In the VMD Main window select *Mouse* → *Move* → *Molecule*. All the coordinates within the 2A65 crystal structure are now moved by clicking with the left mouse button on the transporter in the OpenGL Display window and dragging. Holding the shift key and moving the mouse allows the membrane and water coordinates to be rotated about the point on the screen that is clicked. Using the left mouse button rotates about the *x* or *y* axis of the screen, while using the middle or right button rotates the coordinates about an axis perpendicular to the screen.
2. Using the keyboard and mouse commands indicated above, we now move the crystallized coordinates such that the transporter is positioned correctly in relation to the membrane. The general orientation of the transporter in the lipid membrane is described based on experimental data obtained by Yamashita et al. [31]. More exact placement is obtained by positioning to the greatest extent possible (1) the nonpolar residues (white) at the membrane interface in contact with the hydrophobic lipid tails of the POPE membrane and (2) the polar residues (green) at the membrane interface either in contact with the polar head groups our out of the transverse plane of the lipid bilayer (Fig. 4).

3. When satisfied with the placement of the transporter in relation to the membrane coordinates, in the VMD Main window select *File→Save Coordinates.* Under *Save data from* use the dropdown menu to select the molecule containing the correctly positioned 2A65 coordinates. Under *Selected atoms* select *all.* Ensure *pdb* is selected under *File type.* Click *Save*, navigate to an appropriate directory and enter a file name to save the lipid coordinates. Finally, click *Save* again. The PDB and PSF files generated for the membrane and membrane solvation are automatically saved to the VMD working directory when the membrane structure is generated. Since no modifications were made to the membrane it is not necessary to re-save these files.

3.5 Prepare the Starting Protein Coordinates

In the case of 2A65, in addition to including the coordinates for the amino acids that make up the transporter, the Protein Data Bank entry includes crystallized coordinates for ions found in the binding pocket, the substrate leucine, water molecules, and a B-octylglucoside (BOG) residue. In our simulation, we wish to include the crystallized transporter with ions, but remove the substrate, water molecules, and B-octylglucoside residues. The substrate is removed in this example in order to sample transporter conformations that occur in the absence of substrate. The B-octylglucoside residues are removed since they are not likely to influence the dynamic process of substrate transport, the interest of this study. It is important that decisions in regard to experimental design, such as that mentioned previously, be made with consideration for underlying biology and specific study objectives.

In order to accomplish this, we will use a text editor to modify the crystal coordinates saved after alignment with the membrane in Subheading 3.4. While it is not essential, it can be helpful from an organizational perspective to save the coordinates for each individual subunit/chain in new separate PDB files, keeping only lines containing coordinate information. If several chains are to be included in a single coordinate file, it is necessary to add the TER keyword between each chain, although this is not needed between water molecules. In the case of LeuT (PDB 2A65), we create two new separate PDB files from the membrane aligned coordinates, one containing the coordinate lines for the residues of the transporter and one containing the coordinate lines for the binding pocket ions (with each line of ion coordinate information separated by the TER card).

At this stage it is also important to consider special residues with modifiable protonation states, such as cysteine and histidine residues. When PDB coordinate files are initially read into LEaP to produce input files for production calculations in AMBER, regular protonated cysteine residues should be named CYS, deprotonated or metal atom bound cysteine residues should be named CYM, and cysteine residues involved in disulphide bridges should be named CYX. Similarly, histidine residues should be named HIE when protonated in the epsilon position, HID when protonated in the delta position, and HIP when protonated in both the epsilon and delta positions. In the LeuT crystal structure, there are no cysteine residues, however, we replace the residue name HIS with the residue name HID.

It is helpful also to remove all hydrogen atoms included in the newly made coordinate files. Coordinates for hydrogen atoms in NMR structures may be inaccurate. Furthermore, NMR naming conventions are different from PDB naming conventions. Still, if needed, it is possible to use LEaP to correct the nonstandard hydrogen names. In the present example, all hydrogen atoms will be positioned using LEaP and then energy minimized prior to cMD simulation.

Finally, note that some PDB files contain alternate coordinates for specific residues. When these coordinates are read into AMBER using LEaP, only the A conformation is used. If a different coordinate set is needed, this can be accomplished by modifying the PDB file to contain only the coordinates of interest.

3.6 Build Missing Residues

Review of the headers in the 2A65 PDB file indicates that the four most N- and C-terminal residues and residues N133 and A134 are missing. We will build the missing non-terminal residues (N133 and A134) into our structure, but leave out the terminal residues. This task can be accomplished in UCSF Chimera and MODELLER as follows:

1. In UCSF Chimera load the membrane aligned 2A65 PDB file by selecting *File→ Open* and navigating to the PDB structure. While it is often not necessary to specify the "file type" it is a good practice to do so to avoid any potential unexpected errors.
2. Select *Tools→ Structure Editing→ Model/Refine Loops*. This brings up sequence information for the protein. Missing residues are highlighted with red boxes. Selecting *Model/Refine Loops* will also open UCSF Chimera's interface to MODELLER.
3. In the Model Loops/Refine Structure window select *non-terminal missing structure*. This selects residues that are found in the PDB SEQRES record, but missing from the PDB file coordinates. This selection only selects residues constrained at both ends by existing structures, so the N- and C-terminal missing residues will not be built.
4. Other building parameters can also be modified. In this example, we allow 0 residues adjacent to the missing regions to move, generate 1 model, use the Discrete Optimized Protein Energy (DOPE) energy score [32], and run MODELLER

using the Web service. After selecting *Ok*, progress toward completion of this step can be tracked in the lower left hand side of the main UCSF Chimera window.

5. The new coordinates can be saved by selecting *File → Save PDB*. Give the file a new name by adding text in the box to the right of *File name*, select the newly created model in the box to the right of *Save models* and select *Save*.
6. Using a text editor, open the newly saved PDB file containing the missing residues, navigate to and highlight the newly created residues and cut/paste the new ATOM information into the appropriate place in the previously prepared PDB file containing only the transporter coordinates. Note that since we have not built the N-terminal residues, the PDB file output by UCSF Chimera has renumbered the residues beginning in the first position. Be careful to select and copy the correct residues.
7. Renumber the new residues and atoms appropriately in the transporter coordinate file being prepared for input to AMBER. While renumbering the two newly created residues is straightforward to accomplish manually, renumbering the atoms manually is a daunting task. To assist in doing this, the Fortran 90 source code provided below can be used to compile a program that reads in a PDB file containing 4056 atoms and outputs a PDB file with identical information but appropriately numbered atoms. Different PDB manipulation scripts can be written in any procedural programing language of choice to assist in performing more complex alterations to various components of a PDB file.

```
program renumber_atoms
   implicit none
   character (len = 5) :: first
   character (len = 6) :: old
   character (len = 64) :: last
   integer :: i = 1
   do i = 1, 4056
     read (5,"(a5,a6,a64)"), first, old, last
     write (6,"(a5,i6,a64)"), first, i, last
   end do
end program
```

3.7 Remove Overlapping Membrane Structures

In Subheading 3.4 PDB and PSF files were generated for a lipid bilayer with solvated polar head groups. In Subheadings 3.5 and 3.6 the LeuT transporter and binding pocket ions were correctly positioned within the bilayer and the missing transporter residues were built respectively. There are still, however, lipid and water molecules that overlap with the transporter structure. In this section VMD will be used to delete any lipid or water residues that overlap or are too close to the transmembrane protein.

First, a protein structure file (PSF) is generated from the LeuT only PDB file created in Subheading 3.6. This is done in VMD using the automatic PSF generator as follows:

1. Open VMD and in the VMD Main window select *Extensions→ Tk Console*. In the VMD TkConsole that opens, navigate to the directory where the LeuT PDB file generated in Subheading 3.6 is located by using the "cd" command followed by the path to the appropriate directory and pressing enter.
2. Load the PDB file by entering the command "*mol new filename.pdb*" and pressing enter.
3. In the VMD Main window select *Extensions→ Modeling→ Automatic PSF Builder*.
4. Under "*Step 1: Input and Output Files*" ensure the PDB file loaded in step 2 is selected under "*Molecule*" and that the "*Output basename*" is "*leut_autopsf*".
5. Under "*Step 2: Selections to include in the PSF/PDB*" make sure that "*Everything*" is selected.
6. Under "*Step 3: Segments Identified*" select "*Add a new chain*". Under "*Chain Name*" enter "*LeuT*". The "*First Atom*" and "*Last Atom*" should be changed to the first and last LeuT atom number in the PDB file that was loaded (1 and 4057 respectively). Under "*N terminal patch*" and "*C terminal patch*" add "*NTER*" and "*CTER*" respectively. Finally select "*Add chain*".
7. Under "*Step 4: Patches*" select "*Apply patches and finish PSF*". This will create in the current working directory a PSF and PDB file with the name entered under "*Output basename*".

Next, the previously generated membrane.psf and membrane.pdb files are loaded, the membrane and transporter structures are combined into a single PSF and PDB to allow for comparison, lipid and water molecules that are within a specified distance from the protein are deleted, and the resulting PDB and PSF files for the system with overlapping residues deleted are output. All of these steps are easily accomplished with the following script that can be run from the VMD Tk Console:

```
# Load package
package require psfgen
# Load starting structures
resetpsf
readpsf membrane.psf
coordpdb membrane.pdb
readpsf leut_autopsf.psf
coordpdb leut_autopsf.pdb
```

```
   # Output a combined structure
   writepsf leut_mem.psf
   writepdb leut_mem.pdb
   # Load combined structure
   mol load psf leut_mem.psf pdb leut_mem.pdb
   # Delete lipid and water molecules within 1.2
A of protein
   # Get segids to check
   set check_sel [atomselect top "resname POPE
or water"]
   set check_sel_segs [lsort -unique [$check_
sel get segid]]
   foreach list_seg $check_sel_segs {
     # find atoms
     set current_atom [atomselect top "segid
$list_seg and within 1.2 of protein"]
     # delete residues
     set current_res [lsort -unique [$current_
atom get resid]]
     foreach res $current_res {
       delatom $list_seg $res
     }
   }
   # Output modified structure
   writepsf leut_mem_no_overlap.psf
   writepdb leut_mem_no_overlap.pdb
   # Load modified structure for viewing
   mol load psf leut_mem_no_overlap.psf pdb
leut_mem_no_overlap.pdb
```

The above script is executed in VMD by inserting the commands in a text document and saving the file to VMD's current working directory. Within the Tk Console, the script is then run by executing the command "source filename", where "filename" is the name of the file the above script is saved as.

Next, assess the resulting structure. The script above can be modified such that the cutoff distance at which residues are deleted is changed. When satisfied with the resulting structure, save PDB files for the membrane alone and the membrane solvation alone as follows. In the VMD Main window select *File → Save Coordinates.* Under *Save data from* use the dropdown menu to select the correct molecule. Under *Selected atoms* select *resname POPE and not hydrogen*. Ensure *pdb* is selected under *File type*. Click *Save*, navigate to an appropriate directory and enter a file name to save the lipid coordinates. Finally, click *Save* again. Repeat this process again to save the water molecules alone by changing the *Selected atoms* to *waters and not hydrogen*.

3.8 Generate a Nonstandard POPE Lipid Unit Using Antechamber

If we loaded the files generated thus far into AMBER's LEaP program using a standard protein force field, the POPE lipid residue would not be recognized. In this next section we will use the Antechamber tool set that comes with AMBER in order to generate an input file that is necessary in order to include the POPE lipid structure in our simulation. The Antechamber tool set works in conjunction with the general AMBER force field (GAFF), a force field that is designed to have general atom types such that it can provide broad applicability to a variety of different molecules. It can also be used in conjunction with traditional AMBER force fields that may be more appropriate for the rest of the molecules in a simulation. As such, Antechamber and GAFF are highly useful entities. The Antechamber tool set automatically identifies atom and bond types, judges atomic equivalence, provides a residue topology file, and suggests reasonable alternatives for any resulting missing force field parameters. Given the highly automated nature of this process, however, it is critical to carefully evaluate the resulting output to verify suitability.

Here we will use Antechamber to assign atom types and charges for the POPE residue. To begin, open the PDF file generated previously containing the coordinate information for the POPE lipid membrane. Copy and paste all the lines that together make up a single POPE residue (there are 125 ATOM lines that together represent a single POPE residue) into a new text file called pope.pdb. Next, in order to use antechamber to create the "prepin" file necessary to define the new POPE unit in LEaP, run the following command:

```
$AMBERHOME/bin/antechamber -i pope.pdb -fi pdb
-o pope.prepin -fo prepi -c bcc -s 2
```

The "$AMBERHOME/bin/antechamber" command initiates the antechamber utility. The "-i" command specifies the name of the input file and the "-fi" command specifies the format of the input file; antechamber accepts several other input file formats. The "-o" command specifies the name of the output file and the "-fo prepi" command tells antechamber to output the file in the PREP format used internally by LEaP. "-c bcc" directs antechamber to calculate the atomic point charges using the BCC charge model. Finally, "-s 2" specifies the level of verbosity that antechamber is to provide.

Completion of this step will result in several new files in the directory where the script is executed. The files in capital letters are intermediate files useful for troubleshooting; should analysis of the following produce the intended results they can safely be deleted. The divcon.out file results from quantum mechanics calculations necessary in order to determine the point charges for individual atoms. It is useful to assess this file to ensure that this process has completed without error. Finally, the "pope.prepin" file is the file necessary to load the new POPE unit into LEaP. This file contains a definition of the POPE unit including connectivity, charges, and atom type information. Importantly, the atom types in this file (column 3) are all listed in lower case; this is intended to distinguish from the atom types to be evaluated using traditional AMBER force filed parameters, which are defined in upper case.

The GAFF is designed to be extensive and therefore include parameters for a large combination of parameters. It is possible, however, that a new unit defined in Antechamber may contain a combination of atom types that have not been parameterized. In order to check for this the parmchk executable is called:

```
$AMBERHOME/bin/parmchk -i pope.prepin -f prepi
-o pope.frcmod
```

This command compares the parameters that are necessary for the new POPE unit to those that are available in the GAFF. Should any parameter be missing antechamber will either empirically calculate the parameter or find a similar, analogous parameter; the results of which are then listed in the ".frcmod" file. Should it be impossible to define parameters in this way, Antechamber will add a place holder and a comment indicating that revision is required, in which case other methods of parameterization should be pursued. It is important to carefully assess the rational that is used for any parameters obtained in the ".frcmod" file. The "pope.frcmod" file obtained here is shown below:

```
      remark goes here
      MASS
      BOND
      ANGLE
      DIHE
      IMPROPER
      c3-o -c -os 10.5 180.0 2.0 General improper
torsional angle (2 general atom types)
      c2-c3-c2-ha 1.1 180.0 2.0 Using default value
      NONBON
```

3.9 Solvate the System, Add Missing Atoms, and Generate Input Files for Production Calculations

Prior to performing production calculations such as energy minimizations or MD simulations in AMBER, it is necessary to define: (1) a "prmtop" file that contains the required force field parameters and a description of the molecular topology and (2) an "inpcrd" file that contains the atomic coordinates. An "inpcrd" file can also contain atomic velocities and periodic box information if defined. Here, AMBER's LEaP program will be used to generate the files necessary for production calculations.

In order to realistically model protein behavior, it is furthermore necessary to solvate the system with water molecules and ions. Any missing atoms including hydrogen atoms and N- and C-terminus specific atoms which have not previously been included can be added at this point. Conveniently, LEaP will perform all of these tasks. The molecular coordinates and parameter information developed as described above will be used as a starting point.

Here, tleap, the command line version of LEaP will be used. The script provided below will read in the coordinate and parameter information generated thus far and define a periodic box. Next, water molecules will be added, maintaining a 1.5 Å or greater distance from any existing solute and forming a box that extends 12 Å in the Z axis from any existing solute. Sodium and chloride atoms will then be added within the box surrounding solute. The Coulombic potential on a grid will be used to determine placement and should steric conflict occur with a water molecule, the water molecule will be removed and the ion will be added in its place. The number of sodium and chloride ions are selected to mimic physiological ion concentrations and also to meet the requirement of an overall neutral system. Finally, the resulting system is evaluated for errors and a PDB file as well as the necessary "prmtop" and "inpcrd" files are output for the system as a whole.

```
# Load force field parameters
source leaprc.gaff
source leaprc.ff99SB
# Load Antechamber files - from 3.8
loadAmberPrep pope.prepin
loadAmberParams pope.frcmod
# Load positioned binding pocket ions - from 3.5
bpions = loadpdb "bp_ions_positioned.pdb"
# Load positioned LeuT with missing residues - from 3.6
leut = loadpdb "leut_positioned_allres.pdb"
# Load lipids and lipid solvation - from 3.7
pope = loadpdb "pope.pdb"
popewater = loadpdb "pope_water.pdb"
# Combine into one unit
system = combine {bpions leut pope popewater}
# Set periodic box
setBox system vdw
# Solvate the system along the z axis; maintain a distance of 1.5 A from solute
solvateBox system TIP3PBOX {0 0 12} 1.5
# Add ions to produce a neutral system
addIons system Na+ 56 Cl- 60
# Examine the final unit
check system
charge system
desc system
# Output files
savePdb system leut_system.pdb
```

```
   saveamberparm   system   leut_system.prmtop
leut_system.inpcrd
   quit
```

With all the necessary files in the current working directory and the above script saved to a text file called "leap.in", the following command is used to execute the script:

```
tleap –f leap.in
```

3.10 Perform Energy Minimizations

Next energy minimizations are performed in order to "relax" the system. This is necessary to eliminate any energetically unfavorable interactions that may occur in the process of building the system. For example, hydrogen atoms added as described above are positioned according to a predefined geometry and therefore the resulting coordinates may have steric conflicts or overlap with other residues. The minimization steps eliminate such problems, providing an energetically favorable, more stable starting point for molecular dynamics simulation.

Using the "prmtop" and "inpcrd" files generated in Subheading 3.9, AMBER's sander can be used to perform the energy minimization. The following "minimize.in" file is used to direct sander to perform an energy minimization:

```
Minimization of solvent
 &cntrl
  imin   = 1,
  maxcyc = 1000,
  ncyc   = 400,
  ntb    = 1,
  ntr    = 1,
  cut    = 12
 /
Hold protein and lipids fixed
250.0
RES 1 636
 END
 END
```

The "&cntrl" command must begin with a leading black space and specifies the type of namelist that follows. The minimization starts at step 1 (imin = 1) and continues for 1,000 steps (maxcy = 1,000). The first 400 steps utilizes the steepest decent algorithm, appropriate for quickly reducing the largest strain, and the remaining 600 steps utilizes the conjugate gradient algorithm which is more appropriate for converging to a minima (ncyc = 400). Constant volume periodic boundary conditions will be used (ntb = 1). Position restraints will be used (ntr = 1) using a force constant of 250 kcal/mol/Å^2 to restrain the lipid and protein residues (residue numbers 1–636).

The following command will execute the energy minimization:

```
   $AMBERHOME/bin/sander -O -i minimize.in -o
minimize.out –c leut_system.inpcrd -p leut_sys-
tem.prmtop -r leut_system_min.rst
```

Where "-O" specifies that output files should be overwrite any existing files, "-I" specifies the name of the "mdin" file, "-o" specifies the name of the output file, "-c" specifies the starting "inpcrd" file, "-p" specifies the "prmtop" file, and "-r" specifies the name of the final coordinates following minimization.

Beginning with the "rst" file from the previous minimization, the following "mdin" file can be used to minimize the lipids as well:

```
Minimization of solvent and lipids
 &cntrl
  imin   = 1,
  maxcyc = 1000,
  ncyc   = 400,
  ntb    = 1,
  ntr    = 1,
  cut    = 12
 /
Hold protein fixed
250.0
RES 1 511
 END
 END
```

Finally, all atoms in the simulation are minimized with the following "mdin" file:

```
Minimization of all atoms
 &cntrl
  imin   = 1,
  maxcyc = 1000,
  ncyc   = 400,
  ntb    = 1,
  ntr    = 0,
  cut    = 12
 /
```

3.11 Heat the System and Perform an Initial cMD Equilibration

Following energy minimization the system is ready for cMD equilibration. The system temperature is linearly increased from 0 to 310 K (tempi = 0.0; temp0 = 310.0) in order to prevent excessive and sudden solute fluctuations. A weak restraint is placed on the protein and lipid residues to further aid in this regard. A Langevin temperature equilibration scheme (ntt = 3) will be used to equalize and maintain the system temperature using a collision frequency of 1.0 ps^{-1} (gamma_ln = 1.0).

While equilibration will ultimately be performed using constant pressure and temperature parameters, as the system is heating up the pressure that is calculated can be inaccurate, leading to issues if constant pressure parameters are employed. The use of restraints with constant pressure parameters further causes problems. Accordingly, the system is initially equilibrated at constant volume (ntb = 1) and then later transitioned to constant pressure parameters. Furthermore, since the hydrogen atom motion in solute is unlikely to affect the overall protein dynamics, a SHAKE algorithm is used which fixes all bonds involving hydrogen (ntc = 2; ntf = 2), reducing computational burden and allowing the time step to be increased to 2 fs without introducing any instability (if hydrogen atoms were allowed to oscillate the would provide the highest frequency oscillation in the system and therefore determine the maximum time step).

The following "heat.in" script can be used to heat the system:

```
   cMD heating of the system with restraints on
protein and lipid residues
    &cntrl
     imin = 0, irest  = 0, ntx = 1,
     ntb = 1,
     cut = 12.0,
     ntr = 1,
     ntc = 2, ntf    = 2,
     tempi = 0.0, temp0 = 310.0,
     ntt = 3, gamma_ln = 1.0,
     nstlim = 20000, dt = 0.002
     ntpr = 2000, ntwx = 2000, ntwr = 2000
    /
   Keep  protein  and  lipids  fixed  with  weak
restraints
   10.0
   RES 1 636
   END
   END
```

In the "heat.in" script above the minimization is turned off (imin = 0) and initial velocities are randomly assigned from a Boltzmann distribution (irest = 0; ntx = 1). A cutoff of 12 Å is used (cut = 12). 20,000 cMD steps will be performed (nstlim = 20,000) using a time step of 2 fs per step (dt = 0.002), resulting in a total simulation time of 40 ps. The output file (ntpr), trajectory file (ntwx) and restart file (ntwr) will be written to every 2,000 steps. Finally, weak position restraint (ntr = 1) using a force constant of 10 kcal/mol/Å^2 will be used to minimize lipid and protein residues movement (residue numbers 1–636). The following command will initiate heating of the system:

```
   $AMBERHOME/bin/sander -O -i heat.in -o heat.
out  -c  leut_system_min3.rst  -p  leut_system.
prmtop -r leut_system_heated.rst
```

After heating the system, a short cMD equilibration using constant temperature and pressure parameters in necessary to obtain information about the system that will be required for aMD production runs.

The following "cMD.in" script can be used to accomplish this:

```
   cMD equilibration to obtain parameters nec-
essary for aMD production runs
    &cntrl
     imin = 0, irest  = 1, ntx = 7,
     ntb = 2, pres0 = 1.0, ntp = 2, taup = 2.0,
     cut = 12.0,
     ntr = 0,
     ntc = 2, ntf = 2,
     tempi  = 310.0, temp0  = 310.0,
     ntt = 3, gamma_ln = 1.0,
     nstlim = 500000, dt = 0.002
     ntpr = 2000, ntwx = 2000, ntwr = 2000
    /
```

In the "cMD.in" script above, the parameters that have changed in comparison to the "heat.in" script are defined as follows. Since the simulation is being restarted, the time step is read in from the previous run (irest = 1) and the coordinates being read in are in ASCIII restart format (ntx = 7). Anisotropic pressure scaling will be used (ntp = 2) to maintain a constant pressure (ntb = 2) with an average pressure of 1 atm (pres0 = 1.0) and a relaxation time of 2 ps (taup = 2.0). No position restraints are used (ntr = 0). One nanosecond of simulation time will be obtained (nstlim = 500000, dt = 0.002). The following command can be used to execute the script:

```
   $AMBERHOME/bin/sander -O -i cMD.in -o cMD.
out −c leut_system_heated.rst -p leut_system.
prmtop -r leut_system_cMD_equil.rst
```

3.12 Perform aMD Production Runs

The cMD equilibrated system and data obtained from the cMD equilibration can now be used for aMD production runs. aMD production runs are conducted in a very similar fashion to the cMD equilibration described in Subheading 3.11, with the only difference being the definition of additional aMD specific parameters. The following "aMD.in" script can be used for aMD production runs:

```
   aMD production run
    &cntrl
     imin=0, irest=1, ntx=7,
     ntb = 2, pres0 = 1.0, ntp = 2, taup = 2.0,
     cut = 12.0,
     ntr = 0,
     ntc = 2, ntf = 2,
     tempi  = 310.0, temp0  = 310.0,
     ntt = 3, gamma_ln = 1.0,
     nstlim = 500000, dt = 0.002
     ntpr = 2000, ntwx = 2000, ntwr = 2000
     iamd = 3,
     alphaD = 357.7,
     EthreshD = 12988.9,
     alphaP= 10494.2,
     EthreshP = -94241.8
    /
```

In the "aMD.in" script above, the simulation parameters are the same as in the previously used "cMD.in" with the addition of the following aMD specific parameters. The aMD implementation in AMBER allows for the possibility of boosting the torsional terms of the potential only (iamd = 2), the whole potential (iamd = 1), or the whole potential with an additional boost to the torsions (iamd = 3). alphaD is defined as the product of 0.2, 3.5 kcal/mol/residue, and the number of protein residues (511 residues are in the LeuT example here). EthreshD is defined as the product of 3.5 kcal/mol/residue and the number of protein residues, added to the average dihedral based on cMD equilibration (11,200.4 for the system in this example). alphaP is defined as the product of 0.2 and the total atoms (52,471 total atoms in this example). EthreshP is defined as the sum of alphaP and the average EPtot from cMD equilibration (−104,736 for the system in this example).

The following command can be used to execute the "aMD.in" script:

```
   $AMBERHOME/bin/sander -O -i aMD.in -o aMD_01.
out –c leut_system_cMD_equil.rst -p leut_sys-
tem.prmtop -r leut_system_aMD_01l.rst
```

Several additional aMD production runs can be performed, each time beginning with the restart coordinates output from the previous run.

3.13 Prepare Data for PCA Analysis

When sufficient aMD production runs have completed, principal component analysis is used to analyze the results from the aMD simulations. The statistical program R is a platform for performing statistical calculations with large data sets. Through the use of the Bio3D add on package, R is able to perform analysis of PDB and DCD files. To start using the Bio3D package with R to analyze the data, the data must be formatted into a Bio3D useable style. The first thing to note is that Bio3D works best with protein only data, so any other molecules will need to be removed (lipids, waters, ions, etc.). The second note is that Bio3D does not utilize parameter-topology files from AMBER (PRMTOP) or protein structure files from NAMD (PSF); instead, Bio3D uses information from the PDB file to determine the protein structure. The final note on Bio3D is that it requires DCD trajectories for processing (the default binary for NAMD), so AMBER trajectories will need to be converted.

1. Use VMD and load in the structure file (PSF or PRMTOP): *File→ New Molecule* then load in the PRMTOP file. VMD will recognize that it is an AMBER topology file but not whether or not it is one with a periodic boundary conditions. For our LeuT simulations, and probably under most circumstances, it will be necessary to change the file type to *AMBER coordinates with Periodic Box*.
2. Load the trajectories into the created molecule: Right click the molecule and select *Load data into molecule*. Select the trajectory file and *load in*. Depending on the size of the trajectory and available computer resources, this may need to be broken into smaller steps or trajectory frames will need to be skipped.
3. Change the visualization to show the protein only: *Graphics→ Representations*. Select the molecule and change the atom selection to the desired atoms. Replacing the selection all with the phrase protein will usually be sufficient, however this may change depending on the configuration of the topology file.
4. Save a PDB file of the first frame of the trajectory: Right click the molecule from the main window and click *Save Coordinates*. Change the type to *PDB* and change the first and last frame to *0*. Use the drop down menu to select the atom selection specified in **step 3** above. This will be the PDB Bio3D will use to determine the protein structure.
5. Create a DCD file of the trajectory frames: Follow the steps above to Save coordinates, but change type to *DCD* and make sure the first frame is *0* and the last frame is the last frame loaded into the system.

3.14 Using Bio3D and R to Calculate the PCA

In an R terminal, load in the Bio3D library (ensure the package has previously been installed). Read in the PDB file of the first frame as the structure and then the trajectory from the DCD file.

```
library(bio3d)
pdb <-read.pdb("LeuT_frame1.pdb")
dcd <- read.dcd("LeuT_trj.dcd")
```

Select the atoms of interest for the PCA analysis. The *atom.select* command pulls the indices of atoms which correspond to the atom selection. For our analysis, we choose to use all alpha carbon positions of select residues. The elety parameter of the atom.select command specifies the atom type ("CA" = alpha carbon), and the resno parameter can be used to select residues by number.

```
ca.ind <- atom.select(pdb, elety="CA")
```

In order to eliminate translations and rotations the trajectory structures will need to be fitted and superposed to the PBD structure. The *fit.xyz* command can be used to accomplish this. The *fit.xyz* command takes two necessary input parameters—the structure to use as a reference and the structures to fit to the reference. It should be noted that the *fit.xyz* command only takes coordinates as input, so the *$xyz* object of the PDB will need to be used. The bracket argument pulls the atom selections from above as the coordinates to fit. The output of the *fit.xyz* command will be a superposed trajectory of the selected atoms only.

```
trj.fit    <-    fit.xyz(pdb$xyz[ca.ind$xyz],
dcd[,ca.ind$xyz])
```

The PCA of the coordinates can be taken based on the trajectory with the *pca.xyz* command.

```
trj.pca <- pca.xyz(trj.fit)
```

With the Bio3D package installed, the plot command has been overloaded to create a default PCA plot with four graphs. Three are the z-scores of the first three principal components plotted against each other in two dimensions. The last is a scree plot representing how much of the variance of the data set is captured by each principal component (Fig. 5).

```
plot(trj.pca)
```

The plot points can be computationally clustered and colored by cluster. This can be done by creating a distance matrix of the principal components of interest (principal components 1, 2, and 3 were used in this analysis).

```
d <- dist(trj.pca$z[,1:3])
```

A dendrogram of the distance matrix can be calculated and plotted for visualization. The plot of the dendrogram can be used to determine the number of clusters desired from the analysis.

```
hc <- hclust(d)
plot(hc)
```

The *cutree* command can be used to create a color vector which will color each point based on the number of groups desired. It takes two arguments, the dendrogram and k which is the desired number of clusters. Here, the LeuT data appeared to fall into seven clusters. The PCA plots can be colored by replotting the PCA and using the output from the *cutree* command as an argument to the col parameter (Fig. 6). For structure analysis, representative structures from each cluster can be isolated and analyzed in molecular visualization software of choice like VMD.

```
grps <- cutree(hc,k=7)
plot(trj.pca,col=grps)
```

3.15 Validation and Use of the PCA

PCA of aMD data is useful for finding protein segments that are most involved in structural changes. The RMSF (root mean square fluctuations) calculation can be used to determine how much each residue moves during the trajectory.

```
rf <- rmsf(trj.fit)
```

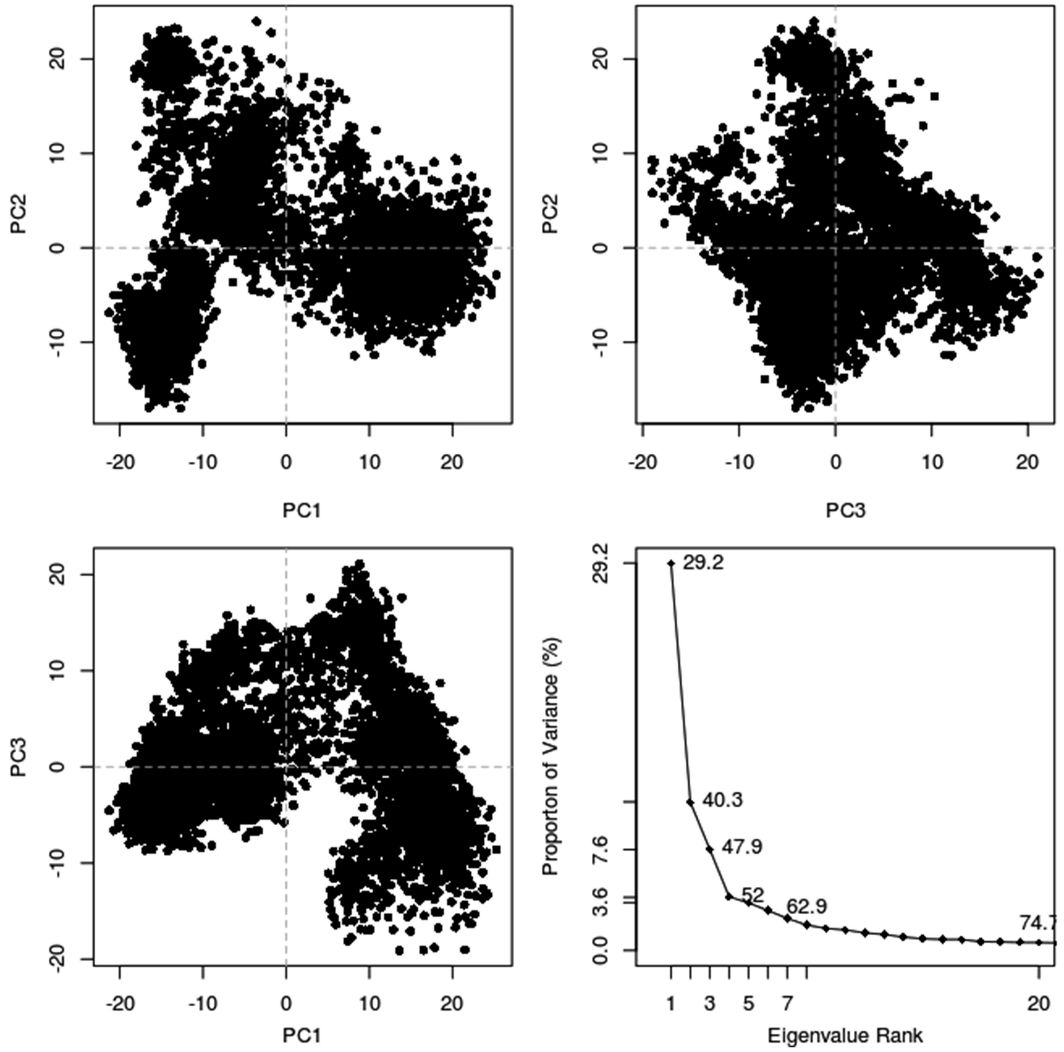


Fig. 5 Bio3D plot of principal component data

The *pca.xyz* command calculated a matrix with as many principal components as number of data frames entered into it, but the question of how many principal components to use remains. One tool for determining how many to use is the scree plot which shows how much of the data is captured by each principal component, but for the LeuT analysis described here, less than 50 % of the variance is captured after the three most influential principal components. The RMSF can also be utilized in order to determine whether all of the captured fluctuations within those first three components is a sufficient representation of the structural changes involved.

The RMSF can be used to validate the principal components by overlaying the RMSF with the *$au* object from the PCA calculation. The *$au* object is the atomic vector which shows how much each residue contributes to a principal component. The goal is to

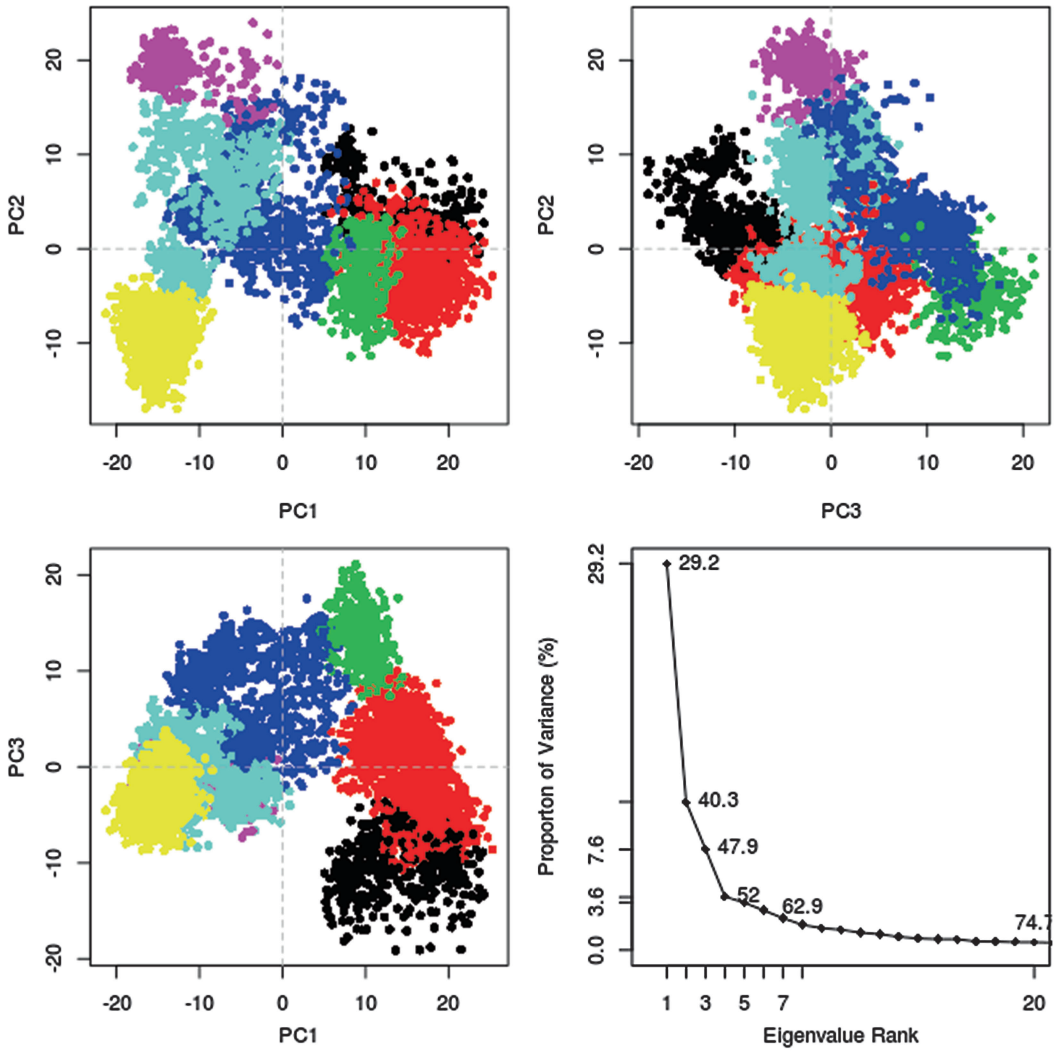


Fig. 6 Bio3D plot of PCA data colored by cluster after calculation of cluster groups

obtain a graph where each residue's RMSF value is captured by at least one of the selected principal components. If a residue shows on the RMSF but is not captured in one of the *$au* objects, then another principal component will need to be used (Fig. 7).

```
   barplot(rf,col="purple",border="purple")
   par(new=TRUE)
   plot(trj.pca$au[,1],type="l",col="orange",
lwd=3)
   points(trj.pca$au[,2],type="l",col="green",
lwd=3)
   points(trj.pca$au[,3],type="l",col="skyblue
",lwd=3)
```

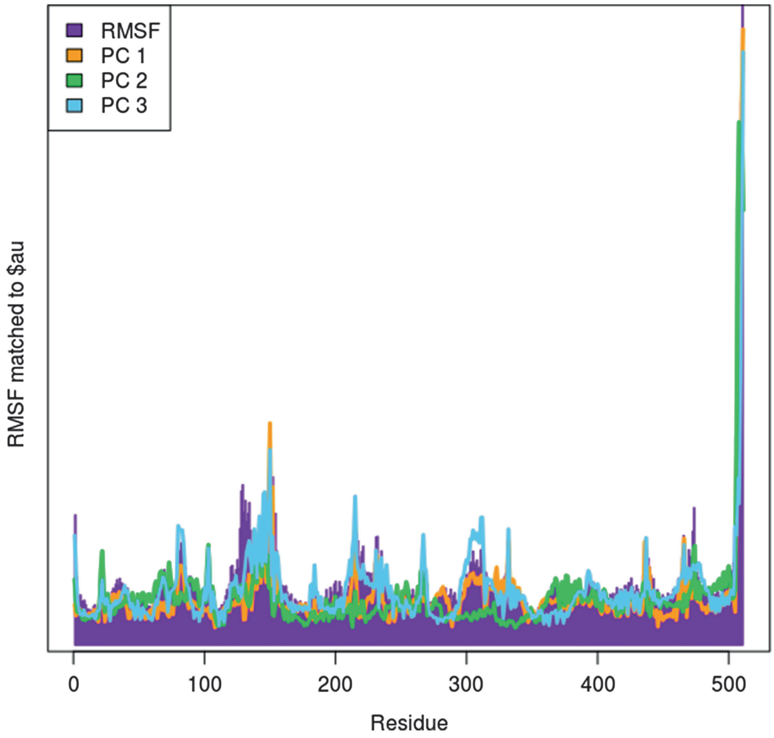


Fig. 7 Root mean square fluctuations graph superposed with the $au vectors representing the amount of variance captured by each principal component. This graph has all residues included, and the carboxy terminus appears to dominate the principal component

```
   legend.labels = c("RMSF", "PC 1", "PC 2", "PC 3")
   legend.cols = c("purple", "orange", "green",
"skyblue")
   legend("topleft",legend.labels, fill = leg-
end.cols)
```

Looking at Fig. 7 from the above example, sometimes a single residue or group of residues may dominate the entire PCA and prevent the PCA from capturing other residue fluctuations. In the LeuT example here, it appears that the last few residues of TM12 at the carboxy terminus of the protein contribute a large portion of the PC. However, these residues mostly just vibrate in the cytosol, and we can therefore exclude these residues from the analysis and rerun the PCA. This can be done by editing the indices used with the atom.select command. Then check the *$au* object with the RMSF again (Fig. 8).

```
   ind.adj  <-  atom.select(pdb,elety="CA",re
sno=1:505)
   trj.fit.adj  <-  fit.xyz(pdb$xyz[ind.adj$xyz],
dcd[,ind.adj$xyz])
   trj.pca.adj <- pca.xyz(trj.fit.adj)
   rf.adj <- rmsf(trj.fit.adj)
```

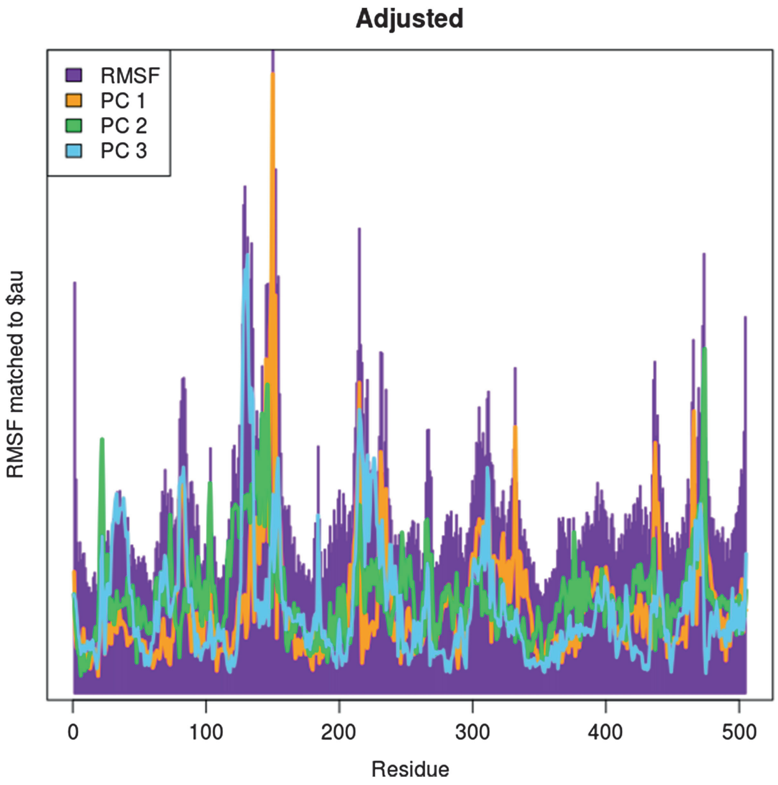


Fig. 8 RMSF graph superposed with the $au vectors of the principal components after adjusting the PCA by removing residues 506–512

```
   barplot(rf.adj,col="purple",border="purple"
,main="Adjusted")
   par(new=TRUE)
   plot(trj.pca.adj$au[,1],type="l",col="orang
e",lwd=3)
   points(trj.pca.adj$au[,2],type="l",col="gre
en",lwd=3)
   points(trj.pca.adj$au[,3],type="l",col="sky
blue",lwd=3)
```

Once a suitable RMSF plot has been obtained, the PCA can be reclustered and replotted to visualize the new PCA result (Fig. 9).

```
   d <- dist(trj.pca.adj$z[,1:3])
   hc <- hclust(d)
   grps <- cutree(hc,k=7)
   plot(trj.pca.adj,col=grps)
```

The fluctuations captured by each principal component can also be visualized as a trajectory in VMD. Use the *mktrj.pca* command to generate a VMD trajectory for visualizing captured fluctuations in each principle component desired. Load the PDB file into VMD and change the representation to *Trace* or *Tube* since the PCA only printed out the alpha carbons.

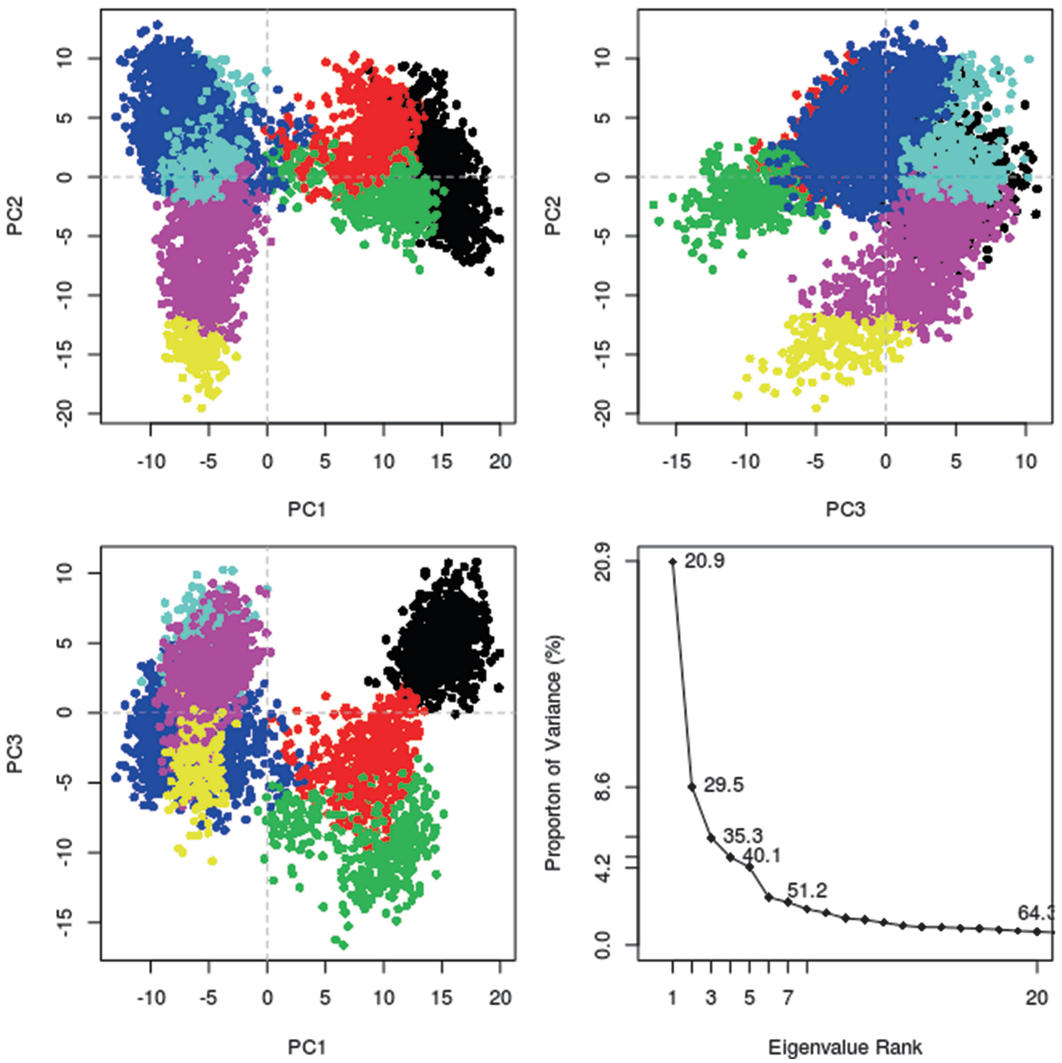


Fig. 9 Adjusted PCA plot of LeuT with residues 506–512 removed and each point colored by cluster

```
mktrj.pca(trj.pca,pc=1,mag=1, file="PC1.pdb")
```

As mentioned above in the background, it may be necessary to reweigh the PCA plots to determine the validity of each point. The following steps are necessary in order to accomplish this:

1. Extract the $\beta\Delta V[\mathbf{r}(t_i)]$ for each structural point from the simulation log files using text manipulation tools like grep and awk.
2. Load these values into R.
3. Calculate $e^{\beta\Delta V[\mathbf{r}(t_i)]}$ for the vector.
4. Plot any two desired principal components against each other and generate a contour plot using the $e^{\beta\Delta V[\mathbf{r}(t_i)]}$ vector as a coloring parameter (or any other desired method of plotting three dimensional data in R).

3.16 Compare the Resulting Conformations with Additional Structural Data

Additional structural coordinates, such as various crystallized LeuT structures thought to be representative of portions of the substrate transport cycle, can be projected into the PCA space using the homology functionality within Bio3D. To start, read the structures to be projected onto the PCA into R and align to the reference PDB.

```
crys <- pdbaln(c(pdb,other pdb files))
```

The alignment of the crystal files and the alignment of the trajectory files must match. Unfortunately, since the alignment changes the index values of the reference PDB to match the sequence, it is necessary to match the index values of the aligned sequence to the trajectory. There is a function (*pdb2aln*) to do this planned for new releases of Bio3D; however, at the time of this writing, this function is not yet available in the general Bio3D package.

Gap inspection (for missing residues) may be necessary using the *gap.inspect* command. In this case, the PCA may need to be recalculated if residues previously used have gaps. One way to avoid having to recalculate the PCA is to do the PCA of the crystals alone and project the trajectory into the crystal PCA space. The following example will assume projecting the crystals onto the trajectory PCA space.

1. Run a fit to the reference PDB:

```
crys.fit <- fit.xyz(crys$xyz[1, any index values
such as gap positions or specific residues], crys
$xyz[2:length(crys$xyz[,1]),same index values])
```

2. Project the crystals into the trajectory PCA space:

```
crys.pca <- pca.project(crys.fit, trj.pca)
plot(trj.pca)
points(crys.pca[,1],crys.pcs[,2], col= "black")
```

References

1. Lindorff-Larsen K, Piana S, Dror RO, Shaw DE (2011) How fast-folding proteins fold. Science 334(6055):517–520. doi:10.1126/science.1208351
2. Shaw DE, Maragakis P, Lindorff-Larsen K, Piana S, Dror RO, Eastwood MP, Bank JA, Jumper JM, Salmon JK, Shan Y, Wriggers W (2010) Atomic-level characterization of the structural dynamics of proteins. Science 330(6002):341–346. doi:10.1126/science.1187409
3. Genchev GZ, Kallberg M, Gursoy G, Mittal A, Dubey L, Perisic O, Feng G, Langlois R, Lu H (2009) Mechanical signaling on the single protein level studied using steered molecular dynamics. Cell Biochem Biophys 55(3):141–152. doi:10.1007/s12013-009-9064-5
4. Baker JL, Biais N, Tama F (2013) Steered molecular dynamics simulations of a type IV pilus probe initial stages of a force-induced conformational transition. PLoS Comput Biol 9(4):e1003032. doi:10.1371/journal.pcbi.1003032
5. Forti F, Boechi L, Estrin DA, Marti MA (2011) Comparing and combining implicit ligand sampling with multiple steered molecular

dynamics to study ligand migration processes in heme proteins. J Comput Chem. doi:10.1002/jcc.21805

6. Li W, Shen J, Liu G, Tang Y, Hoshino T (2011) Exploring coumarin egress channels in human cytochrome P450 2A6 by random acceleration and steered molecular dynamics simulations. Proteins 79(1):271–281. doi:10.1002/prot.22880
7. Shen M, Guan J, Xu L, Yu Y, He J, Jones GW, Song Y (2012) Steered molecular dynamics simulations on the binding of the appendant structure and helix-beta2 in domain-swapped human cystatin C dimer. J Biomol Struct Dyn 30(6):652–661. doi:10.1080/07391102.2012.689698
8. Xu L, Hasin N, Shen M, He J, Xue Y, Zhou X, Perrett S, Song Y, Jones GW (2013) Using steered molecular dynamics to predict and assess Hsp70 substrate-binding domain mutants that alter prion propagation. PLoS Comput Biol 9(1):e1002896. doi:10.1371/journal.pcbi.1002896
9. Hamelberg D, Mongan J, McCammon JA (2004) Accelerated molecular dynamics: a promising and efficient simulation method for biomolecules. J Chem Phys 120(24):11919–11929. doi:10.1063/1.1755656
10. Bucher D, Grant BJ, Markwick PR, McCammon JA (2011) Accessing a hidden conformation of the maltose binding protein using accelerated molecular dynamics. PLoS Comput Biol 7(4):e1002034. doi:10.1371/journal.pcbi.1002034
11. Grant BJ, Gorfe AA, McCammon JA (2009) Ras conformational switching: simulating nucleotide-dependent conformational transitions with accelerated molecular dynamics. PLoS Comput Biol 5(3):e1000325. doi:10.1371/journal.pcbi.1000325
12. Mucksch C, Urbassek HM (2013) Enhancing protein adsorption simulations by using accelerated molecular dynamics. PLoS One 8(6):e64883. doi:10.1371/journal.pone.0064883
13. de Oliveira CA, Grant BJ, Zhou M, McCammon JA (2011) Large-scale conformational changes of Trypanosoma cruzi proline racemase predicted by accelerated molecular dynamics simulation. PLoS Comput Biol 7(10):e1002178. doi:10.1371/journal.pcbi.1002178
14. Salmon L, Pierce L, Grimm A, Ortega Roldan JL, Mollica L, Jensen MR, van Nuland N, Markwick PR, McCammon JA, Blackledge M (2012) Multi-timescale conformational dynamics of the SH3 domain of CD2-associated protein using NMR spectroscopy and accelerated molecular dynamics. Angew Chem 51(25):6103–6106. doi:10.1002/anie.201202026
15. Thomas JR, Gedeon PC, Grant BJ, Madura JD (2012) LeuT conformational sampling utilizing accelerated molecular dynamics and principal component analysis. Biophys J 103(1):L1–L3. doi:10.1016/j.bpj.2012.05.002
16. Phillips JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, Chipot C, Skeel RD, Kale L, Schulten K (2005) Scalable molecular dynamics with NAMD. J Comput Chem 26(16):1781–1802. doi:10.1002/jcc.20289
17. Wang Y, Harrison CB, Schulten K, McCammon JA (2011) Implementation of accelerated molecular dynamics in NAMD. Comput Sci Discov 4(1):pii: 015002, doi:10.1088/1749-4699/4/1/015002
18. Case DA, Darden TA, Cheatham TE, Simmerling CL, Wang J, Duke RE, Luo R, Walker RC, Zhang W, Merz KM, Roberts B, Hayik S, Roitberg A, Seabra G, Swails J, Goetz AW, Kolossváry I, Wong KF, Paesani F, Vanicek J, Wolf RM, Liu J, Wu X, Brozell SR, Steinbrecher T, Gohlke H, Cai Q, Ye X, Wang J, Hsieh MJ, Cui G, Roe DR, Mathews DH, Seetin MG, Salomon-Ferrer R, Sagui C, Babin V, Luchko T, Gusarov S, Kovalenko A, Kollman PA (2012) AMBER 12. University of California, San Francisco
19. Pierce LC, Salomon-Ferrer R, Augusto FOC, McCammon JA, Walker RC (2012) Routine access to millisecond time scale events with accelerated molecular dynamics. J Chem Theory Comput 8(9):2997–3002. doi:10.1021/ct300284c
20. Voter AF (1997) Hyperdynamics: accelerated molecular dynamics of infrequent events. Phys Rev Lett 78(20):3908–3911
21. Voter AF (1997) A method for accelerating the molecular dynamics simulation of infrequent events. J Chem Phys 106(11):4665–4677. doi:10.1063/1.473503
22. Steiner MM, Genilloud PA, Wilkins JW (1998) Simple bias potential for boosting molecular dynamics with the hyperdynamics scheme. Phys Rev B 57(17):10236–10239
23. Rahman JA, Tully JC (2002) Puddle-skimming: an efficient sampling of multidimensional configuration space. J Chem Phys 116(20):8750–8760. doi:10.1063/1.1469605
24. Shen T, Hamelberg D (2008) A statistical analysis of the precision of reweighting-based simulations. J Chem Phys 129(3):034103. doi:10.1063/1.2944250
25. Xin Y, Doshi U, Hamelberg D (2010) Examining the limits of time reweighting and Kramers' rate theory to obtain correct kinetics from accelerated

molecular dynamics. J Chem Phys 132(22):224101. doi:10.1063/1.3432761

26. Gedeon PC, Indarte M, Surratt CK, Madura JD (2010) Molecular dynamics of leucine and dopamine transporter proteins in a model cell membrane lipid bilayer. Proteins 78(4):797–811. doi:10.1002/prot.22601

27. Pettersen EF, Goddard TD, Huang CC, Couch GS, Greenblatt DM, Meng EC, Ferrin TE (2004) UCSF Chimera—a visualization system for exploratory research and analysis. J Comput Chem 25(13):1605–1612. doi:10.1002/jcc.20084

28. Sali A, Blundell TL (1993) Comparative protein modelling by satisfaction of spatial restraints. J Mol Biol 234(3):779–815. doi:10.1006/jmbi.1993.1626

29. Humphrey W, Dalke A, Schulten K (1996) VMD: visual molecular dynamics. J Mol Graph 14(1):33–38, 27-38

30. Grant BJ, Rodrigues AP, ElSawy KM, McCammon JA, Caves LS (2006) Bio3d: an R package for the comparative analysis of protein structures. Bioinformatics 22(21):2695–2696. doi:10.1093/bioinformatics/btl461

31. Yamashita A, Singh SK, Kawate T, Jin Y, Gouaux E (2005) Crystal structure of a bacterial homologue of Na+/Cl−-dependent neurotransmitter transporters. Nature 437(7056):215–223. doi:10.1038/nature03978

32. Shen MY, Sali A (2006) Statistical potential for assessment and prediction of protein structures. Protein 15(11):2507–2524. doi:10.1110/ps.062416606

33. Aksimentiev A, Sotomayor M, Wells D (2012) Membrane proteins tutorial. Theoretical and Computational Biophysics Group, University of Illinois at Urbana-Champaign, Champaign

Chapter 13

Simulations and Experiments in Protein Folding

Giovanni Settanni

Abstract

The interplay between simulations and experiments of protein folding has largely contributed to the elucidation of many important aspects of the phenomenon. In this chapter, I briefly describe the experiments which provide information on the kinetics of the protein folding process, and help to characterize the folding transition state. Then, I show how to probe the kinetics of protein folding using molecular dynamics simulations, how to compare the simulations with the experiments and how to help and rationalize the latter, ultimately offering a molecular picture of the process. After the production of suitable molecular dynamics simulation data in the form of trajectories, the procedure involves sequentially the identification of the stable states of the protein, the identification of the transition pathways connecting the stable states, the identification of the transition state conformations, comparison with experimental results, and finally, the identification of the molecular determinants or reaction coordinates of the folding process, that is, the features that clearly help distinguishing the transition state from the stable states.

Key words Kinetics, Transition state, Committor, Phi-value, Clustering, Kinetic network

1 Introduction

Since its discovery [1], the phenomenon of reversible folding of proteins has engaged generations of scientists from many different fields. After decades of research the emerging picture often used to describe the protein folding process, at least in its simplest declination, is that of a first order phase transition between the native state of the protein, and the unfolded/denatured state, the former being enthalpically stabilized by favorable interactions internal to the protein chain and with the solvent, the latter being entropically stabilized by the large number of conformations that the protein chain can sample within this state.

This rationalization of the protein folding process, led to more focused attempts to identify the determinants of the process. Indeed, a first order phase transition implies the crossing of a free energy barrier, which, then, represents a bottleneck in the transition process. The height of the free energy barrier, which can be measured by experiments on the folding kinetics, and the way it changes

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_13, © Springer Science+Business Media New York 2015

depending on the various components of the protein-solution system (protein sequence, protein structure, environmental conditions, etc.) has been used to assess which elements play a major role and how that happens. In this framework a special role is played by phi-value analysis.

In phi-value analysis [2, 3], the structure of a given protein is perturbed by the introduction of conservative point mutations along the sequence and the effect of the perturbation on the folding free energy barrier is assessed by a combination of equilibrium and kinetic experiments. The equilibrium experiments are used to measure with high accuracy the stability of wild-type and mutant protein. The kinetics experiments are used to assess the folding rate, which is related to the height of the free energy barrier through an Arrhenius relation $k_{\mathrm{obs}} \propto \exp - \Delta G_{\ddagger-U}/k_{\mathrm{b}}T$, where k_{obs} is the observed folding rate, $\Delta G_{\ddagger-U}$ is the height of the free energy barrier for folding, i.e., the free energy difference between the transition and unfolded state, k_{b} is the Boltzmann constant, T the temperature. The folding phi-value of the mutated amino acid is thus the ratio between the change in height of the free energy barrier for folding and the change of the protein stability upon mutation.

$$\Phi_{\mathrm{f}} = \frac{\Delta G^{\mathrm{mut}}_{\ddagger-\mathrm{U}} - \Delta G^{WT}_{\ddagger-\mathrm{U}}}{\Delta G^{\mathrm{mut}}_{F-U} - \Delta G^{WT}_{F-U}} = \frac{\Delta\Delta G_{\ddagger-U}}{\Delta\Delta G_{F-U}} = \ln\left(\frac{k^{\mathrm{mut}}_{f}}{k^{WT}_{f}}\right) / \, \Delta\Delta G_{F-U} \qquad (1)$$

where ΔG_{F-U} is the folding free energy, i.e., the free energy difference between folded and unfolded state (*see* **Note 1**). In the classical interpretation, phi-values close to one mean that the mutation has produced the same effect on the free energy of the transition and folded state (the free energy of the unfolded state is used as reference), thus, the conformation of the mutated residue must be similar in the two states. On the other hand, a phi-value close to zero means that the mutation affected mostly the free energy of the folded state and not the transition state, thus the conformation of the residue in the transition state must be similar to its unfolded state. Intermediate and non-classical ($\Phi < 0$ or $\Phi > 1$) phi-values cannot be interpreted straightforwardly and each case requires specific attention.

Systematic measurements of phi-values of many amino acids, sometime termed alanine/glycine scans, are being made available for many proteins. These measurements help to identify either which amino acids take part in the formation of a folding nucleus around which the overall folding process takes place (the amino acids with large Φ), or the presence of a diffuse transition state, where there are no amino acids with large Φ. Early results [4] have shown that in many cases it is possible to correlate the phi-value of an amino acid to the fraction of native atomic contacts formed at the transition states. This relationship offers both a way to test simulations, and to interpret simulation data where experimental data are not available [5].

Simulations allow for the interpretation of phi-value data in terms of molecular structures of the various states of the protein. Indeed, simulations, after being validated with experimental data, allow, for example, for verifying which interactions are present in the transition state vs. the native state or for investigating the presence of residual structure in the unfolded state, that is, interactions that are present also in the unfolded state [5–9]. Ultimately, simulations can be used to find out which structural observables represent good reaction coordinates for the folding process, in other words, the observables able to distinguish the transition state from the native and from the unfolded states [10]. The simulation approach, however, is hampered by the elusive nature of the transition state, which, being a saddle point in the free energy landscape of the protein, with considerably higher free energy than the stable states, is not normally densely populated and/or easy to sample.

Although nowadays there exist specialized hardware capable of simulating classical atomistic models of relatively small proteins with the surrounding solvent molecules up to the millisecond time scale [11] (a fully quantum-mechanical treatment of the folding dynamics of proteins is at present unthinkable), the time scales of the folding process of proteins, sometimes reaching hours, are often too large for such an approach on standard high performance hardware (i.e., linux clusters, gpu workstations). For this reason, a large variety of methods have been devised to overcome this barrier. One possible approach is the use of simplified coarse-grained force fields, where interactions between atoms are substituted by a potential function designed to have a deep minimum corresponding to the native state of the protein. These models, referred to generally as Gō models [12], have been shown to reproduce some important characteristics of the folding process of proteins [7, 8, 13–15]. These characteristics depend for a large part on the topology of the protein, i.e., the way each amino acid interacts with the other amino acids along the sequence. The scope of these models, however, does not allow for addressing interesting questions about non-native interactions, which are excluded a priori, or the specific contribution to the interactions due to the chemical nature of each group of atoms, which is not preserved in these models. For this reason I will focus on classical atomistic models of proteins and peptides and in particular on the use of these models in molecular dynamics simulations of the folding/unfolding process, aimed to reveal the structure of the transition state ensemble.

In what follows, I give an overview of the steps necessary to (a) produce a molecular dynamics simulation suitable to study the folding process of a protein system; (b) identify the stable states, and transition events; (c) identify the transition states; (d) compare with experimental data and find good reaction coordinates for folding. These various steps will be exemplified by means of two test cases.

2 Materials

All the simulations and analysis are performed on Linux workstations/clusters. In case study 1 simulations are performed using the program GROMACS [16]. In case study 2 CHARMM [17] is used to run the simulations. In either case, analysis of the trajectories is performed using the program WORDOM [18]. Shell scripts and AWK scripts are used for data analysis. The programs GNUPLOT (http://www.gnuplot.info) and GRACE (http://plasma-gate.weizmann.ac.il/Grace/) are used for plotting graphs. The program VMD [19] is used for visualization of protein structures and trajectories. The native conformation of the simulated Trpz1 peptide is obtained from the Protein Data Bank, with pdbid 1LE0 [20].

3 Methods

3.1 Models, Force Fields, Simulations Techniques

The aim of our simulations is to collect as many folding/unfolding transition events as possible, which provide the largest part of the information we need to characterize the transition state. These events are rare and may occur on time scales that cannot be sampled by standard molecular dynamics simulations at constant temperature. In our two test cases we have adopted several expedients to overcome this problem.

Case study 1: [10] High temperature is used to speed up the unfolding process (*see* **Note 2**). The Trpz1, a 12-residue peptide, is simulated at high temperature using a standard force field OPLSAA [21]. Several unfolding simulations are collected at each temperature starting from the experimental native state with different initial velocities. Each of the simulations must be long enough to show at least an unfolding event. In the case of Trpz1, which is a very small peptide, this occurs within 100 ns at most of the simulated temperatures (200 ns at 375 K). Other simulations parameters follow. The peptide is immersed in a periodic cubic simulation box leaving at least 8 Å between the peptide and the box boundaries. The cutoff for van der Waal and electrostatic interactions is set to 10 Å. The box is filled with 1666 SPC water molecules [22]. Simulations are run at several temperatures between 375 and 450 K and pressure is set to 1 atm (*see* **Note 3**). Temperature and pressure are regulated by the Berendsen algorithm [23]. The time step is set to 2 fs and all covalent bonds are kept rigid using the LINCS algorithm [24]. Snapshots of the coordinates of the system are saved every 20 ps into trajectory files.

Case study 2: [25] The effects of the aqueous solvent are modeled by a mean-field approximation, avoiding the explicit representation of solvent molecules in the simulations. The same peptide Trpz1 is simulated using the charmm19 [17] united atom force field (i.e., only polar hydrogen atoms are explicitly represented) close to the transition temperature of the peptide (330 K). The SASA model is adopted [26] as a mean-field approximation of the solvent. The lack of friction due to the absence of explicit water molecules helps to speed up the folding/unfolding kinetics of the peptide without significantly affecting the thermodynamics of the system [27]. The temperature is kept constant by the Berendsen thermostat [23]. The SHAKE algorithm [28] is used to fix covalent bond lengths, allowing for a 2 fs time step. In about 20 μs of cumulated trajectories from ten different simulation runs, it is possible to collect hundreds of folding/unfolding events. The coordinates of the system are saved every 20 ps into trajectory files.

In both case studies, the production runs are preceded by a minimization, heating and equilibration phase. The simulations are relatively straightforward and do not require specially designed input files; we refer to Chapter 1 of this book or the CHARMM and GROMACS online documentation for their preparation. In either case, at the end of the simulations, the resulting trajectory files (either in .xtc format or in .dcd format, for GROMACS and CHARMM, respectively) need to be listed along with their complete path in a file called trjlist.txt.

3.2 Identifying the Stable States of the System

Transition events between the stable states of the system can be identified only after the stable states of the system have been identified. To this extent, a set of observables relevant for the description of the protein/peptide are selected and the time series of these observables are measured along the trajectories. In the case of the Trpz1 peptide, the root mean square deviation (RMSD) of the Cα atoms from the native conformation and the number of native hydrogen bonds formed along the backbone (HB) represent relevant observables. Many ways are available to perform these measurements. I will show how to use the program WORDOM (http://wordom.sourceforge.net/), which is particularly fast and easy to handle. WORDOM analysis can be invoked using the following syntax:

```
$ wordom -iA <inputfile>.wrd -imol trpz.pdb -itrj trjlist.txt > <output>.dat
```

where the wordom input file .wrd contains the following for RMSD calculations:

```
BEGIN    rmsd         #specify which observable needs to be measured (RMSD in this case)
--TITLE  carmsd       #an arbitrary title for the calculation
--SELE  /*/*/CA       #the selection of atoms for the calculation /<segid>/<resid>/<atomname>
#                     #in this case all the CA atoms are selected
END
```

or the following for HB calculations:

```
BEGIN contacts        #compute contacts
--TITLE HB            #on output column HB lists for each snapshot the number of the
#                     #contacts satisfying the selection conditions listed below
--SELE /*/1/O : /*/12/H : 2.6#contact selection condition: contact if distance between atom O
#                            #on residue 1 and H on residue  12 is smaller than 2.6 Angstrom
--SELE /*/3/H : /*/10/O : 2.6
--SELE /*/3/O : /*/10/H : 2.6
--SELE /*/5/H : /*/8/O : 2.6
--SMOOTH              #replaces the classical contact step function with a smooth sigmoidal
#                     function
END
```

In the output of these calculations, the time step along the trajectory and the observable of the corresponding conformation are listed.

A further important analysis that needs to be carried out at this point is a cluster analysis. In cluster analysis, similar conformations observed along the trajectory are grouped together into clusters. This will help to understand which conformations are more populated, and how the trajectory flows through these sets of conformations. The degree of similarity can be measured using several metrics, e.g., RMSD, distance-RMSD, contact-map difference etc. Here we adopt the distance-RMSD (dRMSD) of Cα and Cβ atoms, that is, for each conformation we compute the distances between all the Cα and Cβ atoms and store them in a matrix, then the dRMSD between two conformations is the Euclidean distance between the corresponding distance matrices. The clustering is performed using the "leader" algorithm [29] (alternatively, many other algorithms can be used). Briefly, the conformations along the trajectory are sequentially analyzed. If the analyzed conformation is within a cutoff distance from any of the cluster centers already identified, then it is added to that cluster, otherwise it is used to identify the center of a new cluster. WORDOM can be used for the clustering, the input file in this case will be:

```
BEGIN cluster         # request clustering calculation
--TITLE c1            # arbitrary title
--SELE /*/*/C[A|B]    # select all the CA and CB atoms
--DISTANCE drms       # the distance-RMSD is used for calculations
--METHOD leader       # the leader algorithm is used
--CUTOFF 0.8          # the clustering cutoff in Angstrom
END
```

The cutoff for clustering must be chosen small enough so that conformations within the same cluster belong to the same free energy minimum. At the same time the cutoff must not be too small otherwise we will obtain many clusters containing very few conformations, and this may negatively affect our subsequent calculations (*see* **Note 4**). In the output of this calculation, the time step of each trajectory snapshot is listed along with its corresponding cluster center.

Depending on the algorithm used for clustering, the volume of each cluster in conformation space may vary (*see* **Note 5**). Assuming that it fluctuates around a certain fixed value, we expect that when the trajectory visits an energy-stabilized free energy minimum (i.e., the native state) the volume of the sampled conformation space, i.e., the sum of the volume of all the clusters visited so far, will not increase too much, as the system fluctuates around the same minimum conformation. On the contrary, when the trajectory will explore an entropy-stabilized free energy minimum (i.e., the unfolded state), the sampled conformation space volume will increase significantly, as it will be very unlikely, given the high dimensionality of the conformation space, that very similar conformations are sampled repeatedly in such a high-entropy state and it will be

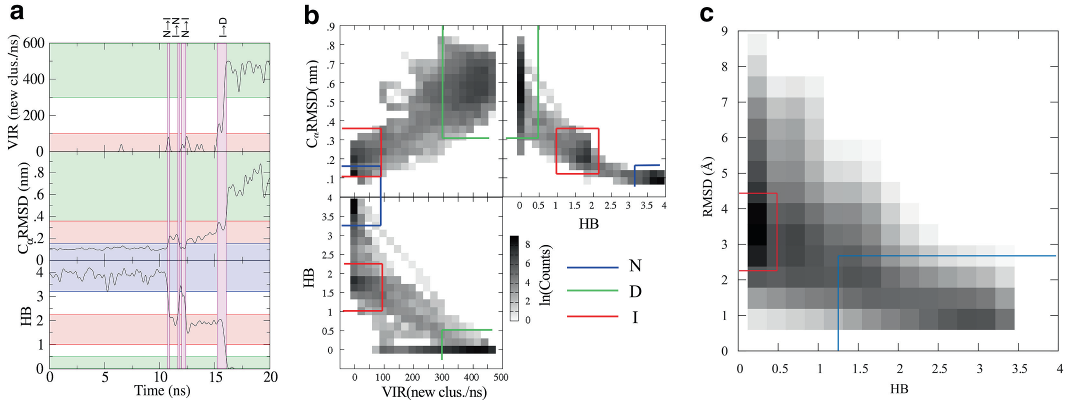


Fig. 1 (**a**) representative time series of the observables HB, RMSD and VIR for case study 1, simulations at 450 K. The horizontal lines mark the boundaries of each state as identified from the histograms of the observables. These boundaries mark the starting and ending points of transition pathways (regions shaded in magenta). (**b**) Histogram of the observables from case study 1, simulations at 450 K. *Lines* around the populated regions indicate the boundaries of the stable states of the peptide. Adapted from ref. 10 with permission from Elsevier. (**c**) Histogram of the observables for case study 2. The boundaries of the native (*blue*) and denatured (*red*) state are shown

also unlikely that we have sampled this state to saturation. The "leader" clustering algorithm, then, where new clusters are defined as the trajectory visits conformations further away from those already visited, provides a natural way to quantify this. It allows us to define a further observable linked to the speed of conformational sampling: the volume increase rate (VIR), i.e., the number of new clusters observed in the unit time, which is supposedly large in the unfolded state and small in the native state (*see* **Note 6**).

Histograms of the observables allow for the identification of separate populated regions, which correspond to the stable states of the peptide (Fig. 1b, c). The two dimensional histograms can be produced by pasting the time series of the observables (Fig. 1a) in a single file and then binning the data in 2D bins (e.g., using an AWK script). The binned data can be plotted using GNUPLOT in pm3d mode. Alternatively, 1D histograms can be produced using the histogram analysis tool in GRACE. In case study 1 (Fig. 1b), three populated states can be identified, the fully native N, the denatured state D and an intermediate state I. In case study 2, two broadly populated states have been identified, the native and the unfolded state (*see* **Note 7**). The minor differences between the stable states in the two case studies are due to the different simulation setups (different temperatures and force field).

It is also convenient to store in separate trajectory files all the snaphots of the original trajectory corresponding to each identified state. This can be done using WORDOM with the following syntax:

```
$ wordom -F listA.F -imol trpz.pdb -itrj trjlist.txt -otrj stateA.dcd
```

where the listA.F file is just a list of snapshots corresponding to the state A. The listA.F file can be generated using a simple AWK script to parse the time series of the observables and select those snapshots that are within the boundaries of the required state. The output file stateA.dcd is a standard .dcd trajectory file listing all the snaphots corresponding to the state (obviously, the chronological information in those trajectory files have no meaning).

3.3 Identifying Folding/Unfolding Transition Events

Transition events occur when the trajectory leaves a stable state and reaches a different stable state. To quantitatively capture these events, we draw a border line enclosing the most populated part of each stable state as it occurs in the simulations (Fig. 1b, c) and then we look for events where the trajectory crosses the border of one state in the outwards direction and the border of another state in the inward direction (Fig. 1a). Is it possible that the chosen observables do not describe the system optimally. In that case fast "recrossing" events may be observed, i.e., the system may leave the region of observables' space corresponding to one state, reach the region belonging to another state and quickly go back in the initial state. These events are not true transitions, they are due to the inability of the observables to completely separate the two states. The use of several independent observables and a conservative choice of the borders of the states so that only the most populated regions are enclosed, minimizes the probability to observe recrossings.

In case study one, where we identified 3 stable states, in principle we could observe six different transition events, i.e., from N to I, and back, from I to D and back and from N to D and back. In reality, we observe only four kinds of transitions N to I and back and I to D and back, although we have very few representatives of the backward (refolding) transitions. Analysis of the average transition times shows that the N to I transition is the rate limiting step, for this reason, in what follows, we will focus on the determination of the transition state for the N to I transition.

3.4 Identifying the Transition State Ensemble

For the identification of the transition state ensemble (TSE) in a two-state transition it is useful to introduce the concept of committor p_c of a conformation, i.e., the probability that the simulations starting from the conformation reach one state before the other [30]. The TSE is defined as the set of conformations that are equally committed to end in one state or the other, so their committor (sometime called folding probability) p_c equals approximately 0.5. In other words, simulations starting from conformations of the TSE have a 50 % probability to reach one of the two states before the other. Because of this definition, the committor represents an optimal reaction coordinate, in the sense that it allows to distinguish each phase of the transition process, that is, the "reactant" state, the "product" state and the transition state in between. The committor of conformations sampled along our simulations can be computed by starting many additional simulations from the given conformation and counting the fraction that reaches one of the states before the other. These calculations are computationally very expensive as we need at least 20 trial runs for each conformation to obtain a relatively precise estimate of p_c (*see* **Note 8**) and each run needs to be long enough to allow the system to commit to either state. Strategies need to be employed to reduce as much as possible those calculations.

Strategy 1: The committor is either precisely 0 or 1 for conformations in the unfolded or folded state, respectively. It is different from those values only for conformations along transition pathways and it has generally a monotonic behavior along these pathways [31]. In case study 1 we can limit the committor calculations to few conformations selected along the transition pathways and use a binary search approach to converge to those conformations with committor ~0.5 (i.e., the TSE). In practice, this approach requires a Linux shell script and can be solved stepwise:

(a) Select the conformation for committor calculation along transition pathways using a binary search strategy, targeting conformations with committor ~0.5 (i.e., select a conformation at the midpoint of the trajectory between two conformations one with $p_c > 0.5$ and the other with $p_c < 0.5$).

(b) Extract the selected conformation from the trajectory and use it to start 20 (or more) short (2 ns for Trpz1) simulations.

(c) Measure the observables (RMSD, HB, VIR) along the committor simulations and use them do determine which stable state was reached first.

(d) Evaluate the committor as the fraction of the simulations that reached one of the stable states first (*see* **Note 9**). Go back to (a).

Strategy 2: Instead of computing the committor for single conformations, we can assume that similar conformations (i.e., those with a small pairwise dRMSD) have also similar committors. Then, we can measure the committor of the clusters which will approximate those of their member conformations [32]. We do that without starting any new committor calculation, rather we use the simulation data that we have already collected [33]. The latter must contain information about a sufficient number of transition events in order to get a precise estimate of the cluster committors. We adopt this strategy in case study 2. In practice, this approach requires the following steps:

(a) The data about cluster membership and observables (RMSD, HB) along the trajectory are placed in a single file.

(b) A script is made (e.g., using awk) that analyzes the file from the previous step and assesses which stable state is reached first along the trajectory after it leaves each conformation.

The script also counts the fraction of the conformations in each cluster that reach one state before the other. This is the cluster committor.

(c) The cluster committor is finally printed along the trajectory, i.e., each snapshot is assigned with its cluster committor.

At the end of these calculations, in both case study 1 and 2, we obtain the committor for at least several relevant conformations along the transition pathways. In particular, with both strategies we end up with a set of conformations with a committor ~0.5 (Fig. 2).

3.5 Validating and Characterizing the Transition State Ensemble

In case study 1, we have measured the committor for few conformations along the transition pathways then we need to find out, by analyzing those conformations, if there are observables that may correlate with the committor. We identified several possible observables and tested them by measuring the degree of correlation with the committor (Fig. 3). In practice, this can be achieved by:

(a) Using wordom or VMD to measure possible observables along the trajectory.

(b) Extracting (e.g., using awk) the value of the observables for the conformations with known committor.

(c) Plotting the committor vs the observable.

In Fig. 3 only a fraction of all the tested observables is shown. Visual inspection of the structures with $p_c \sim 0.5$ and comparison with those from the two stable states must be used to guide the selection of the relevant observables. As it is often the case, the observable that best represents the transition is a complex combination of factors that describes the reciprocal position of several groups of atoms. In the case of Trpz1 we found that the reciprocal position of the four tryptophan side chains is crucial in determining the stage of the transition. When the distance between the side chains of Trp9 and Trp4 becomes smaller than the distance between Trp4 and Trp2, then the committor increases. In other words, when the Trp side chains align with each other as in the native state, although the packing may not yet be native, then the committor increases and those conformations are more likely to fold rather than unfold.

In case study 2, we have measured an approximate committor $\acute{p}_c$ for all the sampled conformations, then we can test if this approximation represent a good reaction variable. To this extent we adopt the procedure suggested by Hummer and coworkers [34] where we compute the conditional probability $p(TP|\acute{p}_c)$ that conformations with a given $\acute{p}_c$ belong to a transition pathway. In practice, we need to:

(a) Combine the $\acute{p}_c$ data with the RMSD and HB data in one file, so that for every sequential snapshot of the trajectory we have at the same time $\acute{p}_c$, RMSD and HB.

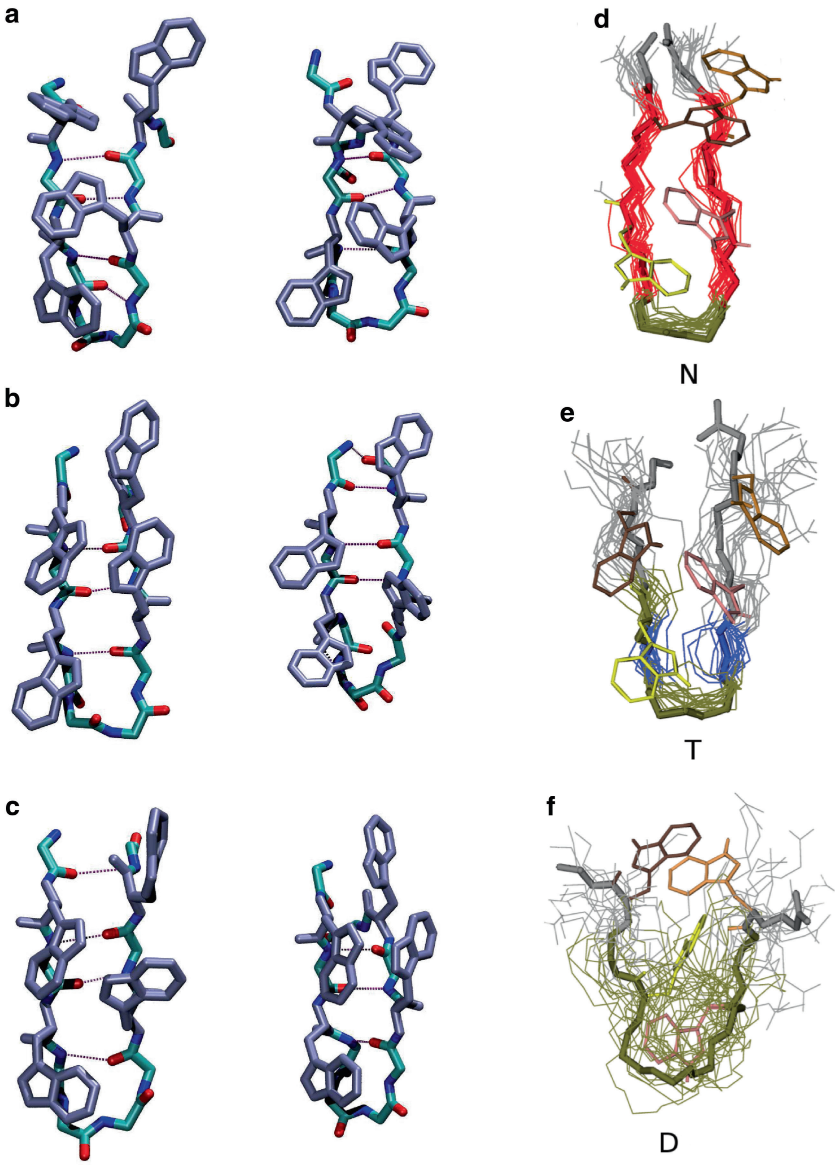


Fig. 2 Conformations of the Trpz1 peptide with $p_c > 0.7$ (**a**), $p_c \sim 0.5$ (**b**) and $p_c < 0.2$ (**c**) from case study 1, simulations at 450 K. Adapted from ref. 10 with permission from Elsevier. Superimposed conformations of the native (**d**), the TS (**e**) and the denatured state (**f**) in case study 2. Adapted with permission from ref. 25. Copyright (2011) American Chemical Society. In all cases the Trp side chains have been rendered as licorice sticks, while the other side chains have been hidden for clarity. In **d**–**f** the backbone has been colored according to the prevalent secondary structure (*red* extended beta-sheet, bl*ue* beta-bridge, *yellow* turn, *gray* random coil)

(b) Write a script (e.g., using awk), that bins the conformations according to $\acute{p}_c$ and counts the fraction of conformations in each $\acute{p}_c$ bin that belong to a transition pathway (i.e., $p(TP|\acute{p}_c)$).

(c) Plot the resulting $p(TP|\acute{p}_c)$.

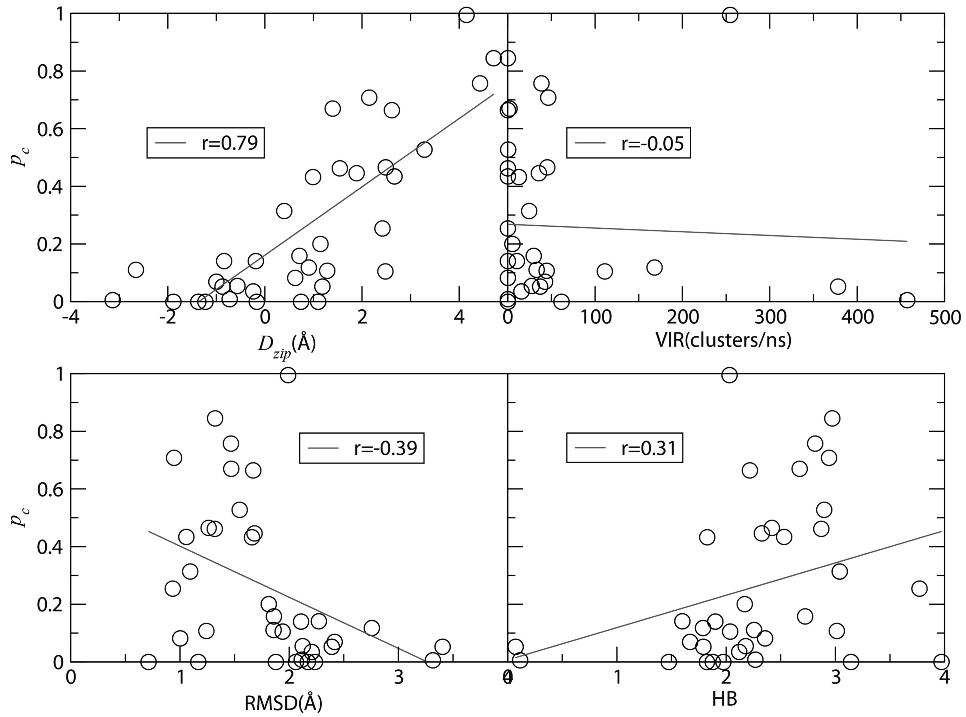


Fig. 3 Committor p_c (folding probability) plotted as a function of several observables for case study 1. The difference in the distance between the $C\gamma$ atom paris of Trp^2-Trp^4 and Trp^9-Trp^4 (D_{zip}) has a good correlation with p_c especially in the transition region, thus it may represent a good reaction coordinate

In the ideal diffusive case where p_c is the true committor and optimal reaction coordinate, $p(TP|p_c)$ is a bell shaped curve reaching a maximum of 0.5 at the transition state [34]. The results that we obtain with our approximate $\acute{p}_c$ are shown in Fig. 4. The bell shaped curve distribution of $\acute{p}_c$ from our simulations reach values in between 0.48 and 0.49, as expected from a very good reaction coordinate. These results demonstrate that the approximate committor $\acute{p}_c$ is almost an optimal reaction coordinate, as it can separate particularly well the product and reactant and the intermediate species, including the elusive TSE.

3.6 Comparison with Experiment

In the previous two paragraphs we have shown how it is possible to check for internal consistency the protein folding model emerging from the simulations. We still need a way to compare with available experimental data and/or make predictions about possible experiments regarding the folding kinetics we are trying to characterize. As mentioned earlier, phi-value analysis is often the experimental technique employed for studying the folding kinetics of proteins. Thus, we need a way to translate the structural information about

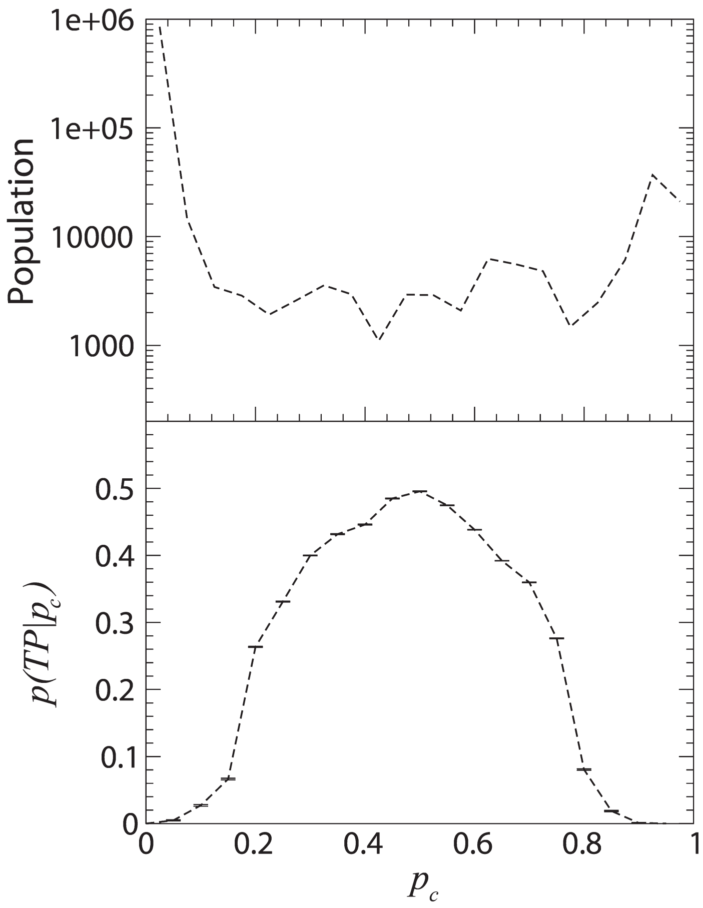


Fig. 4 (**a**) Distribution of p'_c for case study 2. (**b**) The conditional probability to be on a transition pathway given the value of p'_c ($p(TP|p'_c)$). Adapted with permission from ref. 25. Copyright 2011 American Chemical Society

the TSE from our simulations, to something like phi-values. Early studies on protein engineering showed that stability changes upon mutation correlate with the number of atomic contacts removed/added with the mutation [4, 35]. This led to assume that phi-values, that are a ratio of free energy changes in TSE and in native state upon mutation, would correlate with the fraction of native contacts formed by the residue in the TSE [5]. Then, a structural phi-value SΦ(R) for residue R can be defined from the simulations as:

$$\mathrm{S}\Phi(R) = {\sum_{i \in NC(R)} \rho_{\mathrm{TSE}}(i)} \Big/ {\sum_{i \in NC(R)} \rho_N(i)} \tag{2}$$

where the sums are extended to all the native atomic contacts $NC(R)$ of residue R. $\rho_N(i)$ and $\rho_{TSE}(i)$ are the fraction of conformations where the contacts i is formed in the native (N) and TS ensemble, respectively. The assumption about the correspondence of the phi-value Φ and the structural phi-value SΦ has been later verified by measuring in silico the folding kinetics of a model peptide and its mutants using the techniques described above [33].

In practice, to measure SΦ along the trajectory, we need to count the atomic contacts along the trajectory. As per commonly adopted definition, an atomic contact is present when the distance between a pair of heavy atoms is lower than 6.0 Å. For the contact calculations, we initially include all the possible pairs of heavy atoms, with the exclusion of those involving the same residue, nearest neighbor residues and backbone atoms. To do that we create an AWK script that reads the pdb file of the peptide and generates a WORDOM input file listing all the mentioned pairs of atoms. The resulting WORDOM input file has the following structure:

```
BEGIN contacts                   # the contacts module is invoked
--TITLE a2CBa1CA                 # an informative  title for the contact
--SELE /*/2/CB:/*/1/CA:6.0       # the selected pair of atoms followed by the
#                                distance cutoff
--TITLE a2CBa1CB                 # repeat for all possible pair of atoms in the
#                                peptide
--SELE /*/2/CB:/*/1/CB:6.0       # ...
--TITLE a2CBa1OG
--SELE /*/2/CB:/*/1/OG:6.0
...

END
```

WORDOM, with the previous input, generates a large output file with as many columns as pairs of atoms and as many lines as snapshots in the trajectory. To limit the extent of the output, we do not run the analysis on the whole trajectory but only on the subsets of interest. In particular we run it on the state trajectory file of the native state and of the transition state, separately. First, we identify the native atomic contacts as those that are present in at least 2/3 of the snapshots of the native state ($\rho_N > 2/3$). Then we measure the ρ_{TSE} and ρ_N only for those contacts and we apply Eq. 2 to obtain the SΦs. The latter steps can be achieved with AWK scripts.

The SΦs of our simulated systems provide a prediction of the experimental phi values that can be verified, if those are available. In the case of the Trpz1 peptide this is not the case, as the peptide is very small and the folding transition is very broad. However, kinetics measurements are available for similar peptides like Trpz2, Trzp3, Trpz4 and the GB1 hairpin [36, 37]. In these experiments several characteristics of the peptides have been compared and their effect on the kinetics evaluated. The main outcome of the experiments is that mutations on the turn region mostly affect the folding rate, while mutations of the hydrophobic core (i.e., the four Tryptophan side chains in Trpz1) affect mostly the unfolding rate.

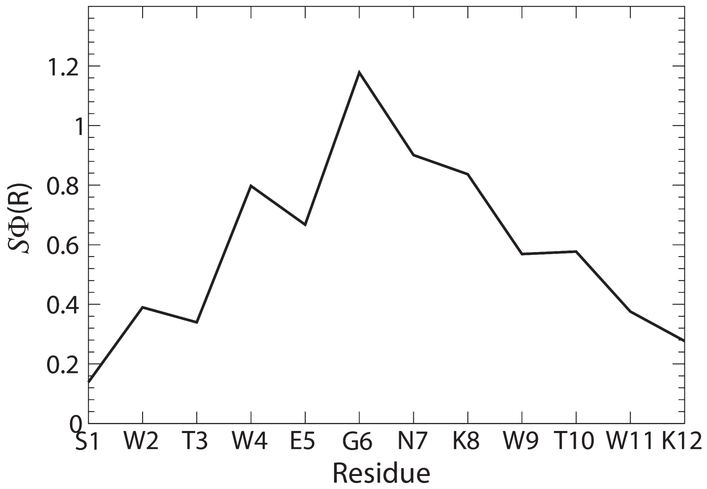


Fig. 5 Structural phi-values SΦ(*R*) for the TSE identified in case study 2. Adapted with permission from ref. 25. Copyright 2011 American Chemical Society

Simulations data can be used to rationalize those experiments. Indeed, the structural phi-values obtained from the simulations of case study 2 (Fig. 5) are large in the turn region and lower for regions increasingly distant from the turn. Similarly, in case study 1 the TSE conformations (Fig. 2a) show a hydrophobic core with a non-native packing of side-chains (but native-like reciprocal distances) and a natively in-register turn region thanks to the closure of the loop between Thr3 and Thr10. Thus, the turn region is formed already in the TSE (but not in the denatured/unfolded state), so that any change in the turn region would mostly affect the free energy difference between the TSE, where the turn is present and the unfolded state, where the turn is not present, and as a consequence, the folding rate. A similar argument explains the effects of perturbations to the hydrophobic core region: the packing of the hydrophobic residues is not-native like in the TSE so that any change of these residues will mostly affect the free energy difference between the native state, where they are natively packed, and the TSE, where they are not, ultimately affecting the unfolding rate.

Concluding, I have presented detailed procedures to simulate, using MD, the kinetics of folding of proteins/peptides and obtain data directly comparable with the relevant experiments, i.e., phi-value analysis and/or folding kinetics measurements. I have shown how to practically apply the procedures to real data and pointed out possible difficulties dependent on the system under investigation. The information provided here should serve the reader to extend the methodology to an increasingly larger variety of systems where both computational and experimental investigations are possible. All the mentioned scripts and input files are available for download upon request to the author (settanni@uni-mainz.de).

4 Notes

1. Experimentally, the unfolding phi-value is often preferred to the folding phi-value as it is affected by smaller uncertainties $\Phi_{\mathrm{unf}} = \frac{\Delta\Delta G_{\ddagger-F}}{\Delta\Delta G_{U-F}} = \ln\left(\frac{k_{\mathrm{unf}}^{\mathrm{mut}}}{k_{\mathrm{unf}}^{\mathrm{WT}}}\right) / \Delta\Delta G_{U-F}$. The two phis are linked by the following relationship: $\Phi_{\mathrm{f}} = 1 - \Phi_{\mathrm{unf}}$.
2. In this kind of simulations, mostly unfolding transitions are observed, although by citing the reversibility of the Newtonian dynamics, they have been assumed to provide information on the folding pathway, as well [5]. In general, however, such a large change in environmental conditions may induce changes in the free energy landscape of the protein and on the folding/unfolding pathways. Extensive test of the method with the experiments has shown that this does not occur often, possibly because of the robustness of the folding/unfolding pathways of proteins that are the end result of a very long evolution process.
3. Notwithstanding the conditions used for the simulations, the limited size of the system and the simulation protocol involving a slow heating phase prevent the observation of the liquid-vapor phase transition of water.
4. For the choice of the right cutoff , when the native state of the peptide is known, our suggested strategy consists in measuring the dRMSD with respect to the native state along the trajectory and producing an histogram of dRMSD values. This histogram typically shows either a small peak at low dRMSD or, at least, a shoulder. That peak is due to the other conformations of the native state observed along the trajectory and its location in terms of dRMSD is a good starting point for the clustering cutoff because the native state is possibly the narrowest free energy minimum to be observed in the simulations.
5. In the present case, clusters are portions of *n*-spheres in the space of distance matrices, where n is the number of independent elements of the distance matrix $n = N(N-1)/2$ with N the number of atoms used for the distance matrix.
6. The number of visited clusters changes in a discrete way. To measure the VIR, we smooth the number of visited clusters by convoluting it with a narrow Gaussian kernel (width 100 ps) and then taking the derivative.
7. In reality, for the case study 2 it is possible to identify four different states using a Markov-state-model approach, however from the slowest relaxation in the system it is possible to establish that the largest free energy barrier separates the denatured states from the variably folded states. Within those broad

states, it is possible to identify further and smaller free energy barriers, which help distinguish the states of the peptide. This complex analysis of the kinetics of protein folding is out of the scope of this book chapter and the interested reader should refer to the original publication [25] for further details.

8. The committor p_c of a conformation follow a scaled binomial distribution, thus the standard error on the mean value is equal to $\sqrt{p_c(1-p_c)/N_t}$ where N_t is the number of trials and it is largest at committor ~0.5, that is, at the TSE.
9. A more refined approach consist of defining a separatrix in between the boundaries of the two states and measuring the fraction of commitment runs on the "folding" side of the separatrix as a function of the elapsed time along the run. This value follows an exponential relaxation which converges to the p_c (*see* ref. 10 for details).

References

1. Anfinsen CB, Haber E, Sela M, White FH Jr (1961) The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. Proc Natl Acad Sci U S A 47:1309–1314
2. Matouschek A, Kellis JT Jr, Serrano L, Fersht AR (1989) Mapping the transition state and pathway of protein folding by protein engineering. Nature 340(6229):122–126
3. Fersht AR, Matouschek A, Serrano L (1992) The folding of an enzyme. I. Theory of protein engineering analysis of stability and pathway of protein folding. J Mol Biol 224(3):771–782
4. Jackson SE, Moracci M, elMasry N, Johnson CM, Fersht AR (1993) Effect of cavity-creating mutations in the hydrophobic core of chymotrypsin inhibitor 2. Biochemistry 32(42): 11259–11269
5. Li A, Daggett V (1996) Identification and characterization of the unfolding transition state of chymotrypsin inhibitor 2 by molecular dynamics simulations. J Mol Biol 257(2):412–429
6. Gsponer J, Caflisch A (2002) Molecular dynamics simulations of protein folding from the transition state. Proc Natl Acad Sci U S A 99(10):6719–6724
7. Vendruscolo M, Paci E, Dobson CM, Karplus M (2001) Three key residues form a critical contact network in a protein folding transition state. Nature 409(6820):641–645
8. Clementi C, Nymeyer H, Onuchic JN (2000) Topological and energetic factors: what determines the structural details of the transition state ensemble and "en-route" intermediates for protein folding? An investigation for small globular proteins. J Mol Biol 298(5): 937–953
9. Settanni G, Gsponer J, Caflisch A (2004) Formation of the folding nucleus of an SH3 domain investigated by loosely coupled molecular dynamics simulations. Biophys J 86(3): 1691–1701
10. Settanni G, Fersht AR (2008) High temperature unfolding simulations of the TRPZ1 peptide. Biophys J 94(11):4444–4453
11. Shaw DE, Deneroff MM, Dror RO, Kuskin JS, Larson RH, Salmon JK et al (2008) Anton, a special-purpose machine for molecular dynamics simulation. Commun ACM 51(7):91–97
12. Abe H, Go N (1981) Noninteracting local-structure model of folding and unfolding transition in globular proteins. II. Application to two-dimensional lattice proteins. Biopolymers 20(5):1013–1031
13. Klimov DK, Thirumalai D (2000) Mechanisms and kinetics of beta-hairpin formation. Proc Natl Acad Sci U S A 97(6):2544–2549
14. Settanni G, Cattaneo A, Maritan A (2001) Role of native-state topology in the stabilization of intracellular antibodies. Biophys J 81(5):2935–2945
15. Settanni G, Hoang TX, Micheletti C, Maritan A (2002) Folding pathways of prion and doppel. Biophys J 83(6):3533–3541
16. Lindahl E, Hess B, van der Spoel D (2001) GROMACS 3.0: a package for molecular simulation and trajectory analysis. J Mol Model 7(8):306–317

17. Brooks BR, Bruccoleri RE, Olafson BD, States DJ, Swaminathan S, Karplus M (1983) Charmm—a program for macromolecular energy, minimization, and dynamics calculations. J Comput Chem 4(2):187–217
18. Seeber M, Cecchini M, Rao F, Settanni G, Caflisch A (2007) Wordom: a program for efficient analysis of molecular dynamics simulations. Bioinformatics 23(19): 2625–2627
19. Humphrey W, Dalke A, Schulten K (1996) VMD: visual molecular dynamics. J Mol Graph 14(1):33–38, 27–38
20. Cochran AG, Skelton NJ, Starovasnik MA (2001) Tryptophan zippers: stable, monomeric beta-hairpins. Proc Natl Acad Sci U S A 98(10):5578–5583
21. Jorgensen WL, Maxwell DS, TiradoRives J (1996) Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. J Am Chem Soc 118(45):11225–11236
22. Berendsen HJC, Postma JPM, van Gunsteren WF, Hermans, J (1981) Intermolecular foces. In: Pullman B (ed). Reidel, Dordrecht, The Netherlands
23. Berendsen HJC, Postma JPM, Vangunsteren WF, Dinola A, Haak JR (1984) Molecular-dynamics with coupling to an external bath. J Chem Phys 81(8):3684–3690
24. Hess B, Bekker H, Berendsen HJC, Fraaije JGEM (1997) LINCS: a linear constraint solver for molecular simulations. J Comput Chem 18(12):1463–1472
25. Radford IH, Fersht AR, Settanni G (2011) Combination of Markov state models and kinetic networks for the analysis of molecular dynamics simulations of peptide folding. J Phys Chem B 115(22):7459–7471
26. Ferrara P, Apostolakis J, Caflisch A (2002) Evaluation of a fast implicit solvent model for molecular dynamics simulations. Proteins 46(1): 24–33
27. Cavalli A, Ferrara P, Caflisch A (2002) Weak temperature dependence of the free energy surface and folding pathways of structured peptides. Proteins 47(3):305–314
28. Ryckaert JP, Ciccotti G, Berendsen HJC (1977) Numerical-integration of Cartesian equations of motion of a system with constraints—molecular-dynamics of N-alkanes. J Comput Phys 23(3):327–341
29. Hartigan JA (1975) Clustering algorithms. Wiley series in probability and mathematical statistics. Wiley, New York. xiii, 351 p
30. Du R, Pande VS, Grosberg AY, Tanaka T, Shakhnovich ES (1998) On the transition coordinate for protein folding. J Chem Phys 108(1):334–350
31. Frenkel D, Smit B (2002) Understanding molecular simulation : from algorithms to applications, 2nd ed. Computational science series. Academic, San Diego. xxii, 638 p
32. Rao F, Settanni G, Guarnera E, Caflisch A (2005) Estimation of protein folding probability from equilibrium simulations. J Chem Phys 122(18):184901
33. Settanni G, Rao F, Caflisch A (2005) Phi-value analysis by molecular dynamics simulations of reversible folding. Proc Natl Acad Sci U S A 102(3):628–633
34. Best RB, Hummer G (2005) Reaction coordinates and rates from transition paths. Proc Natl Acad Sci U S A 102(19):6732–6737
35. Serrano L, Matouschek A, Fersht AR (1992) The folding of an enzyme. III. Structure of the transition state for unfolding of barnase analysed by a protein engineering procedure. J Mol Biol 224(3):805–818
36. Du D, Zhu Y, Huang CY, Gai F (2004) Understanding the key factors that control the rate of beta-hairpin folding. Proc Natl Acad Sci U S A 101(45):15915–15920
37. Du D, Tucker MJ, Gai F (2006) Understanding the mechanism of beta-hairpin folding via phi-value analysis. Biochemistry 45(8):2668–2678

Part III

Protein Structure Determination

Chapter 14

Comparative Modeling of Proteins

Gerald H. Lushington

Abstract

Much of the biochemistry that underlies health, medicine, and numerous biotechnology applications is regulated by proteins, whereby the ability of proteins to effect such processes is dictated by the three-dimensional structural assembly of the proteins. Thus, a detailed understanding of biochemistry requires not only knowledge of the constituent sequence of proteins, but also a detailed understanding of how that sequence folds spatially. Three-dimensional analysis of protein structures is thus proving to be a critical mode of biological and medical discovery in the early twenty-first century, providing fundamental insight into function that produces useful biochemistry and dysfunction that leads to disease. The large number of distinct proteins precludes rigorous laboratory characterization of the complete structural proteome, but fortunately efficient in silico structure prediction is possible for many proteins that have not been experimentally characterized. One technique that continues to provide accurate and efficient protein structure predictions, called comparative modeling, has become a critical tool in many biological disciplines. The discussion herein is an updated version of a previous 2008 treatise focusing on the general philosophy of comparative modeling methods and on specific strategies for successfully achieving reliable and accurate models. The chapter discusses basic aspects of template selection, sequence alignment, spatial alignment, loop and gap modeling, side chain modeling, structural refinement and validation, and provides an important new discussion on automated computational tools for protein structure prediction.

Key words Proteins, Comparative modeling, Homology, Threading, Sequence alignment, Structure alignment, Loop modeling, Structure refinement, Structure validation

Abbreviations

AA	Amino acid (plural: AAs)
BSE	Bovine spongiform encephalopathy
CASP	Critical assessment of techniques for protein structure prediction
CATH	Class architecture, topology, and homologous superfamily
Cα	Alpha carbon on the amino acid backbone
DNA	Deoxyribonucleic acid
H-bond	Hydrogen-bond
MD	Molecular dynamics
NMR	Nuclear magnetic resonance
N_R	Number of residues
PDB	Protein data bank

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_14, © Springer Science+Business Media New York 2015

PrPC	Prion protein cellular
PrPSc	Prion protein scrapie
ps	Picosecond(s)
PSI	Protein structure initiative
RMSD	Root-mean-squared deviation
3D	Three-dimensional

1 Introduction

One of the wonders of life is its tremendous diversity, not only in terms of organisms of vastly different sizes and characteristics but even within any one single organism. The lowly bivalve, for example, is composed of different tissues that range across an incredible breadth of color, transparency, flexibility, hardness, adhesiveness, and electrical conductivity. Such variation is owed primarily to proteins: a class of materials composed of a modest set of ~20 distinct amino acids (AA) building blocks that evolution has chemically permuted into the most diverse collection of unique, naturally occurring substances of any molecular class in existence. By varying the length and sequence of constituent AA chains, proteins can assemble to form the fundamental matrices of materials that are harder than stone, softer than soap, as translucent as glass, as opaque as soot, soluble in water or grease, excellent conductors or insulators, and among the most efficient known fluorophores, capacitors, diodes, and catalysts known to man. One of the most important keys to rationalizing, exploiting, and refining such attributes, and seeking corrective measures when they go awry, is a detailed understanding of the three-dimensional (3D) assembled structure(s) of proteins, for it is in this form that proteins adopt their unique functional properties and exert their intended influence on their surrounding environment.

Beginning with the first atomic-level resolution of a protein structure (whale myoglobin by Kendrew et al. [1]), 3D protein models have provided a wealth of insight into biomolecular properties and processes, inspiring a growing thirst for structural detail. However, whereas biomolecular sequencing has become highly amenable to efficient, high throughput characterization, the experimental resolution of the 3D protein structure remains time-consuming and frequently poses significant challenges for traditional characterization techniques such as X-ray crystallography, with eventual success contingent on luck, persistence, ingenuity, or through the application of innovative techniques [2]. Seminal early work by Anfinsen that will be discussed later, implied, however, that the structure of any protein was uniquely dictated by its constituent sequence [3], which suggests that the combination of a comprehensive catalog of protein sequences and an accurate understanding of the relationship between sequence and structure could provide the basis for correspondingly comprehensive understanding of the structural proteome. In this spirit, the Protein Structure Initiative (PSI) (http://www.nigms.nih.gov/Research/FeaturedPrograms/PSI/) was initiated in 2000 with the expressed goal of making "*the three-dimensional, atomic-level structures of most proteins easily obtainable from knowledge of their corresponding DNA sequences.*" As for the human genome project, such objectives are of a scope (around 100,000 proteins, not counting post-translational modifications and conformational variants) that requires that conventional resolution methods such as X-ray crystallography and NMR by supplemented by efficient and analytically rigorous computational modeling techniques that perceive and exploit relationships between primary AA sequence and the biologically observed 3D structural manifestation of the protein. This chapter thus offers a brief discussion of in silico protein structure prediction, focusing mainly on comparative modeling. Whereas other papers (e.g., Baker and Šali [4], Martí-Renom et al. [5]) provide comprehensive reviews of the underlying method development and research achievements in the field, this chapter discusses the motivation for using comparative modeling and outlines the practical considerations to be made in assembling such a model. In the years since the original version of this chapter was composed [6], computational developments have taken place that have substantially changed the practice of computational structural biology: many of the protocols described within the original text have been effectively implemented as systematically automated protocols that are often capable of producing plausible (and often excellent) results with minimal human intervention. This revised version recognizes this valuable service as an important contribution to the efficient acquisition of knowledge and has added a section which profiles some of the best current (ca. 2013) resources for automated comparative modeling. However, prudent application of automated protocols still requires the confidence of understanding the underlying manipulations and computations, thus the meat of this chapter remains a detailed discussion of how protein structure predictions are computationally effected.

2 Conceptual Basis for Comparative Modeling

A key inspiration for projects such as the PSI was realization that the human genome sequencing project was not the panacea for biological understanding that some had hoped for, but rather established a basis that helped to illuminate factors other than the genetic coding portions of our DNA that determine tissue function and differentiation, inter- and intra-species diversity, medical causality, and other key issues of organism-scale biology. A more apt currency for appreciating the diversity inherent in life may lie in proteomics wherein one can identify specific units responsible for a given biological function or dysfunction, quantify their relative abundance as a function of tissue behavior, and thus help to unravel the underlying molecular mechanisms. A key source of insight for the latter is protein structure. Combined with simple rules of physics (e.g., thermodynamics and electrostatics) and chemistry (bond formation/breaking), 3D molecular structure information can be readily extrapolated toward understanding (or confidently predicting) molecular physical attributes, inter-molecular association, enzyme function, and structural response. Structural information thus serves as a basis for rationalizing relationships between constituent molecules and specific functional or dysfunctional processes evident in an organism.

Pharmaceutical research frequently follows the paradigm of discovering a biomolecular target for a given disease, characterizing the target, and finding ways of beneficially modulating its behavior. Since most targets are proteins, a tool that often plays a key role in the first step is comparison of protein expression signatures for diseased and healthy tissue samples. Proteins whose presence or function is noticeably amplified or suppressed in dysfunctional tissue are flagged as potential diagnostic biomarkers and they and their close partners in biochemical pathways often prove to be a fertile source of possible therapeutic targets. Therapeutics are then sought to either restore normal healthy balance in target function or to compensate for imbalances. Knowledge of protein 3D structure can have some applicability in rationalizing protein–protein interaction partners in the search for prospective targets, but is especially invaluable for the latter step of therapeutics discovery, providing key insight into potential receptors that might be pharmacologically targeted, as well as providing a basis for optimizing drug candidates according to their propensity for binding to specific target receptor. Insight into nontarget protein structures is also very useful for intuiting potential drug side effects, as proteins with receptors similar to the target are at pronounced risk of inadvertent modulation by chemicals designed for the latter. Broad understanding of protein structures should further aid in expression-based target identification by helping to pinpoint protein–protein interactions that may obscure whether a deviant expression profile in a protein is a primary cause of a given dysfunction or a secondary symptom. Specifically, knowing 3D structures can inform us of which proteins should colocalize in a given cellular environment, and which are likely to engage in direct (i.e., actual physical association) or indirect (e.g., exchange of metabolite or transmitter) interactions.

Protein 3D structures may be obtained in a number of ways. Most high accuracy structures in the protein databank (PDB; http://www.rcsb.org/pdb) were resolved through crystallographic techniques, with the rest mainly arising from NMR analysis.

While some proteins have been resolved via both techniques, crystallography is best suited to high-resolution determination of structures for soluble, predominantly globular, proteins most amenable to crystallization, whereas NMR affords more a dynamic interpretation of the structure (i.e., multiple conformers are usually resolved from the same data) and is viable for many proteins that resist crystallization, albeit generally at a lower level of resolution. Theoretically, the entire proteome should be accessible to either NMR or crystallography, although the lipophilicity of transmembrane proteins poses experimental complications while some of the most massive proteins still entail major challenges of data deconvolution. While the trajectory of decades of structural biology offers hope that technical advances will eventually conquer the remaining obstacles, it is more uncertain whether experimental resolution of the entire known proteome will ever be achieved, as both crystallographic and NMR remain technically very demanding, time-consuming, and expensive. The underlying assumption of the PSI is that computational modeling may yet accomplish what is spectroscopically impractical.

The main schemes for computational protein structure prediction include: (1) self assembly simulations, (2) associative models based on sequence pattern recognition, and (3) comparative modeling. All of these methods were inspired in part by seminal findings of Christian Anfinsen et al. that an unraveled ribonuclease AA chain could, in plain solution, coalesce within a reasonable time frame (minutes to hours) to form a protein functionally indistinguishable from native in vivo ribonuclease [3]. This implied that: (a) a proteins can be uniquely identified by AA sequence, (b) this sequence uniquely encodes the in vivo protein function, and (c) the AA sequence is capable of consistent self-assembly into functional form, based exclusively on intramolecular interactions among sequence AAs plus interatomic interactions with surrounding solvent. Observation (c) thus directly suggested that assembly simulations could be a reliable mode for protein structure prediction, whereas observations (a) and (b) were key to the eventual formulation of pattern recognition and comparative modeling techniques.

Practical formulation of associative and comparative modeling methods required a substantial basis in empirical understanding of protein structure. One key development was the accumulation of a reasonable volume of protein sequence data beginning in the early 1950s, leading to gradual elucidation of strong correlative patterns between protein sequence similarity on one hand and analogous function on the other. This permitted the classification of proteins into families and superfamilies [7], with an underlying assumption being that members of the same protein superfamily are all evolutionarily related (a.k.a., homologous), and members of the same family are closely related.

A second important precursor to protein structure prediction was the acquisition of crystal structures from the late 1950s onwards. These structures generally validated the earlier family and superfamily classifications in that proteins with similar sequences and function were usually found to have commensurately similar 3D structure. Analysis of structural data also revealed that all proteins tended to assemble as a sum of a limited number of unique substructure forms. At a fine grained level, it was noted that all proteins tended to adopt similar H-bond stabilized features such as alpha helices, beta sheets, and a number of distinct turn and hairpin structures, collectively referred to as *secondary structure*. At coarser levels, it also became apparent that the full manifold of protein structures could be classified into a relatively small number of unique *folds*: characteristic self-stabilizing collections of secondary structure elements. Estimates for the precise number of distinct folds that one should expect to find in the proteome have varied significantly over the past 20 years [8] but may be converging to approximately 2,000. In a practical sense, while there are almost certainly unique protein classes that have not yet been structurally resolved, the number of distinct, experimentally characterized and validated folds has recently stagnated: as of July 2013, the lists of distinct folds registered by CATH [9] and SCOP [10] from the body of PDB structures had not been added to since 2009 (1,282) and 2008 (1,393) respectively, where the parenthetical numbers reflect the number of distinct folds registered in each of these two classification schemes. In the heyday of the PSI (ending in around 2007), approximately 20 % of newly solved protein structures corresponded to new folds, but that ratio has obviously dwindled, perhaps indicating that the remaining portion of fold space might be dominated by challenging (e.g., poor solubility, or substantially disordered) proteins that are inherently resistant to conventional crystallographic and NMR techniques.

In light of the current state of the practice, it thus appears that comparative modeling might be destined to applicability within a subset of the proteome, but fortunately within this regime the underlying basis for the technique is solid. Not surprisingly, proteins within the same family (as classified by sequence) almost invariably correspond to similar fold classifications. Such correspondence between sequence similarity trends versus structural similarity completes the basis for both associative and comparative modeling techniques. Specifically, associative models are based on probabilistic tendencies of certain AA combinations to adopt a given secondary structure and tendencies of certain secondary structure elements to adopt a specific fold, whereas the more general tendency of proteins with similar sequences to adopt similar 3D structures is the foundation for comparative modeling.

While each of the different structure prediction method is potentially applicable to modeling a given target protein, each has limitations to heed. Self-assembly simulations, commonly known as protein folding models, are especially computationally demanding due to huge numbers of unique conformers that must be considered for even modest-sized proteins. With such large conformer manifolds, one may also have to empirically choose between multiple unique but comparably favorable structures. A more severe problem exists for the fraction of protein population whose thermodynamically optimal conformer is not the one observed in vivo, the most famous example of which being the prion PrPC whose native form appears to be higher in energy than a rogue form, PrPSc. Presence of the latter rogue form of the protein in living tissue is believed to catalyze the exothermic refolding of healthy PrPC to the biologically inutile PrPSc, with the result being prion diseases such as scrapie, Creutzfeldt-Jakob Disease or BSE [11]. Given the greater thermodynamic stability of the latter, a good folding algorithm based on the underlying free energy effects assumed by Anfinsen to drive the assembly process [3] should converge to the rogue PrPSc form rather than the biologically healthy PrPC. In truth, energy is often not the sole driving force behind assembly, in that life forms have developed sophisticated tools called chaperonins that perform protein assembly quality control, correcting misfolds and sometimes preferentially effecting assembly of higher energy structural forms. Relevant interactions between chaperonins and their client polypeptides are very complex and the associated mechanisms of mediated folding are still in the process of being elucidated [12, 13]. Folding simulations that explicitly consider chaperonin contributions have shown promise in de novo structure prediction [12–14]; however, it is not yet clear whether they will sustain general success over the diverse collection of proteins that require some form of folding mediation.

Accurate associative modeling is also fairly challenging due to prospects for compounding errors. Even for the most sophisticated prediction algorithms, the first step of secondary structure prediction from raw sequence remains limited to about 80 % success on a per-residue basis [15], thus leading for a medium-sized protein (hundreds of AAs) to dozens of local errors at the outset. Since local errors in backbone torsion can easily extrapolate to large errors in the global structure (e.g., mislocation of entire lobes), many proteins are poorly predicted. Furthermore, the stated 80 % success rate applies primarily to globular proteins, and indications are that similar predictions for transmembrane proteins are worse on average [16] largely because there is a much smaller pool of secondary structure features resolved membrane-spanning proteins available for training predictive models.

Assessment of results from the Critical Assessment of Techniques for Protein Structure Prediction (CASP) competitions *reveal that the average predictive performance of* comparative models remains consistently superior to nontemplate methods [17]. For example, in the 6th CASP competition (2004), on the full set of systems for which both comparative and folding models were generated, with fully automated comparative models achieving a mean RMSD (root mean-squared deviation) relative of 2.17 Å relative to experimental crystal structures and further improvement to 2.09 Å RMSD given human intervention in template selection, alignment modification, etc. By contrast, folding methods (with and without intervention) attained mean RMSD values of 2.41 and 2.51 Å respectively. Comparative models are also relatively computationally expedient; however, their dependence on quality 3D templates with structural and functional similarity to the target precludes many targets from consideration. Fortunately, the current body of structurally resolved representative fold structures is believed to encompass the majority (perhaps two thirds) of all distinct protein classes, thus in the majority of scientific studies it should be possible to identify a prospective template or modeling cases.

3 Methods

In practice, comparative modeling is best viewed not as one technique but rather as a strategy for assembling information from various component methods (including assembly and associative techniques) toward a unified 3D structure prediction. In general, these component steps can be approximately summarized as follows:

1. identify template proteins with structural similarity to the target as gauged (optimally) from sequence-based homology, or from physicochemical similarity.
2. align the target sequence with all relevant template sequences according to the same arguments of homology or physicochemical similarity employed in **step 1**.
3. spatially align all of the template structures into a single framework, and use the sequence alignment to project the target protein backbone onto this framework.
4. estimate structures for target protein fragments that are ill-represented by the template manifold, or else omit them from the predicted structure.
5. align target side chains with analogous side chains from the template structures, or intelligently guess their disposition according to known spatial and torsional preferences.
6. refine unphysical contacts and strains via conformational searches, and,
7. evaluate the final relaxed model for physical tenability.

Each step above entails various methodological and strategic considerations, some of which provide opportunities for iterative feedback to prior steps, as is shown graphically in Fig. 1. These considerations will be elaborated upon in the remainder of this section.

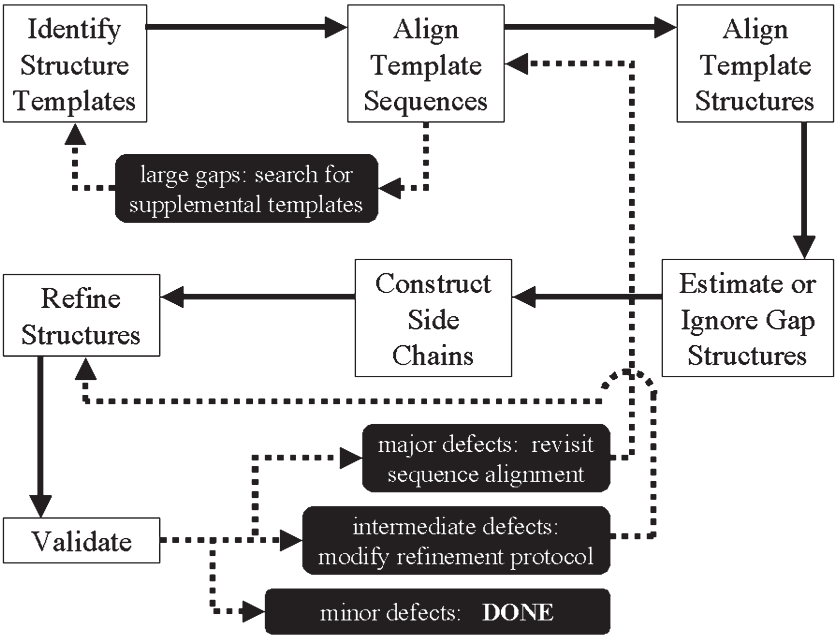

Identify Structure Templates
Align Template Sequences
Align Template Structures
large gaps: search for supplemental templates
Refine Structures
Construct Side Chains
Estimate or Ignore Gap Structures
major defects: revisit sequence alignment
Validate
intermediate defects: modify refinement protocol
minor defects: DONE

Fig. 1 Flow diagram for comparative modeling of proteins showing standard process (*solid arrows*) and feedback/refinement mechanisms (*dashed arrows*)

3.1 Template Identification

The two important objectives in identifying templates are quality and quantity. Specifically, one wishes to find templates with a high quality match to your target of interest so that they offer a strong likelihood of corresponding to similar structures, which will thus provide an apt framework for assembling your target structure. Secondly, unless one is able to uncover a very high quality template that covers the full span of interesting folds in the target with a high degree of homology, it is very useful to identify multiple possible templates. The strength in numbers is derived from several factors: the different templates often provide a useful source of validation in that they should display comparable structure in regions of functional overlap, and templates that cover distinct nonoverlapping regions of the target can collectively serve as a complementary basis with which to stitch together a more complete model of the target that might be possible from a single template alone. There is even some benefit in the case where multiple templates cover largely the same regions and features of the target: if different plausible templates exhibit some compositional diversity but have similar 3D folding patterns, the quality of the resulting sequence alignment and comparative model has been generally found to benefit since multiple sequence alignments are typically more accurate than pair-wise alignments, and employing simultaneous input from multiple structures may tend to average out local structural aberrations.

When identifying prospective templates via sequence homology, a reasonable template-target sequence identity is important for minimizing the risk of errors in sequence alignment and for maximizing the likelihood that the entire template fold is to the target. The widely used minimum sequence identity criterion for homologous templates is 30 % sequence identity over the extent of the mutual target-template alignment: this criterion is expected to provide reasonable confidence that the target and template do indeed share a common fold. The precise origin of the 30 % identity criterion is difficult to trace, in that the seminal paper typically cited for template identification, by Chothia and Lesk, offered a more conservative criterion of 50 % identity, although their objective was to find highly similar structures with less than 1.0 Å RMSD in the position of backbone atoms [18]. Conversely, others have claimed that identity levels as low as 20–22 % are frequently still viable [19], provided one has clear empirical evidence that suggests a clear functional analogy between the target and template proteins. Nonetheless, the empirical 30 % criterion is broadly accepted, has stood the test of careful analyses [20] and, when complemented by a careful, well-scrutinized sequence alignment, generally affords confidence for a plausible structure prediction.

While many targets exist for which no templates with at least 30 % sequence identity are available, it has been estimated via fold statistics that close to two thirds of all possible targets of interest should have a template of reasonable structural similarity already present in the PDB. This wealth of templates may be explained from the fact that structural conservation can often be largely retained even in cases of very distant ancestral commonality. Furthermore, shared structure is also possible courtesy of evolutionary convergence (i.e., unrelated proteins may gradually adapt similar structures due to innate functional benefits that are uniquely available to such conformations) or in spite of evolutionary divergence (i.e., a protein retains a desirable conformation attained by a distant ancestor, in spite of an extensive mutational history that reduces the sequence identity to well below 30 %). Thus if no template meeting the 30 % sequence identity is identified through homology searches, a reasonable model may still be achieved via a technique known as "threading" that evaluates target sequences against a library of unique fold representatives (a collection of structurally unique proteins such as those collected in the CATH database [9]) according to residue by residue similarity in terms of shape, spatial volume, hydrophilicity/hydrophobicity, helix- or sheet-forming propensity, etc. Functionally special residues such as prolines (whose closed-ring structure induces a kink in AA chains), cysteines (well known for their fold-stabilizing disulfide bonds), and ionic residues (aspartate, glutamate, lysine, and arginine all tend to occur predominantly near solvent accessible surfaces of proteins, except in cases where they form salt bridges) are frequently

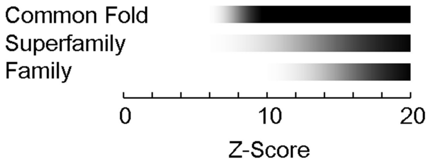


Fig. 2 Observed correspondence between PROSPECT *Z*-scores computed for pairs of proteins and the likelihood of their sharing a common fold or having a familial relationship

given additional weight in any assessment. As sequence identity is generally less useful for assessing threading templates, a consensus *Z*-score is instead used to gauge statistical significance for the quality of one template candidate relative to others being considered. *Z*-score scales vary across the manifold of different threading programs, but typically provide an indication of how likely it is that a given target/template pair are actually members of a common protein family or super family, whether they merely share a common fold, or whether they have no obvious relationship. An example of such ranges, as reported for the widely used PROSPECT-II program [21], is provided in Fig. 2.

The *Z*-score provides a reasonable scheme for identifying the plausible templates, but has a margin of error that must be heeded. A study contrasting performance of PROSPECT-II with other threading programs found that the top-scoring PROSPECT-II match was a valid fold-conserving template 84.1 % of the time when the template manifold contained species within the same family as the target, 52.6 % of the time when superfamilial (but no familial) templates were available, and 27.7 % if the best templates merely shared a common fold [21]. These numbers improved to 88.2, 64.8 and 50.3 % when examining the top five matches for possible valid templates. These are reasonably successful ratios relative to competing threading models at the time, but do highlight the possibility that invalid templates may achieve high scores, and that one should scrutinize multiple top-scoring candidates, looking for templates known to prefer similar regions of cells as the target species, known to perform functions at least vaguely similar to the target, etc.

A more recent and more rigorous assessment was performed by Brylinski and Lingam [22] explored the capacity of threading programs to correctly identify multiple legitimate templates by choosing not just the top-scoring candidate but also successively ranked options, while recognizing the increasing risk of selecting false positives as one scans down the list. Surveying ten popular threading techniques via searches for distant analogs (i.e., any templates with sequence identity of greater than 40 % relative to the query were considered to be trivial matches and thus were omitted from statistics) of a diverse collection of test molecules, the HHPred method [23] was found to give the best performance of any single model, correctly identifying 50 % of real structural analogs (true positives) before incorrectly selecting as many as 5 % of those prospective templates that were not actually structurally related (i.e., false positives). The eThread method, which selected templates based on a consensus of the ten distinct models, was able to improve on this performance by achieving on average a true positive rate of 60 % before suffering a 5 % false positive selection. In cases where a reasonable sampling of templates has been identified, a high degree of modeling performance can be attained by comparing the structures of the different templates and discarding obvious outliers (i.e., poor alignment with the bulk of the selected templates), but scenarios where few high scoring templates are identified run the risk of producing poor structure predictions.

Note that in cases where templates have been identified by homology, but the sequence identity is somewhat below the safe range (i.e., less than 40 %, and especially if less than 30 %), threading can provide an excellent point of validation. If threading analysis on the target template does not rank the target at a level that seems to assure a conserved fold, the putatively homologous template should be scrutinized with care and skepticism.

Finally, although one may rely substantially on the numerical guidance of sequence identity or threading *Z*-scores for identification and specification of useful templates, there are aspects of the process that can benefit from informed human intervention. For example, if you seek to model a protein in a specific functional conformation (e.g., many proteins exhibit sizable structural variations between activated and inhibited states) it is important to bias your template manifold toward that specific conformation. Whenever possible, one should avoid using multiple templates that span conformationally different states as this may produce a final model reflective of an unphysical hybrid of those states. Finally, for metalloproteins, where the presence or absence of the metal ions can substantially impact the resulting protein conformation, one should identify the desired metal ion state of the target and choose templates with analogous ion residency.

3.2 Sequence Alignment

Most of the various programs commonly used for template identification generally also yield a tentative sequence alignment relative to the target. In homologous cases with greater than 50 % target-template sequence conservation over the mutually aligned portion of the structure, it is generally assumed that the alignment prediction algorithm will produce a qualitatively reliable alignment with only modest local misalignments (no positional errors more than several residues). Over a data set of broadly varying protein similarity, the PROSPECT-II assessment of threading reliability was that the program could achieve about a 60 % average accuracy in prediction the alignment position of any given residue, and typically located each residue within 4 AA positions of the correct spot around 80 % of the time [21]. In fairly strong threading models (e.g., PROSPECT *Z*-scores >10) one can likely assume better performance, perhaps on par with good homology alignments. However in all cases, and especially those with poorer sequence identity or *Z*-scores, careful manual validation is a good policy. This can be achieved by comparing the alignment relative to the known 3D (template) structure to identify any of the following cases:

- sizeable gaps (i.e., greater than 2 or 3 AAs) present in template core regions.
- gaps greater than 1 AA in known template sheets or helices.
- target prolines located within known template helices.
- positional displacement of more than 1 or 2 AAs for template cysteine residues known to engage in disulfide bonds.
- displacement of more than 2–3 AAs for template ionic residues known to form salt bridges.

Alignments containing instances such as those above can yield unphysical comparative models for which fold conservation may be less favored or even no longer feasible. Manual adjustments may thus be made to alleviate such errors, at the expense of opening or extending gaps in exterior loops, as long as relatively modest penalties are incurred in the overall alignment score. Any templates that have a significant number of alignment problems that cannot be alleviated through minor manual adjustment should be viewed with skepticism. Furthermore, the number of irreconcilable alignment problems observed for a given target/template pair is a good source of evaluating and prioritizing that template's suitability relative to other candidates.

The presence of template gaps in regions that should correspond solvent exposed loops in the target species is rarely a key criterion in evaluating a template or a target-template alignment. With the exception of large deletions that might lead to excessive loop shortening (which might produce untenable strain in the model), the impact of minor modifications to such loops has minimal influence on the basic fold of a protein, thus from an evolutionary perspective they tend to have the largest frequency of noncritical point mutations, insertions, and deletions among homologs. If the gap in such a loop is relative small (less than 5 AAs), and is not believed to play an important role in properties of interest (i.e., not being part of a known receptor or protein–protein interaction site) it might be justifiably omitted, although it is often more satisfying to piece it together according to empirical loop libraries (e.g., [24, 25]) or as an arbitrary structure such as a beta turn, with the expectation that its relatively large mobility will be respond well to subsequent refinement steps. For gaps that are significantly longer than 10 AAs, however, the structural integrity of your predicted model is best served by finding a legitimate template for the unrepresented loop itself. This can entail addition of a lower ranked template that may be less suitable for global alignment, but that does afford reasonable alignment to the gap regions. If such a template is believed to have regions other than the gap in question that are inappropriate for modeling the rest of the target (i.e., very poor alignment, or known to have an inappropriate conformation), one may discard all of the template except that specifically corresponding to the gap region.

3.3 Spatial Alignment of the Target

Once templates have been selected and a sequence alignment is in place, construction of a preliminary structural model of the target is straightforward and can usually be accomplished reliably with black-box processes implemented in most comparative modeling programs. The assembly methods typically construct a backbone model first, and then incorporate side chains into the resulting framework. Some methods may assemble the core region of a protein (solvent-inaccessible portions of the structure plus conserved secondary structure elements) first, then treat exposed loops. Most backbone modeling methods begin by superimposing all templates onto a common framework and computing a consensus backbone defined by mean positions of corresponding Cα's. One of three different schemes is then typically used: (1) rigid body assembly of target fragments corresponding to nongapped portions of the target/template alignment as is implemented in the COMPOSER program [26], (2) segment matching whereby target protein fragments are projected onto the consensus backbone with torsional angles being established through reference to fragment polypeptide libraries (e.g., SEGMOD [27]), or (3) construction of a compromise model that minimizes steric and torsional restraints of the target backbone as it is projected onto the template framework (e.g., MODELLER [28]). All of the above methods have proponents, and it is unclear whether the ultimate model accuracy varies much as a function of method choice, in that subsequent refinement is usually required regardless of technique. The restraints-based formalism is probably the most widely used by comparative modelers at this point.

One major complication to the assembly process may arise when the target protein contains a terminal domain that is represented by a different template than is used for the main core of the protein. In such cases, it may be unclear from the relevant templates how the terminal domain should pack onto the core framework. One method that we have applied for such a scenario [29] is to assemble the main core and the terminal domain as separate models, and to estimate the preferred core—terminus packing via protein–protein docking analysis as performed by GRAMM [30], ZDock [31] and other programs. Protein-protein docking methods typically offer a suitability score for each predicted complex, thus helping to guide the selection of complex to serve as the packing model. Another practical and critical consideration is whether a predicted complex places the two units in a position where it is possible to chemically rejoin the broken AA chain without unphysical strain on the backbone. Given a very low probability of predicting a complex that perfectly places the final AA of one unit in covalent binding distance to the initial AA of the next unit, our strategy has been to omit a portion of the target sequence within the core-terminus boundary region during initial model construction for these two domains, with the intention of re-integrating this portion as a flexibly modeled gap (see next section). Specifically, our recommended protocol is to omit a number of residues (say N_R) that are not predicted to be part of a defined secondary structure element (i.e., unstructured coil), and then constrain the choice of docked complexes to those that place the final Cα of the first domain within a distance of less than $N_R \times 3.0$ Å from the first Cα of the next domain (i.e., somewhat less than the maximum unstrained Cα–Cα distance of approximately $N_R \times 3.8$ Å). This method has not yet been exhaustively validated, but is based on reasonable logic and provides a potentially workable solution to an otherwise very challenging problem.

3.4 Loop and Gap Modeling

While the previous backbone assembly step should yield structure predictions for many (hopefully most) of the loops in your target, it is common for some to be represented by poor sequence conservation or to appear in gapped regions of the alignment. Gaps covering 5 AA positions or less pose minimal concerns as they can typically be patched in with reasonable accuracy by referring to polypeptide structure libraries (e.g., [24, 25, 27]), but gaps of greater length are difficult to reliably model via methods other than template comparison. In cases where no template for the gap can be found from either homology searches or threading, it is still possible to attempt a prediction based either on the same peptide libraries available for short loops or an algorithmic scheme akin to protein folding strategies (e.g., for molecular dynamics *see* Chapter 1 and for Monte Carlo conformational searches *see* Chapter 2). Neither of these strategies is guaranteed to produce a model with close correspondence to its real optimal structure, especially for gap lengths larger than 10 AAs. In general, however, many loops without a suitable template remain unresolved precisely because they inherently have high conformational mobility. In such cases, nonoptimal conformers may well still correspond to in vivo accessible structures.

3.5 Side Chain Modeling

Conserved disulfide bonds and salt bridges are typically incorporated into the target model directly during the backbone assembly process. Beyond this, side chain positions for highly conserved residues may also be inferred directly from the template, although their conformations are known to vary significantly from one protein to another when sequence identity is less than 50 %, and are considered to be poorly conserved for identities less than 30 % [32]. Fortunately a number of effective side-chain packing algorithms have been developed and validated (*see* Huang et al. [33] for an excellent review), and are often implemented as black-box features in comparative modeling programs. For the fastidious, the graphical program PyMol [34] provides an excellent qualitative side chain sampling and evaluation tool embedded within the program's mutagenesis toolkit. This tool permits users to click their way through a manifold of distinct, plausible side chain conformers, which each structure graphically rendered according to favorable and unfavorable contacts with the surrounding protein. Deepview [35] offers a somewhat similar mutation module that, although a bit less user friendly, provides the added benefit of molecular mechanics side chain optimization.

3.6 Refinement

The nature of comparative modeling, whereby protein structures are predicted by analogy to different but related proteins, invariably leads to some degree of structural error. If the template (or template manifold) contains no sequence gaps relative to the target and only minor variations in sequence, the resulting error may be less than the resolution accuracy of the template(s) thus further correction to the predicted target structure would be unnecessary. However if the target-template alignment contains gaps or has sequence identity less than 70 %, some structural refinement is advisable. In cases where some templates have been selected to cover regions of the sequence poorly represented by primary templates, boundary effects arise where one template may significantly perturb the projected target backbone derived from other templates and vice versa. Library models used for loop patching lack specific environment information necessary to adapt the loop to the target of interest and thus yield imperfect backbones. It is finally important to note that most backbone errors are amplified in the predicted side chain conformation. Fortunately, most models that have been assembled with care and without unrealistic assumptions will possess only modest errors that may be corrected in a physically natural manner. In vivo, a protein that has been perturbed from its native conformation due to some minor disturbance (e.g., intermolecular interaction and change in ionic state) stands a reasonable chance of reverting to normal structure when a stable normal environment is restored. By analogy, one may consider the deviations arising from a careful but imperfect protein assembly to be comparable to an environmental perturbation, thus any simulation that subjects the protein model to conditions akin to a normal in vivo environment should encourage reversion to a reasonable structure. Molecular dynamics (MD) methods are exceptionally well suited to this task: protein simulation is one of the most common MD applications (*see* Chapter 1), and excellent force fields (Chapter 4) have thus been developed for simulating the interactions of proteins (and their constituent cofactors, ions, etc.) with solvent environments comparable to in vivo conditions. Various reviews on the MD strategies, techniques, and resources applicable to protein structure prediction and refinement are available (e.g., Refs. [36, 37]). The main drawback to MD simulations is computational expense, however, thus when seeking to mitigate unphysical structural aspects arising from model assembly one may find that constant-temperature simulations may not accomplish all necessary refinements in a viable time frame. An alternative is to perform simulated annealing calculations wherein the protein is gradually warmed up to a fairly hot temperature (1,000 K is a practical upper limit for short pulse heating; for temperatures hotter than this or for heating durations longer than 10 ps, one may observe potentially irreversible unphysical structural changes) and then slowly cooled back to ambient temperature. After multiple thermal cycles of this sort (typically 5–10, with each cycle lasting from 10–100 ps), most moderate errors in the original structure are corrected, generally leaving a fairly plausible conformer. Some comparative modeling programs such as MODELLER [28] contain an embedded MD code and perform annealing as an option of the structure prediction process.

Special caution is due when modeling transmembrane proteins. Many MD methods and parameters have been tailored for the specific case of soluble proteins immersed in a polar (generally aqueous) media, and may yield unphysical conformations for inherently hydrophobic membrane-binding portions of a protein. Special methods for simulating transmembrane proteins are discussed in Chapters 8–11 of this volume and e.g., in a review by Im and Brooks [38].

3.7 Validation

Any protein structure determination, from a rough comparative model to a high-resolution synchrotron-based crystal structure, will differ somewhat from the real in vivo conformation. A number of validation tools have thus arisen to evaluate structural models and detect aspects that appear to differ conformationally from standard bond distance, angle, torsion, or contact ranges derived from extensive assessment of known structures. Conveniently, many of the best validation tools are available online, e.g.:

- SAVES (http://nihserver.mbi.ucla.edu/SAVES/): uses multiple distinct servers running PROCHECK, WHAT_CHECK, ERRAT, VERIFY3D, and PROVE to examine amino acid stereochemistry, appropriateness of nonbonded contacts, secondary structure populations, and packing.
- Harmony (http://caps.ncbs.res.in/harmony/): evaluates structures according to backbone conformation, solvent accessibility, and hydrogen bonding.

- MolProbity (http://molprobity.biochem.duke.edu): validation relative to Ramachandran distributions, steric clashes and hydrogen bonding [39].
- PROSESS (http://www.prosess.ca/): validates structures according to various measures including covalent geometry, nonbonded packing, clashes, His & Asn side chain flips, and global and local atomic-level energies [40].

In comparing the above tools, one finds some redundant checks, but each of these utilities has unique features (albeit SAVES is a unique aggregation of other separate services), thus giving consideration to the feedback from all of these services is a valuable step toward producing a high quality model. Among the various warnings that are likely to be issued for a comparative model, some minor problems may be alleviated by more refinement, however care should be exercised in that over-refinement is often itself a source of error. In the most serious cases, excessive use of simulated annealing can produce effects akin to protein denaturation, while a more moderate scenario could involve removal of conformational attributes that might have corresponded in the template to intermolecular interactions (e.g., association with another protein) that might have an analogous influence on the function of the target system. More serious errors may be indicative of poor (hopefully correctable) choices in the original sequence alignment or loop assembly in the region highlighted. Other issues may arise not from the modeling process but rather from using templates that themselves contain errors. To reduce instances of the latter, one may perform similar validation analysis on candidate templates prior to use and thus screen out inferior structures.

Judgments on whether a validation warning warrants countermeasures hinge on how severe the error appears to be, whether it lies in a particular interesting region of the molecule, and on whether it might impact analysis planned for the model. For small molecule docking studies, accurate structures are required in the receptor region, and side chain conformations can be critical, thus one would likely try to alleviate any appreciable error within reasonable nonbonding radius (typically about 8 Å) of the putative binding site. Protein-protein docking studies and MD analyses are generally much less sensitive.

4 Automated Structure Predictions

The years since the first incarnation of this chapter appeared have been witness to a key development in the field of comparative modeling: automated modeling. The practice of computational protein structure prediction has become greatly popularized, in part thanks to a proliferation of user-friendly and generally effective protein structure prediction services. This is a natural outgrowth of the prestigious CASP competitions that have placed a premium on the development of intelligent web-based services that can automatically implement the complete comparative protocol based only on user specification of a specific target protein sequence. There are few computational science tasks as complex as protein structure prediction that have been so effectively implemented as black box protocols.

The primary focus of this chapter (and indeed much of this book) has been practical instruction on how biological researchers can effectively implement complex computational analyses toward better understanding of numerous facets of protein modeling. To some extent the availability of black box solutions diminishes the need for detailed *how-to* instruction, especially if one can generate quality results based on a process as trivial as "*go to site X, paste in your protein sequence, hit "compute" and wait for the results to appear.*" What value, then, is retained from detailed instructions such as those presented herein?

The primary value of a rigorous and fundamental methodological understanding is the ability to objectively perceive whether a protocol is sensible and provides a good probability of generating accurate structures. In most cases, black box services such as the ones that will be briefly reviewed in this section do permit diligent users to examine log files that clearly specify the specific manipulations undertaken *en route* to the final answer. Users should not use black box services that do not conveniently enable such queries, and should make the effort to evaluate the chosen templates, alignments, and refinement strategies at least once for any given target protein. Since black boxes can be as fallible as humans, careful quality control remains important: if validation data is provided by the automated service, the user should scrutinize it for every run that produces a structure that the user plans to use for subsequent analysis. If detailed validation is not provided automatically, users should subject the generated structures to various independent structure validation services discussed in the previous section.

While there are too many servers that perform automated comparative protein modeling to treat each comprehensively in this chapter, it is helpful to at least list some of the most noteworthy:

SwissModel (http://swissmodel.expasy.org/): Perhaps the oldest structure prediction server, SwissModel emerged in 1995 as a value-added service in conjunction with the SwissProt collection of curated protein structure templates [41]. It remains widely used, but is limited to identification of templates with unequivocal homology to the query sequence, which constrains the scope of the service.

I-TASSER (http://zhanglab.ccmb.med.umich.edu/I-TASSER/): Currently the state of the practice protein structure prediction service, receiving top server accolades in each of the CASP7, 8, 9 and 10 competitions, I-TASSER constructs models based on multiple threading alignments [42]. The server is extensively used and relies on an actively updated template library.

RaptorX (http://raptorx.uchicago.edu/about/): An excellent server for cases where the best templates have very low sequence identity relative to the query protein [43], RaptorX is the latest version of a line of servers from the Jinbo Xu group. RaptorX is complemented by various analytical tools including protein binding site prediction, residue contact map analysis, etc.

HHPred (http://toolkit.tuebingen.mpg.de/hhpred): Strictly based on homology modeling (i.e., technically not employing threading), HHPred achieves high quality structure predictions [44] for most of the targets well addressed by threading (it consistently ranks well in CASP competitions) while achieving a very high degree of computational efficiency.

ROBETTA (http://robetta.bakerlab.org/): A unique combination of homology modeling and ab initio self assembly routines, ROBETTA is able to stitch together plausible models of proteins that may be missing sizable portions when modeled by comparative modeling alone [45]. The self assembly portion of this service is performed by ROSETTA which, although much more computationally demanding than purely comparative modeling, employs steering algorithms that render it much more efficient than conventional molecular dynamics methods.

The question of whether it is better to build comparative models in an automated manner or to rely on human rationalization is an issue worthy of concluding this chapter. Fairly thoughtful studies have concluded that although fully automated models perform nearly as well on average as those that with operator intervention, expert involvement is nonetheless beneficial [46, 47], especially in guiding the somewhat instinctive processes of prioritizing machine-recommended templates (in particular, determining which templates correspond to functional states analogous to that which one wishes to model), checking automated alignments, and recognizing functional motifs whose conservation is critical to generating a model that adheres to a function of interest. Conversely, automated threading procedures are often capable of identifying compositionally dissimilar but functionally valid templates that might be perceived even by domain experts, and are probably better equipped to perform the corresponding (often nonintuitive) sequence alignment. Furthermore, a well-validated automated model will probably outperform nonexperts on average. The basic conclusion, perhaps, is that human and machine are converging on a synergistic partnership, with the relative roles of each varying according to the complexities of each case, but remaining complementary. Regardless of the underlying protocol, a well-constructed and validated structure model can open many doors for subsequent analysis. In addition to valuable insight derived from simple visual inspection, the model can form a reliable basis for many other modeling analyses, as are discussed extensively in other chapters of this book.

References

1. Kendrew JC, Bodo G, Dintzis HM, Parrish RG, Wyckoff H, Phillips DC (1958) A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. Nature 181:662–666
2. Konermann L, Pan Y (2012) Exploring membrane protein structural features by oxidative labeling and mass spectrometry. Expert Rev Proteomics 9:497–504
3. Anfinsen CB, Redfield RR, Choate WI, Page J, Carroll WR (1965) Studies on the gross structure, cross-linkages, and terminal sequences in ribonuclease. J Biol Chem 207:201–210
4. Baker D, Šali A (2001) Protein structure prediction and structural genomics. Science 294: 93–96
5. Martí-Renom MA, Stuart AC, Fiser A, Sánchez R, Melo R, Sali A (2000) Comparative protein structure modeling of genes and genomes. Annu Rev Biophys Biomol Struct 29:291–325
6. Lushington GH (2008) Comparative modeling of proteins. Methods Mol Biol 443:199–212
7. Dayhoff MO (1972) Atlas of protein sequence and structure. National Biomedical Research Foundation, Georgetown University, Washington DC
8. Schaeffer RD, Daggett V (2011) Protein folds and protein folding. Protein Eng Des Sel 24:11–19
9. Pearl F, Todd A, Sillitoe I, Dibley M, Redfern O, Lewis T, Bennett C, Marsden R, Grant A, Lee D, Akpor A, Maibaum M, Harrison A, Dallman T, Reeves G, Diboun I, Addou S, Lise S, Johnston C, Sillero A, Thornton J, Orengo C (2005) The CATH domain structure database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. Nucleic Acids Res 33:D247–D251
10. Murzin AG, Brenner SE, Hubbard T, Chothia C (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 247:536–540
11. Prusiner SB (1991) Molecular biology of prion diseases. Science 252:1515–1522
12. Takagi F, Koga N, Takada S (2003) How protein thermodynamics and folding mechanisms are altered by the chaperonin cage: molecular simulations. Proc Natl Acad Sci U S A 100: 11367–11372
13. Kmiecik S, Kolinski A (2011) Simulation of chaperonin effect on protein folding: a shift from nucleation–condensation to framework mechanism. J Am Chem Soc 133:10283–10289
14. Baumketner A, Jewett A, Shea JE (2003) Effects of confinement in chaperonin assisted protein folding: rate enhancement by decreasing the roughness of the folding energy landscape. J Mol Biol 332:701–13
15. Dor O, Zhou Y (2006) Achieving 80% tenfold cross-validated accuracy for secondary structure prediction by large-scale training. Proteins 66:838–845
16. Rost B (2001) Protein secondary structure prediction continues to rise. J Struct Biol 134: 204–218
17. Kryshtafovych A, Fidelis K (2009) Protein structure prediction and model quality assessment. Drug Discov Today 14:386–393
18. Chothia C, Lesk AM (1986) The relation between the divergence of sequence and structure in proteins. EMBO J 5:823–826
19. Chung SY, Subbiah S (1996) How similar must a template protein be for homology modeling by side-chain packing methods? Pac Symp Biocomput 126–141
20. Forrest LR, Tang CL, Honig B (2006) On the accuracy of homology modeling and sequence alignment methods applied to membrane proteins. Biophys J 91:508–517
21. Dongsup K, Xu D, Guo JT, Elrott K, Xi Y (2003) PROSPECT II: protein structure prediction program for genome-scale applications. Protein Eng 16:641–650
22. Brylinski M, Lingam D (2012) eThread: a highly optimized machine learning-based approach to meta-threading and the modeling of protein tertiary structures. PLoS One 7:e50200
23. Soding J (2005) Protein homology detection by HMM-HMM comparison. Bioinformatics 21:951–960
24. Fernandez-Fuentes N, Oliva B, Fiser A (2006) A supersecondary structure library and search algorithm for modeling loops in protein structures. Nucleic Acids Res 34:2085–2097

25. Kolodny R, Koehl P, Guibas L, Levitt M (2002) Small libraries of protein fragments model native protein structures accurately. J Mol Biol 323:297–307
26. Sutcliffe MJ, Haneef I, Carney D, Blundell TL (1987) Knowledge-based modelling of homologous proteins. Part I. Three dimensional frameworks derived from the simultaneous superposition of multiple structures. Protein Eng 1:377–384
27. Levitt M (1992) Accurate modeling of protein conformation by automatic segment matching. J Mol Biol 226:507–533
28. Šali A, Blundell TL (1993) Comparative protein modeling by satisfaction of spatial restraints. J Mol Biol 234:779–815
29. Lushington GH, Zaidi A, Michaelis ML (2005) Theoretically predicted structures of plasma membrane Ca^{2+}-ATPase and their susceptibilities to oxidation. J Mol Graph Model 24: 175–185
30. Tovchigrechko A, Vakser IA (2005) Development and testing of an automated approach to protein docking. Proteins 60: 296–301
31. Wiehe K, Pierce B, Mintseris J, Tong W, Anderson R, Chen R, Weng Z (2005) ZDOCK and RDOCK performance in CAPRI rounds 3, 4, and 5. Proteins 60:207–221
32. Chung SY, Subbiah S (1996) A structural explanation for the twilight zone of protein sequence homology. Structure 4:1123–1127
33. Huang ES, Koehl P, Leavitt M, Pappu RV, Ponder JW (1998) Accuracy of side-chain prediction upon the near-native protein backbones developed by ab initio folding methods. Proteins 33:204–217
34. PyMol v. 1.6 (2013) http://www.pymol.org/
35. Johansson MU, Zoete V, Michielin O, Guex N (2012) Defining and searching for structural motifs using DeepView/Swiss-PdbViewer. BMC Bioinformatics 13:173
36. Caflisch A, Paci E (2005) Molecular dynamics simulations to study protein folding and unfolding. In: Buchner J, Kiefhaber T (eds) Protein folding handbook, vol 2. Weinheim, Wiley-VCH Verlag GmbH, pp 1143–1169
37. Karplus M, Kuriyan J (2005) Molecular dynamics and protein function. Proc Natl Acad Sci USA 102:6679–6685
38. Im W, Brooks CL III (2005) Interfacial folding and membrane insertion of designed peptides studied by molecular dynamics simulations. Proc Natl Acad Sci USA 102: 6771–6776
39. Chen VB, Arendall WB, Headd JJ, Keedy DA, Immormino RM, Kapral GJ, Murray LW, Richardson JS, Richardson DC (2010) MolProbity: all-atom structure validation for macromolecular crystallography. Acta Crystallogr D Biol Crystallogr D66:12–21
40. Berjanskii M, Liang Y, Zhou J, Tang P, Stothard P, Zhou Y, Cruz J, Macdonell C, Lin G, Lu P, Wishart DS (2010) PROSESS: a protein structure evaluation suite and server. Nucleic Acids Res 38:W633–W640 Webserver edition
41. Peitsch MC (1995) Protein modeling by E-mail. Biotechnology 13:658–660
42. Roy A, Kucukural A, Zhang Y (2010) I-TASSER: a unified platform for automated protein structure and function prediction. Nat Protoc 5:725–738
43. Källberg M, Wang H, Wang S, Peng J, Wang Z, Lu H, Xu J (2012) Template-based protein structure modeling using the RaptorX web server. Nat Protoc 7:1511–1522
44. Hildebrand A, Remmert M, Biegert A, Soding J (2009) Fast and accurate automatic structure prediction with HHpred". Proteins 77:128–32
45. Song Y, DiMaio F, Yu-Ruei Wang R, Kim D, Miles C, Brunette TJ, Thompson J, Baker D (2013) High resolution comparative modeling with RosettaCM 21(10):1735–42
46. Das R, Qian B, Raman S, Vernon R, Thompson J, Bradley P, Khare S, Tyka MD, Bhat D, Chivian D, Kim DE, Sheffler WH, Malmstrom L, Wollacott AM, Wang C, Andre I, Baker D (2007) Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@home. Proteins 69:118–128
47. Zhang Y (2009) I-TASSER: fully automated protein structure prediction in CASP8. Proteins 77:100–113

Chapter 15

De Novo Membrane Protein Structure Prediction

Timothy Nugent

Abstract

Recent advances in identifying residue–residue contacts from large multiple sequence alignments have enabled impressive gains to be made in the field of protein structure prediction. In this chapter, we discuss these advances and provide a step-by-step guide to applying the latest tools to the de novo modelling of alpha-helical transmembrane proteins. As a practical example, we demonstrate the process of building an accurate 3D model of a G protein-coupled receptor, correctly orientated in the membrane, using only its primary protein sequence.

Key words Transmembrane protein, De novo modelling, Contact prediction, Structural bioinformatics

1 Introduction

Membrane proteins are encoded by approximately 30 % of the genes of a typical genome and perform crucial roles in a diverse range of essential biological processes including transport of ions and small molecules, intercellular communication and signal transduction. They are also important drug targets, with estimates suggesting that about 60 % of current drug targets are membrane proteins [1]. Recently, there has been encouraging progress in structure determination led by structural genomics initiatives that explicitly target integral membrane proteins, resulting in increasing coverage of important protein families [2, 3]. Despite this, coverage of membrane protein fold space remains sparse as only about 1 % of structures in the Protein Data Bank (PDB) describe membrane proteins, of which about 300 are unique. However, the technical difficulties associated with purification and structure determination by X-ray crystallography and NMR spectroscopy are likely to prohibit a rapid increase in these numbers. Computational structure prediction therefore provides a vital alternative approach with which to further our understanding of both the structure and function of this important class of proteins.

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_15, © Springer Science+Business Media New York 2015

Historically, tools to analyse membrane proteins have focused on topology prediction methods which typically use machine learning algorithms trained on the limited structural data that is available [4–10]. Predicting membrane protein structure by homology modelling can be highly effective when a suitable template is available, especially when membrane protein-specific tools are used [11, 12]. Given that fold preservation in transmembrane regions requires less sequence conservation than for globular proteins, it should be possible to obtain relatively accurate 3D models even at low (<20 %) sequence identity [13], although the small number of solved membrane protein structures will limit the number of families that such methods are applicable to. De novo modelling approaches, which attempt to build 3D models from sequence information alone, have typically relied upon knowledge-based potentials derived from the statistical analysis of known structures. A number of membrane protein de novo modelling tools exist including FILM [14] and Rosetta Membrane [15–17].

FILM (Folding In Lipid Membranes) is based on the globular protein structure prediction method FRAGFOLD [18, 19] which performs a conformational search guided by simulated annealing using highly resolved super secondary structural fragments to assemble the tertiary fold. FILM adds a knowledge-based membrane potential to the standard FRAGFOLD energy terms (pairwise, solvation, steric, and hydrogen bonding), derived from the statistical analysis of a data set of 640 transmembrane helices with experimentally defined topologies. By assessing the relative frequencies of each amino acid at fixed distances from the membrane centre, the membrane term is calculated by transforming these values using the inverse Boltzmann equation. Applying the method to small membrane proteins of known 3D structure showed that it was capable of predicting both the helix topology and the conformations of these proteins at a reasonable accuracy level. However, reproducing the compactness of large transmembrane helix bundles was challenging, since transmembrane helix bundles are usually not optimally compact, although neighboring helices are closely packed. Subsequent modification of the method allowed the prediction of larger bundles by incorporating lipid exposure prediction into the potential function, allowing models of seven transmembrane helix bacteriorhodopsin and rhodposin to be generated with 6–7 Å root mean square deviation (rmsd) to the experimentally determined structure [20].

RosettaMembrane, a modification of the Rosetta method [21, 22], which also assembles folds using fragments of know structures using simulated annealing, added terms to the energy function that described intraprotein and protein–solvent interactions in the anisotropic membrane environment. The method describes interactions between protein residues in atomic detail while applying continuum solvent models to the water, hydrophobic core, and lipid head group regions of the membrane. RosettaMembrane was able to successfully predict the structures of 12 small transmembrane protein domains (up to 150 residues) to within 4 Å rmsd of the native structures, with near-atomic resolution predictions (<2.5 Å rmsd) achieved for three domains, suggesting that the model captures the essential physical properties that govern the solvation and stability of membrane proteins. More recently, the method was extended to incorporate distance constraints into the predictions by constraining helix–helix interactions, derived from both experimental data or predicted from sequence [23–25]. This allowed larger (90–300 residues) structures with more complicated topologies to be successfully modelled to within 4 Å rmsd in the best four cases, with results indicating that only a single constraint was sometimes sufficient to enrich the population of near-native models.

While the use of knowledge-based potentials derived from statistical analyses of known membrane protein structures has been the standard approach for de novo structure prediction, the field has changed dramatically over the last 5 years as new methods capable of accurately inferring residue–residue contacts from large multiple sequence alignments (MSAs) have emerged, allowing the direct computation of structures from sequence. The progress of these methods has been largely driven by the rapid growth in the size of sequence databases; while 5 years ago a database may have contained a few hundred sequences for a typical protein family, the number can easily be ten times as many today [26]. It is this wealth of sequence data that allows detection of correlated mutations between sites in MSAs. The key idea behind correlated mutations is that residues proximal in 3D space are likely to impose constraints on each other, which should lead to a correlation in their substitution patterns in a MSA. Should either residue mutate, the stability of the contact might be disrupted, therefore impacting on the stability of the structure. Should one or both residues mutate to a more physicochemically complementary pairing, the contact is more likely to be retained. Residue pairs that form contacts in the native structure are therefore seen to coevolve in tandem, and it is this property that modern contact prediction methods seek to exploit.

Many different strategies have been applied to predicting contacts from sequence data, typically based on the direct recognition of substitution patterns, and often using machine learning-based approaches. However, the major obstacle has been in dealing with indirect coupling effects: should a direct physical coupling exist between sites AB and BC, an apparent coupling may emerge between AC even though no direct interaction exists. It is this indirect coupling that has the effect of blurring the signal and has been the bottleneck in prediction accuracy. A recent implementation by Lapedes et al. [27] dealt with this chaining problem by applying a maximum entropy approach but at a high computational cost.

Another method, Direct Coupling Analysis (DCA), reduced the problem to one of maximum entropy inference of the strength of the interaction parameters between columns in the MSA, and applied a heuristic message passing approach to determine the solution of the contact weights [28]. This allowed the approach of Lapedes et al. to be put to practical use, with prediction accuracy achieving sufficient quality to be useful in structure prediction [29]. Another method, PSICOV, is based on the properties of the inverse covariance (or precision) matrix, which will only contain nonzero elements where there is direct coupling influence. Calculating the inverse covariance matrix, while not always possible, is usually costly, therefore PSICOV constrains the solution based on sparsity using the graphical lasso method [30, 31]. Where sufficient numbers of aligned sequences were available, PSICOV was able to predict contacts with an accuracy approaching 80 % even for long-range contacts—those separated by >23 residues in the sequence—sufficient to identify to the native fold for medium sized (<200 residue) globular proteins [32]. More recently, the plmDCA method [33], which uses a pseudolikelihood method applied to the Potts models, was shown to significantly outperform existing DCA-based approaches, while a consensus approach PconsC that combines PSICOV and plmDCA predictions using a random forest classifier was shown to provide a relative improvement of 20 % in comparison to the best single method [34].

The prediction accuracy of these recent methods has led to the development of a number of de novo structure prediction methods capable of generating accurate models for even large domains, guided primarily by predicted contacts. EVfold [35], which uses DCA in combination with the CNS molecular dynamics software suite to generate 3D models, was able to determine accurate structures for fifteen targets ranging in size from 50 to 260 residues to within 2.7–4.8 Å rmsd of their native structures over at least two-thirds of the protein. Evfold_membrane [36], which incorporates predicted transmembrane topology into the model, was able to generate model with correct folds (TM-score >0.5, *see* **Note 1**) amongst the top ten scoring candidate models in 22 out of 25 targets. The latest version of FILM, FILM3, replaced the statistical potential with a single scoring function based on predicted contacts and their estimated probabilities [37]. Benchmark results using contacts predicted by PSICOV indicated that models with TM-scores >0.5 could be generated for 25 out of 28 membrane protein targets with complex topologies and an average length over 300 residues. In the best case, it was possible to build a model of cytochrome c oxidase polypeptide I with a TM-score >0.75 over all 514 residues. While these results are extremely encouraging, data suggests that even with perfect distance constraints, folding methods are unable to generate models less than 2 Å rmsd of the native structure, suggesting that protein refinement will play an important role in generating higher accuracy models.

2 Methods

This chapter will describe the process of generating a 3D model of an alpha-helical membrane protein starting with only its primary sequence and in the absence of structural homologues. A number of steps are required that will be discussed in detail: (1) predicting residue–residue contacts, (2) predicting secondary structure and transmembrane topology, (3) generating candidate 3D structures, (4) recombining candidate structures to generate a final model, (5) refinement, (6) orientating of the refined model in the membrane, and (7) model quality assessment. To reproduce these steps you will need access to a UNIX/Linux workstation with a number of software packages and databases installed. While we will focus on tools developed in-house at UCL, many of the methods can easily be substituted or combined with other programs. As a target sequence, we will use the 329-residue bovine rhodopsin protein (PDB code 1GZM), a prototypical G protein-coupled receptor (GPCR).

2.1 Predicting Residue–Residue Contacts

The key to building de novo 3D models using FILM3 is the use of predicted contacts, which are generated here with PSICOV (*see* **Note 2**). The input file for PSICOV is a large and diverse MSA. In the original PSICOV paper, the example with the lowest number of sequences had 511 homologous sequences, so an alignment size upwards of this is critical for accurate predictions. Alignments smaller than this, or with low sequence diversity, may result in PSICOV failing to converge on a solution or aborting, although in both cases this behaviour can be overridden. A number of programs can be used to generate MSAs; here we will use HHblits (*see* **Note 3**) which uses profile hidden Markov models to generate alignments [38]. To build a MSA from the target sequence (in FASTA file format) using a Uniprot [39] database, use the following command:

```
./hhblits -i 1gzmA.fasta -d uniprot20_2013_03
-oa3m 1gzmA.a3m -mact 0 -n 3 -diff inf -cov 60
```

The options generate a MSA in a3m format (-oa3m), setting the maximum accuracy (MAC) algorithm realignment threshold to 0 therefore generating quasi-global alignments, running three iterations (-n), filtering both query and databases to generate the most diverse alignment (-diff), and setting the minimum coverage of the target sequence to 60 % (-cov). Full details of HHblits command line parameters can be found by passing the "–help all" flag. The following command can then be used to convert the a3m formatted MSA into the PSICOV input format, which simply consists of a single sequence per row, with the target sequence—which must contain no gaps—as the first line. Duplicate rows are also removed:

```
egrep -v "^>" 1gzmA.a3m | sed 's/[a-z]//g' |
sort -u > 1gzmA.aln
```

Another method that can be used to build a MSA is jackhmmer, part of the HMMER 3.0 package [40] (*see* **Note 4**). To build a MSA using jackhmmer against a UNIREF100 database, use the following command:

```
./jackhmmer -N 3 -E 1e-6--ncE 1e-6--otextw -A
1gzmA.sto 1gzmA.fasta uniref100.fasta
```

Here the command line flags set the number of iterations to 3 (-N), and the E-value threshold to 1e-6. The resulting Stockholm formatted file (-A) can be converted to a3m format with the reformat.pl Perl script, which is included in the HH-suite package:

```
./reformat.pl 1gzmA.sto 1gzmA.a3m
```

This can then be converted to the PSICOV format using the previous egrep command. If you use an alternative method to generate the MSA which produces a FASTA formatted alignment file, this can be converted to the PSICOV format using the fasta2aln tool included with PSICOV:

```
./fasta2aln 1gzmA.msa.fasta > 1gzmA.aln
```

Having successfully generated the MSA, PSICOV can be run with the following command:

```
./psicov -p -d 0.03 1gzmA.aln > 1gzmA.con
```

Here, the command line options output positive predictive values (PPV) rather than raw output scores (-p) and set the precision matrix sparsity to 0.03 (-d)—justified by the observation that on average only ~3 % of all residue pairs are observed to be in direct contact. The resulting output file is in 5 column CASP RR format (http://predictioncenter.org/casp10/index.cgi?page=format):

```
i j d1 d2 p
```

Where i and j are residue numbers, d1 and d2 are real numbers where values of d1 = 0 and d2 = 8 indicate a predicted contact, and p (probability) is the PSICOV PPV. When passed to FILM3, only contacts with a probability >0.5 will be used to build the model, so this list of contacts can easily be replaced or augmented with contacts determined by experimental methods or other predictors such as plmDCA or PconsC, ensuring that the probability is above this threshold. Contacts that are more confidently predicted will result in a lower pseudo-energy when satisfied in the 3D model.

2.2 Predicting Secondary Structure and Transmembrane Topology

Fragment selection by FILM3 is based upon predicted secondary structure using PSIPRED and transmembrane helix predictions using MEMSAT-SVM. Predictions are combined using a decision tree scheme ensuring that transmembrane helix positions are enforced, additionally inserting a small amount of coil in the centre of predicted transmembrane loops if it did not already exist.

Where transmembrane loop regions are predicted, helix, coil or helix/coil is predicted, depending on the PSIPRED confidence. To generate this ensemble secondary structure string, the PSIPRED and MEMSAT-SVM predictions are combined with the Combine-MEMSAT-SVM-PSIPRED.pl Perl script, which is included in the FILM3 package. To generate the PSIPRED prediction (*see* **Note 5**), run it passing in the target FASTA file:

```
./runpsipred 1gzmA.fasta
```

This will generate two output files containing the predicted secondary structure and neural network probabilities used to make the prediction—1gzmA.ss2 and 1gzmA.horiz.

Run the MEMSAT-SVM (*see* **Note 6**) Perl script as follows:

```
./run_memsat-svm.pl 1gzmA.fasta
```

A number of output files are produced including 1gzmA.memsat_svm, which contains the predicted transmembrane topology, and 1gzmA_SVM_ALL.out, which contains the raw support vector machine (SVM) scores. To generate the ensemble secondary structure, place these four files in the same directory as the Combine-MEMSAT-SVM-PSIPRED.pl Perl script and run it passing in just the base name of the target:

```
./Combine-MEMSAT-SVM-PSIPRED.pl 1gzmA
```

This will generate two files called 1gzmA.ess and 1gzmA.zcoord. The third line of 1gzmA.ess contains the ensemble secondary structure string, where C = coil, H = helix, E = sheet, h = helix or coil, and ? = unknown. This string can easily be modified by hand if you want to enforce a different secondary structure using alternative methods, or the transmembrane helix boundaries in the "Topology" line at the end of the 1gzmA.memsat_svm file can be modified and the Combine-MEMSAT-SVM-PSIPRED.pl re-run. The file 1gzmA.zcoord contains predicted *Z*-axis (perpendicular to the Cartesian plane formed by the membrane surface) coordinates for each residue, which are linearly extrapolated from the transmembrane topology prediction. These are used to provide minimum distance constraints to FILM3, ensuring that the meandering nature of the target protein's transmembrane topology is enforced. These coordinates can easily be replaced by those generated by more elaborate schemes such as ZPRED [41].

2.3 Generating Candidate 3D Structures Using FILM3

The FILM3 input file, 1gzmA.nfpar, has the following format:

```
----------1gzmA.nfpar----------
ALNFILE 1gzmA_FILM3.aln
INITEMP 0.6
MAXSTEPS 20000000
POOLSIZE 9
```

```
TRATIO 0.6
MAXFRAGS 5
MAXFRAGS2 25
CONFILE 1gzmA.con
ZFILE 1gzmA.zcoord
-------------------------------
```

Three files are referenced: the first is an alignment file, although the format is slightly different to that used by PSICOV. Instead, the first three lines from the 1gzmA.ess file are added to the beginning of the PSICOV alignment file. Additionally, the second line is set to the size of the PSICOV alignment file. The file can therefore be generated as follows:

```
head -3 1gzmA.ess > 1gzmA_FILM3.aln
cat 1gzmA.aln >> 1gzmA_FILM3.aln
```

Then simply change the second line in 1gzmA_FILM3.aln to the alignment size, which can be determined with:

```
cat 1gzmA.aln | wc –l
```

The file should then appear as follows, assuming the original PSICOV alignment file contained 1,969 sequences:

```
----------1gzmA_FILM3.aln----------
1gzmA
1969
CCCCCCCChhhhCCCCCCCCCCCCCCCCCCCCChHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHhhCCCCChhHH…
MNGTEGPNFYVPFSNKTGVVRSPFEAPQYYLAEPWQFSMLAAYMFL
LIMLGFPINFLTLYVTVQHKKLRTPLNY…
etc.
-----------------------------------
```

The other two files contained in 1gzmA.nfpar are the contacts predicted by PSICOV—1gzmA.con—and the predicted *Z*-axis coordinate file—1gzmA.zcoord. In the FILM3 paper, the use of *Z*-coordinate distance constraints generated lower energy models in eight cases out of 28; the use of these coordinates can therefore be suppressed by removing or commenting out the corresponding line. The remaining options in the 1gzmA.nfpar control aspects of the Replica Exchange Monte Carlo function, which in general can be left at their default values. The MAXSTEPS parameter is the total number of fragment swaps to make, POOLSIZE is the number of replica conformations to use, and INITEMP and TRATIO indicate the starting temperature and temperate ratio between each replica. MAXFRAG and MAXFRAG2 are the minimum and maximum number of fragments to use for each residue. With the input file complete, FILM3 can be run as follows:

```
./film3 1gzmA.nfpar > 1gzmA.pdb
```

The FILM3 binary (*see* **Note 7**) expects to find four files in the working directory—atomdist.dat, rotalib.dat, tdb.dat, and tor.lst—these contain atomic distance, side-chain and fragment library information, and are include in the FILM3 data directory. Running FILM3 once will generate a single model. For the following recombination step, an ensemble of say 100 models or more are required, and usually the more models that are generated, the more accurate the final model will be. We would recommend generating 200 models per target—100 models with *Z*-coordinate distance constraints, and 100 models without. Below we assume that the models are named 1gzmA_i.pdb, with i running from 1 to the number of models. While FILM3 can be run on a single computer, generating large numbers of models is typically achieved using a cluster of machines.

2.4 Recombining Candidate Structures to Generate a Final Model

Models generated by FILM3 have their energy written in the HEADER record of the PDB file. Rather than simply selecting the final model with the lowest energy, a combinatorial refinement step can be used to generate a single final model from the ensemble of models. Here, the lowest energy model is identified and the next 100 lowest energy models, selected from the pooled set of Z-coordinate constrained and unconstrained models, are fitted to it by rigid body superposition. The recombination randomly selects fragments from the ensemble and transfers them on to the lowest energy structure to see if a lower energy model can be produced. This greedy search procedure is repeated until no further improvement in energy is observed. To generate the ensemble, use the super_models.csh script included in the FILM3 package. This uses the ProFit tool for superpositioning (*see* **Note 8**), but many alternatives exists. Once installed, identify the lowest energy model by parsing the HEADER records:

```
cat *.pdb | grep HEADER | sort –n | tail -1 |
cut -c 21-30 | xargs -i grep -e {} *.pdb
```

Assuming this command identifies 1gzmA_3.pdb as the lowest energy target, and the 100 lowest energy candidate structures are in a directory named “models/,” we now modify the super_models.csh script accordingly:

```
----------super_models.csh----------
set refpdb = models/1gzmA_3.pdb
# Superpose all models found in models
directory
```

```
foreach pdb (models/1gzmA_*.pdb)
echo -n "ref " $refpdb > profit.cmds
echo "" >>profit.cmds
echo -n "mob " $pdb >>profit.cmds
echo "" >>profit.cmds
echo "atoms CA" >>profit.cmds
echo "fit" >>profit.cmds
echo "write temp.pdb" >>profit.cmds
echo "quit" >>profit.cmds
# Assume that profit command is in command path
profit < profit.cmds >/dev/null
cat temp.pdb
echo "END"
end
------------------------------------
```

Then run the scripts as follows to generate the ensemble:

```
csh super_models.csh > 1gzmA_ensemble.pdb
```

Now run contactrecomb, included in the FILM3 package, passing it the PSICOV contact file and the ensemble secondary structure file that were generated previously. This command will generate a single recombined model, 1gzmA_recomb.pdb:

```
./contactrecomb 1gzmA_ensemble.pdb 1gzmA.con
1gzmA.ess 1gzmA_recomb.pdb
```

2.5 Model Refinement

After recombination, the final model can be refined using MODELLER (*see* **Note 9**) to produce reasonable loop and side chain conformations [42]. The recombined model is simply used as a template for MODELLER, but with additional secondary structure restraints applied to regions predicted to be alpha-helical by MEMSAT-SVM. The following two files need to be generated, firstly an alignment file that can be read by MODELLER:

```
----------1gzmA_refine.ali----------
>P1;1gzmA_recomb
structureX:1gzmA_recomb::::PDB::0.00:0.00
MNGTEGPNFYVPFSNKTGVVRSPFEAPQYYLAEPWQFSMLAAYM
FLLIMLGFPINFLTLYVTVQHKKLRTPLNYILLNLAVADLFMVFG
GFTTTLYTSLHGYFVFGPTGCNLEGFFATLGGEIALWSLVVLAIER
YVVVCKPMSNFRFGENHAIMGVAFTWVMALACAAPPLVGWSRY
IPEGMQCSCGIDYYTPHEETNNESFVIYMFVVHFIIPLIVIFFCY
GQLVFTVKEAAAQQQESATTQKAEKEVTRMVIIMVIAFLICWLPY
```

```
AGVAFYIFTHQGSDFGPIFMTIPAFFAKTSAVYNPVIYIMMNKQ
FRNCMVTTLCCGKNDDE*
>P1;SEQ
sequence:SEQ::::SEQ::0.00:0.00
MNGTEGPNFYVPFSNKTGVVRSPFEAPQYYLAEPWQFSMLAA
YMFLLIMLGFPINFLTLYVTVQHKKLRTPLNYILLNLAVADL
FMVFGGFTTTLYTSLHGYFVFGPTGCNLEGFFATLGGEIALW
SLVVLAIERYVVVCKPMSNFRFGENHAIMGVAFTWVMALACA
APPLVGWSRYIPEGMQCSCGIDYYTPHEETNNESFVIYMFVV
HFIIPLIVIFFCYGQLVFTVKEAAAQQQESATTQKAEKEV
TRMVIIMVIAFLICWLPYAGVAFYIFTHQGSDFGPIFMTIPAF
FAKTSAVYNPVIYIMMNKQFRNCMVTTLCCGKNDDE*
------------------------------------
```

In this file, the name of the structure in the first and second lines must match the name of the final recombined model, in this case 1gzmA_recomb. The MODELLER Python script is as follows:

```
----------1gzmA_refine.py----------
from modeller import *
from modeller.automodel import *
log.verbose()
env = environ()
class MyModel(automodel):
    def special_restraints(self, aln):
        rsr = self.restraints
        at = self.atoms
        rsr.add(secondary_structure.alpha(self.
        residue_range(39,63)))
        rsr.add(secondary_structure.alpha(self.
        residue_range(73,96)))
        rsr.add(secondary_structure.alpha(self.
        residue_range(109,133)))
        rsr.add(secondary_structure.alpha(self.
        residue_range(154,173)))
        rsr.add(secondary_structure.alpha(self.
        residue_range(203,224)))
        rsr.add(secondary_structure.alpha(self.
        residue_range(252,274)))
        rsr.add(secondary_structure.alpha(self.
        residue_range(287,309)))
a = MyModel(env, alnfile = '1gzmA_refine.ali',
knowns = '1gzmA_recomb', sequence = '1gzmA_
recomb')
```

```
a.starting_model= 1
a.ending_model  = 1
a.md_level = refine.slow
a.make()
-----------------------------------
```

Here, the residue ranges to which alpha-helical secondary structure restraints are applied, according to MEMSAT-SVM transmembrane helix boundary predictions, can be added using the rsr.add command. The alignment file and model name must be referenced accordingly on the following line. The actual refinement step is initiated by the refine.slow command, which uses molecular dynamics with simulated annealing [43]. Finally, run the script using Python:

```
python 1gzmA_refine.py
```

This will generate the final recombined and refined model (Fig. 1).

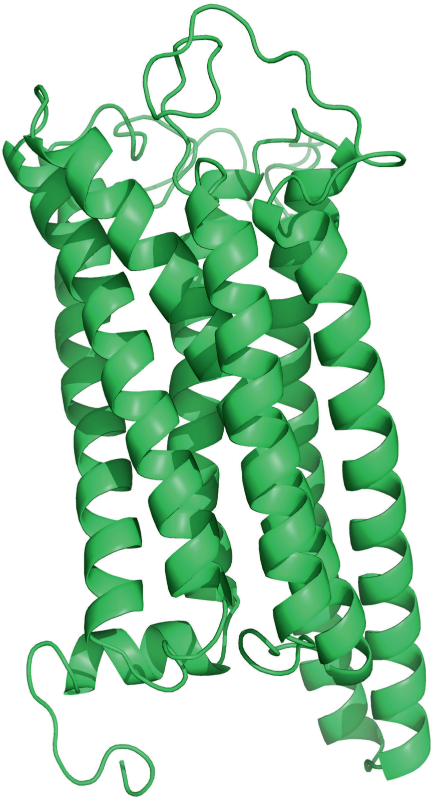


Fig. 1 The final recombined and refined model of rhodopsin

2.6 Orientation of the Model in the Membrane

Identifying the correct orientation of the model within the lipid bilayer allows us to study the complex relationship between sequence, structure, and the lipid environment. Using MEMEMBED (*see* **Note 10**), which couples a knowledge-based membrane potential, calculated by the statistical analysis of transmembrane protein structures, with a combination of genetic and direct search algorithms, we can quickly and accurately orientate membrane proteins within the lipid bilayer [44]. To orientate the model, run the following command:

```
./memembed -n out -s 3 -q 1 1gzmA_refined.pdb
```

This should generate a file called 1gzmA_refined_EMBED.pdb (Fig. 2). Here, the optional flags tell the program that the N-terminus of the model has an extracellular location (-n out), as determined by the MEMSAT-SVM prediction, the "–s 3" flag tells the program to search for the optimal orientation using a genetic algorithm, repeating the search five times, and then returning the lowest energy orientation, while the "-q 1" flag will optimise the hydrophobic thickness of the bilayer after orientation, returning a value in angstroms, and positioning the membrane leaflets appropriately rather than at the default values of –15 and 15 Å.

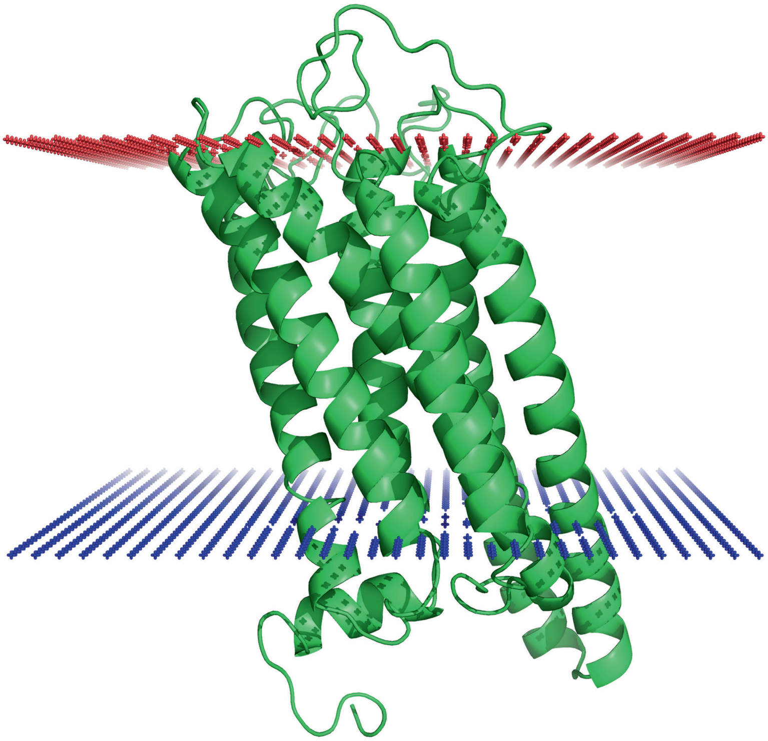


Fig. 2 The model orientated in the membrane. The *blue plane* indicates the membrane inner leaflet; the *red plane* is the membrane outer leaflet

By passing in the model's predicted transmembrane helix boundaries, the tilt angle relative to the membrane for each helix and the model as whole can also be calculated for the orientated model:

```
./memembed -z -r "A" -t 39,63,73,96,109,133,154
,173,203,224,252,274,287,309
1gzmA_refined_EMBED.pdb
```

The "–z" flag tells the program to calculate tilt angles using the comma separated helix boundary list provided by "–t". The "–r" flag indicates which chain the topology refers to—if the model has no chain identifier, pass an empty character in quotes.

Finally, if PyMOL [45] is installed, the included Python script can easily be used to generate an image of the orientated structure:

```
./pymol_membrane_image.py  1gzmA_refined_EMBED.
pdb 1gzmA_refined_EMBED.png
```

In addition to MEMEMBED, a number alternative methods exists that can position structures accurately in the membrane. These include the TMDET, E_z-3D, and PPM servers [46–48].

2.7 Model Quality Assessment

Making an assessment of the quality of a model is clearly important step since it provides users with a measure of confidence in the prediction. Unfortunately, for de novo models, this is challenging due to the absence of an experimental structure to compare against. There are however a number of ways in which an assessment of model quality can be made. Firstly, the estimated precision of contacts predicted by PSICOV can be used. In development of FILM3, it was possible to build models with TM-scores >0.5, therefore indicating approximately the correct fold [49], for 26 out of 28 targets where there were at least 20 predicted contacts with a PPV >0.5. The two targets that met this criteria but had a TM-score <0.5 were both part of a larger complex and appear to be heavily stabilised by additional chains in their native states. The second method is to calculate the mean pairwise TM-score between all models in the ensemble of candidate structures. Where predicted contacts are sufficient to determine the correct fold, there should be a high degree of similarity in the ensemble. We found that the mean pairwise TM-score showed a strong correlation with the TM-score of the final model, therefore allowing the final model TM-score to be predicted using linear regression. You can use the film3mqap program to calculate these values using the ensemble that was generated previously:

```
./film3mqap 1gzmA_ensemble.pdb
```

This will generate a mean pairwise TM-score for the ensemble and a predicted TM-score for the final model—a value >0.5 indicating the model probably has the correct fold. The program also computes a pseudo-temperature-factor from the ensemble of models

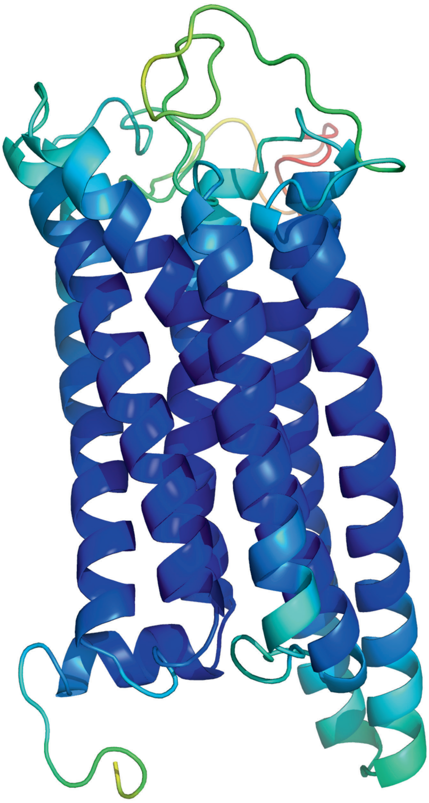


Fig. 3 The final model coloured by pseudo-temperature-factor. *Warmer colours* indicate high variation in the ensemble. These regions, particularly the N-terminus at the *top* of the image, correspond to areas of the contact map where there are few predicted contacts

and writes these to an output file (Fig. 3). When displayed in a graphics package such as PyMOL with the residues are coloured by temperature, regions in the ensemble that are clearly defined and thus more likely to be correct can easily be identified:

```
./film3mqap 1gzmA_ensemble.pdb 1gzmA_refined.pdb
1gzmA_refined_bfactors.pdb
```

It is also possible to use the predicted transmembrane topology to assess models. Modern topology prediction methods can achieve accuracies approaching 90 % on certain data sets; therefore, ensuring that the model displays the correct topology is a relatively straightforward way of assessing its quality. By loading the orientated structure into PyMOL, the predicted transmembrane helices can be selected and coloured as follows:

```
select tmh, resi 39-63 + resi 73-96 + resi 109-
133 + resi 54-173 + resi 203-224 + resi 252-274
+ resi 287-309; color orange, tmh
```

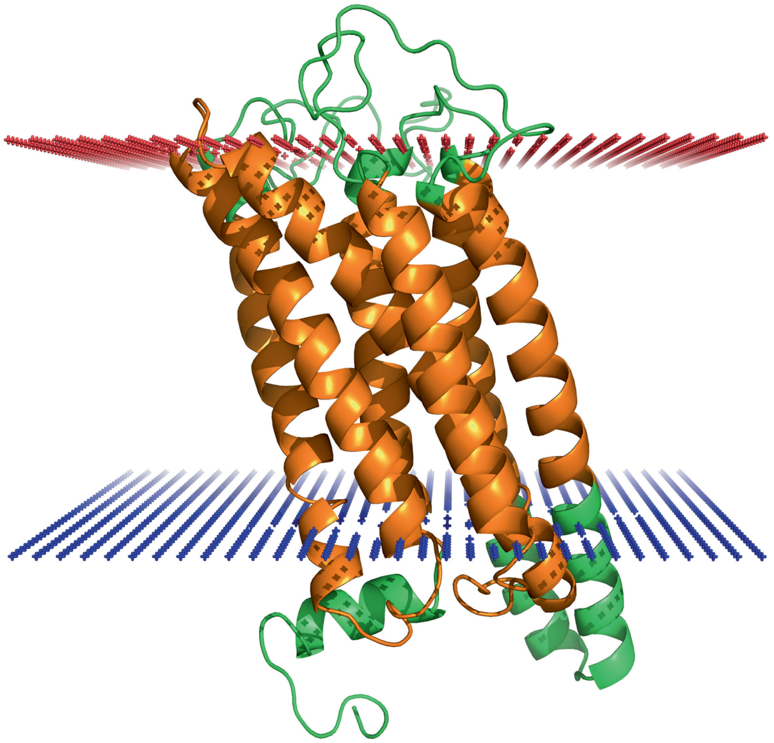


Fig. 4 Transmembrane helices are coloured in *orange*. It is clear that all seven helices lie within the membrane plane

It should be fairly clear if any predicted transmembrane helices lie outside the plane of the membrane (Fig. 4). While it is unreasonable to expect perfect bundles of helices lying perpendicular to the membrane, excessive tilt angles should be inspected closely. At this stage, it may be worth regenerating the ensemble using different subsets of candidate structures. For example, the *Z*-coordinate distance constraints are only useful in a minority of cases. If the topology of the final model built from an ensemble which included *Z*-coordinate constrained candidates was implausible, it may be worth generating the ensemble using only the unconstrained candidates. Similarly, the recombination step produced better models in only 18 out of 28 targets. It is probably wise to inspect the lowest energy candidate model, and compare this with the recombined structure.

2.8 Managing Expectation

The example used here, bovine rhodopsin, represents a good target, consisting of a single transmembrane domain with enough aligned sequences to produce a reliable model. However, other targets may prove more challenging for a number of reasons. Firstly, if the target membrane domain is part of a multi-domain protein, it will need to be parsed from the rest of the sequence before generating alignments, predicting contacts and modelling using FILM3. A number of tools including DomPred [50] can be used for this task. Care also needs to be taken when modelling chains that are part of complexes. In the cases of homo-multimeric chains, it can be expected that both inter and intrachain contacts will coevolve, and contact predictors should identify both types. Currently, we are unable to distinguish between the two, which is likely to be problematic for methods such as FILM3 that will attempt to satisfy as many predicted contacts as possible, with the expectation that they are all intrachain contacts. In general, chains that are stabilised via interactions with other monomers in a complex tend to produce poor models and should be avoided where possible. Other targets that are likely to be challenging are those which undergo significant conformational change upon activation, for example transporters that adopt distinct alternate conformations, therefore requiring different sets of contacts to stabilise each state. Again, the FILM3 objective function will attempt to satisfy such multiple sets of contacts simultaneously, therefore producing a model that may represent the average of two conformations. On the other hand, targets which undergo relatively little conformational change upon activation are likely to produce good models.

3 Conclusions

This chapter should provide a useful introduction to de novo 3D modelling of alpha-helical membrane protein using predicted contacts. Presented here are a number of powerful tools that, in combination, are capable of generating accurate models of large transmembrane protein domains. Such models should be particularly useful for directing experimental studies on families where structural data is unavailable. It is clear that the use of contacts predicted by methods such as PSICOV provide extremely powerful constraints for de novo modelling, and it is likely that this strategy will become applicable to even more protein families as sequence databases continue to grow. We estimate that the PFAM database [51] contains more than 500 single architecture transmembrane domains with >400 aligned sequences—enough to accurately predict contacts using methods such as PSICOV, plmDCA, or PconsC—but no experimentally determined 3D structure. Applying FILM3 to these families has the potential to significantly expand our knowledge of transmembrane fold space, and it is likely that many of these families will be of significant biomedical and pharmacological interest.

4 Notes

1. The TM-score is intended to be a more accurate measure of structural alignment compared to rmsd or GDT. Scores are in the range (0, 1], with 1 indicating a perfect match between two structures, scores below 0.20 typically correspond to randomly chosen unrelated proteins, while scores >0.5 are roughly the same fold [49].
2. PSICOV can be downloaded from http://bioinfadmin.cs.ucl.ac.uk/downloads/PSICOV/. Follow the included compilation instructions to build the PSICOV binary.
3. HHblits binaries and source code and accompanying databases can be downloaded from: http://toolkit.genzentrum.lmu.de/hhblits/.
4. HMMER binaries and source code can be downloaded from http://hmmer.janelia.org/.
5. PSIPRED can be downloaded from http://bioinfadmin.cs.ucl.ac.uk/downloads/psipred/. The NCBI toolkit (ftp://ftp.ncbi.nih.gov) and PSI-BLAST (ftp://ftp.ncbi.nih.gov/blast) are also required. Configure the PSIPRED script by adding the NCBI binary directory and database paths. Follow the included compilation instructions to build the PSIPRED binary.
6. Download MEMSAT-SVM from http://bioinfadmin.cs.ucl.ac.uk/downloads/memsat-svm/, configuring it in exactly the same way as PSIPRED.
7. FILM3 can be downloaded from http:vbioinfadmin.cs.ucl.ac.uk/downloads/FILM3/. Compile the three programs as per the instructions.
8. ProFit can be downloaded from http:vwww.bioinf.org.uk/software/profit/.
9. Download MODELLER from http:vsalilab.org/modeller/. You will need to register to receive the license key required to run it.
10. MEMEMBED can be downloaded from: http://bioinf.cs.ucl.ac.uk/downloads/memembed/. Follow the included compilation instructions to build the MEMEMBED binary.

References

1. Hopkins AL, Groom CR (2002) The druggable genome. Nat Rev Drug Discov 1:727–730
2. Kloppmann E, Punta M, Rost B (2012) Structural genomics plucks high-hanging membrane proteins. Curr Opin Struct Biol 22:326–332
3. Pieper U, Schlessinger A, Kloppmann E et al (2013) Coordinating the impact of structural genomics on the human α-helical transmembrane proteome. Nat Struct Mol Biol 20:135–138
4. Käll L, Krogh A, Sonnhammer ELL (2005) An HMM posterior decoder for sequence feature

prediction that includes homology information. Bioinformatics 21(Suppl 1):i251–i257

5. Bernsel A, Viklund H, Falk J et al (2008) Prediction of membrane-protein topology from first principles. Proc Natl Acad Sci U S A 105:7177–7181
6. Viklund H, Bernsel A, Skwark M et al (2008) SPOCTOPUS: a combined predictor of signal peptides and membrane protein topology. Bioinformatics 24:2928–2929
7. Viklund H, Elofsson A (2008) OCTOPUS: improving topology prediction by two-track ANN-based preference scores and an extended topological grammar. Bioinformatics 24: 1662–1668
8. Bernsel A, Viklund H, Hennerdal A et al (2009) TOPCONS: consensus prediction of membrane protein topology. Nucleic Acids Res 37:W465–W468
9. Nugent T, Jones DT (2009) Transmembrane protein topology prediction using support vector machines. BMC Bioinformatics 10:159
10. Nugent T, Jones DT (2012) Detecting pore-lining regions in transmembrane protein sequences. BMC Bioinformatics 13:169
11. Kelm S, Shi J, Deane CM (2010) MEDELLER: homology-based coordinate generation for membrane proteins. Bioinformatics 26: 2833–2840
12. Hill JR, Deane CM (2013) MP-T: improving membrane protein alignment for structure prediction. Bioinformatics 29:54–61
13. Olivella M, Gonzalez A, Pardo L et al (2013) Relation between sequence and structure in membrane proteins. Bioinformatics 29: 1589–1592
14. Pellegrini-Calace M, Carotti A, Jones DT (2003) Folding in lipid membranes (FILM): a novel method for the prediction of small membrane protein 3D structures. Proteins 50:537–545
15. Yarov-Yarovoy V, Schonbrun J, Baker D (2006) Multipass membrane protein structure prediction using Rosetta. Proteins 62:1010–1025
16. Barth P, Wallner B, Baker D (2009) Prediction of membrane protein structures with complex topologies using limited constraints. Proc Natl Acad Sci U S A 106:1409–1414
17. Barth P, Schonbrun J, Baker D (2007) Toward high-resolution prediction and design of transmembrane helical protein structures. Proc Natl Acad Sci U S A 104:15682–15687
18. Jones DT (1997) Successful ab initio prediction of the tertiary structure of NK-lysin using multiple sequences and recognized supersecondary structural motifs. Proteins Suppl 1:185–191
19. Jones DT, McGuffin LJ (2003) Assembling novel protein folds from super-secondary structural fragments. Proteins 53(Suppl 6):480–485
20. Hurwitz N, Pellegrini-Calace M, Jones DT (2006) Towards genome-scale structure prediction for transmembrane proteins. Phil Trans Roy Soc Lond B Biol Sci 361:465–475
21. Simons KT, Bonneau R, Ruczinski I et al (1999) Ab initio protein structure prediction of CASP III targets using ROSETTA. Proteins Suppl 3:171–176
22. Rohl CA, Strauss CEM, Misura KMS et al (2004) Protein structure prediction using Rosetta. Methods Enzymol 383:66–93
23. Fuchs A, Kirschner A, Frishman D (2009) Prediction of helix-helix contacts and interacting helices in polytopic membrane proteins using neural networks. Proteins 74:857–871
24. Lo A, Chiu Y-Y, Rødland EA et al (2009) Predicting helix-helix interactions from residue contacts in membrane proteins. Bioinformatics 25:996–1003
25. Nugent T, Jones DT (2010) Predicting transmembrane helix packing arrangements using residue contacts and a force-directed algorithm. PLoS Comput Biol 6:e1000714
26. Sadowski MI, Taylor WR (2013) Prediction of protein contacts from correlated sequence substitutions. Sci Prog 96:33–42
27. Lapedes A, Giraud B, Liu L et al (1999) Correlated mutations in models of protein sequences: phylogenetic and structural effects. Stat Mol Biol Genet 33:236–256
28. Weigt M, White RA, Szurmant H et al (2008) Identification of direct residue contacts in protein–protein interaction by message passing. Proc Natl Acad Sci U S A 106(1):67–72
29. Taylor WR, Sadowski MI (2011) Structural constraints on the covariance matrix derived from multiple aligned protein sequences. PLoS One 6:e28265
30. Friedman J, Hastie T, Tibshirani R (2008) Sparse inverse covariance estimation with the graphical lasso. Biostatistics 9:432–441
31. Jones DT, Buchan DWA, Cozzetto D et al (2012) PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. Bioinformatics 28:184–190
32. Taylor WR, Jones DT, Sadowski MI (2012) Protein topology from predicted residue contacts. Protein Sci 21:299–305
33. Ekeberg M, Lövkvist C, Lan Y et al (2013) Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models. Phys Rev E Stat Nonlin Soft Matter Phys 87:012707

34. Skwark MJ, Abdel-Rehim A, Elofsson A (2013) PconsC: combination of direct information methods and alignments improves contact prediction. Bioinformatics 29:1815–1816
35. Marks DS, Colwell LJ, Sheridan R et al (2011) Protein 3D structure computed from evolutionary sequence variation. PLoS One 6:e28766
36. Hopf TA, Colwell LJ, Sheridan R et al (2012) Three-dimensional structures of membrane proteins from genomic sequencing. Cell 149: 1607–1621
37. Nugent T, Jones DT (2012) Accurate de novo structure prediction of large transmembrane protein domains using fragment-assembly and correlated mutation analysis. Proc Natl Acad Sci U S A 109:E1540–E1547
38. Remmert M, Biegert A, Hauser A et al (2011) HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nat Methods 9:173–175
39. Magrane M, Consortium U (2011) UniProt knowledgebase: a hub of integrated protein data. J Biol Databases Curat, Database, p 2011
40. Finn RD, Clements J, Eddy SR (2011) HMMER web server: interactive sequence similarity searching. Nucleic Acids Res 39:W29–W37
41. Granseth E, Viklund H, Elofsson A (2006) ZPRED: predicting the distance to the membrane center for residues in -helical membrane proteins. Bioinformatics 22: e191–e196
42. Martí-Renom MA, Stuart AC, Fiser A et al (2000) Comparative protein structure modeling of genes and genomes. Annu Rev Biophys Biomol Struct 29:291–325
43. Sali A, Blundell TL (1993) Comparative protein modelling by satisfaction of spatial restraints. J Mol Biol 234:779–815
44. Nugent T, Jones DT (2013) Membrane protein orientation and refinement using a knowledge-based statistical potential. BMC Bioinformatics 14:276
45. Schrödinger L (2010) The PyMOL molecular graphics system, version 1.3r1
46. Tusnády GE, Dosztányi Z, Simon I (2005) TMDET: web server for detecting transmembrane regions of proteins by using their 3D coordinates. Bioinformatics 21:1276–1277
47. Senes A, Chadi DC, Law PB et al (2007) E(z), a depth-dependent potential for assessing the energies of insertion of amino acid side-chains into membranes: derivation and applications to determining the orientation of transmembrane and interfacial helices. J Mol Biol 366:436–448
48. Lomize MA, Pogozheva ID, Joo H et al (2012) OPM database and PPM web server: resources for positioning of proteins in membranes. Nucleic Acids Res 40:D370–D376
49. Xu J, Zhang Y (2010) How significant is a protein structure similarity with TM-score = 0.5? Bioinformatics 26:889–895
50. Bryson K, Cozzetto D, Jones DT (2007) Computer-assisted protein domain boundary prediction using the DomPred server. Curr Protein Pept Sci 8:181–188
51. Punta M, Coggill PC, Eberhardt RY et al (2012) The Pfam protein families database. Nucleic Acids Res 40:D290–D301

Chapter 16

NMR-Based Modeling and Refinement of Protein 3D Structures

Wim F. Vranken, Geerten W. Vuister, and Alexandre M.J.J. Bonvin

Abstract

NMR is a well-established method to characterize the structure and dynamics of biomolecules in solution. High-quality structures can now be produced thanks to both experimental advances and computational developments that incorporate new NMR parameters and improved protocols and force fields in the structure calculation and refinement process. In this chapter, we give a short overview of the various types of NMR data that can provide structural information, and then focus on the structure calculation methodology itself. We discuss and illustrate with tutorial examples "classical" structure calculation, refinement, and structure validation approaches.

Key words NMR, Structure calculation, Structure refinement, Structure validation

1 Introduction

The first step of a structure determination by NMR spectroscopy consists of the acquisition of NMR data, typically using heteronuclear multidimensional experiments, that allow the assignment of all atoms/spins of a molecule (^{1}H, ^{15}N, ^{13}C) to their chemical shift values (Fig. 1). Once this chemical shift assignment step is completed, ^{13}C- and ^{15}N-edited 3D NOESY spectra are generally used to obtain inter-atomic distances from nuclear Overhauser effects (NOE). These NOESY spectra provide the most detailed structural information that can be obtained from NMR and are still the most common core data used to define the 3D structure of the protein [1, 2]. In addition to distance information, other parameters, such as J-couplings [3], residual dipolar couplings (RDCs) [4], paramagnetic relaxation enhancements (PRE) and pseudo-contact shifts [5] can be measured, providing additional information to define the protein structure. Recent developments have enabled the calculation of the structures of relatively small

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_16, © Springer Science+Business Media New York 2015

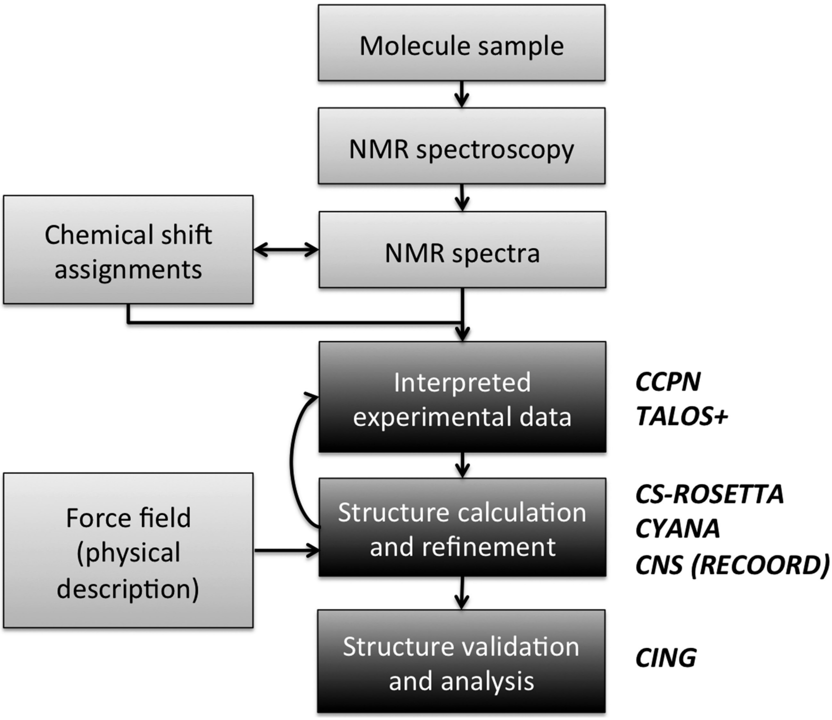

Molecule sample
NMR spectroscopy
Chemical shift assignments
NMR spectra
Interpreted experimental data
CCPN
TALOS+
Force field (physical description)
Structure calculation and refinement
CS-ROSETTA
CYANA
CNS (RECOORD)
Structure validation and analysis
CING

Fig. 1 Schematic overview of the structure calculation process in NMR. This chapter deals with the boxes in *black*, with indication of the software covered

proteins (less than ~12 kDa) from chemical shift values alone [6–8], a procedure that will be briefly outlined in this paper.

The experimental NMR parameters are converted to in silico restraints and 3D structures are generated from restrained molecular dynamics simulations following usually some form of molecular dynamics simulated annealing scheme (MD/SA) [9]. Multiple structures are calculated in this way, starting from the same experimental data but different random starting conditions. Provided that enough data of sufficient quality are available, the structures will converge onto the same overall fold. These structures are nowadays often further refined in explicit solvent (water), which has been shown to significantly improve their quality [10, 11]. Finally, the structures that best satisfy as many experimental restraints as possible, together with proper general chemical properties of proteins (such as bond lengths and angles), are then selected to form an ensemble of structures that represents the definitive solution of the structure calculation process.

In this chapter we will discuss the "classical" NMR structure calculation, refinement, and validation methods, with some reference to new chemical shift-based approaches. These will be illustrated with tutorial examples making use of the programs CYANA

[12] and CNS [13, 14] using the RECOORD [11] approach for water refinement [10], and the CS-ROSETTA protocol [7] based on chemical shift data. These are followed by a description of structure validation with the program CING [15].

2 Theory

This section gives a very brief overview of the NMR data relevant in the determination of 3D protein structures.

2.1 NMR Structural Information Sources

Several NMR parameters providing structural information can be measured for use in structure calculations and refinement; these are briefly described here.

2.1.1 Chemical Shifts

The first step in analyzing NMR experiments is to assign the observed resonances, each of which has a particular chemical shift value, to atoms in the molecule being studied. Although chemical shifts are very sensitive probes of the chemical environment of a spin, their dependency on the 3D structure is complex and their usage in 3D structure calculation is still evolving. A common use is to determine per-amino acid residue secondary structure preferences using the deviations of actual chemical shifts values from those of random coil peptides. This approach can be used to restrict the local conformation of a residue to a given region of the Ramachandran plot, either through torsion angle restraints [16] or by special database potential functions [17]. The last years have also seen the development of methods that can calculate protein structures from chemical shifts only [6–8]. In those, the chemical shifts are typically used to select peptide fragments that are then assembled to produce a 3D model. One of these, CS-ROSETTA [7], will be briefly illustrated in this chapter.

Many software programs are now available that perform the reverse operation, i.e. predict chemical shifts from in silico 3D structures (ShiftX [18], ShiftX2 [19], SHIFTS [20], SHIFTCALC [21], PROSHIFT [22] and SPARTA+ [23]). They can be used for example for structure validation.

2.1.2 NOEs

Classical protein structure determination by NMR relies on obtaining a dense network of distance restraints derived from nuclear Overhauser effects (NOEs) between nearby hydrogen atoms in a protein [1, 2]. Together, these restraints provide the essential information for defining the tertiary structure of a protein.

The NOE originates from cross-relaxation between dipolar coupled spins as a result of through-space spin–spin interactions that result in the transfer of magnetization from one spin to another. The NOE approximately scales with the distance r between the two spins as $1/r^6$. Because of this $1/r^6$ dependency, NOEs are only detected between protons less than 5–6 Å away in space and are thus strongly biased towards the shorter distances. In addition, they are sensitive to dynamics averaging and suffer from so-called spin diffusion effects. Nevertheless, the large number of potential NOE-derived restraints renders them very valuable when treated appropriately.

2.1.3 J-Couplings

Scalar or J-couplings are mediated through chemical bonds connecting two spins. Particularly informative are the vicinal ^{3}J scalar coupling constants between atoms separated by three covalent bonds, which are correlated to the enclosed torsion angle, Θ, by the empirical Karplus equation [24]. In particular, ^{3}J(HN–Hα) and ^{3}J(Hα–Hβ) give information about the backbone φ-angle and the side-chain χ_1 angle in an amino acid, respectively. For amino acid with diastereotopic protons (Hβ2/Hβ3), the use of ^{3}J(Hα–Hβ) coupling does require their stereospecific assignments.

In contrast to the NOEs, scalar coupling constants only provide information on the local conformation of a polypeptide chain. J-couplings are commonly converted into dihedral angle restraints [1] or directly used as J-coupling restraints [25, 26] in NMR structure calculations. Nowadays, the less accurate dihedral angle restraints predicted from chemical shift values are in more common use.

2.1.4 Hydrogen Bonds

Some hydrogens, such as those present in the backbone amide groups, can chemically exchange with hydrogens from the water (H_2O) solvent. Experiments where the H_2O solvent is replaced by the proton–NMR-inactive D_2O can determine which of these hydrogens only exchange slowly; such hydrogens are assumed to be protected from the solvent and/or involved in a hydrogen bond [27]. Identification of the acceptor atom requires either additional experimentation or implicit assumptions. The latter is based on NOEs and/or the regularity of secondary structures and the inferred hydrogen bond restraints should be used with caution. Hydrogen bonds can also be detected directly from cross-hydrogen bond scalar coupling measured from constant time HNCO spectra [28, 29], thus providing accurate restraints for structure calculations [30]. Hydrogen bond restraints are usually introduced into the structure calculation protocol as distance restraints, typically by confining the donor-hydrogen/acceptor distance to a given range.

2.1.5 Residual Dipolar Couplings

Residual dipolar couplings (RDCs) are now a well-established source of structural information [31, 32]. They can be measured in solution by weakly aligning the molecule using a variety of methods [33]. RDCs provide orientational information of the internuclear vector of the two atoms for which the RDC is measured relative to three globally defined axes in the molecule, i.e. those of the alignment tensor. Note that if a RDC is measured between two atoms that are not at a fixed distance from each other, there is also a distance dependence and hence usually only RDCs measured for inter-nuclear vectors with a fixed distance are used in the structure calculations. Residual dipolar couplings can be added as orientational restraints to the target function of the structure calculation algorithm [34].

2.1.6 Diffusion Anisotropy

For non-isotropical tumbling molecules, NMR relaxation data contain orientational information comparable to RDCs as result of the diffusion anisotropy [35]. NMR relaxation is characterized by relaxation times T_1 and T_2, and the T_1/T_2 ratio can be used to define diffusion anisotropy restraints in NMR structure calculations [36]. Again the orientation information comes from the angles of inter-nuclear vectors in an external frame, which, in the case of diffusion anisotropy data, corresponds to the orientational diffusion tensor frame. In practice, ^{15}N T_1 and T_{2-} relaxation data are most often used.

2.1.7 Paramagnetic Relaxation Effects

If a paramagnetic metal ion is present in a protein, or if it is introduced via for example a chelating agent chemically bound to the protein, the NMR signals of the nuclei in a shell around it will be affected [37] by several effects including contact and pseudo-contact shifts, relaxation rate enhancements, and cross-correlation effects. Analogously to RDC and diffusion anisotropy, these, depending on their type, can provide both distance and orientation information which can be converted into restraints to be used in various structure calculation softwares [38, 39].

2.2 Structure Calculation Software

The experimental information sources discussed above can be converted into restraints that can be used in the structure calculation process. Several computer programs have provisions for using the experimental NMR restraints; the most commonly used ones are CNS [13, 14], Xplor-NIH [40] and CYANA [12, 41], although many others are available, e.g. SCULPTOR [42], the SANDER module of AMBER [43], GROMACS [44] and YASARA [45].

Structure calculations in essence transform the experimental data (as restraints) into in silico atomic coordinate information. The calculations are usually based on some molecular dynamic simulated annealing protocol performed in torsion angle and/or Cartesian space, followed by a final refinement phase in explicit solvent (water). A general feature of all these protocols is the usage of a “target function”: lower values of this function for a calculated structure indicates better agreement with the experimental data and with known molecular information. This molecular information is defined by a force field that contains physical energy terms for interactions such as van der Waals interactions and electrostatics, as well as terms describing the molecular geometry such as bond lengths, bond angles, etc. During the initial stages of a structure calculation often the description of some of these terms is simplified to increase the computational speed and/or simplify the energy landscape to be search. For example, long-range nonbonded interactions are reduced to only repulsions between atoms and electrostatic interactions are at first neglected. A full nonbonded representation, including van der Waals (Lennard-Jones) and electrostatic (Coulomb) interactions, is then typically reintroduced for final refinement in explicit solvent [10].

2.3 Structure Selection and Structural Quality

Typically a large pool of structures is generated during the structure calculation process, from which a final ensemble of "best" structures is then selected. This choice of an ensemble of structures, rather than one single one, reflects the uncertainty in the experimental NMR data: often structures that agree with the experimental data equally well but differ locally (such as in loop regions) can be obtained. The most widely used structure selection procedure is based on the agreement with the experimental data (rather arbitrarily defined as a small number of restraint violations) and a low (overall) energy of the structures. Typically ensembles containing the 20 lowest energy models are selected, although this number is arbitrary. Ideally, the selected ensemble should represent the available conformational space accessible to the structure while simultaneously satisfying the experimental restraints. From this ensemble, a representative structure is usually defined; no real consensus exists, however, on how it should be selected. The wwPDB NMR validation taskforce recommends to select the structure that differs the least from all other structures within the ensemble, i.e. the mediod (the structure with the lowest atom coordinate RMSD from all other structures)[46].

The final ensemble is subsequently subjected to structure validation procedures in order to verify its quality. It is useful to distinguish the well-defined from the ill-defined regions of the ensemble during this process: these can be defined with, for example, CYRANGE [47], FindCore [48] or circular variance methods [49]. In practice, the quality indicators that are most commonly used to assess especially the well-defined regions of an NMR ensemble are [50]:

- the goodness of fit to the experimental data, by analyzing restraint violations;
- the precision of the ensemble, measured by positional root mean square deviation (RMSD);
- several physical and stereochemical quality indicators that assess the local and overall quality of protein structures, many of them based on knowledge from high-resolution X-ray structures.

Table 1 lists the most commonly used validation programs; some of which will be described later.

Table 1
Internet resources of NMR-related programs and databases mentioned in this chapter

Software	Internet address	Purpose
CNS	http://cns.csb.yale.edu/v1.3	Multilevel hierachical approach for the most commonly used algorithms in macromolecular structure determination (NMR, crystallography)
RECOORD	http://www.ebi.ac.uk/pdbe-apps/nmr/recoord/	Database of recalculated NMR structures with the CNS scripts used in the tutorial example
PDBe	http://www.pdbe.org/	An information portal to biological macromolecules structures
BMRB	http://www.bmrb.wisc.edu	Biological magnetic resonance data bank
CCPN	http://www.ccpn.ac.uk	A collaborative computing project for NMR
CING	http://nmr.cmbi.ru.nl/icing/	A server to validate the quality of NMR structures
PSVS	http://psvs-1_3.nesg.org/	A server to validate the quality of NMR structures
TALOS+	http://spin.niddk.nih.gov/bax/nmrserver/talos/	Protein backbone angle restraints from searching a database for chemical shift and sequence homology
CYANA	http://www.cyana.org	Structure calculation program (paid license required)
Below accessible via WeNMR framework		
CS-ROSETTA	http://haddock.science.uu.nl/enmr/services/CS-ROSETTA3	Structure calculation program using only chemical shift information
TALOS+	http://haddock.science.uu.nl/enmr/services/TALOS	Protein backbone angle restraints from searching a database for chemical shift and sequence homology
CYANA	http://www.enmr.eu/webportal/cyana.html	Structure calculation program (paid license required)
FormatConverter	http://haddock.chem.uu.nl/enmr/format-converter.html	Conversion between file formats (this web version limited compared to local install)

3 Methods

In this tutorial section we describe the procedures to generate various types of NMR restraints and how they can then be used in structure calculations using CYANA or CNS with the RECOORD scripts. The CS-ROSETTA chemical shift-only approach is also discussed, followed by a description of structure validation using the CING webserver. A classical, NOE distance-based structure determination is recommended when possible since it typically leads to more accurate structures. The protocol to follow is however more complex than that of chemical shift-based methods like CS-ROSETTA. The latter is very simple to use and has been shown to generate accurate models for small systems [7, 51]. The commands to be executed are in bold Courier font with gray shading. Information about the various programs and web pages used in this tutorial can be found in Table 1. As an example project for these tutorials we provide a target from the CASD-NMR experiment [52, 53], OR36 (pdb 2LCI). This protein is 128 amino acids long. Besides backbone chemical shift data, NOE data are also provided that can be used in the other tutorials.

3.1 NMR Data and File Formats

A common problem in NMR is the abundance of file formats to store the NMR data (such as spectral data, chemical shifts, and peaks lists) and resulting restraint files for structure calculation. This makes data exchange between different programs a tedious process. The CCPN Data Model for macromolecular NMR [54] is intended to cover all data needed for macromolecular NMR spectroscopy from the initial experimental data to the final validation. The ccpNmr FormatConverter application allows consistent conversion between a large variety of data formats *via* import to and export from CCPN, and we recommend its local installation (*see* **Note 1**) and use for this purpose. The general steps to follow for importing data files into the FormatConverter from scratch are:

1. Start the FormatConverter on the command line:

```
formatConverter
```

2. Under the *Project* menu option, click **New** and enter a name for your project; you will not be able to use the FormatConverter until you have done this.
3. Under the *Import* menu option, go to *Single files*, select the type of data you want to import, then the name of the software the data comes from. A window will pop up, click on the **Select file** button, navigate to the file, and click **Select**. Click **IMPORT** to start the importing process.
4. Depending on the type of data, you might get popups that ask you to validate information or provide additional input. Press on the **?** button in these popups for additional information.
5. After successful import, repeat the process from **step 3**. Due to the way the CCPN framework stores information, you will need to import both the sequence of your molecule and experimental information to create a complete CCPN project. At this point you will get a prompt to initiate the linkResonances process. Click **Yes** when asked, and **Yes** again to perfom this

process automatically (if possible). For more information on this process, see the links in **Note 1**.

6. Under the *Project* menu option, click **Save** or **Save as** to store your project. The data will be saved in a directory with subdirectories containing a set of XML files; all these have to stay together to remain valid for the CCPN framework.

If you created a CCPN project or downloaded a publicly available one (*see* **Note 1**) you can export data using the following general steps:

1. Under the *Project* menu option, click **Open**, navigate to the main directory containing your CCPN project in the pop-up window, and click **Open**.
2. Under the *Export* menu option, select the format/software to which you want to export the data. In the pop-up window, select the type of data you want to export to file, then select the specific data item(s) from the CCPN project. Click on the **Select export file** button, identify the file name you want to export the data to, and click **Select**. Click **Export … file** to start the export process.
3. Depending on the type of format and data, you might get popups that ask you to validate information or provide additional input. Press on the **?** button in these popups for additional information.

3.2 TALOS+: Chemical Shift-Derived Dihedral Angle Restraints

J-coupling and secondary chemical shifts can be used to define restraints on the torsion angles of the chemical bonds, typically ϕ, ψ, and χ_1 angles can be generated and included. They can be calculated applying the Karplus equation [24] to the measured J-couplings, or estimated from chemical shifts in programs such as TALOS+ [55], DANGLE [56] or CSI [57]. The chemical shift-based methods are now the most commonly used and have good accuracy; DANGLE can be run from the CCPN software, but we will describe here the more commonly used TALOS+ approach.

TALOS+ predicts ϕ and ψ backbone torsion angles from a combination of available chemical shifts (Hα, Cα, Cβ, CO, N) for a protein sequence; it is an empirical method supported by a database of existing information on chemical shifts and backbone torsion angles. Before executing TALOS+, it is essential to ensure that your chemical shifts are correctly referenced (*see* **Note 2**), and that you have the correct input file (*see* **Note 3**). We here describe the use of TALOS + from the web-based server; other options are available (local installation or the WeNMR server, *see* Subheading 3.3).

1. Connect to the TALOS+ Web server: http://spin.niddk.nih.gov/bax/nmrserver/talos.
2. Under the *Chemical Shift Input* tab, click **Browse** and select your chemical shift input file (*see* **Note 3**).

3. Under the *Prediction Options* tab, you can deselect the **Apply Offset Correction** tick mark if you are sure your chemical shifts are correctly referenced (*see* **Note 2**).
4. Under the *Submission Details* tab, enter your email address (twice) and click **Submit** to start TALOS+. The screen will change, and you will after a few minutes receive an email from the "NMR Server Agent" with details on where to retrieve the results.
5. Create a directory where you want to store the results, open the email, and save all six files to this directory (detailed information on these files is available from the TALOS+ server).
6. In order to convert the TALOS+ predictions to accurate dihedral angle restraints, they have to be manually inspected. Connect to the jRAMA+ online viewer: http://spin.niddk.nih.gov/bax/software/TALOS+/JRAMA+.

   The first time you access this page you have to explicitly state that you trust this server; click the tick mark box followed by **Run**.
7. Click on the TALOS+ image in the top left corner; a pop-up will appear. In this popup, click on the **File** menu option (top left) and select **Open Prediction Files**. In the file window that appears, navigate to the directory you created to store the TALOS+ results, and select the pred.tab file. A set of other windows will now appear; the *RCI-S2 and Secondary Structure Plot* gives an overview of the chemical shift predicted backbone dynamics (from RCI [58]) and secondary structure for your protein.
8. Go to the window that contains your protein sequence with a color-coded box for each amino acid. The color coding indicates the following:
   (a) **Green**: the prediction is "Good" for this residue and can be used to create a dihedral angle restraint.
   (b) **Yellow**: the prediction is "Ambiguous" and should be manually inspected before use.
   (c) **Red**: the prediction is "Bad" and should not be used.
   (d) **Blue**: the residue is highly "Dynamic" and a prediction is not possible.

   If you click on a box, the other windows will update to show detailed information about the prediction for this residue, and you can examine the ϕ/ψ distributions of the detected matches and override the TALOS+ decisions on which residues should be included in the prediction and which ones are outliers. You can change the prediction status of this residue selecting a box at the bottom of the main window—only residues with "Good" status will be written out the dihedral restraint file as discussed in the next point.

9. When you are satisfied that only well-predicted residues have “Good” status, you can export the dihedral restraints from the **Tools** menu option. Write out both Xplor and Cyana angle restraint files.

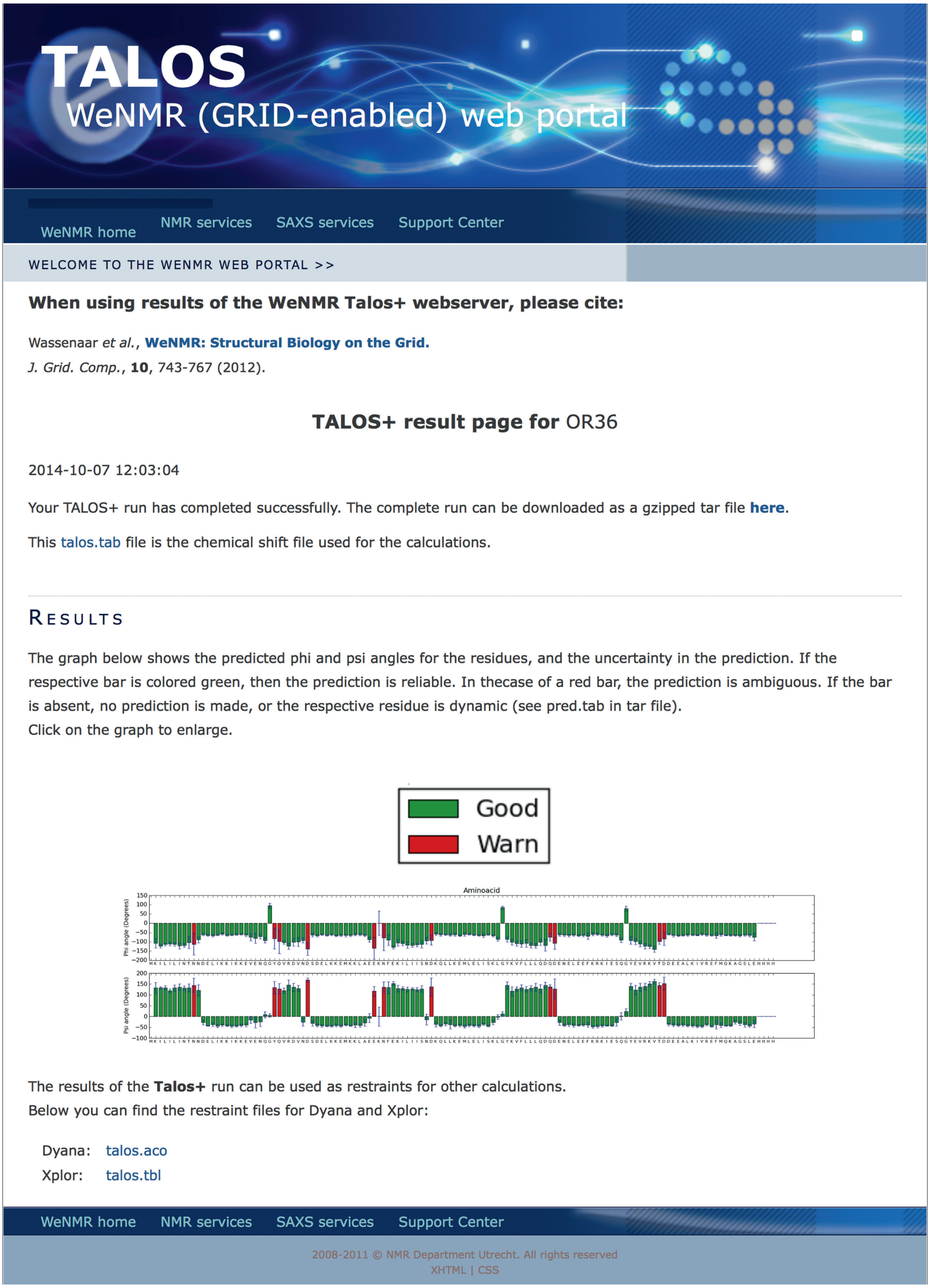
3.3 CS-ROSETTA: Chemical Shift-Based Structure Calculation

The CS-ROSETTA3 protocol [51] makes use of the original CS-ROSETTA protocol [7], using a new fragment selection method (Vernont et al. personal communication) and Rosetta 3.3, and is powered by the WeNMR Grid [59]; *see* **Note 4** for more information on the WeNMR project and how you can access its resources. This tutorial guides you through the steps for setting up a basic CS-ROSETTA run, and how to interpret the results. First create a working directory, download the supplementary data associated with this chapter from the *Springer extra* Website at http://extras.springer.com, and unpack it with the following commands:

```
mkdir OR36
cd OR36
tar xvfz OR36_blind.tar.gz
```

Before submitting your chemical shift list to CS-ROSETTA, it is recommended to check the flexibility of your protein. If there are flexible ends, we suggest you truncate them from your chemical shift list. CS-ROSETTA predicts structure, so performs best for parts of a protein that have structure. To identify flexible residues, run the TALOS+ server as described before or via the WeNMR infrastructure:

1. Connect to the WeNMR TALOS+ Web server: http://haddock.science.uu.nl/enmr/services/TALOS.
2. Give a name to your run.
3. Select as file type to submit: BMRB3.1.
4. Select as chemical shift file to submit the “`originalData/bmrb31.str`” file from the OR36 directory you created previously.
5. Click on **Submit.**
6. You will be presented with a link to the result page. The result page for this system (Fig. 2) shows the predicted phi and psi angles for the residues, and the uncertainty in the prediction. If the respective bar is colored green, then the prediction is reliable. If the bar is red, the prediction is ambiguous. If the bar is absent, no prediction is made, or the respective residue is dynamic. Several consecutive dynamic residues suggest the presence of flexible parts. TALOS+ does not predict the torsion angles for the first and last residue in the sequence. Our target in this tutorial case seems thus to be ordered (i.e. does



TALOS
WeNMR (GRID-enabled) web portal
WeNMR home
NMR services
SAXS services
Support Center
WELCOME TO THE WENMR WEB PORTAL >>

When using results of the WeNMR Talos+ webserver, please cite:

Wassenaar *et al.*, **WeNMR: Structural Biology on the Grid.**
J. Grid. Comp., **10**, 743-767 (2012).

TALOS+ result page for OR36

2014-10-07 12:03:04

Your TALOS+ run has completed successfully. The complete run can be downloaded as a gzipped tar file **here**.

This talos.tab file is the chemical shift file used for the calculations.

RESULTS

The graph below shows the predicted phi and psi angles for the residues, and the uncertainty in the prediction. If the respective bar is colored green, then the prediction is reliable. In thecase of a red bar, the prediction is ambiguous. If the bar is absent, no prediction is made, or the respective residue is dynamic (see pred.tab in tar file).
Click on the graph to enlarge.

🟩 Good
🟥 Warn
Aminoacid
Phi angle (Degrees)
150
100
50
0
−50
−100
−150
−200
Psi angle (Degrees)
200
150
100
50
0
−50
−100
M K I L I L I N T N N D E L I K K I K K E V E N Q G Y Q V R D V N D S D E L K K E M K K L A E E K N F E K I L I I S N D K Q L L K E M L E L I S K L G Y K V F L L Q D Q D E N E L E E F K R K I E S Q G Y E V R K V T D D E E A L K I V R E F M Q K A G S L E H H H H

The results of the **Talos+** run can be used as restraints for other calculations.
Below you can find the restraint files for Dyana and Xplor:

Dyana: talos.aco
Xplor: talos.tbl

2008-2011 © NMR Department Utrecht. All rights reserved
XHTML | CSS

Fig. 2 Example of TALOS output from the WeNMR server based on the chemical shifts for entry 2lci. (The server is accessible via the “NMR” ->“Chemical Shifts” menu of www.wenmr.eu)

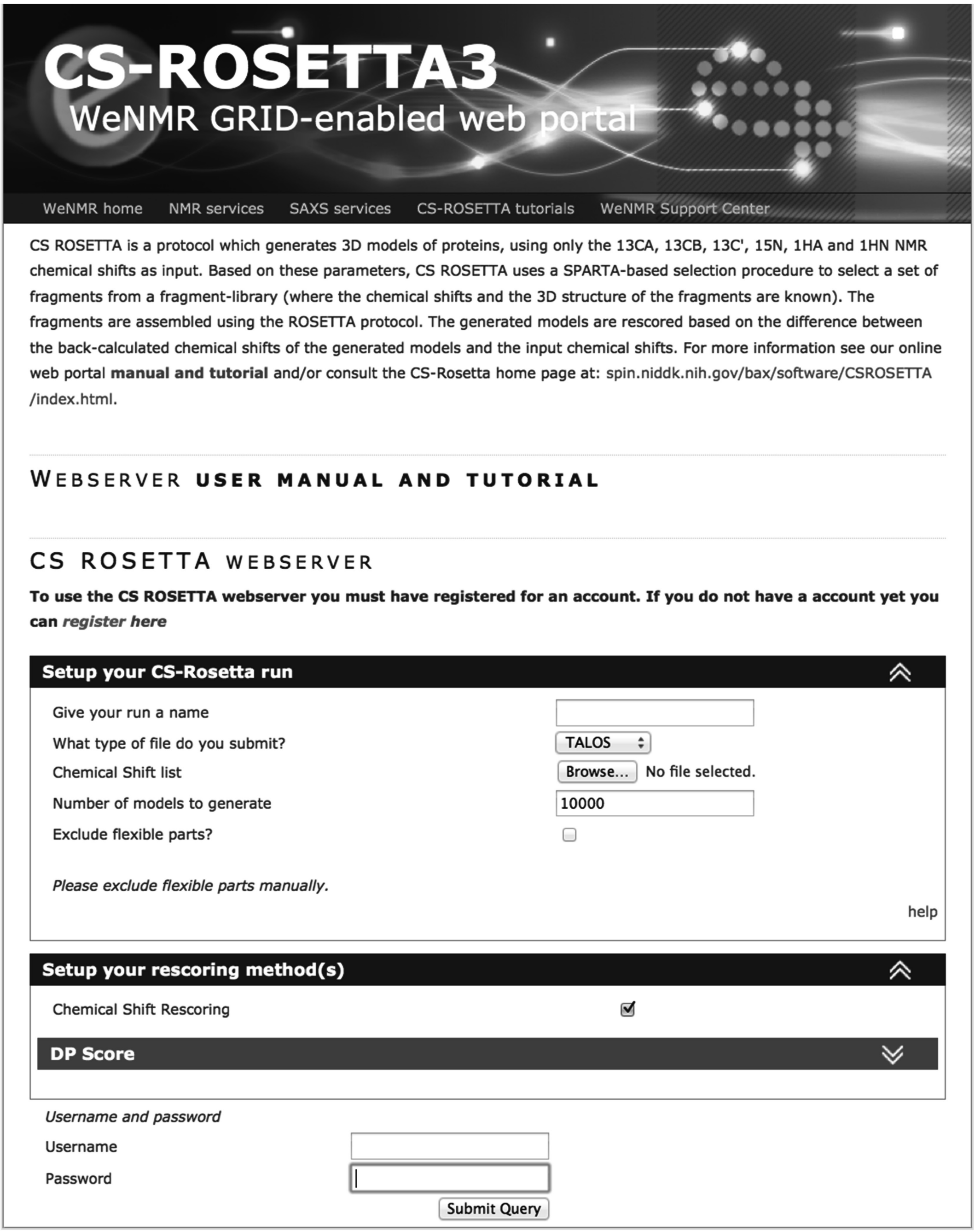
not seem to contain flexible ends). The last Histidines in the sequence are not predicted because no chemical shifts are available for those.

7. Before proceeding save the "`talos.tab`" file provided on the results page. We will use it for the CS-ROSETTA server submission. Edit this file and remove from the sequence the last six histidines since no chemical shifts are available for them. Also delete the chemical shift entries for residue 131 (one of the last histidines). It does not make sense to include them for CS-ROSETTA calculations.

Now set up and run the CS-ROSETTA calculations:

1. Connect to the WeNMR CS-Rosetta3 web portal:

   http://haddock.science.uu.nl/enmr/services/CS-ROSETTA3.
2. Figure 3 shows the web form you have to fill in. There are seven fields you can fill in:
   (a) The run name: has to be unique, has to be alpha numerical, and maximally 20 characters long.
   (b) The data format of the chemical shift list: you can supply your chemical shift list in three data formats: TALOS, BMRB2.1, and BMRB3.1.
   (c) The chemical shift list: the location of the chemical shift file on your local computer.
   (d) The number of models to generate: 10,000 is the maximum for a default account.
   (e) Option to automatically exclude flexible tails: we suggest you to do this manually as explained above.
   (f) Rescoring options: After generating your models, they have to be scored. ROSETTA3.3 does this with an all atom energy score (raw score). There are two additional rescoring algorithms: chemical shift (CS) rescoring, and DP rescoring (based on unassigned NOE peak lists) [60]. For DP rescoring different types of NOE data can be supplied (outside the scope of this tutorial).
3. To set up your tutorial run, choose the following options for the above:
   (a) Run name: OR36.
   (b) Type of file to submit: TALOS.
   (c) Chemical shift list: use the "`talos.tab`" file you just saved and edited.
   (d) Leave the number of models to their default value.



CS ROSETTA is a protocol which generates 3D models of proteins, using only the 13CA, 13CB, 13C', 15N, 1HA and 1HN NMR chemical shifts as input. Based on these parameters, CS ROSETTA uses a SPARTA-based selection procedure to select a set of fragments from a fragment-library (where the chemical shifts and the 3D structure of the fragments are known). The fragments are assembled using the ROSETTA protocol. The generated models are rescored based on the difference between the back-calculated chemical shifts of the generated models and the input chemical shifts. For more information see our online web portal **manual and tutorial** and/or consult the CS-Rosetta home page at: spin.niddk.nih.gov/bax/software/CSROSETTA /index.html.

WEBSERVER **USER MANUAL AND TUTORIAL**

CS ROSETTA WEBSERVER

To use the CS ROSETTA webserver you must have registered for an account. If you do not have a account yet you can *register here*

Setup your CS-Rosetta run

Give your run a name	
What type of file do you submit?	TALOS
Chemical Shift list	Browse... No file selected.
Number of models to generate	10000
Exclude flexible parts?	☐

Please exclude flexible parts manually.

help

Setup your rescoring method(s)

Chemical Shift Rescoring ☑

DP Score

Username and password

Username

Password

Submit Query

Fig. 3 CS-ROSETTA web form on the WeNMR server. (The server is accessible via the "NMR" -> "Structure Calculation" menu of www.wenmr.eu)

(e) Leave the remove flexible parts box unticked.

(f) Chemical shift rescoring is selected by default; leave this.

(g) Enter your username and password and submit the calculations.

After a successful submission you get a link to your personal result page (which is also mailed to you), on which you can track your run, and find the results when the run will have completed (which can take a few days depending on the system size and load on the server). The result page present the top five models based on various scoring scheme and also indicates the reliability of the predictions based on convergence and energetics criteria.

You can directly validate the CS-ROSETTA calculation results using iCING (*see* Subheading 3.6).

3.4 CYANA Structure Calculation from NOESY Peak Lists

A cross-peak in a NOESY spectrum indicates spatial proximity between two atoms; under idealized asumptions the intensity of the NOE peak (I) is proportional to the interatomic distance (r) as $I \sim 1/r^6$. Each NOESY peak can therefore be converted into an interatomic distance after determination of the proportionality constant (a process called "distance calibration"). Because the intensity–distance relationship is only approximate in practice, due to multiple complicating effects such as dynamics and the presence of many interacting spins, and as result of limitations in most structure calculation protocols, a distance range is usually assumed when converting the NOE peak intensity into a distance restraint. This range tends to be between 1.8 and 6.0 angstroms, with various ranges depending on the intensity of the NOE peaks.

An NOE peak can only be used as a distance restraint if it can be assigned to atoms in the molecule based on their chemical shift values: this step is of crucial importance, as the structure calculation process may not be able to deal with too many wrong or highly ambiguous (when many atoms with similar chemical shift values can be assigned to an NOE) assignments. The manual assignment of NOEs is an intensive and time-consuming job: many protocols are available to do this automatically starting from spectra [61] or peak lists [12, 62, 63]. We will here describe the procedure starting from peak lists as implemented in CYANA 2.1: note that CYANA requires a license, but because of its speed and ease of implementation it is in our view the most convenient software to use at this stage.

1. Create a directory for the structure calculation with CYANA. The following files have to be present in this directory:
   (a) A sequence file with the protein amino acid sequence in XEASY/CYANA format (*see* **Note 5**). We will refer to this file as `protein.seq`, and assume it has 100 residues.
   (b) A chemical shift file in XEASY/CYANA format (*see* **Note 5**). We will refer to this file as `shifts.prot`.
   (c) A dihedral angle file (e.g. from TALOS+ in Subheading 3.2). We will refer to this file as `dihedrals.aco`.

(d) One or more peak lists from NOESY spectra in XEASY/ CYANA format (*see* **Note 5**). We will refer to these files as `peaks1.xpk`, `peaks2.xpk`, … .

(e) A CYANA initialization file (`init.cya`) containing the lines:

```
rmsdrange:=1..100
cyanalib
read seq protein.seq
```

Note that you can adjust the RMSD range to only include well-defined regions of the protein structure (if this information is known).

(f) A CYANA file with information for the structure calculation run (`AUTO.cya`) containing the lines:

```
peaks       := peaks1.xpk,peaks2.xpk # NOESY peak lists
prot        := shifts.prot           # Chemical shifts
constraints := dihedrals.aco         # Restraints
tolerance   := 0.030,0.040,0.25      # Tolerances for assignment
structures  := 100,20                # Initial/final structures
steps       := 10000                 # Calculation steps
randomseed  := 34983434              # Calculation seed

noeassign peaks=$peaks prot=$prot autoaco
```

The tolerances determine which atoms (based on their individual chemical shift values) are assigned to the NOESY peaks; wider tolerances will result in CYANA detecting more possible assignments for the NOESY peaks. If the tolerances are too narrow, less assignment possibilities will be found, but correct assignments might be missed. The above protocol assumes that the NOESY peaks are not yet assigned; additional protocols are available from the CYANA Web site and other resources.

2. Start CYANA with the command:

```
cyana
```

The program should automatically pick up all the relevant files and will assign the peak lists and calculate structures based on its internal procedures.

3. If CYANA successfully finished, you will see a set of files with *final* in the name that contain the following information:

(a) `final.pdb`: The atom coordinates of all models in the final structure ensemble.

(b) `final.upl`: An overview file containing information on the structure calculation. Particularly relevant here is the list of violated constraints: if they are violated in most structures (full sequence of+or * signs) you should go back to the original peak lists with the program you were using for spectrum analysis and check whether the NOE peak and/or assignments make sense.

(c) final.aco, final.upl: Constraint files used in final calculation round.

(d) protein-final.prot**:** An XEASY file containing the final chemical shift assignments.

You should also check for "*** WARNING" messages in the AUTO.out file to make sure there were no problems during the calculations. Finally, the files ending in –cycle7.peaks and –cycle7-ref.peaks contain the original peak lists with information adapted based on final information from the structure calculation.

4. You can directly validate the CYANA calculation results using iCING (*see* Subheading 3.6). As CYANA uses a limited force field, we also recommend to water-refine the final structures with the protocol from Subheading 3.5 (*see* **Notes 5** and 7 on how to generate the CNS input via CCPN).

3.5 NMR Structure Refinement with CNS/RECOORD

For the structure calculation part we are going to describe the use of the program CNS [13, 14] with a simulated annealing protocol derived from ARIA [64] followed by refinement in explicit solvent [10]. All the scripts mentioned in this section can be downloaded from the RECOORD [11] Webpage (*see* Table 1).

1. **Download**: Create a folder where you will run the calculations, download there the tar file containing the RECOORD scripts and decompress it:

```
mkdir struct-calc
cd struct-calc/
wget http://www.ebi.ac.uk/pdbe-
apps/nmr/data/recoord/RECOORDscripts-cns1.3.tgz
tar xzfv RECOORDscripts-cns1.3.tgz
```

In case the wget command does not work, use a Web browser to download the scripts manually from the RECOORD webpage (*see* Table 1). This tutorial uses CNS version 1.3.

2. **Initialise**: Before starting the calculations, you need to set up your current path for the scripts to work. In order to do this, you need to edit the file "`changeScriptsDir.sh`" located in RECOORDscripts-cns1.3 and change the directory path for `newDir` in line 8 with your current path and execute it:

```
./changeScriptsDir.sh
```

3. **Get data**: Make sure you have a working version of CNS set up; the last step is then to create a working directory and assigning a project name for the protein you are working on. This project name will be used to generate the file names at the different stages of the protocol. We will use as example the OR36 data for the PDB 2LCI structure with the corresponding NMR restraints available for this entry from the BioMagResBank (BMRB) [65]. First download and rename the PDB structure file:

```
mkdir 2lci
cd 2lci
wget http://www.ebi.ac.uk/pdbe-srv/view/files/2lci.ent.gz
gunzip pdb2lci.ent.gz
mv pdb2lci.ent 2lci.pdb
```

Then get the NMR restraints from this link:

```
http://www.bmrb.wisc.edu/servlets/MRGridServlet?pdb_id=2l
ci&min_items=0&block_text_type=3-converted-DOCR
```

In the result table, click on the number in the "distance" row under the "XPLOR/CNS" column, then click on the link in the "mrblock_id" column. Copy and paste these restraints in a text file called unambig.tbl in the 2lci/ directory (*see* **Note 6**). Alternatively you can export CNS restraints from CCPN projects (*see* **Note 7**) or use the data for OR36 directly from the "`restraints/cns/`" directory of the example project.

4. **Generation of molecular topology files**: We can generate the molecular topology either from the primary sequence or from a PDB coordinate file, depending on availability (*see* **Note 8**). We will use here the downloaded PDB file:

```
../RECOORDscripts-cns1.3/generate.sh 2lci.pdb
```

A topology file called `2lci_cns.mtf` will be generated (note that this name is based on the name of the input file and will be different when you try other examples). You should check the `ERRORS_generate` file created inside the `2lci/` folder: in this particular case, you can see that the script reported many nomenclature errors which can be ignored at this stage.

A new pdb file called 2lci_cns.pdb is also generated with the proper CNS nomenclature. You can display this structure in your favorite molecule viewer to verify it is correct.

5. **Generation of extended starting structure:** The next step is the generation of an extended starting conformation which will be used as input in the simulated annealing protocol:

```
../RECOORDscripts-cns1.3/generate_extended.sh 2lci_cns.mtf
```

The extended structure is in the file 2lci_cns_extended.pdb. You should check the ERRORS_generate_extended file for errors, and again check the generated file in your favourite molecule viewer.

6. **Simulated annealing stage**: For the structure calculation itself, we can use three different types of restraints (if available): unambig.tbl (NOE distance restraints), hbonds.tbl (hydrogen bond restraints) and dihedrals.tbl (dihedral angle restraints). The script "annealing.sh" will generate a CNS parameter file (run.cns) with all details and specifications for the structure calculation protocol, and will start the calculation. This script should be run from a higher level than the previous two:

```
cd ..
RECOORDscripts-cns1.3/annealing.sh 2lci
```

Individual job files will be generated and executed for each model you want to calculate. By default two models will be generated in the created str/ folder, with name similar to 2lci_cns_[1-2].pdb. The CNS input and output files can be found in the directory cnsRef/, together with possible error files (*see* **Note 9**). The header of every PDB file generated contains information about violations and energy values.

7. **Water refinement stage:** Once the simulated annealing phase is finished and all resulting structures have been written into the str/ directory, we can proceed to water refinement:

```
RECOORDscripts-cns1.3/re_h2o.sh 2lci
```

In the str/ directory, a new directory called wt/ will be created, the best energy structures will be copied there and subsequently refined (*see* **Note 10**).

3.6 Structure Validation and Quality Assessment

The 3D protein structures are the end result of a highly complex procedure involving interpretation of experimental data and transformation of this data to in silico atomic coordinates. It is therefore essential that these structures are well-validated and analyzed to ensure their correctness before they can be used as models that represent the conformation of the protein in solution.

This validation and analysis happens on two levels: comparison of the structures to original experimental data such as restraints and chemical shifts, and assessment of the physical quality of the in silico molecule description in terms of known physicochemical properties, such as atom bonds, angles, dihedrals, packing, hydrogen-bond, and electrostatics. Many programs are available to do parts of these validations (*see* Vuister et al. [49] for a recent overview); we will here focus on CING [15], a recently developed package that assembles the results of different validation programs and presents them in a joint, interactive, Web-2.0-based validation report. As an equivalent alternative, PSVS can be used [66].

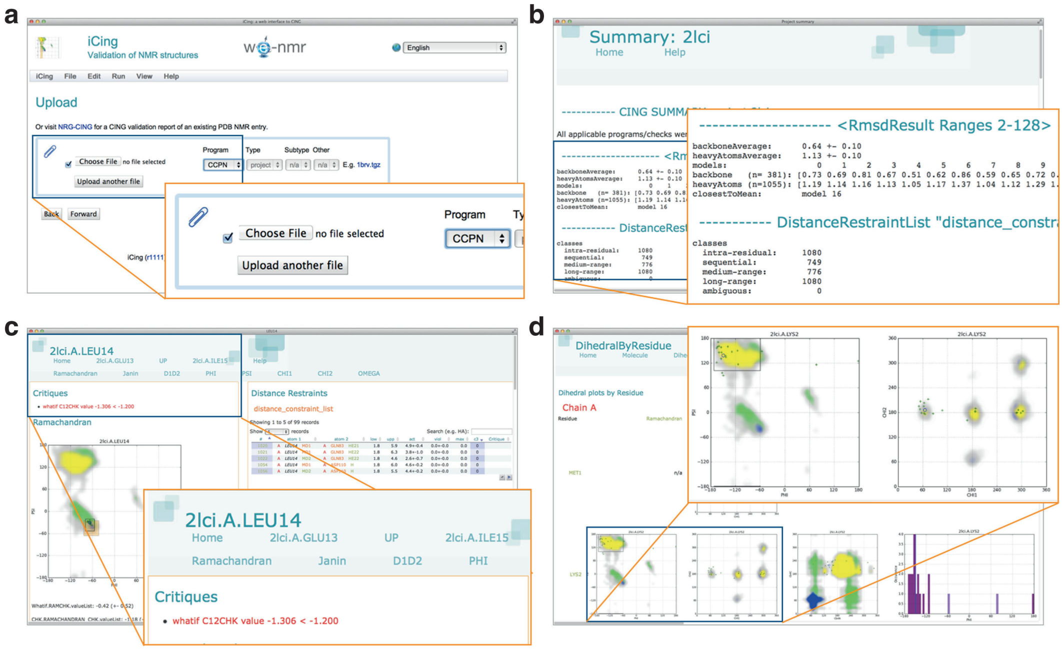
The CING package is free to download and install; however, its large number of dependencies on external software makes this a nontrivial exercise. A virtual linux image with all public-domain software installed can be requested from the authors, requiring only the installation of the external licensed packages. Here, we will use the recommended iCing web server to validate the structures.

The iCing server can accept structure coordinates in the form of PDB files. However, it is highly advisable to also supply the server with the experimental data. The prefered input format of the iCing server is a self-contained CCPN project with the structure, restraints, chemical shifts, peaks, etc, compressed as a .tgz file. Such a file can be generated and submitted directly from the ccpNmr Analysis program or can be generated using the FormatConverter (*see* **Notes 1, 5,** and 7).

1. **Access the iCING server**: To upload the CCPN project data, first create an archive file of your project:

```
tar cvfz myCcpnProjectArchive.tgz myCcpnProjectDirectory/
```

Point your browser to http://nmr.le.ac.uk and select the iCing option. In the upload panel (Fig. 4a) use the "Choose File" button, select your ccpn.tgz file, which will then be uploaded. For "Program" we select CCPN (the default). Note that iCing also natively accepts PDB formatted coordinate files, multiple files in Cyana format as a .tgz file (*see* **Note 11**) or a CING project as a .tgz file. Selecting "Forward" allows one to proceed to the next panel with "Criteria". Leave these


a
iCing
Validation of NMR structures
we-nmr
English
iCing File Edit Run View Help
Upload
Or visit NRG-CING for a CING validation report of an existing PDB NMR entry.
Program Type Subtype Other
Choose File no file selected
CCPN project n/a n/a E.g. 1brv.tgz
Upload another file
Back Forward
iCing (r1111)
b
Summary: 2lci
Home Help
c
2lci.A.LEU14
Home 2lci.A.GLU13 UP 2lci.A.ILE15 Help
Ramachandran Janin D1D2 PHI PSI CHI1 CHI2 OMEGA
Critiques
Distance Restraints
distance_constraint_list
Ramachandran
d
DihedralByResidue
Home Molecule
Dihedral plots by Residue
Chain A
2lci.A.LYS2

```
---------- CING SUMMA
All applicable programs/checks wer

------------------- <RmsdResult Ranges 2-128>
backboneAverage:        0.64 +- 0.10
heavyAtomsAverage:      1.13 +- 0.10
models:                    0    1    2    3    4    5    6    7    8    9
backbone    (n= 381): [0.73 0.69 0.81 0.67 0.51 0.62 0.86 0.59 0.65 0.72 0.
heavyAtoms  (n=1055): [1.19 1.14 1.16 1.13 1.05 1.17 1.37 1.04 1.12 1.29 1.
closestToMean:          model 16

---------- DistanceRestraintList "distance_constr
classes
  intra-residual:      1080
  sequential:           749
  medium-range:         776
  long-range:          1080
  ambiguous:              0
```

• whatif C12CHK value -1.306 < -1.200

Fig. 4 iCing server and results. (**a**) iCing server upload page. Selection area is indicated by the *red box*. (**b**) Summary page for entry 2lci. (**c**) Residue-specific page for Leu14 of entry 2lci. The residue has a red ROG score as result of poor sidechain χ1-χ2 dihedral angle conformation (Janin plot not visible), Many related elements (previous and next residues, restraints, chemical shifts, etc.) can directly be accessed through the links on this page. (**d**) DihedralByResidue plots. A quick overview is obtained by scrolling down. Individual residue pages can be accessed through the links

set to their default values. Selecting "`Forward`" again, yields the "`Options`" panel that allows for selection of residues, or models in the ensemble. You can use this to set a range of residues to consider if you want to override the CING routines for determining this automatically. Generally, this is not required. Pressing "`Forward`" once more yield the Run-panel.

2. **Running CING**: Press "`Submit`" to start the CING validation analysis. The Output panel will appear, which can be manually updated to display the progress in the analysis. A typical CING validation analysis will take ~10–15 min. This is caused by the many programs that are run and the extensive graphical output that will be generated. When the program is finished, a link is presented to the on-line webpages with the results. The whole CING project file that includes all the data, the results from the individual programs and the all the webpages can also be downloaded. The results are stored anonymously on the iCing server and are automatically removed after 2 days.

3. **Using the CING results**: CING generates easily accessible, comprehensive, interactive HTML/Javascript-based validation

reports and directs the NMR spectroscopist to areas of the structure that have problems. Hyperlinks connect all elements at the different levels; e.g. the full molecule, chains, residues, atoms, as well as restraints, peaks, and chemical shifts. Lists are interactive entities that can be reordered and filtered by query. CING uses at all data levels a simple Red–Orange–Green (ROG) scoring, which depends upon the combined analysis of all results and allows CING to direct the user to important issues:

(a) **Red**: Potentially serious issues with the structures.

(b) **Orange**: Potential issues with the structures.

(c) **Green**: No issues detected.

In addition to this classification mechanism, CING displays the validation results in direct relation to the experimental data.

4. **Structure level results**: The link to the on-line webpages with the results first presents an overview of the various sections of the CING report. A link to the *summary* is best examined first (Fig. 4b). Here, an overview is given of the rmsd values within the ensemble, the overall CING [15], WHATIF [67] and PROCHECK_NMR [68] scores, as well as the results of the restraints analysis. CING also analyses, based upon circular variance criteria of the backbone dihedral angles ϕ and ψ, which residues should be considered well-defined (*see* **Note 12**); only these regions are used for the superposition of the structures and the related rmsd values, as are the PROCHECK_NMR scores and one of the two overal CING ROG scores; the other pertains to the full molecule. An ensemble of structures that has low green (<~20 %) and high red (>~50 %) overall ROG score should be labeled as highly troublesome.

5. **Residue level results**: Within the CING philosophy, a large emphasis is placed upon the validation of the individual residues. NMR assignment strategies are almost exclusively residue-based, NMR related parameters are residue dependent and the local nature of the NMR-derived restraints also correlates well with a residue-based approach. Structural properties can also be conveniently summarized at the residue level, and hence all residue-specifc elements are specified on the residue pages. Residue-specific Ramachandran plots (Fig. 4c) are generated for each residue with all the individual conformers of the ensemble marked and ϕ, ψ restraints, if present, indicated. Inspection of the sidechain χ1–χ2 dihedral angle distributions of the conformers in the ensemble is possible by means of the Janin plot, as also automatically generated by CING for each residue. The D1D2 plots display the conformation of the residue relative to the preceding and following residue in the chain. A residue page also contains all the experimental restraint

data involving the featured residue, as well as the "`critiques`" defining its ROG scores. Thus, a residue page presents a comprehensive account of all relevant information pertaining to a specific residue. Hyperlinks connect the page to all other relevant pages; e.g. the previous and next residue in the chain, but also all residues connected through restraints or all its atoms and their chemical shifts. The analysis of the conformation of residues and identification of potential problems can also conveniently be done using the "`Dihedral plots per residue`" page (Fig. 4d), which displays the relevant Ramachandran, Janin and D1D2 plots of all residues sequentially, in one scrollable interface from which the relevant residue can also be selected.

6. **Other information**: It is useful to display the residue-specific ROG scores mapped onto a structure of the molecule. For this, macros for the structure visualization programs JMOL [69], YASARA [45], PyMol (The PyMOL Molecular Graphics System, Schrödinger, LLC.) and MOLMOL [70] are provided. They can be accessed by following the "`Programs flat->Macros`" link from the home page. Closely clustered red/orange residues in the structure are highly suspect and warrant further investigation.

   The CING results are based, in part, upon the results of the programs PROCHECK_NMR [68] and WHATIF [67]. Direct links to the output of these programs are also provided.

4 Notes

1. To use the FormatConverter, install the latest stable version of the CCPN software, which can be downloaded from http://www.ccpn.ac.uk/downloads/stable. A less versatile but easier to use alternative is the FormatConverter web version at http://haddock.chem.uu.nl/enmr/format-converter.html. Further information on the FormatConverter is available from http://www.ccpn.ac.uk/software/fcfolder, as well as a detailed tutorial at http://www.ccpn.ac.uk/software/tutorials/intro. Examples of full CCPN projects that can be used for structure calculation are available from the RECOORD project at http://www.ebi.ac.uk/pdbe-apps/nmr/recoord/ by clicking the *CCPN projects* link near the bottom of the page.
2. Chemical shifts are calculated from absolute frequencies relative to a reference frequency; this way the positions of NMR resonances can be expressed independently of the magnetic field strength. If this reference frequency is not correctly set all chemical shift values calculated with it will be equally offset from their true value. There are a host of programs available to check if

chemical shifts are correctly referenced, and to suggest a correction to obtain their true value ("re-referencing" of the chemical shifts): AVS [71], LACS [72, 73], SPARTA+ [23], PANAV [74], CheckShift [75], ShiftX2 [19], and VASCO [76].

3. Load a CCPN project with the FormatConverter or import a sequence and chemical shift list. Go to the *Export* menu and select *TALOS*. Open the "*project export menu*", click on **Select export file** and define the file you want to write to. The chemical shift data to export can be selected next to *Select shift list to export:* (in the example OR36 project this will be "`1:bmrb21.str`", indicating a shift list with CCPN serial number `1` and name `bmrb21.str`). Click **Export project file**, then **OK** in the popup window (you can here change the sequence numbering should you wish to do so), then **OK** again if the export finished successfully. The exported TALOS file should contain sequence as well as chemical shift information and can be uploaded to the TALOS+ server. The procedure for CS-ROSETTA is the same, although in this case it is useful to first run TALOS+ and then, in the input file to be uploaded to the CS-ROSETTA server, first manually remove N- and C-terminal regions from the sequence that have no chemical shifts, as well as residues that are labeled as "Dynamic" by TALOS.

4. WeNMR [59] provides an extensive infrastructure that allows users to run up-to-date and computationally demanding software online. In order to use it, you will have to obtain a personal X509 certificate and register with the WeNMR project and virtual organization. Instructions for this can be found at the following site: http://www.wenmr.eu/wenmr/access/registration. Relevant software for this protocol available on WeNMR is indicated in Table 1.

5. (a) Export of sequence, peak list and chemical shift data for CYANA. First load a CCPN project with these data with the FormatConverter (*e.g.* the example OR36 project). (i) Sequence export: Go to the *Export* menu and select *Cyana*. Open the "*sequence export menu*" and select the molecule for which you want to export the sequence next to *Select chains to export:* (in the example OR36 project this will be "`Molecularsystem:'A'`", indicating a molecule with chain code "`A`" in the molecular system `Molecularsystem).` Click on **Select export file** and define the file you want to write to, then select the CYANA version next to *Cyana version:* Click **Export sequence file**, then **OK** in the popup window (you can here change the sequence numbering should you wish to do so), then **OK** again if the export finished successfully. (ii) Chemical shift list export: Go to the *Export* menu and select *XEasy*. Open the "*shift export menu*" and select the chemical shift list you want to export next to *Select shift list to*

export: (in the example OR36 project this is "`1:bmrb21.str`"). Click on **Select export file** and define the file you want to write to, then **select the button** next to *Use CYANA2.1 atom names:* if you are using the CYANA 2.1 version or higher. Click **Export shifts file**, then **OK** in the popup window, and **OK** again if the export finished successfully. (iii) Peak list export: Go to the *Export* menu and select *XEasy*. Open the "*peaks export menu*" and select the peak list you want to export next to *Select peak list to export:* (in the example OR36 project there will be three peak lists, ending in "`nnoe_raw`", "`alinoe_raw`" in "`aronoe_raw`"; the text in front of this indicates the CCPN experiment and processing information). Click on **Select export file** and define the file you want to write to, then **select the button** next to *Write as CYANA format:* Click **Export peaks file**, then **OK** in the popup window. In the next window you can define the order of the peak list dimensions in the output file, press **OK** when this is fine. Finally, press **OK** again if the export finished successfully. Note that for reasons particular to the XEASY format assignments in your peak list can only be written out if you first exported a chemical shift list.

(b) Import of coordinates, distance and dihedral restraint lists from CYANA. Create a new CCPN project or load an existing one in the FormatConverter. (i) Coordinate import: Go to the *Import* menu, *Single files*, *Coordinates* and select *PseudoPdb* (note that selecting *Pdb* will only import official files from the PDB). Click on **Select files**, select the file (for OR36, try "`restraints/coordinates.pdb`") and click **Select**. Click **IMPORT**. Then two situations are possible. (1) If your sequence was already loaded into the CCPN previously, click **OK** in the popup window after selecting the correct CCPN molecular system, **OK** again after successful import. (2) If you have no sequence in your project, you will first have to create one matching the molecule from the coordinates. In the first popup window you can change the sequence information and/or molecule name. Click **OK** when done, in the next popup give a name to your molecular system and press **OK**, then click **No** in the next popup unless you have a homomultimer. You might then get additional popups asking you to disambiguate atom names from the coordinate data. Finally click **OK** again after successful import (ii) Distance restraints: Go to the *Import* menu, *Single files*, *Distance constraints* and select *Cyana*. Click on **Select file**, select the file (for OR36, try "`restraints/cyana/distances.upl`") and click **Select**. You can also select the matching lower distance limits "`distances.lol`" file at **Select file (optional)**. Select the correct CYANA version (use 2.1 for the most recent versions), and click **IMPORT**. In the next popup you will be able to select the name of an existing CCPN structure generation, or create a new one. In the next popup give your distance restraint list a name; another popup will appear asking you to read in a lower distance limit file in case you haven't specified it. Click **OK** again after successful import. Finally, click **No** for the popup asking you to run linkResonances—we will do this after the next step. (iii) Dihedral restraints: Go to the *Import* menu, *Single files*, *Dihedral constraints* and select *Cyana*. Click on **Select file**, select the file (for OR36, try "`restraints/cyana/dihedrals.aco`") and click **Select**. Select the correct CYANA version (use 2.1 for the most recent versions), and click **IMPORT**. In the next popup you will be able to select the name of an existing CCPN structure generation, or create a new one. In the next popup give your dihedral restraint list a name; click **OK** again after successful import. When you are done importing all connected restraint files, click **Yes** to the linkResonances question, then **Yes** again (*see* Subheading 3.1, **step 5**).

6. If dihedrals or any other types of restraints are available, they can be obtained in a similar way. The names assigned will be dihedrals.tbl and hbonds.tbl.

7. Export of distance and dihedral restraint lists as well as coordinates (PDB-like files) for CNS/XPLOR. First load a CCPN project with these data with the FormatConverter (e.g. the example OR36_withRestraints project). (i) Distance restraints export: Go to the *Export* menu and select *Cns*. Open the "*distanceConstraints export menu*" and select the distance constraint list for which you want to export the sequence next to *Select distance constraint list to export:* (in the example OR36 project this will be "`1:1:distance constraint list`", indicating a distance constraint list with serial number "`1`" in the structure generation with serial number "`1`"). Click on **Select export file** and define the file you want to write to. Click **Export distanceConstraints file**, then **OK** in the popup window (you can here change the chain codes and sequence numbering should you wish to do so), then **OK** again if the export finished successfully. (ii) Dihedral restraints export: This is the same procedure as for distance restraints except use the *dihedralConstraints* menu, and in the example OR36 project select "`1:2:dihedral constraint list`"`).` (iii) Coordinates export: Open the "*coordinates export menu*" and select the structure(s) you want to export next to *Select structures to export:* in the example OR36 project there will be one structure, "`MolecularSystem:1:model_1`", indicating a set of coordinates for "`model_1`" relating to molecular system "`MolecularSystem`" in structure ensemble with serial "`1`". Click on **Select export file** and define the file you want to write to, then click **Export coordinates file**.

8. This only works if a PDB coordinates file is available. Otherwise use generate_seq.inp and generate_template.inp from CNS to create such a PDB.
9. Once everything is set up in a proper way, you can edit the script and change the protocol parameters where necessary. You can for example change the number of models to generate. It is set to 2 by default, but more common numbers would be 100 or 200. For more complex systems, you can switch to a longer annealing protocol, by doubling the number of steps to be carried out. Depending if you are going to use a cluster or your own computer, you should change the submit command. Remember also to change the sleep time between submitting jobs, especially if you are not using a cluster and you do not want to have 100 jobs running in your computer at the same time! In such a case choose a sleep time that matches the time needed for one structure calculation.
10. You should also edit this script and change the number of structures to refine since by default it is set to only 1. Increase this number to 25 models. They will be assigned names like `2lci_cns_w_[1-25].pdb`. The CNS input and output will be directed to the directory `cnsWtRef/`.
11. Preparing Cyana input to iCing. The Cyana `myMolecule.cyana.tgz` of `myMolecule` should contain `myMolecule.seq` (the cyana sequence file), `myMolecule.prot` (the corresponding cyana prot file), `myMolecule.pdb` (the Cyana generated structure), `myMolecule.aco` (optional dihedral restraints), `myMolecule.lol` (optional lower bounds), `myMolecul.upl` (optional upper bounds).
12. The precision of a structure can be estimated by measuring the conformational variance over an ensemble of models. Usually, this variance has been expressed as the positional root-mean-square deviation (rmsd) of the individual models from the mean structure. This parameter is useful for estimating the precision of the calculation, but does not report on the accuracy. The later can only be calculated if a standard reference is available.

References

1. Wuthrich K (1986) Nmr of proteins and nucleic acids. Wiley, New York, NY
2. Neuhaus D, Williamson MP (2000) The nuclear overhauser effect in structural and conformational analysis. Wiley, New York, NY
3. Altona C (1996) Vicinal coupling constants and conformation of biomolecules. In Harris DMG, a K R (eds) Encyclopedia of nuclear magnetic resonance. Wiley, London. pp 4909–4922
4. Bax A, Kontaxis G, Tjandra N (2001) Dipolar couplings in macromolecular structure determination. Methods Enzymol 339:127–174
5. Bertini I, Luchinat C, Parigi G (2012) Towards mechanistic systems biology. Wiley-VCH Verlag GmbH, Weinheim, Germany. pp 154–171
6. Cavalli A, Salvatella X, Dobson CM, Vendruscolo M (2007) Protein structure determination from NMR chemical shifts. Proc Natl Acad Sci U S A 104:9615–9620

7. Shen Y et al (2008) Consistent blind protein structure generation from NMR chemical shift data. Proc Natl Acad Sci U S A 105:4685–4690
8. Wishart DS et al (2008) CS23D: a web server for rapid protein structure generation using NMR chemical shifts and sequence data. Nucleic Acids Res 36:W496–W502
9. Güntert P (1998) Structure calculation of biological macromolecules from NMR data. Q Rev Biophys 31:145–237
10. Linge JP, Williams MA, Spronk CAEM, Bonvin AMJJ, Nilges M (2003) Refinement of protein structures in explicit solvent. Proteins 50:496–506
11. Nederveen AJ et al (2005) RECOORD: a recalculated coordinate database of 500+ proteins from the PDB using restraints from the BioMagResBank. Proteins 59:662–672
12. Güntert P (2004) Automated NMR structure calculation with CYANA. Methods Mol Biol 278:353–378
13. Brünger AT et al (1998) Crystallography & NMR system: a new software suite for macromolecular structure determination. Acta Crystallogr D Biol Crystallogr 54:905–921
14. Brunger AT (2007) Version 1.2 of the Crystallography and NMR system. Nat Protoc 2:2728–2733
15. Doreleijers JF et al (2012) CING: an integrated residue-based structure validation program suite. J Biomol NMR 54:267–283
16. Luginbühl P, Szyperski T, Wüthrich K (1995) Statistical basis for the use of ^{13}cα chemical shifts in protein structure determination. J Magn Res 109:92
17. Kuszewski J, Qin J, Gronenborn AM, Clore GM (1995) The impact of direct refinement against 13C alpha and 13C beta chemical shifts on protein structure determination by NMR. J Magn Reson B 106:92–96
18. Neal S, Nip AM, Zhang H, Wishart DS (2003) Rapid and accurate calculation of protein 1H, 13C and 15N chemical shifts. J Biomol NMR 26:215–240
19. Han B, Liu Y, Ginzinger SW, Wishart DS (2011) SHIFTX2: significantly improved protein chemical shift prediction. J Biomol NMR 50:43–57
20. Xu XP, Case DA (2001) Automated prediction of 15N, 13Calpha, 13Cbeta and 13C' chemical shifts in proteins using a density functional database. J Biomol NMR 21:321–333
21. Williamson MP, Kikuchi J, Asakura T (1995) Application of 1H NMR chemical shifts to measure the quality of protein structures. J Mol Biol 247:541–546
22. Meiler J (2003) PROSHIFT: protein chemical shift prediction using artificial neural networks. J Biomol NMR 26:25–37
23. Shen Y, Bax A (2010) SPARTA+: a modest improvement in empirical NMR chemical shift prediction by means of an artificial neural network. J Biomol NMR 48:13–22
24. Karplus M (1963) Vicinal proton coupling in nuclear magnetic resonance. J Am Chem Soc 85:2870–2871
25. Kim Y, Prestegard JH (1990) Refinement of the NMR structures for acyl carrier protein with scalar coupling data. Proteins 8:377–385
26. Torda AE, Brunne RM, Huber T, Kessler H, Van Gunsteren WF (1993) Structure refinement using time-averaged J-coupling constant restraints. J Biomol NMR 3:55–66
27. Wagner G, Wüthrich K (1982) Amide protein exchange and surface conformation of the basic pancreatic trypsin inhibitor in solution. Studies with two-dimensional nuclear magnetic resonance. J Mol Biol 160:343–361
28. Pervushin K et al (1998) NMR scalar couplings across Watson-Crick base pair hydrogen bonds in DNA observed by transverse relaxation-optimized spectroscopy. Proc Natl Acad Sci U S A 95:14147–14151
29. Cordier F, Rogowski M, Grzesiek S, Bax A (1999) Observation of through-hydrogen-bond 2hJHC' in a perdeuterated protein. J Magnet Res 140:510–512
30. Bonvin AMJJ, Houben K, Guenneugues M, Kaptein R, Boelens R (2001) Rapid protein fold determination using secondary chemical shifts and cross-hydrogen bond 15N-13C“ scalar couplings (3hbJNC”). J Biomol NMR 21:221–233
31. Bax A (2003) Weak alignment offers new NMR opportunities to study protein structure and dynamics. Protein Sci 12:1–16
32. Bax A, Grishaev A (2005) Weak alignment NMR: a hawk-eyed view of biomolecular structure. Curr Opin Struct Biol 15:563–570
33. Prestegard JH, Bougault CM, Kishore AI (2004) Residual dipolar couplings in structure determination of biomolecules. Chem Rev 104:3519–3540
34. Tjandra N, Omichinski JG, Gronenborn AM, Clore GM, Bax A (1997) Use of dipolar 1H-15N and 1H-13C couplings in the structure determination of magnetically oriented macromolecules in solution. Nat Struct Biol 4:732–738
35. Fushman D, Varadan R, Assfalg M (2004) Determining domain orientation in macromolecules by using spin-relaxation and residual

dipolar coupling measurements. Prog Nucl Magn Reson Spectrosc 44:189–214

36. Tjandra N, Garrett DS, Gronenborn AM, Bax A, Clore GM (1997) Defining long range order in NMR structure determination from the dependence of heteronuclear relaxation times on rotational diffusion anisotropy. Nat Struct Biol 4:443–449
37. Bertini I, Luchinat C, Parigi G, Pierattelli R (2005) NMR spectroscopy of paramagnetic metalloproteins. ChemBioChem 6:1536–1549
38. Banci L et al (2004) Paramagnetism-based restraints for Xplor-NIH. J Biomol NMR 28:249–261
39. Bertini I, Luchinat C, Parigi G (2002) Paramagnetic constraints: an aid for quick solution structure determination of paramagnetic metalloproteins. Concepts Magn Reson 14:259–286
40. Schwieters CD, Kuszewski JJ, Tjandra N, Clore GM (2003) The Xplor-NIH NMR molecular structure determination package. J Magnet Res 160:65–73
41. Güntert P, Mumenthaler C, Wüthrich K (1997) Torsion angle dynamics for NMR structure calculation with the new program DYANA. J Mol Biol 273:283–298
42. Hus JC, Marion D, Blackledge M (2000) De novo determination of protein structure by NMR using orientational and long-range order restraints. J Mol Biol 298:927–936
43. Case DA et al (2005) The Amber biomolecular simulation programs. J Comput Chem 26:1668–1688
44. van der Spoel D et al (2005) GROMACS: fast, flexible, and free. J Comput Chem 26: 1701–1718
45. Krieger E, Koraimann G, Vriend G (2002) Increasing the precision of comparative models with YASARA NOVA–a self-parameterizing force field. Proteins 47:393–402
46. Montelione GT et al (2013) Recommendations of the wwPDB NMR validation task force. Structure 21(9):1563–1570
47. Kirchner DK, Güntert P (2011) Objective identification of residue ranges for the superposition of protein structures. BMC Bioinformatics 12:170
48. Mao B, Guan R, Montelione GT (2011) Improved technologies now routinely provide protein NMR structures useful for molecular replacement. Structure 19:757–766
49. Vuister GW, Fogh RH, Hendrickx PMS, Doreleijers JF, Gutmanas A (2013) An overview of tools for the validation of protein NMR structures. J Biomol NMR 58(4):259–285
50. Spronk C, Nabuurs SB, Krieger E (2004) Validation of protein structures derived by NMR spectroscopy. Prog Nucl Magn Reson Spectrosc 45:315–337
51. van der Schot G et al (2013) Improving 3D structure prediction from chemical shift data. J Biomol NMR 57:27–35
52. Rosato A et al (2009) CASD-NMR: critical assessment of automated structure determination by NMR. Nat Methods 6:625–626
53. Rosato A et al (2012) Blind testing of routine, fully automated determination of protein structures from NMR data. Structure 20:227–236
54. Vranken WF et al (2005) The CCPN data model for NMR spectroscopy: development of a software pipeline. Proteins 59:687–696
55. Shen Y, Delaglio F, Cornilescu G, Bax A (2009) TALOS+: a hybrid method for predicting protein backbone torsion angles from NMR chemical shifts. J Biomol NMR 44:213–223
56. Cheung M-S, Maguire ML, Stevens TJ, Broadhurst RW (2010) DANGLE: A Bayesian inferential method for predicting protein backbone dihedral angles and secondary structure. J Magn Reson 202:223–233
57. Wishart DS, Sykes BD (1994) Chemical shifts as a tool for structure determination. Methods Enzymol 239:363–392
58. Berjanskii MV, Wishart DS (2005) A simple method to predict protein flexibility using secondary chemical shifts. J Am Chem Soc 127: 14970–14971
59. Wassenaar TA, et al (2012) WeNMR: structural biology on the grid. J Grid Comp 10:743–767
60. Huang YJ, Powers R, Montelione GT (2005) Protein NMR recall, precision, and F-measure scores (RPF scores): structure quality assessment measures based on information retrieval statistics. J Am Chem Soc 127:1665–1674
61. Guerry P, Herrmann T (2012) Comprehensive automation for NMR structure determination of proteins. Methods Mol Biol 831:429–451
62. Linge JP, Habeck M, Rieping W, Nilges M (2003) ARIA: automated NOE assignment and NMR structure calculation. Bioinformatics 19:315–316
63. Raman S et al (2010) Accurate automated protein NMR structure determination using unassigned NOESY data. J Am Chem Soc 132:202–207
64. Linge JP, O'Donoghue SI, Nilges M (2001) Automated assignment of ambiguous nuclear overhauser effects with ARIA. Methods Enzymol 339:71–90
65. Seavey BR, Farr EA, Westler WM, Markley JL (1991) A relational database for sequence-specific protein NMR data. J Biomol NMR 1:217–236

66. Bhattacharya A, Tejero R, Montelione GT (2007) Evaluating protein structures determined by structural genomics consortia. Proteins 66:778–795

67. Vriend G (1990) WHAT IF: a molecular modeling and drug design program. J Mol Graph 8:52–56

68. Laskowski RA, Rullmann J, MacArthur MW (1996) AQUA and PROCHECK-NMR: programs for checking the quality of protein structures solved by NMR. J Biomol NMR 8:477–486

69. Herráez A (2006) Biomolecules in the computer: Jmol to the rescue. Biochem Mol Biol Educ 34:255–261

70. Koradi R, Billeter M, Wüthrich K (1996) MOLMOL: a program for display and analysis of macromolecular structures. J Mol Graph 14(51–5):29–32

71. Moseley HNB, Sahota G, Montelione GT (2004) Assignment validation software suite for the evaluation and presentation of protein resonance assignment data. J Biomol NMR 28:341–355

72. Wang L, Eghbalnia HR, Bahrami A, Markley JL (2005) Linear analysis of carbon-13 chemical shift differences and its application to the detection and correction of errors in referencing and spin system identifications. J Biomol NMR 32:13–22

73. Wang L, Markley JL (2009) Empirical correlation between protein backbone 15N and 13C secondary chemical shifts and its application to nitrogen chemical shift re-referencing. J Biomol NMR 44:95–99

74. Wang B, Wang Y, Wishart DS (2010) A probabilistic approach for validating protein NMR chemical shift assignments. J Biomol NMR 47:85–99

75. Ginzinger SW, Gerick F, Coles M, Heun V (2007) CheckShift: automatic correction of inconsistent chemical shift referencing. J Biomol NMR 39:223–227

76. Rieping W, Vranken WF (2010) Validation of archived chemical shifts through atomic coordinates. Proteins 78:2482–2489

Part IV

Protein–Ligand Interactions

Chapter 17

Methods for Predicting Protein–Ligand Binding Sites

Zhong-Ru Xie and Ming-Jing Hwang

Abstract

Ligand binding is required for many proteins to function properly. A large number of bioinformatics tools have been developed to predict ligand binding sites as a first step in understanding a protein's function or to facilitate docking computations in virtual screening based drug design. The prediction usually requires only the three-dimensional structure (experimentally determined or computationally modeled) of the target protein to be searched for ligand binding site(s), and Web servers have been built, allowing the free and simple use of prediction tools. In this chapter, we review the underlying concepts of the methods used by various tools, and discuss their different features and the related issues of ligand binding site prediction. Some cautionary notes about the use of these tools are also provided.

Key words Structural bioinformatics, Protein–ligand interaction, Protein surface grid, Molecular probe, Surface pocket and cavity, Ligand binding site prediction, Bioinformatics software and servers

1 Introduction

Interaction with a ligand molecule is essential for many proteins to carry out their biological function. This interaction is generally specific, not only in terms of the molecules involved in the interaction, but also in the location (i.e., the site of ligand binding) in which the interaction takes place. In order to gain knowledge about the interaction and, by extension, the protein's function and how to influence its activity by, for example, designing small molecule drugs, considerable efforts have been made to develop methods that can predict ligand binding sites (LBSs) of proteins computationally, and a very large number of bioinformatics tools are now available for LBS prediction (reviewed in [1–4]). In general, because of the location specificity of LBSs, most of these methods have exploited one or more of four types of properties (evolutionary, geometric, energetic, and statistical) in order to distinguish the binding site from other parts of the protein surface. In this review, we will survey the many LBS prediction methods and classify them on the basis of the site-distinguishing properties they

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_17, © Springer Science+Business Media New York 2015

use into one of the following categories: (1) template-based—those that utilize homologous and/or similar structures with known binding sites, (2) geometry-based—those that perform some kind of geometric computation to identify binding site pockets, (3) energy-based—those that compute the interaction energy usually using imaginary ligands as binding probes, (4) propensity-based—those that compute the propensity of a certain property that shows a statistically significant preference for known LBSs, rather than non-LBSs, and (5) combination-based and others—those that make the prediction based on the results of other methods and those that cannot be easily classified into one of the above categories. We will focus our review on the basic concepts and features of each of these categories; access information for, and notes about, the different methods surveyed are given in Table 1.

2 Methods

2.1 Template-Based Methods

Proteins sharing sequence homology are known to adopt similar three-dimensional (3D) structures and usually perform similar biological functions [5]. This is the basic idea behind all template-based methods for LBS prediction. The first step in these methods is to identify one or more ligand-bound complex structures (to serve as a template to find potential LBSs) that share sufficient sequence similarity with the target protein (the protein in which LBSs are to be predicted). By superimposing the target protein and identified templates, which include information on the location of known LBSs, a consensus site for ligand binding can be revealed and its characteristics as a putative LBS for the target protein evaluated by comparison to those of known LBSs [6–13]. As a result, the similarity between the template(s) and target protein and the accuracy of the sequence and structure alignments used in the procedure can affect the prediction accuracy of the methods in this category. In general, template-based methods tend to yield better accuracy in LBS prediction than methods in other categories (*see* **Note 1**). Thus, for example, in CASP (Critical Assessment of Protein Structure Prediction) competition experiments in the category of LBS prediction [14, 15], most participating groups employed some form of template-based approach [8–10, 16]. However, one caveat about the use of template-based methods is the requisite for a suitable template structure(s) containing known LBSs.

If the structure of the target protein has not yet been experimentally determined, template-based predictions of LBSs are still possible using homology-derived model(s) of the target protein as a substitute. It has been shown [17, 18] that, for about 90 % of proteins studied, at least one structure analog exists in the structure database PDB [19]. Thus, the large majority of all proteins can be structurally modeled using standard homology modeling

Table 1
List of LBS prediction methods

Method[1]	Reference	Web server[2]	Structure viewer[3]	Note
I. Template-based methods				
3DLigandSite	[6, 16]	http://www.sbg.bio.ic.ac.uk/~3dligandsite/	Yes	• Among the most highly scored methods in CASP8 [14]
FINDSITE	[7, 79]	http://www.cssb.biology.gatech.edu/findsite	–	• Evaluated on a large benchmark set containing>1,000 nonredundant structures and structure models • Option for subsequent docking computation
Firestar	[8, 86]	http://firedb.bioinfo.cnio.es/Php/FireStar.php	–	• Among the most highly scored methods in CASP9 [15] • Single chain input
I-TASSER	[9, 87–89]	http://zhanglab.ccmb.med.umich.edu/I-TASSER/	Yes	• Among the most highly scored methods in CASP9 [15] • Needs registration for job submission. • Includes structure prediction and functional annotation
Lee's method	[10]		–	• Among the most highly scored methods in CASP8 [14]
IntFOLD	[11, 90]	http://www.reading.ac.uk/bioinf/IntFOLD/	Yes	• Multiple prediction functions: structure model, domain, disorder, and binding sites, plus quality assessment of models
ProBis	[21, 28]	http://probis.cmm.ki.si	Yes	• Uses graph theory to find local structure similarity by comparison with about 24000 PDB structures
Im's method	[27]		–	• Concept similar to ProBis [21] • Uses spatial proximity to identify the 100 highest scoring templates
II. Geometry-based methods				
LIGSITE^{csc}	[29, 32]	http://projects.biotec.tu-dresden.de/pocket/	–	• Option for re-ranking by conservation score and changing the parameters • Two widely used established datasets: 210 bound structures and 48 pairs of bound/unbound structures • Option for input of single or multiple chains
PocketPicker	[30]		–	• Calculates "buriedness" values for each empty grid and clusters high "buriedness" grids
VICE	[31]		–	• Similar concept to PocketPicker [30], but with new indices, such as "cavityness" and "shaping factor"

(continued)

Table 1 (continued)

Method[1]	Reference	Web server[2]	Structure viewer[3]	Note
SCREEN	[35]	http://bhapp.c2b2.columbia.edu/screen2/cgi-bin/screen2.cgi	–	• Uses two spheres of different radii to identify empty spaces • Uses random forests, a machine learning method, to improve prediction accuracy • Batch job is allowed
POCASA	[36]	http://altair.sci.hokudai.ac.jp/g6/service/pocasa/	Yes	• Similar concept to SCREEN [35] • Excludes "noise points," e.g., points deep inside the protein structure
CASTp	[37–39]	http://sts.bioengr.uic.edu/castp/	Yes	• Maps residue information from Swiss-port [91] and SNP data from OMIM [92] to predicted LBS residues
MSPocket	[40]		–	• Source code available
fpocket	[41, 42]	http://bioserv.rpbs.univ-paris-diderot.fr/cgi-bin/fpocket	Yes	• Multiple functions: pocket prediction, molecular dynamics simulation of the pocket, and viewing of conserved pockets in homologous proteins
PocketDepth	[93]	http://proline.physics.iisc.ernet.in/pocketdepth/	Yes	• Similar concept to LIGSITE^{csc} [29, 32] • Uses a "depth factor" to identify surface grids of a pocket
DEPTH	[94]	http://mspc.bii.a-star.edu.sg/tankp/run_depth.html	Yes	• The main purpose of this server is to compute the depth of protein residues and the solvent accessible surface area (SASA), but it can also be used to predict the ligand binding cavity • Basic assumption: ligand binding residues are often both deep in a cavity and solvent-exposed • Option to adjust operating parameters
DoGSiteScorer	[95]	http://dogsite.zbh.uni-hamburg.de/	Yes	• Uses a variety of geometric and physicochemical properties (volume, depth, surface, ellipsoid main axes, site lining atoms and residues, and functional groups)
III. Energy-based methods				
SiteHound	[46, 53]	http://scbx.mssm.edu/sitehound/sitehound-web/Input.html	Yes	• Option to use one of 4 types of probes (methyl carbon, aromatic carbon, phosphate oxygen, or hydroxyl oxygen) to compute the interaction energy

Q-SiteFinder	[48]	http://www.modelling.leeds.ac.uk/qsitefinder/	Yes	• Compiles a widely used dataset containing 35 pairs of bound/unbound complexes • Uses the methyl group (–CH_3) as probe
Morita's method	[49]		–	• Similar to Q-SiteFinder, with modified probe distribution and clustering
FTSite	[50]	http://ftsite.bu.edu	Yes	• Option to use16 different small molecular probes separately on grids
IV. Propensity-based methods				
Hirayama's method	[54]		–	• Re-ranks the pockets predicted by SURFNET [34] by amino acid composition propensities for LBSs
STP	[56]	http://opus.bch.ed.ac.uk/stp/	–	• Re-ranks the cavities identified by SURFNET [34] according to the number of highly scored atoms • Does not provide the predicted pockets, but assigns a score to each atom
LISE	[57, 58]	http://lise.ibms.sinica.edu.tw	Yes	• Scored by binding site enrichment factors of protein triangles • Tested on two widely used small datasets and a large nonredundant set [57, 58] • Reports success rates for different types of proteins [58].
V. Combination-based methods and others				
ConCavity	[60]		–	• Combines scores from 3 different methods with residue conservation data
MetaPocket 2.0	[62]	http://projects.biotec.tu-dresden.de/metapocket/	–	• Reports the consensus of the results of 8 different methods
MEDock	[65]		–	• Uses docking to search for LBSs
Thornton's method	[67]		–	• Uses a neural network to predict active site residues in enzymes

[1] see Note 7
[2] see Note 8
[3] see Note 9

routines to facilitate LBS prediction (*see* Chapter 16 of this volume). Most template-based LBS prediction servers (e.g., *see* Table 1) can accept the target protein sequence as input, but the accuracy of the prediction will depend on the quality of the derived homology model (*see* **Note 2**).

Since protein structures are more highly conserved than amino acid sequences [20], structure comparison has the potential to discover suitable templates for LBS prediction in cases in which sequence comparison may fail. As structure determines function, some methods, such as ProBis [21], focus on the structural similarity between target and templates rather than sequence homology. However, even if template(s) can be identified by structure comparison, accurate prediction of LBSs still cannot be guaranteed because proteins with a globally similar 3D structure may have different functions and thus distinct LBSs [22, 23]. On the other hand, binding sites for similar ligands or similar enzymatic mechanisms have been found to be conserved in proteins sharing no overall structural similarity [24–26], and this has been harnessed with success by methods utilizing structure comparisons to identify similarity only at the level of local structures, i.e., the area surrounding the binding site [21, 27, 28].

2.2 Geometry-Based Methods

The main task of geometry-based methods is to identify, by computing some types of geometric measures, pocket(s) on the protein surface that can accommodate small ligand molecules. Statistical studies made on protein–ligand complex structures archived in PDB indicate that small molecule ligands tend to bind at deflated regions of the protein surface, in particular, its largest (and/or deepest) cavities [29]. Consequently, most geometry-based methods have focused on identifying the largest pockets in proteins. However, how to determine and identify cavities on the protein surface is a more complicated problem than it might appear at first sight, and, over the years, many diverse and creative approaches have been explored.

The first step in many LBS prediction methods in this category is to find empty space on the protein surface, and one popular approach is to spray grid points on the target protein and find empty grids (those not occupied by protein atoms) [11, 29–32]. For example, LIGSITE^{csc} [29] scans grids in seven directions (*x*, *y*, *z*, and four cubic diagonals) to identify surface-empty-surface connections, then clusters the empty grids from these to identify empty spaces for potential ligand binding. Another approach is to place empty spheres on the protein surface [33, 34]. For example, to find large empty spaces, SURFNET [34] places empty spheres between every pair of protein atoms that have no intervening protein atoms. One variation of the empty sphere approach involves rolling two spheres with different radii on the target protein to generate an inner and outer "surface" and reveal empty spaces, i.e., empty pockets, between these two surfaces [35, 36]. Yet another approach is to compute Delaunay triangulation to find voids on the protein surface [37–41]. For example, fpocket [41, 42] uses the alpha sphere [37] to identify empty spaces on a protein structure (an alpha sphere is a sphere that contacts four atoms and contains no internal atoms).

In the next step, the empty grids, probes, spheres, or voids are clustered to identify the largest pocket (cavity), which is often assigned as the best (top-ranked) predicted LBS. In the development of these methods, many parameters have been devised to help in the assignment of the top-ranked LBSs, including the use of a "buriedness index" [30, 31], "path length" and "shaping index" [31], and "depth factor" [36]. Other properties of protein structures, including the B-factor [43] and packing density [44], have been found to be highly relevant to LBSs and deserve further investigation. However, despite considerable success, the assumption that the largest pocket is the true LBS is not always correct. According to one study, of a rather small set of 210 protein–ligand complex structures surveyed, only about 70 % of ligands were found to bind to the largest cavity in the protein, while 87 % of the ligands bound to one of the three largest cavities [29]. To avoid relying on this inherently incorrect assumption, many of the geometry-based methods resort to re-ranking the identified cavities by incorporating other geometrical, physicochemical, and/or evolutionary (e.g., sequence conservation) properties [29, 35, 41, 45], although re-ranking does not usually dramatically improve the prediction results based on the cavities identified.

2.3 Energy-Based Methods

The aim of energy-based methods is to find patches of the protein surface that are energetically favorable for ligand binding [46–52]. This is usually done by devising a probe molecule and computing the interaction energy between the probe and surrounding protein atoms. Many of the energy-based methods are also grid-based, as they place probes on empty grids of the protein surface in order to perform the energy computation [46–50]. For example, SiteHound uses Molecular Interaction Fields (MIFs) to calculate the interaction energy between a target protein and a probe, grid points with a high interaction energy are then clustered to identify potential LBSs [46]. Generally speaking, energy-based methods are less diverse and much less numerous than geometry-based methods. The main differences between the various energy-based methods are in the design of the probes and how they are distributed on the protein surface. In general, there is a trade-off between prediction quality and computational load, which increases with increased probe complexity. The server SiteHound-web [46, 53] allows users to choose one of four types of probes to compute the interaction energy; the FTSite server uses up to 16 different small molecular probes to identify consensus clusters of grids; the server Q-SiteFinder [48] uses only the methyl group ($-CH_3$) as the probe, while Morita's method [49], which is similar to Q-SiteFinder, improves the prediction by introducing a new probe distribution and clustering scheme.

2.4 Propensity-Based Methods

Another class of methods computes neither geometry nor interaction energy, but the statistics of certain properties for their propensities to be at, or associated with, known LBSs. One typical propensity-based method is Hirayama's method [54], in which the predicted binding pockets generated by a program named Alpha Site Finder [55] are re-ranked by the amino acid composition, which shows small, but statistically significant, differences between LBSs and non-LBSs. Propensity-based methods often re-rank the pockets predicted by other methods, mainly geometry-based methods. For example, the surface triplet propensities (STP) algorithm [56] assigns a propensity score to each atom located in the binding pockets predicted by SURFNET [34], then re-ranks the SURFNET pockets by simply counting the number of high-scoring atoms. In contrast, a new propensity-based method developed in our laboratory [57, 58] does not rely on pockets pre-identified by other methods. This new method was named LISE for "Ligand Interacting and binding Site Enriched protein triangles." In LISE, the protein triangles are a triplet of three protein surface atoms simultaneously interacting with a ligand molecule and the three protein atoms are concomitantly enriched at LBSs and are assigned an enrichment factor deduced from a statistical analysis of a set of protein–ligand complex structures [59].

2.5 Combination-Based Methods and Others

Because geometry and energy are two distinct attributes of LBSs and because different methods may complement and/or compensate each other, it is not surprising that a combination-based approach can prove successful in LBS prediction (*see* **Note 3**). For example, ConCavity [60] uses a weighted equation to combine the "raw scores" from three different methods (LIGSITE^{cs} [29], SURFNET [34], and PocketFinder [61]) with surface residue conservation data to generate a final, composite score for prediction, while MetaPocket 2.0 [62, 63] combines the results of eight methods (LIGSITE^{cs} [29], PASS [33], Q-SiteFinder [48], SURFNET [34], Fpocket [41], GHECOM [64], ConCavity [60], and POCASA [36]) and uses their consensus for the prediction. The publications describing ConCavity and Metapocket 2.0 reported that a better prediction performance was achieved with the combined methods used in these two approaches than with each of the individual methods alone. Finally, there are other methods for predicting LBSs, such as MEDOCK [65] and MolSite [66], which use a genetic algorithm and docking computation, and Thornton's Method [67], which uses neural network learning, to predict LBSs.

2.6 Related Issues

As mentioned above, many creative and useful LBS prediction methods have been developed, especially during the past decade. The success rates of prediction for recently reported methods now routinely exceed 80 % for the top-ranked site and 90 % for the top-three sites, and each reported new method tends to claim a superior performance compared to previously reported methods; however, the comparison has often been made on rather small datasets and, in addition, a head-to-head comparison is difficult to achieve because different methods may employ different evaluation criteria. Other compounding factors, discussed below, are also involved.

Most geometry-based methods are evaluated by a "distance criterion," which checks the distances between a specific single point, usually the geometric center of the predicted LBS pocket, and all the ligand atoms, and dictates that at least one of these distances must be within 4 Å for the prediction to be considered a success [29]. Because the aim of geometry-based methods is to identify the largest pockets of the target protein, pinpointing the exact binding location within a pocket is often not a primary concern. In contrast, since energy-based methods often identify a small patch within a binding pocket as the most energetically favorable site for ligand binding, most energy-based methods are instead assessed using a "precision criterion," which checks the volume percentage of the predicted pocket occupied by the ligand [48, 49]. Although this seems to be more stringent than the "distance criterion," it should be noted that a predicted pocket, even one with a 100 % precision, may cover just a small part of the binding site, and this may not be sufficient to adequately represent the whole binding site, especially in the case of large pockets that can bind diverse ligand molecules. Alternatively, especially in template-based methods, one can evaluate whether each of the predicted ligand-binding amino acids is in contact with (e.g., within a certain distance of) the ligand, thus allowing the prediction's sensitivity and specificity to be calculated from the percentage of amino acid residues that are predicted to be LBS-associated [16, 60]. However, cases abound in which different ligands bind to the same protein in the same pocket, but at different locations, or even in a different pocket, and, thus, a residue considered to be a binding site residue for one ligand can be a non-binding site residue for another, making the computed sensitivity and specificity values somewhat dependent on the particular ligand and also on the particular complex structure used [57, 58].

The issue of how to derive a dataset of protein–ligand complexes with which to evaluate LBS predictions is also complicated. For example, although a certain threshold of amino acid sequence identity (say 30 %) can be used to remove homologous proteins to build a so-called "nonredundant" set, many structures may, as mentioned above, share a similar 3D structure and thus a similar LBS despite having a sequence identify below the threshold, making the dataset not really redundant. An example of this includes the finding that, in a dataset used widely for evaluating LBS prediction methods [29], there are six kinase structures that, along with their LBS pockets, can be well superimposed even though their sequence identities are below the threshold used to remove homologous structures [57]. On the other hand, as also mentioned above, even homologous proteins can have distinct LBSs, so the inclusion of homologous structures does not necessarily make a dataset redundant with regard to LBS prediction. Compounding the issue are the different criteria, such as different chain length and X-ray resolution, used to select protein structures. Thus, for example, we may expect geometry-based methods to perform better than energy-based methods on a dataset of large proteins, since pocket size increases linearly with protein size [48, 68] and geometry-based methods are designed to find large pockets. Furthermore, the experimentally determined protein structures deposited in PDB are heavily biased, as the protein in about 70 % of ligand-bound protein complex structures is a cytosolic enzyme [58], which, compared to other types of protein, is more likely to have a better resolution owing to, for example, interest in using them as targets for drug development. As a result, methods trained on PDB structures, while performing well on cytosolic enzymes, may perform badly on other types of proteins, such as membrane proteins, which are the targets of more than 50 % of the drugs currently in use [69–71], but have a rather small percentage of representation in PDB (*see* **Note 4**). This is also one of the main reasons that, when applied to proteome-wide data, many methods usually produce a much lower accuracy than those reported using small datasets [58].

Most LBS predictions have been carried out using structures of single chains. However, in order to function, many proteins form a complex of multiple homo- or hetero-chains, and their LBS may be located at the interface between protein chains. The biological unit of a protein should therefore be considered for LBS prediction whenever possible (*see* **Note 5**). Biological unit structures can be downloaded from PDBsum [72]. Finally, as more and more new methods are developed, the prediction algorithms involved become more and more sophisticated and often include new indices (shaping index, buriedness, etc.), modified clustering procedures, and novel properties for re-ranking. As a result, more and more parameters have to be optimized. While most LBS prediction servers provide a default set of parameters, users should be aware that some methods have been tested only on a small dataset or even require the use of different parameter values for different types or datasets of proteins (*see* **Note 6**).

3 Conclusions and Prospects

Notwithstanding the existence of promiscuous proteins (those that can bind many different ligand molecules) and promiscuous ligands (those that can bind to different sites on the same or different proteins) [73–76], protein–ligand interactions are generally specific, a fact that underlies the success of the various prediction methods reviewed above which use certain properties to distinguish between LBSs (specific interactions) and other parts of the protein surface (nonspecific interactions). Nevertheless, protein LBSs come in different sizes and shapes and may bind a variety of ligand molecules, making it difficult to evaluate LBS prediction methods, many of which have been developed and tested based only on a small, or biased, set of complex structures. In view of this, it is worth making additional efforts to build LBS databases for specific protein families, such as kinases [77, 78], whereby different types of LBSs can be systematically compared and studied to reveal the common principles of protein–ligand recognition, and this will help in developing a new generation of methods with improved accuracy in predicting LBSs across the spectrum of proteins. In the meantime, for the purpose of facilitating docking computations, it would be useful to have a success-measuring criterion, perhaps less sophisticated than the "distance," "precision," or "residue-based sensitivity/specificity" mentioned above, to evaluate different types of LBS prediction methods based on their ability to correctly identify only the ligand binding pocket and not necessarily the precise location or residues involved.

4 Notes

1. The first step in LBS prediction should be to search for a homologous structure(s). If ligand-bound homologous structure(s) exist in PDB, a template-based prediction is usually quite reliable, particularly if the prediction is supported by other types of methods.
2. With the exception of FINDSITE [7, 79] and perhaps a few others, most methods do not report their prediction accuracy on homology-derived models, although, according to FINDSITE, homology models can be tolerated for template-based LBS predictions.
3. It is generally recommended to use multiple different methods to find consensus and/or to identify potential LBSs for evaluation.
4. Several LBS prediction methods have been developed for specific protein families [80–84]; for target proteins belonging

to these specific protein families, these methods can be more reliable than those reviewed in this chapter. General-purpose methods may have a reduced accuracy in predicting the LBSs of certain specific protein families, such as membrane receptors [58].

5. Some methods and their Web server and/or stand-alone program may only accept single protein chains and may miss potential LBSs at chain–chain interfaces.

6. It may be easier for users to experiment using stand-alone programs, if available, to optimize the parameters for their specific application.

7. A difficult case for LBS prediction is when a significant conformational change occurs upon ligand binding. This is a problem that has not been sufficiently addressed in the literature for LBS prediction.

8. Submitting a batch of jobs to run on an online server is usually not recommended (or not allowed), as it may take up too much computing time. For this type of application, as in large-scale predictions, it is more feasible to run the batch job on a stand-alone version installed on the user's local computer.

9. Many structure viewers, such as Jmol [85], are freely available and can be installed to view the predictions for servers that do not offer an online viewing option.

References

1. Leis S, Schneider S, Zacharias M (2010) In silico prediction of binding sites on proteins. Curr Med Chem 17(15):1550–1562
2. Laurie AT, Jackson RM (2006) Methods for the prediction of protein–ligand binding sites for structure-based drug design and virtual ligand screening. Curr Protein Pept Sci 7(5):395–406
3. Perot S, Sperandio O, Miteva MA, Camproux AC, Villoutreix BO (2010) Druggable pockets and binding site centric chemical space: a paradigm shift in drug discovery. Drug Discov Today 15(15–16):656–667
4. Henrich S, Salo-Ahen OM, Huang B, Rippmann FF, Cruciani G, Wade RC (2010) Computational approaches to identifying and characterizing protein binding sites for ligand design. J Mol Recognit 23(2):209–219
5. Lee D, Redfern O, Orengo C (2007) Predicting protein function from sequence and structure. Nat Rev Mol Cell Biol 8(12):995–1005
6. Wass MN, Kelley LA, Sternberg MJ (2010) 3DLigandSite: predicting ligand-binding sites using similar structures. Nucleic Acids Res 38(Web Server issue):W469–W473
7. Brylinski M, Skolnick J (2008) A threading-based method (FINDSITE) for ligand-binding site prediction and functional annotation. Proc Natl Acad Sci U S A 105(1):129–134
8. Lopez G, Maietta P, Rodriguez JM, Valencia A, Tress ML (2011) Firestar – advances in the prediction of functionally important residues. Nucleic Acids Res 39(Web Server issue):W235–W241
9. Roy A, Kucukural A, Zhang Y (2010) I-TASSER: a unified platform for automated protein structure and function prediction. Nat Protoc 5(4):725–738
10. Oh M, Joo K, Lee J (2009) Protein-binding site prediction based on three-dimensional protein modeling. Proteins 77(Suppl 9): 152–156
11. Roche DB, Tetchner SJ, McGuffin LJ (2011) FunFOLD: an improved automated method for the prediction of ligand binding residues using 3D models of proteins. BMC Bioinformatics 12:160
12. Brylinski M, Feinstein WP (2013) eFindSite: improved prediction of ligand binding sites in

protein models using meta-threading, machine learning and auxiliary ligands. J Comput Aided Mol Des 27(6):551–567

13. Roy A, Zhang Y (2012) Recognizing protein–ligand binding sites by global structural alignment and local geometry refinement. Structure 20(6):987–997
14. Lopez G, Ezkurdia I, Tress ML (2009) Assessment of ligand binding residue predictions in CASP8. Proteins 77(Suppl 9): 138–146
15. Schmidt T, Haas J, Gallo Cassarino T, Schwede T (2011) Assessment of ligand-binding residue predictions in CASP9. Proteins 79(Suppl 10):126–136
16. Wass MN, Sternberg MJ (2009) Prediction of ligand binding sites using homologous structures and conservation at CASP8. Proteins 77(Suppl 9):147–151
17. Zhang Y (2008) Progress and challenges in protein structure prediction. Curr Opin Struct Biol 18(3):342–348
18. Liu J, Rost B (2001) Comparing function and structure between entire proteomes. Protein Sci 10(10):1970–1979
19. Rose PW, Bi C, Bluhm WF, Christie CH, Dimitropoulos D, Dutta S et al (2013) The RCSB Protein Data Bank: new resources for research and education. Nucleic Acids Res 41(Database issue):D475–D482
20. Illergard K, Ardell DH, Elofsson A (2009) Structure is three to ten times more conserved than sequence – a study of structural response in protein cores. Proteins 77(3):499–508
21. Konc J, Janezic D (2010) ProBiS algorithm for detection of structurally similar protein binding sites by local structural alignment. Bioinformatics 26(9):1160–1168
22. Keskin O, Tsai CJ, Wolfson H, Nussinov R (2004) A new, structurally nonredundant, diverse data set of protein–protein interfaces and its implications. Protein Sci 13(4): 1043–1055
23. Keskin O, Nussinov R (2005) Favorable scaffolds: proteins with different sequence, structure and function may associate in similar ways. Protein Eng Des Sel 18(1):11–24
24. Gherardini PF, Wass MN, Helmer-Citterich M, Sternberg MJ (2007) Convergent evolution of enzyme active sites is not a rare phenomenon. J Mol Biol 372(3):817–845
25. Totrov M (2011) Ligand binding site superposition and comparison based on atomic property fields: identification of distant homologues, convergent evolution and PDB-wide clustering of binding sites. BMC Bioinformatics 12(Suppl 1):S35
26. Xie L, Bourne PE (2008) Detecting evolutionary relationships across existing fold space, using sequence order-independent profile-profile alignments. Proc Natl Acad Sci U S A 105(14):5441–5446
27. Lee HS, Im W (2013) Ligand binding site detection by local structure alignment and its performance complementarity. J Chem Inf Model 53(9):2462–2470
28. Konc J, Janezic D (2010) ProBiS: a web server for detection of structurally similar protein binding sites. Nucleic Acids Res 38(Web Server issue):W436–W440
29. Huang B, Schroeder M (2006) LIGSITEcsc: predicting ligand binding sites using the Connolly surface and degree of conservation. BMC Struct Biol 6:19
30. Weisel M, Proschak E, Schneider G (2007) PocketPicker: analysis of ligand binding-sites with shape descriptors. Chem Cent J 1:7
31. Tripathi A, Kellogg GE (2010) A novel and efficient tool for locating and characterizing protein cavities and binding sites. Proteins 78(4):825–842
32. Hendlich M, Rippmann F, Barnickel G (1997) LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. J Mol Graph Model 15(6):359–363, 389
33. Brady GP Jr, Stouten PF (2000) Fast prediction and visualization of protein binding pockets with PASS. J Comput Aided Mol Des 14(4):383–401
34. Laskowski RA (1995) SURFNET: a program for visualizing molecular surfaces, cavities, and intermolecular interactions. J Mol Graph 13(5):323–330, 307–308
35. Nayal M, Honig B (2006) On the nature of cavities on protein surfaces: application to the identification of drug-binding sites. Proteins 63(4):892–906
36. Yu J, Zhou Y, Tanaka I, Yao M (2010) Roll: a new algorithm for the detection of protein pockets and cavities with a rolling probe sphere. Bioinformatics 26(1):46–52
37. Liang J, Edelsbrunner H, Woodward C (1998) Anatomy of protein pockets and cavities: measurement of binding site geometry and implications for ligand design. Protein Sci 7(9): 1884–1897
38. Binkowski TA, Naghibzadeh S, Liang J (2003) CASTp: computed atlas of surface topography of proteins. Nucleic Acids Res 31(13): 3352–3355
39. Dundas J, Ouyang Z, Tseng J, Binkowski A, Turpaz Y, Liang J (2006) CASTp: computed atlas of surface topography of proteins with structural and topographical mapping of

functionally annotated residues. Nucleic Acids Res 34(Web Server issue):W116–W118

40. Zhu H, Pisabarro MT (2011) MSPocket: an orientation-independent algorithm for the detection of ligand binding pockets. Bioinformatics 27(3):351–358
41. Le Guilloux V, Schmidtke P, Tuffery P (2009) Fpocket: an open source platform for ligand pocket detection. BMC Bioinformatics 10:168
42. Schmidtke P, Le Guilloux V, Maupetit J, Tuffery P (2010) fpocket: online tools for protein ensemble pocket detection and tracking. Nucleic Acids Res 38(Web Server issue): W582–W589
43. Yang LW, Bahar I (2005) Coupling between catalytic site and collective dynamics: a requirement for mechanochemical activity of enzymes. Structure 13(6):893–904
44. Shih CH, Chang CM, Lin YS, Lo WC, Hwang JK (2012) Evolutionary information hidden in a single protein structure. Proteins 80(6): 1647–1657
45. Glaser F, Morris RJ, Najmanovich RJ, Laskowski RA, Thornton JM (2006) A method for localizing ligand binding pockets in protein structures. Proteins 62(2):479–488
46. Ghersi D, Sanchez R (2009) EasyMIFS and SiteHound: a toolkit for the identification of ligand-binding sites in protein structures. Bioinformatics 25(23):3185–3186
47. Silberstein M, Dennis S, Brown L, Kortvelyesi T, Clodfelter K, Vajda S (2003) Identification of substrate binding sites in enzymes by computational solvent mapping. J Mol Biol 332(5):1095–1113
48. Laurie AT, Jackson RM (2005) Q-SiteFinder: an energy-based method for the prediction of protein–ligand binding sites. Bioinformatics 21(9):1908–1916
49. Morita M, Nakamura S, Shimizu K (2008) Highly accurate method for ligand-binding site prediction in unbound state (apo) protein structures. Proteins 73(2):468–479
50. Ngan CH, Hall DR, Zerbe B, Grove LE, Kozakov D, Vajda S (2012) FTSite: high accuracy detection of ligand binding sites on unbound protein structures. Bioinformatics 28(2):286–287
51. An J, Totrov M, Abagyan R (2004) Comprehensive identification of “druggable” protein ligand binding sites. Genome Inform 15(2):31–41
52. Cheng G, Qian B, Samudrala R, Baker D (2005) Improvement in protein functional site prediction by distinguishing structural and functional constraints on protein family evolution using computational design. Nucleic Acids Res 33(18):5861–5867
53. Hernandez M, Ghersi D, Sanchez R (2009) SITEHOUND-web: a server for ligand binding site identification in protein structures. Nucleic Acids Res 37(Web Server issue):W413–W416
54. Soga S, Shirai H, Kobori M, Hirayama N (2007) Use of amino acid composition to predict ligand-binding sites. J Chem Inf Model 47(2):400–406
55. Edelsbrunner H, Facello M, Fu R, Liang J (1995) Measuring proteins and voids in proteins. In proceedings of the twenty-eighth Hawaii international conference on system sciences, Vol. 5: Biotechnology Computing, IEEE Computer Society Press, Los Alamitos, CA. pp 256–264
56. Mehio W, Kemp GJ, Taylor P, Walkinshaw MD (2010) Identification of protein binding surfaces using surface triplet propensities. Bioinformatics 26(20):2549–2555
57. Xie ZR, Hwang MJ (2012) Ligand-binding site prediction using ligand-interacting and binding site-enriched protein triangles. Bioinformatics 28(12):1579–1585
58. Xie ZR, Liu CK, Hsiao FC, Yao A, Hwang MJ (2013) LISE: a server using ligand-interacting and site-enriched protein triangles for prediction of ligand-binding sites. Nucleic Acids Res 41(Web Server issue):W292–W296
59. Xie ZR, Hwang MJ (2010) An interaction-motif-based scoring function for protein–ligand docking. BMC Bioinformatics 11:298
60. Capra JA, Laskowski RA, Thornton JM, Singh M, Funkhouser TA (2009) Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3D structure. PLoS Comput Biol 5(12):e1000585
61. An J, Totrov M, Abagyan R (2005) Pocketome via comprehensive identification and classification of ligand binding envelopes. Mol Cell Proteomics 4(6):752–761
62. Zhang Z, Li Y, Lin B, Schroeder M, Huang B (2011) Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction. Bioinformatics 27(15):2083–2088
63. Huang B (2009) MetaPocket: a meta approach to improve protein ligand binding site prediction. OMICS 13(4):325–330
64. Kawabata T (2010) Detection of multiscale pockets on protein surfaces using mathematical morphology. Proteins 78(5):1195–1211
65. Chang DT, Oyang YJ, Lin JH (2005) MEDock: a web server for efficient prediction of ligand binding sites based on a novel optimization algorithm. Nucleic Acids Res 33(Web Server issue):W233–W238
66. Fukunishi Y, Nakamura H (2011) Prediction of ligand-binding sites of proteins by molecular

docking calculation for a random ligand library. Protein Sci 20(1):95–106

67. Gutteridge A, Bartlett GJ, Thornton JM (2003) Using a neural network and spatial clustering to predict the location of active sites in enzymes. J Mol Biol 330(4): 719–734
68. Laskowski RA, Luscombe NM, Swindells MB, Thornton JM (1996) Protein clefts in molecular recognition and function. Protein Sci 5(12):2438–2452
69. Liang X, Zhao J, Hajivandi M, Wu R, Tao J, Amshey JW et al (2006) Quantification of membrane and membrane-bound proteins in normal and malignant breast cancer cells isolated from the same patient with primary breast carcinoma. J Proteome Res 5(10):2632–2641
70. Cole ST (2002) Comparative mycobacterial genomics as a tool for drug target and antigen discovery. Eur Respir J Suppl 36:78s–86s
71. Almen MS, Nordstrom KJ, Fredriksson R, Schioth HB (2009) Mapping the human membrane proteome: a majority of the human membrane proteins can be classified according to function and evolutionary origin. BMC Biol 7:50
72. Laskowski RA (2001) PDBsum: summaries and analyses of PDB structures. Nucleic Acids Res 29(1):221–222
73. Singla N, Goldgur Y, Xu K, Paavilainen S, Nikolov DB, Himanen JP (2010) Crystal structure of the ligand-binding domain of the promiscuous EphA4 receptor reveals two distinct conformations. Biochem Biophys Res Commun 399(4):555–559
74. Ekins S, Kortagere S, Iyer M, Reschly EJ, Lill MA, Redinbo MR et al (2009) Challenges predicting ligand–receptor interactions of promiscuous proteins: the nuclear receptor PXR. PLoS Comput Biol 5(12):e1000594
75. Burris TP, Montrose C, Houck KA, Osborne HE, Bocchinfuso WP, Yaden BC et al (2005) The hypolipidemic natural product guggulsterone is a promiscuous steroid receptor ligand. Mol Pharmacol 67(3):948–954
76. von Eichborn J, Murgueitio MS, Dunkel M, Koerner S, Bourne PE, Preissner R (2011) PROMISCUOUS: a database for network-based drug-repositioning. Nucleic Acids Res 39(Database issue):D1060–D1066
77. Chiu YY, Lin CT, Huang JW, Hsu KC, Tseng JH, You SR et al (2013) KIDFamMap: a database of kinase-inhibitor-disease family maps for kinase inhibitor selectivity and binding mechanisms. Nucleic Acids Res 41(Database issue):D430–D440
78. van Linden OP, Kooistra AJ, Leurs R, de Esch IJ, de Graaf C (2013) KLIFS: a knowledge-based structural database to navigate kinase–ligand interaction space. J Med Chem 57(2):249–277
79. Skolnick J, Brylinski M (2009) FINDSITE: a combined evolution/structure-based approach to protein function prediction. Brief Bioinform 10(4):378–391
80. Gandhi NS, Mancera RL (2012) Prediction of heparin binding sites in bone morphogenetic proteins (BMPs). Biochim Biophys Acta 1824(12):1374–1381
81. Krick R, Busse RA, Scacioc A, Stephan M, Janshoff A, Thumm M et al (2012) Structural and functional characterization of the two phosphoinositide binding sites of PROPPINs, a beta-propeller protein family. Proc Natl Acad Sci U S A 109(30):E2042–E2049
82. Yu DJ, Hu J, Huang Y, Shen HB, Qi Y, Tang ZM et al (2013) TargetATPsite: a template-free method for ATP-binding sites prediction with residue evolution image sparse representation and classifier ensemble. J Comput Chem 34(11):974–985
83. Khare H, Ratnaparkhi V, Chavan S, Jayraman V (2012) Prediction of protein-mannose binding sites using random forest. Bioinformation 8(24):1202–1205
84. Gandhi NS, Freeman C, Parish CR, Mancera RL (2012) Computational analyses of the catalytic and heparin-binding sites and their interactions with glycosaminoglycans in glycoside hydrolase family 79 endo-beta-d-glucuronidase (heparanase). Glycobiology 22(1):35–55
85. Jmol: an open-source Java viewer for chemical structure s in 3D. http://www.jmol.lorg
86. Lopez G, Valencia A, Tress ML (2007) Firestar – prediction of functionally important residues using structural templates and alignment reliability. Nucleic Acids Res 35(Web Server issue):W573–W577
87. Zhang Y (2009) I-TASSER: fully automated protein structure prediction in CASP8. Proteins 77(Suppl 9):100–113
88. Zhang Y (2007) Template-based modeling and free modeling by I-TASSER in CASP7. Proteins 69(Suppl 8):108–117
89. Zhang Y (2008) I-TASSER server for protein 3D structure prediction. BMC Bioinformatics 9:40
90. Roche DB, Buenavista MT, Tetchner SJ, McGuffin LJ (2011) The IntFOLD server: an integrated web resource for protein fold recognition, 3D model quality assessment, intrinsic disorder prediction, domain prediction and ligand binding site prediction. Nucleic Acids Res 39(Web Server issue):W171–W176
91. Boeckmann B, Bairoch A, Apweiler R, Blatter MC, Estreicher A, Gasteiger E et al (2003)

The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. Nucleic Acids Res 31(1):365–370

92. McKusick VA (1998) On the naming of clinical disorders, with particular reference to eponyms. Medicine (Baltimore) 77(1):1–2

93. Kalidas Y, Chandra N (2008) PocketDepth: a new depth based algorithm for identification of ligand binding sites in proteins. J Struct Biol 161(1):31–42

94. Tan KP, Varadarajan R, Madhusudhan MS (2011) DEPTH: a web server to compute depth and predict small-molecule binding cavities in proteins. Nucleic Acids Res 39(Web Server issue):W242–W248

95. Volkamer A, Kuhn D, Rippmann F, Rarey M (2012) DoGSiteScorer: a web server for automatic binding site prediction, analysis and druggability assessment. Bioinformatics 28(15): 2074–2075

Chapter 18

Information-Driven Structural Modelling of Protein–Protein Interactions

João P.G.L.M. Rodrigues, Ezgi Karaca, and Alexandre M.J.J. Bonvin

Abstract

Protein–protein docking aims at predicting the three-dimensional structure of a protein complex starting from the free forms of the individual partners. As assessed in the CAPRI community-wide experiment, the most successful docking algorithms combine pure laws of physics with information derived from various experimental or bioinformatics sources. Of these so-called "information-driven" approaches, HADDOCK stands out as one of the most successful representatives. In this chapter, we briefly summarize which experimental information can be used to drive the docking prediction in HADDOCK, and then focus on the docking protocol itself. We discuss and illustrate with a tutorial example a "classical" protein–protein docking prediction, as well as more recent developments for modelling multi-body systems and large conformational changes.

Key words Biomolecular interactions, Information-driven docking, Conformational changes, Multi-body docking, HADDOCK, Molecular modelling

1 Introduction

Docking is defined as the modelling of the three dimensional (3D) structure of a molecular complex from its known unbound constituents. It was developed to aid in the structural elucidation of transient or weak interactions, which can be challenging to characterize experimentally due to, for example, difficulties in crystallization or because the molecular weight rules out a thorough classical NMR analysis. The advent of explicit treatment of molecular flexibility, together with better and more efficient algorithms for both sampling and scoring, has earned docking a solid reputation amongst experimentalists. In turn, this attention brought new challenges such as the prediction of large molecular assemblies, protein–nucleic acid complexes, high-throughput predictions of entire metabolic pathways, or understanding the molecular origins of binding affinity and specificity [1, 2].

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_18, © Springer Science+Business Media New York 2015

Docking predictions rely usually on a combination of shape complementary and some energy functions to define the conformational landscape of the interacting molecules and identify near-native models, respectively. Since most of these functions are borrowed from molecular dynamics and structure prediction software, they suffer from the same limitations due to their approximate character [3]. To gain the upper hand, available (experimental) information was incorporated into the algorithms to greatly enhance their accuracy. In fact, this approach—information-driven docking—has become the most reliable and successful in the docking community, as shown in the latest assessment of the community-wide docking assessment experiment (CAPRI) [4, 5].

Traditionally, information was incorporated in docking predictions as a post-sampling filter. This filter approach is simple and straightforward: it first blindly generates a large pool of models and then removes or penalizes models that do not agree with the data. Its disadvantage lies in the need for a large number of solutions that should cover, ideally, the entire conformational search space. Often, as in the case of large or extremely flexible systems, the computational cost associated with exhaustive sampling of the search space is prohibitively high. Therefore, an alternative is to incorporate the data directly during the sampling stage—the so-called *information-driven docking*. This effectively biases the algorithm to visit only regions of the interaction space that respect the information contained in the data and thus enriches the pool of generated models with "correct" models, provided of course the information is correct (which can be a pitfall of data-driven approaches) [6].

HADDOCK is an information-driven docking software [7] that was originally adapted from the NMR automated structure determination approach ARIA [8]. It uses CNS (Crystallography and NMR System) [9, 10] as computation engine and, as such, has access to a variety of energy functions that allow inclusion of various NMR-derived parameters such as chemical shift perturbations, NOE distances, residual dipolar couplings, and pseudo-contact shifts to drive the docking prediction (please see the Chapter 16 by Vranken et al. for a description of the various NMR data). Throughout the years, HADDOCK has been extended to support other types of information commonly generated in the "wet-lab" such as mutagenesis, chemical cross-linking, and SAXS, as well as "dry-lab" interface predictions from bioinformatics methods [11–13]. How this information is incorporated into HADDOCK depends on its characteristics. For instance, information such as NMR chemical shift perturbation (CSP), alanine scanning mutagenesis, chemical cross-linking, or EPR distances are translated into distance restraints. These restraints can have different levels of accuracy and ambiguity (one-to-many or one-to-one relationships, i.e., some highly ambiguous, e.g., from CSP, some very specific, e.g., EPR distances). These are generally implemented as an additional term in the energy function used to describe the conformational landscape. Lower resolution sources of information such as SAXS, Cryo-EM, or collision cross-section data (CCS) from ion-mobility mass spectrometry are currently used only for scoring in HADDOCK; this is done by measuring the discrepancy between the structural properties back-calculated from the generated models and the experimentally measured values. Information about the radius of gyration of the molecule can also be extracted from SAXS data; this one-dimensional value can in principle also be restrained during the sampling.

In this chapter, we will focus on *information-driven docking* with HADDOCK and describe how different types of data can be used in the modelling. We will illustrate this in a protocol example making use of HADDOCK and NMR chemical shift perturbation data to model the phosphoryl transfer complex between the signal transducing proteins HPr and IIa(glucose) of *E. coli* (PDB entry 1GGR) [14].

2 Theory

This section briefly discusses various useful information sources, how these can be used to drive docking predictions, and describes the HADDOCK strategy to produce structural models of biomolecular complexes. Since HADDOCK uses CNS for its structure calculations, details on the implementation of particular restraint type is best found in the CNS Web site (http://cns-online.org/v1.3) or related publications [9, 10].

2.1 Sources of Information for Data-Driven Docking

2.1.1 Common NMR Structural Information Sources

The majority of data derived from NMR, such as NOE inter-proton distances, hydrogen bonds, residual dipolar couplings, relaxation diffusion anisotropy, pseudo-contact shifts and paramagnetic relaxation enhancements, can be applied directly to docking. Please refer to the Chapter 16 for a detailed explanation of these restraints.

Some NMR experiments and other techniques particularly useful for the structural elucidation of interactions are shortly described below.

2.1.2 NMR Chemical Shift Perturbations

Chemical Shift Perturbations are a simple strategy to identify regions of a protein that interact with the partner molecule. Atomic nuclei register changes in their chemical environment as small perturbations in their well-defined chemical shifts. By labelling one of the components of the complex and titrating the other partner unlabelled, it is possible to follow which chemical shifts are displaced when compared to the spectra of the isolated partner, since the proximity of the partner will alter the chemical environment of those residues in the interface. The reverse experiment (first partner unlabelled and second labelled) can be done to map the interface on the other partner. The downside of this technique is that it cannot distinguish between perturbations caused by the proximity of the partner molecule and those caused by allosteric effects, conformational changes upon binding, or solvent reshuffling at the interface.

2.1.3 NMR Cross-Saturation

In cross-saturation experiments, one of the interacting partners is $^{2}H/^{15}N$ uniformly labelled. The only observable protons are those that can exchange back with protons of water (e.g., amide protons). Upon irradiation with a radiofrequency pulse, the unlabeled protein protons become instantly saturated due to spin diffusion effects. Afterwards, cross-relaxation phenomena transfer this saturation to neighboring interfacial protons on the labelled protein, reducing their peak intensity in a ^{15}N HSQC spectrum. Reversing the labelling on the partners allows this experiment to pinpoint accurate information on the binding interface. Since this experiment relies on direct through-space interactions, it is more reliable than chemical shift perturbation experiments, particularly for interactions where large conformational changes occur upon binding.

2.1.4 Hydrogen/ Deuterium Exchange

H/D exchange provides information on the solvent accessible residues of a protein. In a deuterated medium, amide protons exposed to the solvent exchange rapidly while those buried by the protein structure do not. Upon interaction, the interface of the proteins also becomes inaccessible to solvent exchange. Following this event by either NMR with ^{15}N HSQC spectra or by mass spectrometry reveals the solvent-accessible surface of the bound complex and, indirectly, the interfacial residues.

2.1.5 NMR Pseudocontact Shifts

Pseudocontact shift (PCS) experiments require a paramagnetic ion attached to the protein, much like paramagnetic relaxation enhancement experiments, and are usually measured in ^{15}N HSQC or ^{13}C HSQC spectra [15]. When comparing a reference (diamagnetic) spectrum with a paramagnetic spectrum recorded in the presence of, for example, a paramagnetic lanthanide ion, PCSs can be measured as the differences in the chemical shifts between both spectra. Intramolecular PCSs can be used to optimize the ΔX tensor parameters of the protein to which the lanthanide is attached, while intermolecular PCSs can be used to obtain the anisotropic tensor ΔX parameters with respect to the second protein. Since both ΔX tensors are theoretically equal, they can be used to derive relative orientations between the two interacting proteins. Furthermore, given the distance-dependence of the PCS effect, this experiment also provides (long-range) distance information between the proteins.

2.1.6 Residual Dipolar Couplings

Residual dipolar couplings (RDCs) are a chemical phenomenon manifested as an increase or decrease in the magnitudes of multiplet splittings that can be seen in undecoupled NMR spectra [16].

These couplings can be measured in solution by inducing a weak alignment of the molecule, which can be done using a variety of methods [17]. RDCs provide information on the orientation of the internuclear vector of the two atoms for which the RDC is measured (e.g., N–C, N–H) with respect to the three global axes of the alignment tensor. This information can be used in the context of docking to orient the binding partners with respect to each other, thus reducing the number of degrees of freedom to be sampled during the docking calculation [18, 19].

2.1.7 Long Distance Information

NOE-derived distances are typically short, below 5–6 Å. Larger distances can be reported by other experimental sources, such as chemical cross-linking detected by mass spectrometry (the distance depends on the linker length and flexibility), EPR (20–80 Å), FRET (~50 Å). These can be used to define upper limits in the docking predictions. These techniques been used, for instance, in the characterization of very large molecular assemblies that are not amenable to NMR [20–22].

2.1.8 Low-Resolution Shape Information (Cryo-EM, SAXS, CCS)

Low-resolution information obtained through Cryo-EM, SAXS, and CCS experiments provide overall shape information that can be used either to limit the conformational space search in sampling stages, or to filter “incorrect” models after the sampling. Cryo-EM information is often simpler and faster to obtain compared to NMR experiments, in particular for very large multi-body assemblies like chaperonins or the nuclear pore complex, since there are less requirements for sample preparation and there is no peak assignment step to be carried out, although it still requires plenty of manual intervention and can become challenging when conformational heterogeneity is present. The resulting electron density maps allow, depending on the resolution, the identification and positioning of the interacting partners in the density map. SAXS provides a scattering curve and allows the calculation of the radius of gyration of the assembly. It can also be used to generate molecular envelopes that can be used in a similar manner as Cryo-EM maps. Collision Cross Section (CCS) data from ion mobility mass spectrometry provides a rotationally averaged two-dimensional projection of the molecules. Currently, HADDOCK implements SAXS (and CCS data) as a filter [23]. These can be best used after the rigid-body energy minimization step, and only for complexes whose components show asymmetry in their molecular shapes.

2.1.9 Bioinformatics Predictions

In the absence of experimental data, there is a wealth of information stored in sequence and structure databases that can be manipulated to provide predictions on the interface of the complex [11]. Of the many algorithms and Web servers available to predict protein interfaces, one is routinely used in tandem with HADDOCK: CPORT [24]. This server makes use of the structure of the free proteins to search for possible interfacial residues by sequence and structure homology. Other approaches based on correlated mutation from evolutionary records have been proposed that predict unambiguous contacts between interacting partners [25, 26].

2.2 Protein–Protein HADDOCKing

2.2.1 Docking Protocol

Docking and flexible refinement in HADDOCK are performed in three successive stages:

- *it0*: Rigid-Body Energy Minimization (RBEM).
- *it1*: Semi-Flexible Simulated Annealing (SA) in Torsion Angle Space (TAD/SA).
- *water*: Restrained Molecular Dynamics in Explicit Solvent.

These are preceded by a structure/topology generation stage that rebuilds missing atoms if necessary and a post-processing stage in which various energy terms, restraint violations, and intermolecular contact are analyzed.

Rigid-Body Energy Minimization (RBEM, it0)

In the initial docking stage, the interacting partners are first separated in space and each is randomly rotated around its center of mass to remove any orientational bias. They are then subjected to a rigid-body energy minimization protocol, where first only the orientation of the partners is optimized, and then both rotations and translations are allowed, effectively resulting in the docking of the molecules. Given the fast calculation of each docking model at this stage, it is typically worth generating a large number of models to cover the interaction space. By default, 1,000 are written to disk, although 10,000 are sampled—each model is the result of five internal docking trials with, for each, the 180°-rotated solution around the normal to the interface being sampled as well. These models are then ranked according to the HADDOCK score (see below), and a fraction of these is selected for further flexible refinement—typically 200.

Semi-flexible Simulated Annealing in Torsion Angle Space (TAD/SA, it1)

The second stage of the HADDOCK protocol fine-tunes each complex by flexible refinement of its interface. This second stage starts with a rigid-body SA step to optimize the orientation of the components. Then, the side chains of the interface—automatically defined for each docking model as all residues within 10 Å of a partner molecule—are allowed to move in a second SA stage. A third and final SA stage optimizes both backbone and side-chains of the interface residues to allow for some conformational rearrangements. Finally, a short energy minimization in Cartesian space relaxes the models. A new ranking of the models is produced at this step, but, usually, all models are allowed to undergo the third and final refinement step in explicit solvent.

Restrained Molecular Dynamics in Explicit Solvent (Water)

By default, both the RBEM and TAD/SA stages do not include any explicit description of the solvent (*see* **Note 1**). The docking is performed in vacuum with a dielectric constant (ε) of 10 for the electrostatic Coulomb energy term (should be set to 78 in case of protein–DNA docking). To improve the network of hydrogen bonds and electrostatic interactions at the interface, as well as increase the realism of the prediction, the third and final step of the HADDOCK protocol is a short restrained molecular dynamics simulation in Cartesian space in a shell of explicit solvent, either TIP3P (water, 8 Å shell) or DMSO (lipid-mimic, 12.5 Å shell).

2.2.2 HADDOCK Score

All structure calculations in HADDOCK are bound to a set of energetics terms, which together form the HADDOCK score. This score is a weighted sum of terms whose weights depend on the stage of the HADDOCK protocol:

- *(it0)* $E = 0.01\ E_{vdW} + 0.1\ E_{elec} + 1.0\ E_{desolv} - 0.01\ BSA + 0.01\ E_{AIR}$
- *(it1)* $E = 1.0\ E_{vdW} + 1.0\ E_{elec} + 1.0\ E_{desolv} - 0.01\ BSA + 0.1\ E_{AIR}$
- *(water)* $E = 1.0\ E_{vdW} + 0.2\ E_{elec} + 1.0\ E_{desolv} + 0.01\ E_{AIR}$

where the van der Waals and electrostatic energy terms, E_{vdW} and E_{elec}, are represented by Lennard–Jones and Coulomb potentials, E_{desolv} is an empirical desolvation term developed by Fernandez-Recio et. al. [27], BSA is the buried surface area of the model in Ångstrom, and E_{AIR} is the energy reflecting the accordance of the model to the input restraints (the distance-based ones). Other terms might be included, such as E_{sym} for symmetry restraints or E_{RDCs} for residual dipolar coupling, depending on the application.

2.2.3 Clustering of Final Solutions

HADDOCK models are not analyzed on a per model basis. Instead, it is assumed that groups of structurally similar models with overall low HADDOCK score are the best representatives of the near-native conformation. To form these groups, HADDOCK offers two clustering algorithms. The default choice in HADDOCK2.2 is a contact-based algorithm that groups models based on the similarity of their contact networks at the interface—fraction of common contacts (FCC) [28]. This is particularly efficient for large molecules, multi-body assemblies, and symmetrical complexes. The other choice is a standard RMSD-based clustering algorithm [29] that uses the backbone interface-ligand RMSD as a distance measure. The backbone interface-ligand RMSD is calculated by first fitting the models on the interface of the first component of the complex (which should be the largest). Then, the RMSD is computed on the interface of the remaining components. The defaults cutoffs for each algorithm are 0.75 for FCC clustering and 7.5 Å for backbone interface-ligand RMSD.

2.3 Restraints Implemented in HADDOCK

2.3.1 Ambiguous Interaction Restraints (AIRs)

Several experimental techniques provide information on residues that are potentially involved in the interaction, but fail to report on the specificity of the residue pair interactions (i.e., they produce surface patches but not the specific pairwise residue contacts). HADDOCK implements this information as Ambiguous Interaction Restraints (AIRs). This concept, similar to ambiguous NOEs [30], is designed to create an attraction between the interfaces during the docking without favoring a particular orientation. To support this implementation, HADDOCK divides interacting residues in two classes—active and passive—that differ in their contribution to the binding event. Active residues are usually those involved directly in the interaction, for which there is strong experimental or predictive information, while passive residues comprise those that are surface accessible neighbors of active residues. Active residues will be forced to be at the interface, otherwise generating a restraint violation, while passive residues may or may not be at the interface (if not, no violation is generated). AIRs are generated between each active residue on one partner and all active and passive residues on the other partner(s). The total number of AIRs is equal to the sum of active residues. False negatives (missing information) are dealt with usually by automatic definition of passive residues, while false positives can be minimized by randomly removing a fraction (50 %, by default) of the restraints for each docking trial. Internally, HADDOCK (or rather CNS) uses lists of active and passive residues for each interacting partner to define an *effective distance* for each active residue. This distance is defined between an active residue i of protein A and all active and passive residues of protein B as,

$$d_{iAB}^{\text{eff}} = \left(\sum_{m_{iA=1}}^{N_{A\text{atom}}} \sum_{k=1}^{N_{\text{res}B}} \sum_{n_{kB=1}}^{N_{B\text{atom}}} \frac{1}{d_{m_{iA}n_{kB}}^{6}} \right)^{-\frac{1}{6}}$$

where $N_{A\text{ atom}}$ indicates all atoms of residue i on protein A, $N_{\text{res}B}$ indicates each active and passive residue on protein B, $N_{B\text{ atom}}$ indicates all atoms of the residue k on protein B, and d is the Euclidean distance between atoms m_{iA} and n_{kB}. The effective distances are restrained to a target value by a flat bottom harmonic potential with upper and lower bounds. This potential switches to a linear potential beyond the upper bound to avoid large forces that could cause the calculations to fail. By default, HADDOCK sets the effective upper bound to 2 Å and the lower bound to 0 Å (the van der Waals energy term prevents potential atom overlap), meaning that if an effective distance d^{eff} greater than 2 Å separates pair of restrained residues, these feel an attractive force. This seemingly small distance is a by-product of the mathematical formula (it grows smaller with the number of distances entering the sum), and in reality translates to real distances of 3–5 Å between atoms of the active/passive residues in the model.

2.3.2 Unambiguous Distance Restraints

When the information is sufficiently accurate to allow unambiguous pairing of atoms or residues between partner molecules, it is possible to incorporate it in HADDOCK as unambiguous distance restraints. These use the same functional form of the distance restraining potential as the AIRs (flat bottom harmonic potential with upper and lower bounds transitioning to a linear potential after the upper bound to avoid large forces). In practice, ambiguous and unambiguous boils down to the syntax in which the restraints are written: either with multiple atom selections linking one residue or group of atoms to many in the partner molecule, or just to one specific pair of atoms. The main difference is that unambiguous restraints are never randomly discarded while ambiguous restraints are by default. Ambiguous and unambiguous distance restraints can therefore be combined and written in any of the two types of distance restraint files. Next to these two types, HADDOCK also allows the definition of a third class of restraints as hydrogen bond restraints. These are treated in a similar manner as unambiguous restraints. Each restraint file can be activated or deactivated at various stages of the protocol.

2.3.3 Symmetry Restraints

If there is a priori information that hints or determines that the complex assumes a symmetrical arrangement, it is possible to incorporate this in the docking prediction and thus restrict quite substantially the conformational search space. HADDOCK supports several types of cyclic and dihedral symmetries, implemented through combinations of symmetrical distance restraints [31, 32]: C2, C3, C4, D2, and C5. Next to the symmetry restraints, non-crystallographic symmetry restraints (NCS restraints) can also be defined: these ensure that two molecules are similar without imposing any symmetry operation between them, equivalent to restraining the RMSD between the two molecules to be zero.

2.3.4 Center of Mass Restraints

To ensure compactness of solutions, for example when using symmetry restraints without experimental information, HADDOCK allows the definition of distance restraints between the geometric centers of mass of the molecules (based only on CA atoms). These so-called center of mass restraints can also be used when there is no information about the binding interface (*ab initio* docking), albeit with a lower chance of success since the only factor in play are the physical terms of the scoring function.

2.3.5 Other Orientational Restraints

Next to the distance-based restraints defined above, HADDOCK supports a variety of orientational restraints from NMR, including residual dipolar couplings [19] and relaxation anisotropy data that are useful in defining the relative orientation of the molecules, and pseudo-contact shifts that provide both distance and orientational information [33]. For details refer to the Chapter 16.

3 Materials

The following Subheading 4 describes an example protocol on the usage of HADDOCK to model the interaction between two proteins by using experimental information. In order to follow the example, the reader must have a number of software programs installed on his/her computer, together with the data provided in the Extra Materials (extras.springer.com). The protocol should be run on a GNU/Linux system (or under Mac OSX).

3.1 HADDOCK v2.2

HADDOCK can be obtained free of charge for academic users at http://nmr.chem.uu.nl/haddock. In principle, little is required to install and configure HADDOCK (detailed instructions are provided with the software). Also, a mailing-list is available to all current and potential users to provide advice and troubleshooting for any problem(s) encountered during installation and usage (http://groups.yahoo.com/group/haddock-discuss). Furthermore, the WeNMR project provides several tutorials, tips, and a help center related to HADDOCK and might also be of interest to all users (http://www.wenmr.eu/wenmr/support/documentation/nmr-services/haddock). Finally, a Web server version of HADDOCK is also available, free of charge for academic users that offers a user-friendly interface to all the powerful features of this docking software (http://www.haddocking.org).

3.2 Crystallography and NMR Suite (CNS) v1.3

CNS can be obtained free of charge for academic users at http://cns-online.org/v1.3. The usage of particular experimental information to drive the docking, such as pseudo-contact shifts (PCS) requires the installation of an additional module and recompilation of CNS. Otherwise, the binaries provided at the Web address are sufficient to run HADDOCK. HADDOCK v2.2 is designed for CNS v1.3, so ensure that the versions of the software are appropriate.

3.3 ProFit

The ProFit software calculates the root mean square deviation of atomic coordinates between molecules. It is designed to "be the ultimate protein least squares fitting program" and it supports a powerful zone selection syntax. It can be obtained free of charge for academic users at the author's Web site (http://www.bioinf.org.uk/software/profit/).

3.4 NACCESS

NACCESS is a software program that uses the Lee & Richards method [34] to calculate the solvent accessible area of a molecule from its three-dimensional PDB coordinates. It is available free of charge to academic and nonprofit organizations (http://bioinf.manchester.ac.uk/naccess/).

3.5 PyMOL

PyMOL is an open-source molecular visualization system offering a large number of features and extensible through Python scripts. It is available in several formats depending on the affiliation and needs of the user at www.pymol.org.

3.6 Grace (xmgrace)

Grace is a simple 2D plotting software program that can be obtained free of charge here: http://plasma-gate.weizmann.ac.il/Grace.

4 Methods

4.1 Modelling of Complexes with HADDOCK

We describe here the use of the HADDOCK2.2 package for the modelling of a protein–protein complex using chemical shift perturbation data. We will use data from the `haddock2.2/examples/e2a-hpr` directory. You should first copy this directory to the directory you are working in (*see* **Note 2**):

```
cp -r $HADDOCK/examples/e2a-hpr .
```

4.1.1 Preparation of PDB Files and Input Data

Make sure that your input models are compliant with the PDB format, particularly, the presence of an `END` statement as the last line of the file. Furthermore, the segment identifier (characters 73–76 in each `ATOM` statement) and chain identifier fields (character 22 in each `ATOM` statement) should be empty strings (i.e., filled with spaces). If you use a crystal structure, make sure that there are no double occupancies or residue insertions. If you are using an ensemble of models, split the file in individual files that contain only one structure (*see* **Note 3**).

As input data, you should combine chemical shift perturbation data (or other data indicating residues at the interface) and solvent accessibility data calculated with NACCESS: use only those residues that have both a high enough chemical shift perturbation (*see* **Note 4**) and a high enough relative accessibility. In the example, the residue solvent accessibilities calculated with NACCESS are already provided in the files `e2a_1F3G.rsa` and `hpr/hpr_rsa_ave.lis` (the latter containing the average for the 10 starting models for hpr). From these files you can select the residues with high enough (e.g., >~40 %) accessibility (*see* **Note 5**). You could calculate the accessibility values yourself using the following command:

```
naccess e2a_1F3G.pdb
```

4.1.2 Definition of Active and Passive Residues

Passive residues are defined as the solvent accessible surface neighbors of active residues. To define and visualize them you can use a molecular visualization program, for example PyMOL,

```
pymol e2a_1F3G.pdb
```

Start by coloring the active residues, for example in red. Then, filter out the residues with a low solvent accessibility, using either the output of NACCESS, recommended, or an embedded tool of the visualization program (e.g., `get_area` command in PyMOL). Next, select all surface neighbors within a certain cutoff radius (e.g., 5 Å), and that are solvent accessible, to define the passive residues and color them for example in green. In the `e2a-hpr` example, several PyMOL scripts are provided with the respective residues already colored according to this scheme: `e2a_pymol_active.pml`, `e2a_pymol_active_passive.pml` and similar for hpr. You can load these scripts in PyMOL using the following commands:

```
pymol e2a_1F3G.pdb
```

Then, in the PyMOL command line, type:

```
@e2a_pymol_active.pml
```

You will use the active and passive residues for both molecules to generate Ambiguous Interaction Restraints (AIRs); for this go to the HADDOCK GenTBL service (http://haddock.chem.uu.nl/services/GenTBL/) and follow the instructions. You should save the resulting file as `ambig.tbl` in the working directory; note that, in the `e2a-hpr` example directory, an example file named `e2a-hpr_air.tbl` is already present and can be used for comparison (*see* **Note 6**).

4.1.3 Setup of a New Run: new.html

To set up a new run, go to the project setup page on http://www.nmr.chem.uu.nl/haddock, click on "start a new project" and follow the instructions. Depending on the experimental data you have available, you can input various data files such as ambiguous restraints, unambiguous restraints, RDCs etc. PCS restraints are not yet supported in the Web site, but an example case is provided with the HADDOCK software. After saving the `new.html` file to disk, type `haddock2.2` in the same directory. This will generate a run directory containing all necessary information to run haddock. An example of a `new.html` file can be found in the `e2a-hpr` directory as `new.html-refe.` (*see* **Note 7**) and is displayed below. Such a file can in principle also be created by manual editing.

```
<html>
<head>
<title>HADDOCK - start</title>
</head>
<body bgcolor=#ffffff>
<h2>Parameters for the start:</h2>
<BR>
<h4><!-- HADDOCK -->
AMBIG_TBL=./e2a-hpr_air.tbl<BR>
HADDOCK_DIR=../../<BR>
N_COMP=2<BR>
PDB_FILE1=./e2aP_1F3G.pdb<BR>
PDB_FILE2=./hpr/hpr_1.pdb<BR>
PDB_LIST2=./hpr-files.list<BR>
PROJECT_DIR=./<BR>
PROT_SEGID_1=A<BR>
PROT_SEGID_2=B<BR>
RUN_NUMBER=1<BR>
submit_save=Save updated parameters<BR>
</h4><!-- HADDOCK -->
</body>
</html>
```

4.1.4 Run.cns

The next step is to define all parameters to perform the docking run. For this, enter the newly created directory:

```
      cd run1
```

You will find a file called `run.cns` containing all the parameters to run the docking, which deserves special attention. You need to edit this file and define a few parameters such as the location of the CNS executable and the queue command to use. Other options such as the semi-flexible segments at the interface, or fully flexible segments (*see* **Note 8**), the number of models to generate at each stage, the clustering algorithm and cutoff, and the force constants for the several energy terms are also defined there. You can edit your `run.cns` file manually or via “Project Setup” on http://www.nmr.chem.uu.nl/haddock. More information is available via the “run.cns” option in the manual section on http://www.nmr.chem.uu.nl/haddock.

4.1.5 Docking Run

To actually start the docking run with HADDOCK type in the directory containing the `run.cns` file (*see* **Note 9**).

```
    haddock2.2 >& haddock.out &
```

As more extensively explained in Subheading 2 before and "the docking" section in the HADDOCK manual, the entire protocol consists of three stages. An initial topology and structure generation step validates and builds the structure files to be used in the docking. The initial models as provided by the user are written to `data/sequence/`.

- *Topology and model generation.* The resulting topologies (*.psf) and coordinates (*.pdb) files are written to the `begin/`directory (*see* **Notes 10** and **11**). There is one output file per chain—`generate_X.out`, where X is the segment identifier given in `run.cns`—that must be checked for errors if there is a problem at this stage (*see* **Note 12**).
- *Randomization and rigid body energy minimization.* The docked models are written to `structures/it0/`. When all models have been generated, HADDOCK will write the PDB files with names sorted according to the HADDOCK score (weights defined in the `run.cns`) to `file.cns`, `file.list,` and `file.nam` in the same directory. The number of trials (`ntrials`, by default 5) and the sampling of 180° rotated solutions (`rotate180_0`, by default true) can be modified in `run.cns`.
- *Semi-flexible simulated annealing.* The best models after rigid body docking (defined at `structures_1` in `run.cns` and by default 200) will be subjected to a semi-flexible simulated annealing (SA) in torsion angle space. The temperatures and number of steps for the various stages are also defined in `run.cns`. The resulting refined models are written into `structures/it1`. The numbering of the file names reflects their rank from the previous step (e.g., `complex_1.pdb` is the refined best ranked structure in `it0` according to the HADDOCK score). At the end of the calculation, HADDOCK generates the `file.cns`, `file.list,` and `file.nam` files as in the previous stage (*see* **Note 13**). At the end of this stage, the models are analyzed and the results are placed in the `structures/it1/analysis` directory (see the analysis sections below).
- *Flexible explicit solvent refinement.* The choice of the solvent in which to refine the models is defined in `run.cns` (`solvent`) and can be either `water` or `dmso`. The resulting models are written in the `structures/it1/water` directory. The numbering in the files here matches that of the previous stage (e.g., `complex_1w.pdb` is the water refined `complex_1.`

pdb of it1). At the end of the explicit solvent refinement, HADDOCK generates the file.cns, file.list, and file.nam files. Finally, the models are analyzed and the results are placed in the structures/it1/water/analysis directory (*see* the analysis sections).

4.1.6 Automatic Analysis

A number of analysis scripts are automatically run after the semi-flexible and explicit solvent refinement stages and the results placed in structures/it1/analysis and structures/it1/water/analysis, respectively. Here we discuss a few of the most relevant output files.

- e2a-hpr_fcc.disp: contains the pairwise FCC matrix; this file is used as input for FCC clustering. If the clustering algorithm is RMSD, then the filename is e2a-hpr_rmsd.disp. The FCC measure, unlike RMSD, is asymmetric (FCC(AB) ≠ FCC(BA)) so it produces a full matrix.
- cluster.out: contains the clusters generated from the abovementioned matrix. The clusters are numbered according to their size (number of models in the cluster) and not according to their HADDOCK score. This is related to the algorithm used to cluster the models.
- noe.disp: contains the number of distance restraints violations per structure and averaged over the ensemble over all distance restraint classes and for each class (unambiguous, ambiguous, hbonds) separately. Similar files are generated when you have RDC (sani.disp), relaxation anisotropy (dani.disp), or PCS (pcs.disp) restraints.
- energies.disp: contains the various energy terms per model and averaged over the ensemble.
- ana_*.lis: there is a set of files called ana*.lis where * can be dihed_viol, dist_viol_all, hbond_viol, hbonds, nbcontacts, noe_viol_all, noe_viol_ambig, noe_viol_unambig. The "viol" refers to violations, and those files contain listings of violations including the number of times a restraint is violated as well as the average distance and violation per restraint. In addition, ana_hbonds.lis gives a listing of hydrogen bonds, and ana_nbcontacts.lis a listing of non-bonded contacts.
- ene-residue.disp: contains intermolecular energies for all interface residues.
- nbcontacts.disp: contains non-bonded contacts.

4.1.7 Manual Analysis

An important part of the analysis needs to be performed manually. A number of analysis scripts and programs are provided in the tools directory. These allow you to collect various statistics on the generated models and more importantly to perform re-clustering of solutions and their analysis on a per-cluster basis.

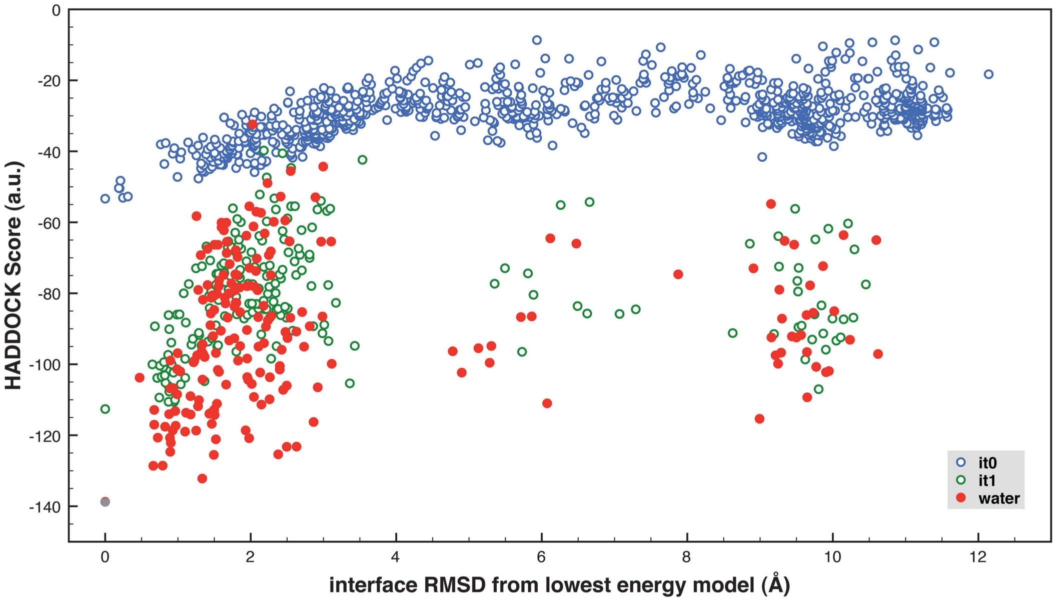


Fig. 1 Plot of HADDOCK scores versus interface RMSD from the lowest energy model for the three stages of the docking protocol (*blue*, *green*, and *red*, for it0, it1, and water refinement, respectively). One can clearly see a funnel at low RMSD values becoming more apparent after flexible refinement

– *Collecting statistics of the models with ana_structure.csh:* This script should be run once the `file.list` file has been created. It extracts various energy terms, violation statistics, and the buried surface area from each PDB file and calculates the RMSD of each structure compared to the lowest energy one (if the location of ProFit is defined (see installation and software links on http://www.nmr.chem.uu.nl/haddock)). The output are several files named "`structures*.stat`" that contain the same information sorted in different ways. Usually, the most important file is `structures_haddock-sorted.stat`. From this file, you can generate a plot of the HADDOCK score as a function of the RMSD to the lowest energy model and investigate if the run produces an "energy funnel," meaning that the low energy models should have small RMSD values and the high energy models should have large RMSD values (*see* Fig. 1). A script called `make_ene-rmsd_graph.csh` is provided in `$HADDOCKTOOLS` and it produces a Grace compatible plot file. Specify two columns to extract data from and a filename:

```
    $HADDOCKTOOLS/make_ene-rmsd_graph.csh    3    2    structures_haddock-
sorted.stat
```

This will generate a file called `ene_rmsd.xmgr`, which you can display using xmgrace:

```
    xmgrace ene_rmsd.xmgr
```

– *Clustering of solutions*: The clustering is run automatically in it1/analysis and it1/water/analysis, based on the criteria defined in the run.cns file. However, try using different cutoffs for the clustering since it is difficult to know a priori the best RMSD/FCC cutoff. This value depends on the system under study and the number of experimental restraints used to drive the docking (*see* **Note 14**). For FCC clustering, the script to use it cluster_fcc.py, while for RMSD clustering, use the C program cluster_struc (this should have been compiled during the installation of HADDOCK). The scripts read the appropriate e2a-hpr_*.disp file containing the pairwise matrix and generate clusters. The usage is (in the analysis directory):

```
python cluster_fcc.py e2a-hpr_fcc.disp cut-off [options] >cluster.out
```

or

```
cluster_struc [-f]  e2a-hpr_rmsd.disp cut-off min_size >cluster.out
```

Here cutoff indicates the FCC/RMSD cutoff to determine if two models belong in the same cluster. For FCC clustering, there are several options that can be modulated (type python cluster_fcc.py -h for the list and their explanation). In the RMSD clustering script, min_size is the minimum number of models in a cluster (typically a number like 4) and -f is an optional full-linkage clustering algorithm.
In either case, the output looks like the following:

```
Cluster 1 -> 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 23 24 27 28 43


Cluster 2 -> 25 26 29 32 34 35 57 71 73 20 21 44 39 46 …
```

The numbers after the arrow correspond to the rank in file.nam. The "sorted" models are also in the analysis directory. For example, 2 corresponds to the second model in the analysis directory, which is the second structure listed in file.list in it1 or it1/water.

– *Analysis of the clusters with ana_clusters.csh:* This script takes the output of the clustering script, by default analysis/cluster.out, to perform an analysis of the various clusters, calculating average energies, RMSDs, and buried surface area per cluster. The following runs the analysis on all clusters:

```
$HADDOCKTOOLS/ana_clusters.csh [-best #] analysis/cluster.out
```

The -best # is an optional argument to generate additional files with cluster averages calculated only on the best (#) ranked models of a cluster according to their HADDOCK score. This recommended option allows removing the dependency of the cluster averages on the size of the respective clusters (*see* **Note 15**). The ana_clusters.csh script analyses the clusters in a similar way as the ana_structures.csh script, but in addition generates average values over the models belonging to one cluster. It creates a number of files for each cluster containing the cluster number clustX in the name. It also creates files containing various averages over the clusters, cluster_xxx.txt; these contain the average and standard deviation of various terms such as intermolecular energy (xxx=ene) etc. Also, files combining all the above information and sorted based on various criteria are provided: clusters.stat that contains the various cluster averages and clusters_xxx-sorted.stat where xxx is the energy term according to which the values are sorted (e.g., xxx=ene for intermolecular energy, etc.). Again, the most relevant output file is clusters_haddock-sorted.stat, or rather clusters_haddock-sorted.stat_bestX.

- *Rerunning the HADDOCK analysis on a cluster basis*: Having performed the cluster analysis, you can now rerun the HADDOCK analysis for the best models of each cluster to obtain violation and energetics details and statistics. To run this analysis, we need the cluster-specific file.nam_clust#, file.list_clust# and file.cns_clust# files. A script in the tools/directory called make_links.csh will move the original file.nam, file.list and file.cns files to file.nam_all, file.list_all, file.cns_all and the same with the analysis directory. It will then create links to the appropriate files (file.nam_clust#, …) and to a new analysis_clust#/directory.

For example, to rerun the analysis for the best 10 models of the first cluster type in the water directory:

```
     $HADDOCKTOOLS/make_links.csh clust1_best10
     cd ../../..
     haddock2.2
```

The cd command brings you back into the main run directory from where you start again HADDOCK. Only the analysis of the best 10 models of the first cluster in the water will be run. Once this is finished, go to the respective analysis directory and inspect the various files. The RMSD from the average models should now be low (check rmsave.disp).

Having run the HADDOCK analysis on a cluster basis for each cluster, you should now have new directories in the water directory, called analysis_clustX_best10. Each of these analysis directories contains now cluster specific statistics. You can also visualize the clusters, using for example PyMOL. We provide a Perl script in the tools/directory, joinpdb, which allows concatenation of the various PDB files into one single ensemble file:

```
     $HADDOCKTOOLS/joinpdb -o e2a-hpr_clust1.pdb e2a-
hprfit_*.pdb
     pymol e2a-hpr_clust1.pdb
```

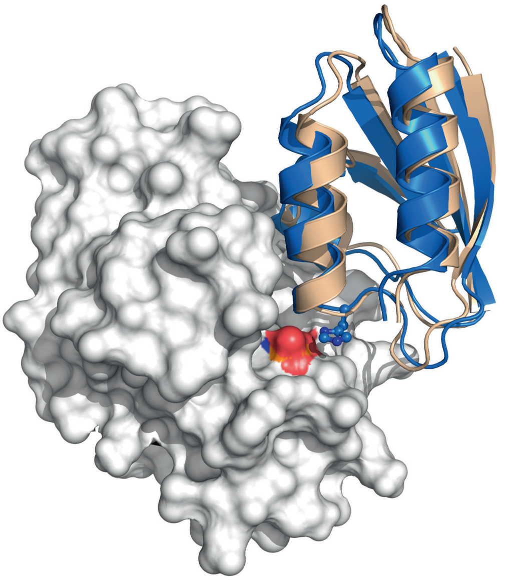


Fig. 2 Superimposition of the top model of the best scoring cluster onto the native structure (PDB ID 1GGR). The molecules were superimposed on backbone atoms of E2A, which is shown in white surface representation with the phosphorylated histidine colored according to the atom types (*blue, red*, and *orange*, for nitrogen, oxygen, and phosphorous, respectively). The HPR molecules are shown in cartoon representation (the model in *blue* and the native in peach) and the histidine residue involved in the phosphate transfer in ball-and-stick. The model is in excellent agreement with the native structure (interface RMSD = 0.97 Å). The proximity of the two histidines across the interface, which was not defined as a restraint in HADDOCK, is consistent with the biological function of this phosphotransferase complex

In general, the top ranked models of the cluster with the lowest HADDOCK score are considered the representatives of the biological system. However, scoring in docking remains a difficult problem and we do recommend, if possible, using additional independent information to validate the results (e.g., mutagenesis data). The selected model should explain as much as possible all what is known about the system (*see* Fig. 2).

4.2 Other Docking Scenarios

Although the previous example illustrates a canonical dimer docking, HADDOCK also supports more advanced protocols. Users can model large macromolecular complexes, address substrate-induced conformational changes, and deal with extremely flexible peptides. These protocols are briefly explained below.

4.2.1 Multi-body Docking

HADDOCK allows users to include up to six molecules and dock them simultaneously [31]. Multi-body HADDOCKing, as this protocol is called, follows the same rules of the original pairwise docking protocol requiring only that each molecule is restrained to at least another one of the system. The restraints between the different molecules are defined with the same syntax described in the case of dimer docking, and can be generated via the Web server indicated before: http://haddock.chem.uu.nl/services/GenTBL/. Here, we recommend the user to also turn on center of mass restraints, in order to ensure the compactness of the resulting models. If there is any cyclic and/or dihedral symmetry present, the user can activate the built-in symmetry restraints and impose them between and/or within each molecule (*see* Subheading 2.3.3).

4.2.2 Flexible Multi-domain Docking

Modelling binding-induced large conformational changes is a major challenge of the docking community, since it requires sampling a vast and intricate conformational space. Unfortunately, addressing such a number of degrees of freedom is often out of reach for most of the current sampling methods, including the one of HADDOCK. To tackle this challenge, we developed a special application of the multi-body docking protocol [35] that divides to conquer: the flexible partner is cut at hinge regions, and thus dissected into rigid domains that allow HADDOCK to sample a wider range of motions during rigid-body energy minimization.

The identification of the hinge regions can be carried out using normal mode analysis, such as that provided by the Web server HingeProt (*see* **Note 16**). To ensure molecular integrity and biological realism, we define connectivity restraints (in the form of distance restraints) between the separated domains. These are first defined with a maximum distance of 10 Å, to allow sampling of large range of motions. They are then shortened to a peptide bond distance (1.3 Å) at the water refinement step. Imposing different connectivity restraints is possible by submitting both `unambig.tbl` and `hbonds.tbl` restraint files (a copy except for the connectivity distance), and changing the stages when these are active in `run.cns` (options `unambig_firstit (0)`, `hbond_firstit (2)` and `unambig_lastit (1)`, `hbind_lastit (2)`). It is also necessary to define the artificial termini as uncharged and the first three residues starting from the "cut" hinge as fully flexible.

4.2.3 Protein–Peptide Docking

Albeit the other end of the size spectrum, small systems such as peptides are also challenging regarding sampling. Their extreme flexibility and the many conformations they can adopt upon binding makes them challenging to model and require usually long molecular dynamics simulations or other advanced sampling methods, none of which is possible or feasible, time-wise, for use in HADDOCK.

To cover the conformational landscape of peptides, we developed a shortcut approach. In this custom-tailored protocol, the peptide is provided as an ensemble of three most common conformations: α-helical, β-strand, and polyproline II (*see* **Note 17**). Additionally, the number of MD steps in the flexible refinement stage needs to be increased fourfold to improve sampling efficiency (from 500/500/1000/1000 to 2000/2000/4000/4000). Finally, the peptide is defined entirely as fully flexible and the clustering algorithms are adapted for small molecules (*see* **Note 14**).

5 Notes

1. HADDOCK has a special feature—solvated docking—that allows water molecules to be introduced at the interface of the complex for entire duration of the docking protocol. This feature should only be used when the experimental information is accurate enough to drive the docking and the interface is expected to be "wet." In short, solvated docking starts by surrounding each molecule by a shell (approximately 4 Å wide) of water molecules, optimized via a short MD simulation, prior to the RBEM stage. After the minimization, all water molecules that are not at the interface are removed. At the interface, only a fraction of the molecules is kept (by default 25 %), with the removal being carried out via a biased Monte Carlo sampling method whose criteria is based on a statistical potential of amino acid–water contact propensities. Finally, energetically unfavorable water molecules (those with a positive intermolecular energy) are removed, which might lead to a complete desolvation of the interface, and another round of RBEM is performed to optimize the final complex. The remaining of the HADDOCK protocol remains unchanged, with the difference that interfacial water molecules might be included in the further refinement. We refer the reader to the following references for an in-depth explanation of solvated docking in HADDOCK: [36–39].
2. HADDOCK must be correctly installed for the `$HADDOCK` environment variable to be defined. Check the installation instructions provided with the software.
3. If your input PDBs contains missing segments, this might lead to domains drifting away during the refinement stage. To avoid this, simply define a few unambiguous distance restraints between CA atoms from the various "sub-domains," setting the actual measured distance as a target distance and the bounds to 0.0. The same can be done to ensure that an ion coordination geometry is properly maintained. Missing residues at the interface or in hinge regions must be handled with extreme care not to compromise the biological integrity of the models. Missing atoms, on the other hand, are not problematic since HADDOCK rebuilds them based on the topology files of the force field, as long as the residue name is defined in them. Also, termini charges are very important for the docking protocol, as they can lead to artificial interactions. By default,

termini are charged but they can be neutralized by using an appropriate linkage file (`protein-allhdg5-4-*.link` files in the `toppar/`directory). For example, to have both termini of molecule *A* uncharged, simply add in run.cns (option `prot_link_A`) the appropriate linkage file (`protein-allhdg5-4-noter.link`). Another important point concerns ions; if proper care is not taken, they can be problematic during the torsion angle dynamics stage. HADDOCK has an inbuilt mechanism that defines artificial bonds to "chelate" the ion to the protein but it relies on proper ion naming. Check these names in the `covalions.cns` script and add yours if necessary. Also, make sure that their name in the PDB file matches the ion names defined in the `ion.top` file in the `toppar/`directory. To avoid that a N- or C-terminal patch be applied to them, they should also be defined in the `topallhdg5.4.pep` file (look for the `"first IONS"` and `"last IONS"` statements).

4. We have developed an automated method to discriminate the significant CSP—SAMPLEX [40]. It compares two sets of chemical shifts from two different samples (e.g., bound/unbound), and using the three-dimensional structure of the molecules, returns the confidence for each residue to be in a perturbed or unperturbed state.

5. The accessibility cutoff is not a hard limit; check the accessibilities and possibly include residues with lower accessibilities but with functionally important groups.

6. The syntax of the restraints is what determines their (un) ambiguous character, not the filename where they are stored: `ambig.tbl`, `unambig.tbl`, or `hbonds.tbl`. This allows for example to mix unambiguous and ambiguous restraints in the same file. The difference lies in the random removal option (`noecv=true`), which is applied only to `ambig.tbl`. In principle, `ambig.tbl`; `unambig.tbl` and `hbonds.tbl` could be used concurrently, for example, to provide extra NOEs or other data (e.g., FMD connectivity restraints) for which one wants to use different force constants or for which there is exceptional certainty.

7. An important setting in `new.html` is the value of `N_COMP`. This should be set equal to the number of components of the complex (2 in case of a dimer, 3 for a trimer, etc.). Note that it can also be set to 1, in which case HADDOCK can be used for refinement instead of docking.

8. HADDOCK allows the definition of fully flexible regions: these are treated as fully flexible throughout all stages, except the initial rigid-body docking. This should be useful for cases where part of a structure are disordered or unstructured or when docking small flexible molecules onto a protein.

9. This command causes HADDOCK to run in the background and all output to be redirected to the `haddock.out` file. If at some stage HADDOCK stops producing new models and the run is not yet finished, unzip and search for error messages in the output files:

   ```
   gunzip  xxx.out.gz
   ```

   `xxx.out.gz` is a particular output file, in which you should look for ERR. Also, kill the current HADDOCK process:

   ```
   ps -ef | grep haddock
   kill -9 id
   ```

   Here `id` is the process id that is returned by the `ps -ef` command.

   You can restart a HADDOCK run, but before doing that, make sure to delete any *.out file from the run directory (*see* **Note 12**). HADDOCK will proceed from where it was in the calculations.

10. The OPLS force field used by HADDOCK is a mixed united/all-atom force field. This means that protons do not have van der Waals parameters of their own. Instead these are accounted for in the heavy atom parameters to which they are attached. In HADDOCK, by default, nonpolar hydrogen atoms are deleted in order to speed up the calculation. This does not affect the resulting models significantly since the missing hydrogen atoms are actually accounted for in the united atoms parameters. You can change this behavior by setting `delenph=true` in `run.cns`. This should be done in case classical NOE distance restraints are used, or in situations where the hydrogen atoms are extremely relevant (e.g., small molecule docking).

11. Topology generation is often the most problematic stage in HADDOCK. While most amino acids and their most common modifications are supported in HADDOCK, small ligands and exotic modifications to amino acids or nucleic acid bases that are not described in the force field will give an error and halt the docking protocol. A list of the supported modified amino acids is given here: http://haddock.science.uu.nl/services/HADDOCK/library.html. For those molecules not in the list, the user is left to generate their own parameterization scheme, using for example PRODRG [41], ACPYPE [42], or ATB [43], and provide the necessary files (topology and parameters in CNS format) in the `toppar/`directory: `ligand.top` and `ligand.par`. Molecule parameterization is not simple though, and must be approached and carried out with extreme care and expertise (for a "best-practices" guide for parameterization, although under a different force-field, check the following reference: [44]).

12. If a particular stage of HADDOCK fails, in case of a model not being generated, the run being stopped accidentally, etc., reissuing the command `haddock2.2` might not be enough. HADDOCK makes use of the output files to control the flow of the docking run. When a particular step is initiated, HADDOCK write a `.job` file and a `.out` file, and when it is completed, the .out file is compressed to a `.out.gz` file. To safely restart a HADDOCK run, remove all `.out` files prior to the issuing of the `haddock2.2` command.
13. A typical error at this stage is that a couple of models in `it1` are not successfully generated. Often, this can be solved by changing the random seed in `run.cns` (`iniseed`, by default 917) and restart HADDOCK (*see* **Note 12**). Otherwise, try to decrease the `timestep` (e.g., 0.001 instead of 0.002) and/or the temperature of the first two SA stages (e.g., 1,000 or 500 K instead of 2,000). HADDOCK, by default, tries this automatically in case a model fails. If none of this works, simply copy the missing models from the `it0` directory so that the run can proceed. This can be done using the `copy-missing.csh` script provided in the `tools` directory with as arguments the file root name and the number of the missing model.
14. If only a small fraction of the models do fall into clusters, try decreasing the cutoff in case of FCC clustering, or increasing it in case of RMSD clustering. If all models fall in one single cluster, and the restraints are not that restrictive, try the reverse. This is particularly relevant for protein–small molecule docking, for which a tailored FCC clustering algorithm using small molecule atoms–protein residue contacts is available. For the RMSD clustering, due to the nature of interface-ligand RMSD, the resulting RMSD values are larger than would be obtained by fitting on all chains of the complex, which explains the large cutoff value that is used by default (7.5 Å).
15. It is better to use matching number of models (e.g., 4) to compare the cluster statistics in order to remove cluster size effects. In our experience, the size of a cluster does not always correlate with its quality/score and as such cannot be used as an indicator of the quality of the cluster.
16. The choice of the hinge region(s) where to "cut" the molecules should be made with the structural integrity of the molecule in mind. As such, hinges are favored if located at the end of α-helices and β-strands, or in loops. The experimental temperature factors should also be taken into account when deciding between possible hinges. The molecule should then be cut at the first peptide bond following the predicted hinge region.
17. The peptide conformations can be generated using PyMOL and setting the φ/ψ dihedral angles according to the desired secondary structure: −57° and −47° for α-helix, −139° and −135° for β-strand, and −78° and −149° for polyproline II.

References

1. Melquiond AS, Karaca E, Kastritis PL et al (2012) Next challenges in protein–protein docking: from proteome to interactome and beyond. Comput Mol Sci 2:642–651
2. Kastritis PL, Bonvin AM (2013) Molecular origins of binding affinity: seeking the Archimedean point. Curr Opin Struct Biol 23(6):868–877
3. Schlick T, Collepardo-Guevara R, Halvorsen LA et al (2011) Biomolecular modeling and simulation: a field coming of age. Quart Rev Biophys 44:191–228
4. Janin J (2013) The targets of CAPRI rounds 20–27. Proteins 81(12):2075–2081
5. Lensink MF, Janin J (2013) Docking, scoring and affinity prediction in CAPRI. Proteins 81(12):2082–2095
6. de Vries SJ, Melquiond ASJ, Kastritis PL et al (2010) Strengths and weaknesses of data-driven docking in critical assessment of prediction of interactions. Proteins 78:3242–3249
7. Dominguez C, Boelens R, Bonvin AMJJ (2003) HADDOCK: a protein–protein docking approach based on biochemical or biophysical information. J Am Chem Soc 125: 1731–1737
8. Linge JP, Habeck M, Rieping W et al (2003) ARIA: automated NOE assignment and NMR structure calculation. Bioinformatics 19:315–316
9. Brünger AT, Adams PD, Clore GM et al (1998) Crystallography & NMR system: a new software suite for macromolecular structure determination. Acta Crystallogr D Biol Crystallogr 54:905–921
10. Brunger AT (2007) Version 1.2 of the crystallography and NMR system. Nat Protocol 2:2728–2733
11. de Vries SJ, Bonvin AMJJ (2008) How proteins get in touch: interface prediction in the study of biomolecular complexes. Curr Protein Pept Sci 9:394–406
12. Karaca E, Bonvin AMJJ (2013) Advances in integrative modeling of biomolecular complexes. Methods 59:372–381
13. Schmitz C, Melquiond AS, de Vries SJ et al (2012) Protein–protein docking with HADDOCK, NMR of biomolecules: towards mechanistic systems biology, 1st edn. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, pp 521–535
14. Wang G, Louis JM, Sondej M et al (2000) Solution structure of the phosphoryl transfer complex between the signal transducing proteins HPr and IIA(glucose) of the Escherichia coli phosphoenolpyruvate:sugar phosphotransferase system. EMBO J 19:5635–5649
15. Bertini I, Calderone V, Cerofolini L et al (2012) The catalytic domain of MMP-1 studied through tagged lanthanides. FEBS Lett 586:557–567
16. Bax A (2003) Weak alignment offers new NMR opportunities to study protein structure and dynamics. Protein Sci 12:1–16
17. Prestegard JH, Bougault CM, Kishore AI (2004) Residual dipolar couplings in structure determination of biomolecules. Chem Rev 104:3519–3540
18. Tjandra N (1997) Direct measurement of distances and angles in biomolecules by NMR in a dilute liquid crystalline medium. Science 278:1111–1114
19. van Dijk ADJ, Fushman D, Bonvin AMJJ (2005) Various strategies of using residual dipolar couplings in NMR-driven protein docking: application to Lys48-linked di-ubiquitin and validation against 15N-relaxation data. Proteins 60:367–381
20. Kalisman N, Adams CM, Levitt M (2012) Subunit order of eukaryotic TRiC/CCT chaperonin by cross-linking, mass spectrometry, and combinatorial homology modeling. Proc Natl Acad Sci U S A 109:2884–2889
21. Choi UB, Strop P, Vrljic M et al (2010) Single-molecule FRET-derived model of the synaptotagmin 1-SNARE fusion complex. Nature 17:318–324
22. Brunger AT, Strop P, Vrljic M et al (2011) Three-dimensional molecular modeling with single molecule FRET. J Struct Biol 173: 497–505
23. Karaca E, Bonvin AMJJ (2013) On the usefulness of ion-mobility mass spectrometry and SAXS data in scoring docking decoys. Acta Cryst D69:683–694, 1–12
24. de Vries SJ, Bonvin AMJJ (2011) CPORT: a consensus interface predictor and its performance in prediction-driven docking with HADDOCK. PloS One 6:e17695
25. Weigt M, White RA, Szurmant H et al (2009) Identification of direct residue contacts in protein–protein interaction by message passing. Proc Natl Acad Sci U S A 106:67–72
26. Marks DS, Hopf TA, Sander C (2012) Protein structure prediction from sequence variation. Nat Biotechnol 30:1072–1080
27. Fernández-Recio J, Totrov M, Abagyan R (2004) Identification of protein–protein interaction

sites from docking energy landscapes. J Mol Biol 335:843–865

28. Rodrigues JPGLM, Trellet M, Schmitz C et al (2012) Clustering biomolecular complexes by residue contacts similarity. Proteins 80: 1810–1817
29. Daura X, Gademann K, Jaun B et al (1999) Peptide folding: when simulation meets experiment. Angew Chem Int Ed 38:236–240
30. Nilges M, O'Donoghue SI (1998) Ambiguous NOEs and automated NOE assignment. Progr Nucl Magn Reson Spectros 32:107–139
31. Karaca E, Melquiond ASJ, de Vries SJ et al (2010) Building macromolecular assemblies by information-driven docking: introducing the HADDOCK multibody docking server. Mol Cell Proteomics 9:1784–1794
32. Nilges M (1993) A calculation strategy for the structure determination of symmetric dimers by 1H NMR. Proteins 17:297–309
33. Schmitz C, Bonvin AMJJ (2011) Protein–protein HADDocking using exclusively pseudocontact shifts. J Biomol NMR 50:263–266
34. Lee B, Richards FM (1971) The interpretation of protein structures: estimation of static accessibility. J Mol Biol 55:379–400
35. Karaca E, Bonvin AMJJ (2011) A multidomain flexible docking approach to deal with large conformational changes in the modeling of biomolecular complexes. Structure 19: 555–565
36. van Dijk ADJ, Bonvin AMJJ (2006) Solvated docking: introducing water into the modelling of biomolecular complexes. Bioinformatics 22:2340–2347
37. Kastritis PL, van Dijk ADJ, Bonvin AMJJ (2012) Explicit treatment of water molecules in data-driven protein–protein docking: the solvated HADDOCKing approach. Methods Mol Biol 819:355–374
38. Kastritis PL, Visscher KM, van Dijk ADJ et al (2013) Solvated protein–protein docking using Kyte-Doolittle-based water preferences. Proteins 81:510–518
39. van Dijk M, Visscher KM, Kastritis PL et al (2013) Solvated protein-DNA docking using HADDOCK. J Biomol NMR 56:51–63
40. Krzeminski M, Loth K, Boelens R et al (2010) SAMPLEX: automatic mapping of perturbed and unperturbed regions of proteins and complexes. BMC Bioinformatics 11:51
41. Schüttelkopf AW, van Aalten DMF (2004) PRODRG: a tool for high-throughput crystallography of protein–ligand complexes. Acta Crystallogr D Biol Crystallogr 60:1355–1363
42. Sousa da Silva AW, Vranken WF (2012) ACPYPE – AnteChamber PYthon Parser interfacE. BMC Res Notes 5:367
43. Malde AK, Zuo L, Breeze M et al (2011) An automated force field topology builder (ATB) and repository: version 1.0. J Chem Theor Comput 7:4026–4037
44. Lemkul JA, Allen WJ, Bevan DR (2010) Practical considerations for building GROMOS-compatible small-molecule topologies. J Chem Inf Model 50:2221–2235

Chapter 19

Identifying Putative Drug Targets and Potential Drug Leads: Starting Points for Virtual Screening and Docking

David S. Wishart

Abstract

The availability of 3D models of both drug leads (small molecule ligands) and drug targets (proteins) is essential to molecular docking and computational drug discovery. This chapter describes a simple approach that can be used to identify both drug leads and drug targets using two popular Web-accessible databases: (1) DrugBank and (2) The Human Metabolome Database. First, it is illustrated how putative drug targets and drug leads for exogenous diseases (i.e., infectious diseases) can be readily identified and their 3D structures selected using only the genomic sequences from pathogenic bacteria or viruses as input. The second part illustrates how putative drug targets and drug leads for endogenous diseases (i.e., noninfectious diseases or chronic conditions) can be identified using similar databases and similar sequence input. This chapter is intended to illustrate how bioinformatics and cheminformatics can work synergistically to help provide the necessary inputs for computer-aided drug design.

Key words Drug, Disease, Drug target, Metabolite, Bioinformatics, Sequence comparison, Chemical similarity, Exogenous disease, Endogenous disease

1 Introduction

As most readers have already seen in previous chapters, protein modeling is a mature field that allows many interesting biological questions to be addressed using only a computer. Insights gained through computational modeling have helped us to better understand proteins and their many important structure–function relationships. While macromolecular modeling has helped enormously to advance basic biology, one of the central justifications for the enormous resources that have gone into this field over the past 30 years is the hope that molecular modeling could, one day, accelerate both drug discovery and drug design [1–3]. Computational drug discovery is a subfield of macromolecular modeling that involves the docking or virtual screening of one or more small-molecule compounds against a chosen protein target.

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_19, © Springer Science+Business Media New York 2015

We distinguish between receptor-based virtual screening and ligand-based virtual screening, of which the latter is a powerful technique to identify new ligands based on a set of existing known ligands. This is, however, outside the scope of this receptor-protein focused book. The small-molecule ligands are called *drug leads* and the protein of interest is called the *drug target*. Both computer-aided docking and receptor-based virtual screening employ a variety of algorithms that allow the small molecule(s) to be rapidly rotated and translated around the protein surface or active site and scored on the basis of their steric fit and/or predicted free energy [4–7]. In more advanced packages the ligand (and even the protein) is allowed to exhibit some conformational flexibility. When an optimal orientation is found or a particularly high scoring molecule is identified, a drug lead or a drug mechanism is said to have been "discovered." The results of these computational experiments are used in an iterative fashion by synthetic organic chemists to help design or select improved lead compounds.

What distinguishes virtual screening from docking is the number of molecules used (screening uses 1,000s, docking uses one), the objective of the search process (screening identifies drug leads, docking identifies active sites or mechanisms), and the robustness or complexity of the docking energy function (docking uses a complex force field, screening does not). There are now many excellent docking and/or virtual screening software packages such as Dock [8], AutoDock [9], Gold [10], Glide [11], and AutoDock Vina [12]. Almost all are freely available. These will be discussed in more detail in the next chapter.

However it is important to remember that before either virtual screening or macromolecular docking can begin, a protein target needs to be identified (and modeled) and a set of potential drug leads needs to be assembled. This chapter describes how both drug targets and drug leads can be identified through several easily accessible Web resources. Specifically we show how putative drug targets for pathogenic viruses or bacteria can be identified directly from their genomic sequences and how the 3D structures of putative drug leads and drug targets can be subsequently extracted from a comprehensive drug and drug-target database called DrugBank [13]. This chapter also illustrates how human drug targets and potential drug leads for prostate cancer can be similarly identified and extracted for docking/screening programs using the Human Metabolome Database (HMDB) [14] and DrugBank. The intent of this chapter is to give readers the necessary input files and know-how to proceed to the next steps (docking and screening) in computational drug discovery.

2 Theory

When medicinal chemists or pharmaceutical scientists think about drugs and drug targets they generally classify them into two separate groups: (1) those that are associated with "endogenous" human diseases and (2) those that are associated infectious or "exogenous" diseases. Endogenous diseases are typically chronic human disorders or conditions that arise due to germ-line mutations (genetic diseases), somatic mutations (cancer), age (atherosclerosis, immune disorders), or some other internal factors. On the other hand, exogenous diseases are typically temporary diseases or conditions that arise from external, nonhuman agents such as viruses, bacteria, fungi, protozoans, or poisonous animals (snakes, insects). The vast majority of drug targets (96 %) and drugs (89 %) are associated with endogenous diseases, while only a tiny minority of drugs targets (4 %) and drugs (11 %) are actually associated with exogenous or infectious diseases [13, 15].

2.1 Identifying Drug Targets and Drug Leads for Exogenous Diseases

The identification of putative drug targets and drug leads for exogenous diseases can take one of two paths, both of which depend substantially on bioinformatics and sequence database comparisons. One can either attempt to identify a completely novel drug target/drug lead or one can attempt to identify a drug target/drug lead that is similar (or even identical) to an existing class of drug targets or drug leads. In both cases, one needs either the complete protein or DNA sequence of the pathogen of interest. Fortunately, with the advent of next-generation DNA sequencing, the entire DNA sequence for hundreds of infectious agents of interest is already known or can be determined in as little as a day.

If one chooses to identify a completely novel drug target or drug lead the task is then to identify those genes or proteins in the genome that are: (1) essential to viability; (2) disease causing; or (3) presented on the surface of the organism. Surface-bound proteins may be identified by sequence analysis by looking for transmembrane segments using such tools as TMHMM [16] or PSORTb [17]. Essential genes, especially for bacteria, may be identified by comparing sequences to existing databases of essential genes such as in the Database of Essential Genes [18]. Likewise disease-causing genes can be identified by comparing sequences between non-pathogenic forms of the microbe with pathogenic forms (say *E. coli* O157 vs. *E. coli* MG1655) or through the identification of pathogenicity islands using tools such as IslandViewer [19]. Alternately essential genes or disease causing genes may be experimentally identified through knock-out mutations or deletions. Generally all viral genes in a viral genome are essential while only 200-300 bacterial genes in a given bacterial genome are essential. Furthermore, among most pathogens, only a small fraction of proteins or genes (<20) are typically disease-causing. Once these "druggable" genes or protein targets are identified, one must select for those that are sufficiently different (<35 % identity) from any human homologues. This prevents any cross-reactivity between the host's proteins and the pathogen's target proteins. It also prevents any potentially adverse drug interactions. After these nonhomologous protein targets are found one can either search/screen for an inhibitory molecule or develop a vaccine (using parts of the surface proteins). When working with completely novel drug targets, it is often difficult to know which lead compounds might work, so widespread chemical library screening is often used.

If one wishes to find matches to an existing class of drug targets or drug leads the task involves identifying those genes or proteins in the genome of the organism that are similar to known drug targets. The underlying assumption is that if a novel virus or a newly identified pathogenic bacterium shares some significant sequence similarity to a protein that is a known drug target from another organism, then the same (or similar) drugs may be used to combat or kill this pathogen. Alternately, these previously known drugs may serve as potential drug leads for further synthetic modification so as to develop more effective therapies for the organism of interest. What is needed for this process to work is a database of known drug target sequences, each of which is linked to a set of associated drugs. Ideally this database should also include the 3D structures (known or predicted) of the drug targets and the drugs themselves. Fortunately such a database exists. It is called DrugBank [13]. DrugBank is a comprehensive drug database or drug encyclopedia that combines detailed drug (i.e., chemical) data with comprehensive drug target (i.e., protein) information. Since it first appeared in 2006 [20], the database has been expanded and updated multiple times [13, 21]. The latest version of DrugBank contains more than 7,300 drug entries including >1,550 FDA approved small molecule and biotech drugs as well as >5,000 experimental drugs. Each compound entry contains detailed structure files in SDF, MOL, and PDB formats. Additionally, nearly 15,000 protein or drug target sequences are linked to these drug entries, many of which have 3D structures or 3D homology models associated with them.

DrugBank supports standard BLAST sequence queries, including appropriately formatted multiple sequence inputs (i.e., complete proteomes). The output from these queries includes the name(s) and hyperlinks to the associated drugs and the 3D structures of the drug targets. Once the drugs are identified, it is possible to use DrugBank again to search for similar drugs (based on structure similarity). The structures of all of these chemical "hits" may be downloaded, either as PDB, MOL, or SDF files. SDF files can be converted to PDB files using the freely available tools MolConverter (ChemAxon), CACTVS [22] or the Cactus

Converter (http://cactus.nci.nih.gov/services/translate/). Thus by using DrugBank it is possible to rapidly obtain 3D structures of putative drug targets and the 3D structures of 100s or even 1,000s of drug leads. These data sets would obviously serve as the basis for docking or virtual screening studies.

2.2 Identifying Drug Targets and Drug Leads for Endogenous Diseases

Identifying drug targets for endogenous diseases is often far more challenging than identifying drug targets for infectious or exogenous diseases. This is because most endogenous human diseases have a complex etiology. With the exception of about 500 [23] relatively rare, monogenic (single gene) disorders, the vast majority of endogenous diseases are multifactorial or partially polygenic (multi-gene) in origin. Nevertheless, with the advent of such techniques as microarray analysis, GWAS (genome wide association studies) or high throughput proteomics, it is now possible to identify large numbers of disease-associated genes relatively rapidly [24]. To date more than 6,500 human disease genes (mutations, polymorphisms, copy number variants) for both monogenic and complex, polygenic diseases have been identified. This information is being catalogued in many online databases such as OMIM [23], the Human Metabolome Database [14], VnD [25], and GeneCards [26]. It is also possible to find disease–gene associations directly through PubMed or other Web servers such as MedGene [27] and PolySearch [28]. A comprehensive list of druggable genes is maintained at the drug–gene interaction database (DGIdb) [29].

Once a list of genes or proteins associated with a given disease is available (along with their sequences) then it is a matter of doing a series of similar kinds of sequence searches against DrugBank as described for Subheading 2.1. However it is also possible to find additional or even novel drug leads by looking through the Human Metabolome Database (HMDB). The HMDB, like DrugBank, is a multipurpose bioinformatics-cheminformatics database containing detailed information about metabolites, their associated enzymes or transporters and their disease-related properties. The utility of the HMDB in drug discovery lies in the fact that most drugs are actually analogs of existing metabolites, cofactors, or signaling molecules. Therefore if one identifies a protein or proteins in a disease-specific pathway that require a certain metabolite or cofactor, then these proteins may prove to be good drug targets and their cofactors or metabolites could prove to be good drug leads. Indeed many inborn errors of metabolism (phenylketonuria, alkaptonuria, and galactosemia) are treated through the addition or removal of metabolites in the diet.

Both DrugBank and the Human Metabolome Database (HMDB) support single and multiple protein sequence queries and both produce results that include the name(s) and hyperlinks to the associated drugs or metabolites and the 3D structures of the corresponding proteins. Once the small molecule leads are identified, it is possible to use DrugBank or HMDB again to search for structurally similar drugs or metabolites. The structures of all these chemical "hits" may be downloaded, either as PDB, MOL, or SDF files. The SDF files can then be converted to PDB files using the freely available tools MolConverter (ChemAxon) or the Cactus Web server. Thus by judiciously using DrugBank and/or HMDB it is possible to rapidly and easily obtain 3D structures of putative drug targets and the 3D structures of numerous drug leads for endogenous diseases.

2.3 Sequence Matching and Chemical Compound Matching

This particular chapter focuses on using two different matching protocols, one for sequence matching and another for chemical structure matching. Sequence matching, or sequence alignment is a central feature to much of bioinformatics while chemical structure matching is a central feature to much of cheminformatics.

Sequence alignment is often based on a technique called dynamic programming. Strictly speaking dynamic programming is an efficient mathematical technique that can be used to find optimal "paths" or routes to multiple destinations or in locating paths that could be combined to score some maximum value. The application of dynamic programming to sequence alignment was first demonstrated more than 40 years ago by Needleman and Wunsch [30]. As these two researchers demonstrated, dynamic programming allows two or more sequences to be efficiently and automatically lined up, permitting gaps to be inserted, extended, or deleted to make an optimal pairwise alignment. In dynamic programming, the two sequences being compared (say sequence A and sequence B) are put on either axis of a table. Sequence A might be on the *X*-axis, while sequence B might be on the *Y*-axis. Each letter in the query sequence is compared to each letter the reference sequence and a number (based on a scoring matrix and a special recursive function) is placed in every box or cell that intersects each pair of letters. Once the table of numbers is filled out, a second stage (called the traceback stage) is undertaken wherein the table is scanned in a diagonal fashion from the lower right to upper left to look for the highest scores. The path that is chosen is actually a series of "maximum" numbers. When all the scores in this optimal path are added together, it gives a quantitative measure of the pairwise sequence similarity while at the same time defining which letters in sequence A should be matched with the letters in sequence B.

Dynamic programming is a relatively slow, memory intensive process. However, it can be sped up considerably. For instance, if look-up tables are used, if advanced statistics are employed, if more than one letter at a time (a "tuple") is scored and if the traceback search is limited to sections close to the diagonal line then you have the essence of the BLAST algorithm [31]. This is the very fast algorithm used to perform most alignments against large sequence databases. It is also the algorithm used in the sequence searches for DrugBank and HMDB.

Chemical compound matching is fundamentally different than sequence matching. Instead of trying to match strings of characters, one tries to match substructures, coordinates or geometric shapes. This is somewhat similar to the idea of structure superposition, which is done with protein structure comparison. However, because the structures of chemical compounds are far more diverse than what is seen for proteins, the structure matching utilities in chemistry have to be slightly more sophisticated. In particular, chemists must use the concept of subgraph isomorphisms [32] and adjacency matrices to identify chemical similarity. For substructure searching the 2D chemical structures of both the query and database compounds must be re-cast as tables that indicate the bond connectivity between each pair of atoms. These tables, which have 1s for connected atoms and 0s for unconnected atoms are called adjacency matrices. The name (adjacency matrix) comes from the fact that they indicate which atoms are adjacent (connected) to each other. Once prepared, the adjacency matrix from the query structure is compared to every adjacency matrix in the database. If substantial sections of the query matrix match to an adjacency matrix (or portion thereof) in the database, then it is likely that the two structures are similar. Different scoring schemes and adjustable threshold cutoffs may be used to distinguish strong matches from weak matches or to identify compounds with particularly important substructures.

As will be seen in the examples to follow, both sequence searching and chemical structure similarity searching can play an important role in drug target and lead compound identification.

3 Methods

For this section we will describe two protocols. One will describe the identification of drug targets and drug leads for a novel retrovirus that exhibits strong similarity to the AIDS virus (HIV) (*see* **Notes 1–8**). The other will describe the identification of drug leads (from a preexisting list of putative drug targets) for prostate cancer.

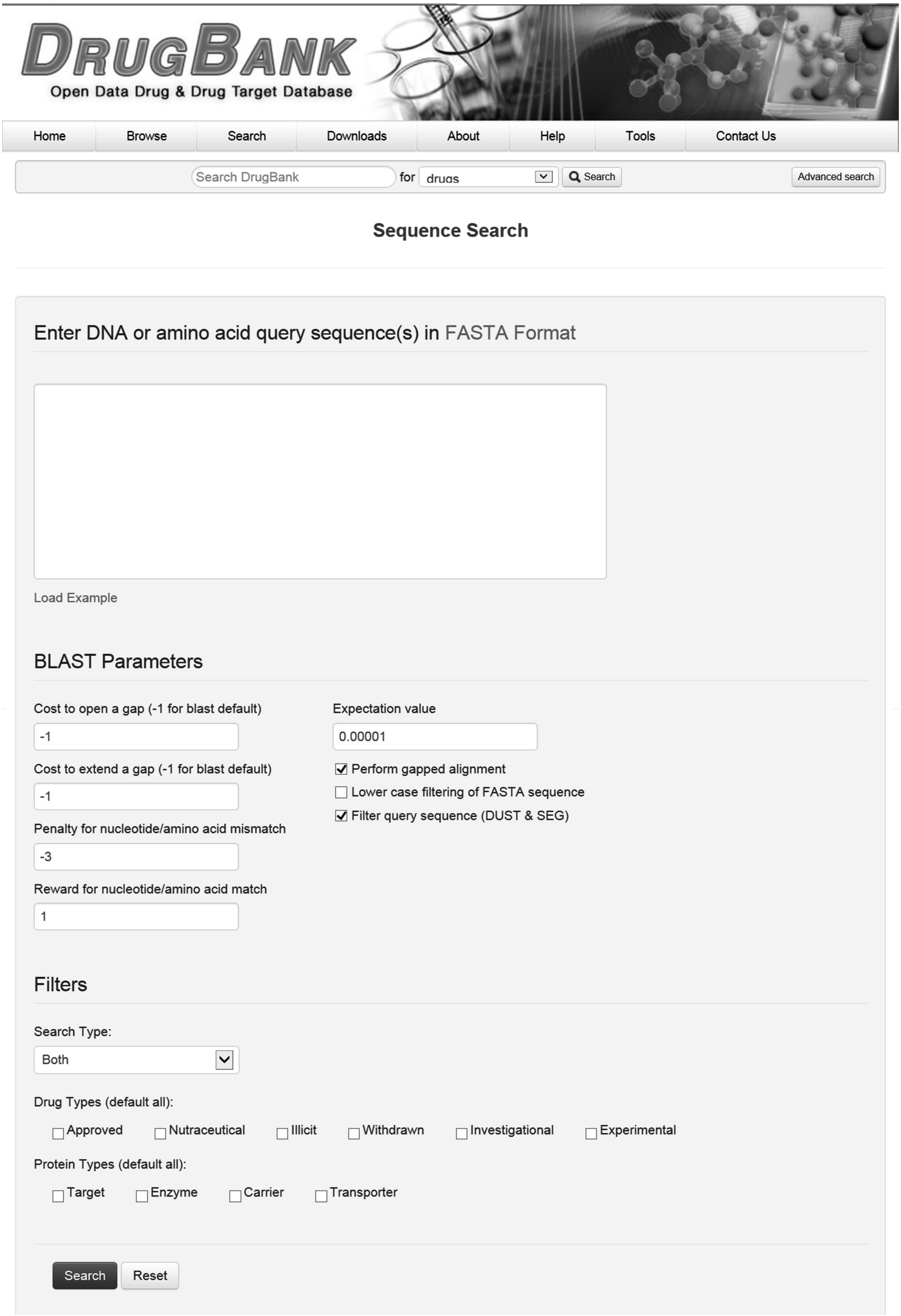
3.1 Identifying Drug Targets and Drug Leads for a Novel Virus

In this example we will use sequence data derived from a recently sequenced, but unnamed virus that exhibits strong sequence similarity to the human immunodeficiency virus (HIV). This particular virus has a total of 15 identifiable open reading frames or polyprotein fragments, which have been fully translated. We will use this sequence information, in combination with DrugBank to identify several drug targets, several drug leads and the necessary coordinate files to conduct rational drug design efforts via docking and virtual screening.

1. Start your local Web browser and go to the DrugBank Web site at http://www.drugbank.ca. The DrugBank homepage should be visible as should the grey menu bar located near the

top of the page with the eight clickable titles **Home, Browse, Search, Downloads, About, Help, Tools, Contact Us**.

2. Click on the **Search** link. A submenu should appear that displays several search options including **ChemQuery, Text Query, Interax Interaction Search, Sequence Search,** and **Data Extractor**. Select the **Sequence Search** option. A window with the title Sequence Search should appear (Fig. 1). As seen in the figure the window contains a standard online BLAST search form with a text box window, with eight different BLASTP parameter settings. There are also options for the **Drug type** and **Database** to be searched, with a variety of options. In almost all cases users can leave everything (except the **Drug type** and **Database** selection) in their default position. A unique feature of the **Sequence Search** program is its capacity to handle multiple FASTA-formatted sequences. This allows users to BLAST multiple sequences—or even entire proteomes.
3. For this example we will be looking for potential drug targets to a newly isolated retrovirus. To obtain the set of sequences to paste into the **Sequence Search** text box, launch a new browser window and go to: http://www.wishartlab.com/molecular-modelingproteins/virus. Click on the **Virus** hyperlink. A list of 15 viral protein sequences should be visible. Select all 15 sequences by clicking a dragging through the window with your mouse. Copy the sequences (using the **Copy** option on your browser or using Ctrl + C).
4. Now click on the **Sequence Search** browser window to activate it and paste the sequences into the **Sequence Search** text box by clicking your mouse in the text box and using the **Paste** option on your browser (or Ctrl + V). You have now pasted 15 different protein sequences from the newly sequenced retrovirus. Use the scroll bars on the right side of the text box to see if all 15 sequences are there (numbered Peptide 1 to Peptide 15).
5. Now select the DrugBank sequence database to search. For this example go down to the bottom of the **Sequence Search** window and select Drug Type "Approved" and Database "Target." This means you will search through all known protein targets of FDA approved drugs. Once this is done, press the **Search** button. Within a few seconds the BLAST search for all 15 input sequences should be completed. The program will return a text-based BLAST summary for each of the 15 proteins that were submitted. The top portion of the **Sequence Search** output consists of a summary of the submitted sequences. Below that is the BLAST result for the first sequence (Peptide 1) listing the E-value, the bit score, the query length, the name of the closest match, and the alignment with the query sequence at the top and the DrugBank database match



Sequence Search

Enter DNA or amino acid query sequence(s) in FASTA Format

BLAST Parameters

Cost to open a gap (-1 for blast default): -1

Cost to extend a gap (-1 for blast default): -1

Penalty for nucleotide/amino acid mismatch: -3

Reward for nucleotide/amino acid match: 1

Expectation value: 0.00001

☑ Perform gapped alignment

□ Lower case filtering of FASTA sequence

☑ Filter query sequence (DUST & SEG)

Filters

Search Type: Both

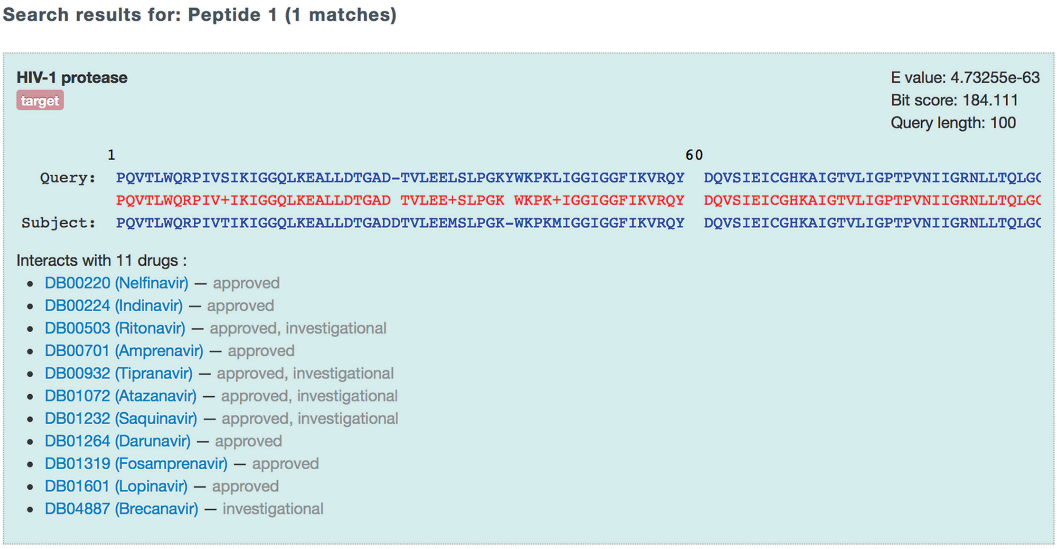
Drug Types (default all):

□ Approved □ Nutraceutical □ Illicit □ Withdrawn □ Investigational □ Experimental

Protein Types (default all):

□ Target □ Enzyme □ Carrier □ Transporter

Fig. 1 Screenshot of the DrugBank BLAST search page



Search results for: Peptide 1 (1 matches)

HIV-1 protease
target

E value: 4.73255e-63
Bit score: 184.111
Query length: 100

```
          1                                                           60
  Query:  PQVTLWQRPIVSIKIGGQLKEALLDTGAD-TVLEELSLPGKYWKPKLIGGIGGFIKVRQY  DQVSIEICGHKAIGTVLIGPTPVNIIGRNLLTQLGC
          PQVTLWQRPIV+IKIGGQLKEALLDTGAD TVLEE+SLPGK WKPK+IGGIGGFIKVRQY  DQVSIEICGHKAIGTVLIGPTPVNIIGRNLLTQLGC
Subject:  PQVTLWQRPIVTIKIGGQLKEALLDTGADDTVLEEMSLPGK-WKPKMIGGIGGFIKVRQY  DQVSIEICGHKAIGTVLIGPTPVNIIGRNLLTQLGC
```

Interacts with 11 drugs :

- DB00220 (Nelfinavir) — approved
- DB00224 (Indinavir) — approved
- DB00503 (Ritonavir) — approved, investigational
- DB00701 (Amprenavir) — approved
- DB00932 (Tipranavir) — approved, investigational
- DB01072 (Atazanavir) — approved, investigational
- DB01232 (Saquinavir) — approved, investigational
- DB01264 (Darunavir) — approved
- DB01319 (Fosamprenavir) — approved
- DB01601 (Lopinavir) — approved
- DB04887 (Brecanavir) — investigational

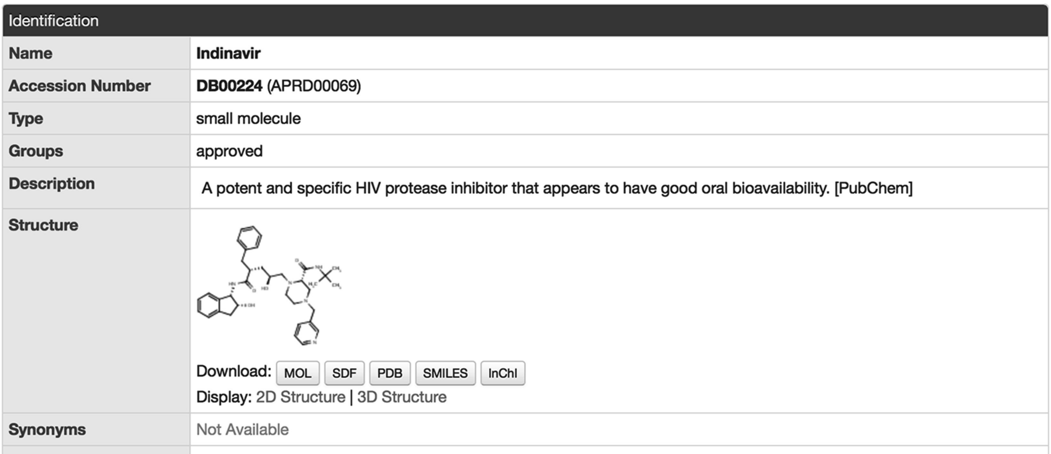
Fig. 2 Screeen shot of the output from the DrugBank BLAST search using the 16 viral protein sequences belonging to a novel retrovirus

below. Matched residues will be displayed in the middle as red letters. Below the alignment is a series of hyperlinks to a number of drug names (*see* Fig. 2). Clicking on any of these drug name hyperlinks will reveal that Peptide 1 may be acted upon by protease inhibitors.

6. For Peptide 1, click on the hyperlink to Indinavir (users may select any one of the many hyperlinks in this list). This should take you to the DrugCard for Indinavir. This page describes the drug and its mode of action in detail and it suggests that Indinavir may be able to target this viral protein target as well.

7. Scroll down further through the **Sequence Search** output page and look for other sequences in this retrovirus that exhibit hits to known DrugBank compounds and for drugs that would be likely to work on these protein targets. In total you should find that there are at least 24 FDA approved drugs for at least four different target proteins in this novel retrovirus.

8. Your task now is to create a library of 3D structures for each of these potential anti-viral drugs. To do so it is necessary to click on each of the drug names and scroll down the DrugCard page that is displayed (Fig. 3). Near the top of each page is a picture of the drug. Below each drug image is a set of hyperlinks indicating Download: MOL, SDF, SMILES, InChI, PDB. Click on the **PDB** link and download the PDB text file of the drug lead (Indinavir in this case). Each of these PDB files was generated



Identification	
Name	**Indinavir**
Accession Number	**DB00224** (APRD00069)
Type	small molecule
Groups	approved
Description	A potent and specific HIV protease inhibitor that appears to have good oral bioavailability. [PubChem]
Structure	Download: MOL SDF PDB SMILES InChI Display: 2D Structure | 3D Structure
Synonyms	Not Available

Fig. 3 A view of the tabular output found in the DrugCard for Indinavir

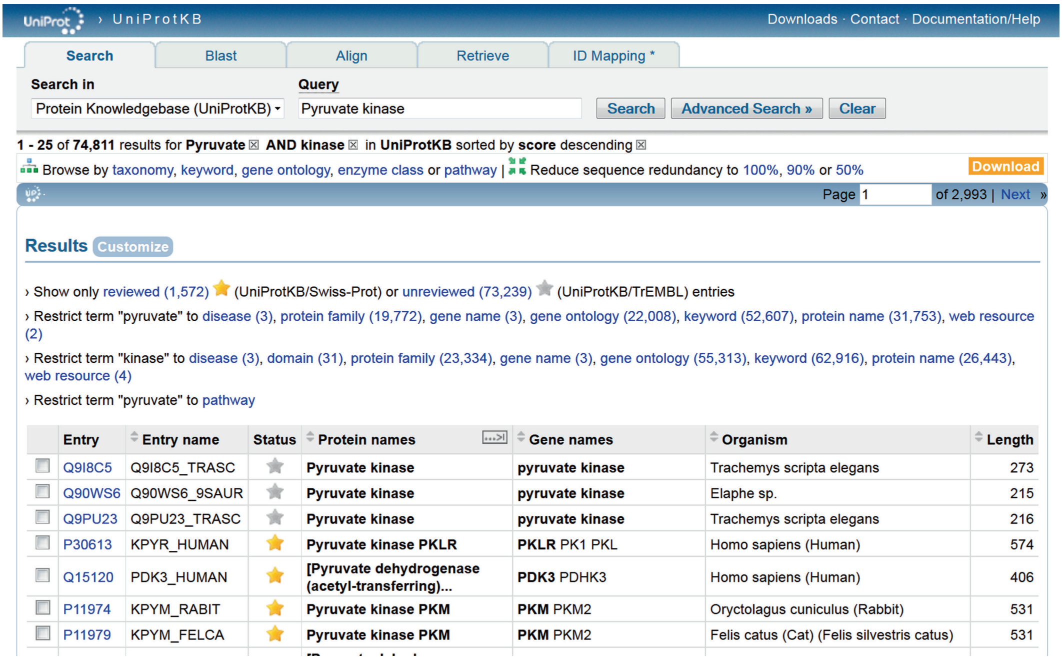
using the MolConverter 3D structure generator. You should repeat this step for all drug structures found in **steps 5**–**7**. You may also obtain additional drug leads and drug structures by going to the top of each DrugCard page and clicking on the button located on the top right corner called **Show Drugs with Similar Structures** for "All" drugs. This will generate a table of chemically similar drugs (including approved, withdrawn and experimental) that may exhibit potential activity against these viral proteins. Download the PDB structures for these compounds as well. You should now have a large collection of PDB files (i.e., 3D structures) of possible drug leads for each of the unique proteins associate with the virus.

9. To perform docking or virtual screening experiments it will be necessary to generate 3D structures of each of the protein targets identified through **steps 5**, **6**. For many of the proteins identified in this exercise it is possible to generate a 3D homology model using Modeller [33], which is a downloadable program or Proteus2 [34] or Swiss-Model (*see* Chapter 16 for more details), which are Web servers. Further information about protein structure modeling is available in Chapters 14 and 15 of this volume. Once you have done this for all the proteins that can be modeled (not all will have 3D homologues) you should have a large collection of PDB files (i.e., 3D structures) of the key drug targets for this virus.
10. Use these two sets of structures (one for the small molecule drug leads, the other for the drug targets) to initiate a virtual screening run or attempt to dock selected compounds into their corresponding protein targets.

3.2 Identifying Drug Targets and Drug Leads for Prostate Cancer

Prostate cancer is the second most common type of cancer in men in North America. It is responsible for more male deaths than any other cancer except lung cancer. It is a disease that generally strikes men over the age of 50, however many factors beyond age, including genetics and diet, have been implicated in its development. In this example we will show how a large list of candidate target proteins can be easily obtained and then quickly reduced. From this list we will show how potential drug candidates or (anti)metabolites may be identified using DrugBank and the HMDB. We will also demonstrate how the necessary coordinate files can be obtained to conduct rational drug design efforts via docking and virtual screening.

1. The first step is to identify a set of disease genes or disease proteins that are upregulated or altered in prostate cancer. This is best done via a literature review. Users could consider using the Prostate Gene Database or PGDB (http://www.urogene.org/pgdb/), a general PubMed literature search, a search through PolySearch [28], data found in the GeneCards database [26] or the NCBI Gene Expression Omnibus [35]. Once an initial set of protein names or gene names has been compiled, one should try to select those proteins that appear to be: (a) enzymes; (b) soluble proteins; (c) able to bind or act upon relatively unique small molecules. The reason for these selection criteria is that if one wants to develop a small molecule drug, the drug target should exhibit some propensity to bind a small molecule. Furthermore, if one wants to perform docking or virtual screening studies, the protein structure needs to be known or at least modeled. Since 99 % of all proteins in the PDB are of soluble proteins or soluble fragments, the need for soluble protein targets is obviously important, although advanced methods for membrane protein structure modeling are covered in this volume. For the purposes of this example, a reasonably good list of candidate protein targets that fit these three criteria is given below:
   (a) Alpha-methyl-acyl-CoA racemase.
   (b) Glyceraldehyde 3-phosphate dehydrogenase.
   (c) Pyruvate kinase dehydrogenase.
   (d) Pyruvate kinase.
   (e) Glycine-*N*-methyl transferase.
   (f) Pipecolic acid oxidase.
   (g) Sarcosine dehydrogenase.
   (h) Hydroxymethylglutaryl-CoA synthase.
   (i) Acetyl-CoA-acetyltransferase.
   (j) 3-Oxo-5-alpha-steroid 4-dehydrogenase 1.
2. To retrieve the protein sequences (which are necessary for the next steps in the analysis) you may start your local Web browser



UniProt › UniProtKB Downloads · Contact · Documentation/Help

Search in: Protein Knowledgebase (UniProtKB)

Query: Pyruvate kinase

1 - 25 of 74,811 results for **Pyruvate** ☒ **AND kinase** ☒ in **UniProtKB** sorted by **score** descending ☒

Browse by taxonomy, keyword, gene ontology, enzyme class or pathway | Reduce sequence redundancy to 100%, 90% or 50%

Page 1 of 2,993 | Next »

Results Customize

› Show only reviewed (1,572) ★ (UniProtKB/Swiss-Prot) or unreviewed (73,239) ☆ (UniProtKB/TrEMBL) entries

› Restrict term "pyruvate" to disease (3), protein family (19,772), gene name (3), gene ontology (22,008), keyword (52,607), protein name (31,753), web resource (2)

› Restrict term "kinase" to disease (3), domain (31), protein family (23,334), gene name (3), gene ontology (55,313), keyword (62,916), protein name (26,443), web resource (4)

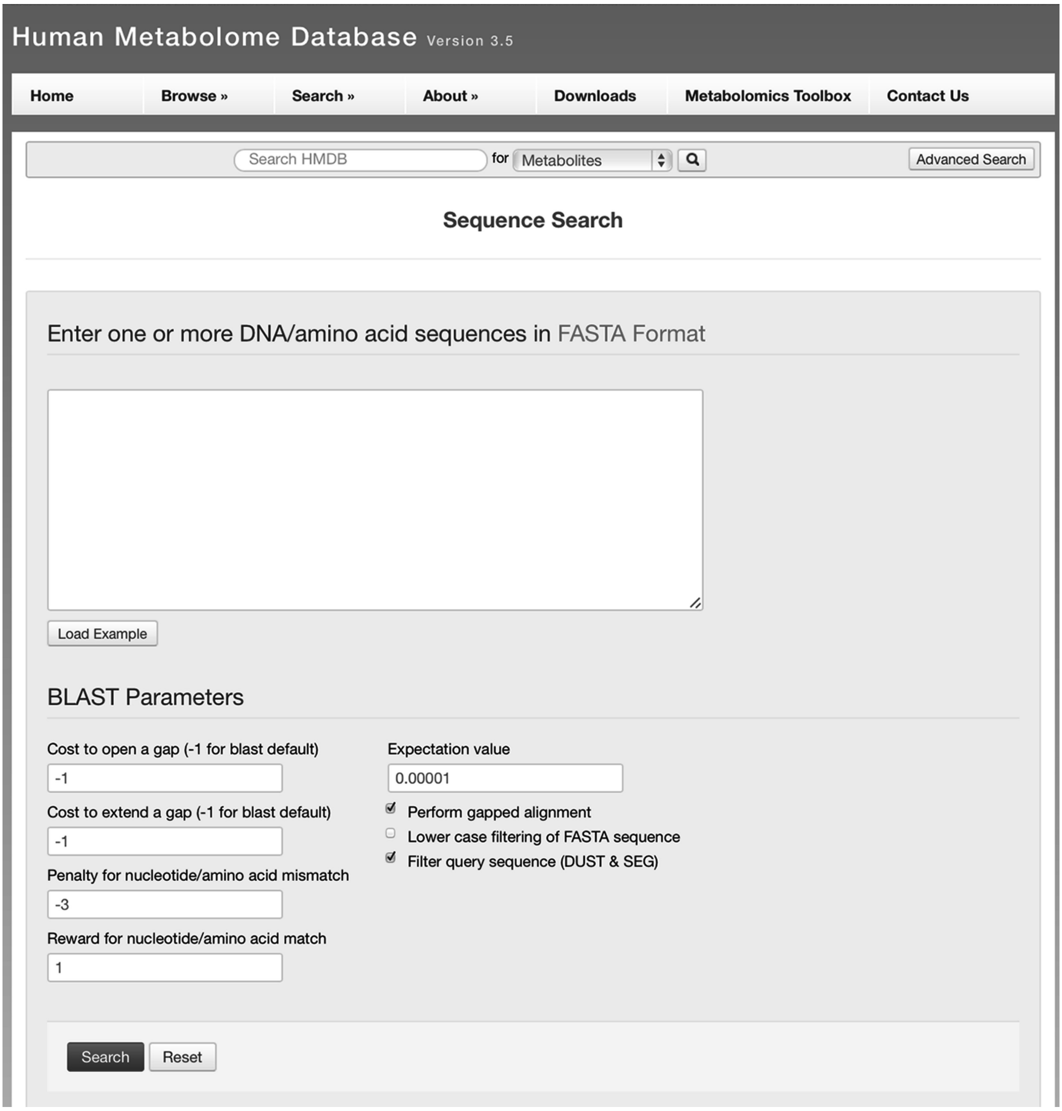
› Restrict term "pyruvate" to pathway

	Entry	Entry name	Status	Protein names	Gene names	Organism	Length
□	Q9I8C5	Q9I8C5_TRASC	☆	**Pyruvate kinase**	**pyruvate kinase**	Trachemys scripta elegans	273
□	Q90WS6	Q90WS6_9SAUR	☆	**Pyruvate kinase**	**pyruvate kinase**	Elaphe sp.	215
□	Q9PU23	Q9PU23_TRASC	☆	**Pyruvate kinase**	**pyruvate kinase**	Trachemys scripta elegans	216
□	P30613	KPYR_HUMAN	★	**Pyruvate kinase PKLR**	**PKLR** PK1 PKL	Homo sapiens (Human)	574
□	Q15120	PDK3_HUMAN	★	**[Pyruvate dehydrogenase (acetyl-transferring)...**	**PDK3** PDHK3	Homo sapiens (Human)	406
□	P11974	KPYM_RABIT	★	**Pyruvate kinase PKM**	**PKM** PKM2	Oryctolagus cuniculus (Rabbit)	531
□	P11979	KPYM_FELCA	★	**Pyruvate kinase PKM**	**PKM** PKM2	Felis catus (Cat) (Felis silvestris catus)	531

Fig. 4 Screenshot of the UniProt databasee search page

and go to the UniProt Web site at http://www.uniprot.org/. In the **Query** box at the top of the page (Fig. 4) type in the name of each protein candidate and press the return key. A list of hits from multiple organisms will appear in a tabular format. Ensure that you select the proteins from *Homo sapiens* only. Click on the corresponding protein name or Uniprot Accession number to open its UniProt protein page. Scroll down the protein page until the Sequence field is reached. A hyperlink with the world "FASTA" should be located just above the sequence. By clicking on this hyperlink it is possible to retrieve a FASTA formatted protein sequence file. This process should be repeated for each protein in the above list. However, to help save time, a FASTA sequence file for all ten proteins is also available for download at http://www.wishartlab.com/molecularmodelingproteins/cancer. To obtain these sequences, click on the **Cancer** hyperlink. Select all ten sequences by clicking a dragging through the window with your mouse. Copy the sequences (using the **Copy** option on your browser or using Ctrl + C) into your computer memory buffer.

3. The next step is aimed at finding metabolites or drugs that may bind, antagonize, inhibit or deactivate these proteins. To find these molecules, launch a new window within your current browser and go to the HMDB Web site at http://www.hmdb.ca. The HMDB homepage should be visible with a simple



Sequence Search

Enter one or more DNA/amino acid sequences in FASTA Format

BLAST Parameters

Cost to open a gap (-1 for blast default): -1

Cost to extend a gap (-1 for blast default): -1

Penalty for nucleotide/amino acid mismatch: -3

Reward for nucleotide/amino acid match: 1

Expectation value: 0.00001

☑ Perform gapped alignment

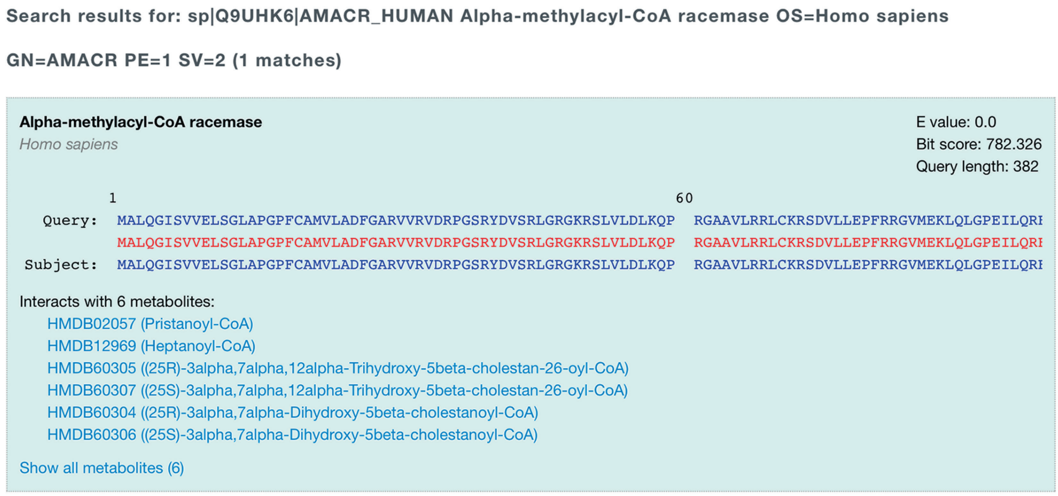
□ Lower case filtering of FASTA sequence

☑ Filter query sequence (DUST & SEG)

Fig. 5 Screenshot of the HMDB BLAST search page

menu bar located near the top of the page with the seven clickable titles **Home, Browse, Search, About, Downloads, Metabolomics Toolbox, and Contact Us**.

4. Click on the **Search** link. A submenu should appear that displays nine different search options including **Chem Query, Text Query, Sequence Search, Advanced Search, etc**. Select the **Sequence Search** option. A window with the title Sequence Search should appear (Fig. 5). As seen in the figure the window contains a standard online BLAST search form with a text box window, with eight different BLASTP parameter settings. A unique feature of the **Sequence Search** program is its capacity



Search results for: sp|Q9UHK6|AMACR_HUMAN Alpha-methylacyl-CoA racemase OS=Homo sapiens GN=AMACR PE=1 SV=2 (1 matches)

Alpha-methylacyl-CoA racemase
Homo sapiens

E value: 0.0
Bit score: 782.326
Query length: 382

```
          1                                                           60
  Query:  MALQGISVVELSGLAPGPFCAMVLADFGARVVRVDRPGSRYDVSRLGRGKRSLVLDLKQP  RGAAVLRRLCKRSDVLLEPFRRGVMEKLQLGPEILQRI
          MALQGISVVELSGLAPGPFCAMVLADFGARVVRVDRPGSRYDVSRLGRGKRSLVLDLKQP  RGAAVLRRLCKRSDVLLEPFRRGVMEKLQLGPEILQRI
Subject:  MALQGISVVELSGLAPGPFCAMVLADFGARVVRVDRPGSRYDVSRLGRGKRSLVLDLKQP  RGAAVLRRLCKRSDVLLEPFRRGVMEKLQLGPEILQRI
```

Interacts with 6 metabolites:

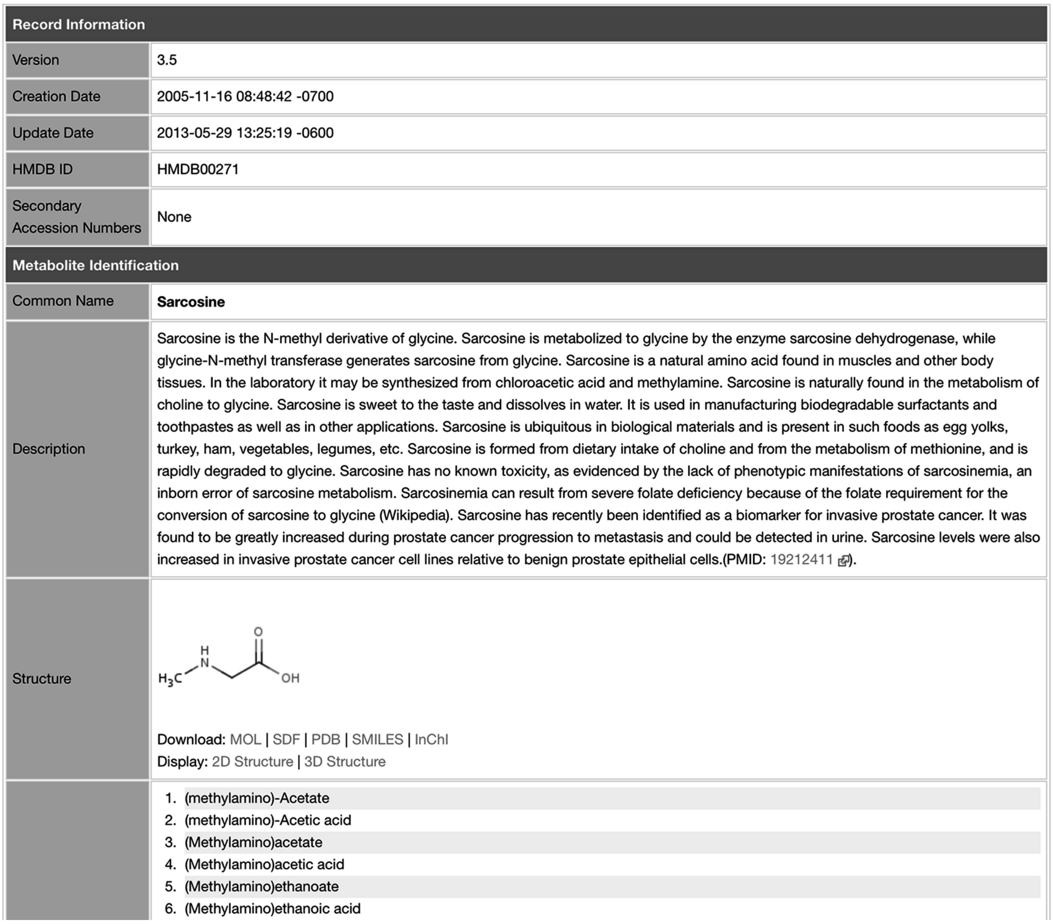
HMDB02057 (Pristanoyl-CoA)
HMDB12969 (Heptanoyl-CoA)
HMDB60305 ((25R)-3alpha,7alpha,12alpha-Trihydroxy-5beta-cholestan-26-oyl-CoA)
HMDB60307 ((25S)-3alpha,7alpha,12alpha-Trihydroxy-5beta-cholestan-26-oyl-CoA)
HMDB60304 ((25R)-3alpha,7alpha-Dihydroxy-5beta-cholestanoyl-CoA)
HMDB60306 ((25S)-3alpha,7alpha-Dihydroxy-5beta-cholestanoyl-CoA)

Show all metabolites (6)

Fig. 6 Screenshot of the output from a BLAST search against the HMDB using the ten protein sequences identified as potential prostate cancer drug targets

to handle multiple FASTA-formatted sequences. This allows users to BLAST multiple sequences—or even entire proteomes.

5. Now click on the **Sequence Search** browser window to activate it and paste the sequences into the **Sequence Search** text box by clicking your mouse in the text box and using the **Paste** option on your browser (or Ctrl + V). You have now pasted ten different protein sequences that are potential drug/metabolite targets. Use the scroll bars on the right side of the text box to see if all ten sequences are there.
6. Once you have confirmed that all ten sequences have been pasted in, press the **Search** button. Within a few seconds the BLAST search for all ten input sequences should be completed. The program will return a text-based BLAST summary for each of the ten proteins that were submitted. The top portion of the **Sequence Search** output consists of a summary of the submitted sequences. Below that is the BLAST result for the first sequence (Alpha-methylacyl-CoA racemase) listing the E-value, the bit score, the query length, the name of the closest match and the alignment with the query sequence at the top and the HMDB database match below. Matched residues will be displayed in the middle as red letters. Below the alignment is a series of hyperlinks to a number of compound names (*see* Fig. 6).
7. Scroll down the list until you see the word "Sarcosine" as one of the metabolites. Click on the word Sarcosine. This should take you to the MetaboCard for Sarcosine. This page describes the metabolite, its structure, its metabolic importance, its metabolic pathways, and the enzymes that act on it.



Record Information	
Version	3.5
Creation Date	2005-11-16 08:48:42 -0700
Update Date	2013-05-29 13:25:19 -0600
HMDB ID	HMDB00271
Secondary Accession Numbers	None
Metabolite Identification	
Common Name	**Sarcosine**
Description	Sarcosine is the N-methyl derivative of glycine. Sarcosine is metabolized to glycine by the enzyme sarcosine dehydrogenase, while glycine-N-methyl transferase generates sarcosine from glycine. Sarcosine is a natural amino acid found in muscles and other body tissues. In the laboratory it may be synthesized from chloroacetic acid and methylamine. Sarcosine is naturally found in the metabolism of choline to glycine. Sarcosine is sweet to the taste and dissolves in water. It is used in manufacturing biodegradable surfactants and toothpastes as well as in other applications. Sarcosine is ubiquitous in biological materials and is present in such foods as egg yolks, turkey, ham, vegetables, legumes, etc. Sarcosine is formed from dietary intake of choline and from the metabolism of methionine, and is rapidly degraded to glycine. Sarcosine has no known toxicity, as evidenced by the lack of phenotypic manifestations of sarcosinemia, an inborn error of sarcosine metabolism. Sarcosinemia can result from severe folate deficiency because of the folate requirement for the conversion of sarcosine to glycine (Wikipedia). Sarcosine has recently been identified as a biomarker for invasive prostate cancer. It was found to be greatly increased during prostate cancer progression to metastasis and could be detected in urine. Sarcosine levels were also increased in invasive prostate cancer cell lines relative to benign prostate epithelial cells.(PMID: 19212411).
Structure	Download: MOL | SDF | PDB | SMILES | InChI Display: 2D Structure | 3D Structure
	1. (methylamino)-Acetate 2. (methylamino)-Acetic acid 3. (Methylamino)acetate 4. (Methylamino)acetic acid 5. (Methylamino)ethanoate 6. (Methylamino)ethanoic acid

Fig. 7 Screenshot of the MetaboCard for Sarcosine. The hyperlinks for the MOL, SDF, and PDB structure files (below the structure) are also visible

8. Scroll down further through the **Sequence Search** output page and look for other sequences that exhibit hits to known human metabolites and for metabolites that would be likely to work on these protein targets. Ideal "lead" metabolites should be larger, polyatomic molecules (not metals) that are nonessential (not ATP). Many of these compounds are substrates or products for enzyme reactions. By overloading an enzyme with a product, it is possible to inhibit its reaction rate. Alternately, by identifying a chemical analog of an enzyme substrate it is possible to completely arrest the activity of the enzyme.

9. Your task now is to create a library of 3D structures for each of these potential anti-cancer drugs. To do so it is necessary to click on each of the metabolite names and scroll down the MetaboCard page that is displayed (Fig. 7). Near the top of each page is a picture of the compound. Below each metabolite

image is a set of hyperlinks indicating Download: MOL, SDF, SMILES, InChI, PDB. Click on the **PDB** link and download the PDB text file of each metabolite of interest. You may also obtain additional drug/metabolite leads and drug structures by going to the top of each MetaboCard page and clicking on the button located on the top right corner called **Show Similar Structures**. This will generate a table of chemically similar compounds that may exhibit potential activity against these proteins. Download the PDB structures for these compounds as well. You should now have a collection of 15–20 PDB files (i.e., 3D structures) of possible drug leads for each of the prostate cancer associated proteins.

10. Users may also want to employ DrugBank (as described in Subheading 3.1) to identify additional drug leads using the ChemQuery tool in this database. Indeed, these efforts would prove to be quite fruitful for this particular example as DrugBank contains a number of well known enzyme antagonists. To generate models for the protein targets, we suggest that users follow **steps 9** and **10**, as described in Subheading 3.1. This will allow them to complete the necessary steps required to set up docking and virtual screening efforts.

4 Notes

1. The examples given in Subheadings 3.1 and 3.2 are realistic but somewhat simplified compared to what might be necessary for "real life" drug discovery. In particular, the identification of drug targets always requires some critical assessment of the utility and viability of the drug target or drug lead. This typically requires a good deal of library research and additional experimentation. For instance, one must determine whether the drug target(s) should be inhibited (therefore requiring an antagonist) or activated (therefore requiring an agonist). As a general rule, the development of antagonists is generally much easier than agonists.
2. It is usually a good idea to determine whether the putative drug target has been previously identified and whether experimental lead compounds have already be explored. Even if a drug target appears viable, one should take particular care to determine if the protein is essential, unique, or conditionally expressed for the associated disease or condition. Nonessential, nonunique, or continuously expressed proteins are generally not good drug targets. Likewise proteins with generally weak affinities (i.e., most carbohydrate binding proteins) or poor turnover rates often turn out to be poor drug targets.
3. The selection of drug leads also requires some careful consideration. While DrugBank, HMDB, and PubChem can

offer many useful suggestions, they are not the only sources for drug leads. Surveys through the literature or careful searches through specialized drug-screening databases can often yield very useful ideas. Once a collection of drug leads has been identified, it is usually prudent to assess the suitability of the compound as a drug. Drug compounds must not be too soluble, too lipophilic, too unstable, or too toxic. These requirements are closely related to their physicochemical properties, which are also related to their Absorption, Distribution, Metabolism, Excretion, and Toxicity (ADMET).

4. ADMET prediction is becoming increasingly common in early-stage drug discovery, drug screening, and drug design. Indeed, many computational chemists would argue that ADMET prediction is something that should *always* be done in the early phases of drug-lead selection. Fortunately there are now a number of software packages, online servers, and standardized rules (Lipinski's rule of five) to determining the likely success or drug-likeness that a compound might have.
5. Among existing tools, AdmetSAR [36] and PreADMET (http://preadmet.bmdrc.org/) probably represent two of the most comprehensive and complete ADMET servers currently available.
6. AdmetSAR is both a server and a database with more than 210,000 literature-derived ADMET data values for nearly 100,000 compounds corresponding to 45 kinds of ADMET-associated properties obtained for different proteins, cell types, and organisms. Through database matches, machine learning classifiers and rule-based regression models derived from its large database and various molecular descriptors, the AdmetSAR server also allows users to predict up to 27 ADMET properties for query compounds. Some of these properties include probabilities for blood–brain barrier penetration, Caco-2 permeability, intestinal absorption, P-gp inhibition/substrate status, CYP isotype inhibitor or substrate status, renal cation transporter substrate status, carcinogenicity, and Ames, fish, or honeybee toxicity. The server accepts SMILES string data as input and rapidly returns a hyperlinked list of values, probabilities, or qualitative classification statements (non-inhibitor, toxic, nontoxic, etc.). Each entry is also hyperlinked to a brief description of the ADMET feature.
7. The PreADMET server (http://preadmet.bmdrc.org/) supports a variety of applications including molecular descriptor calculations (2,000+ values), drug likeness calculations, Caco-2 cell permeability, MDCK cell permeability, human intestinal absorption (HIA), skin permeability, blood–brain barrier permeability, plasma protein binding, Ames toxicity, and rodent

carcinogenicity. The server is nicely designed and provides detailed references and descriptions about the server output and how it should be interpreted.

8. Perhaps the most important point to remember for each of the methods outlined here is that one is generating computer-based predictions. There is no guarantee that any of these predictions (drug targets or drug leads) will turn out to yield a viable therapeutic or even an interesting lead compound. As with any prediction in life science, one must always be prepared to thoroughly test the predictions using in vitro assays and animal models. In many cases the computer predictions will turn out to be wrong. In rare cases, the initial predictions may prove to be quite promising. Nevertheless, the results from any well-constructed wet-bench experiments can and should be used to help guide subsequent steps involved in the computational design, docking, and selection of drug leads.

References

1. Geldenhuys WJ, Gaasch KE, Watson M, Allen DD, Van der Schyf CJ (2006) Optimizing the use of open-source software applications in drug discovery. Drug Discov Today 11:127–132
2. Kirkpatrick DL, Watson S, Ulhaq S (1999) Structure-based drug design: combinatorial chemistry and molecular modeling. Comb Chem High Throughput Screen 2:211–221
3. Wlodawer A, Vondrasek J (1998) Inhibitors of HIV-1 protease: a major success of structure-assisted drug design. Ann Rev Biophys Biomol Struct 27:249–284
4. Mohan V, Gibbs AC, Cummings MD, Jaeger EP, DesJarlais RL (2005) Docking: successes and challenges. Curr Pharm Des 11:323–333
5. Jain AN (2004) Virtual screening in lead discovery and optimization. Curr Opin Drug Discov Devel 7:396–403
6. Sousa SF, Ribeiro AJ, Coimbra JT, Neves RP, Martins SA, Moorthy NS, Fernandes PA, Ramos MJ (2013) Protein-ligand docking in the new millennium – a retrospective of 10 years in the field. Curr Med Chem 20:2296–2314
7. Taylor RD, Jewsbury PJ, Essex JW (2002) A review of protein-small molecule docking methods. J Comput Aided Mol Des 16:151–166
8. Shoichet BK, Kuntz ID (1993) Matching chemistry and shape in molecular docking. Protein Eng 6:723–732
9. Goodsell DS, Morris GM, Olson AJ (1996) Automated docking of flexible ligands: applications of AutoDock. J Mol Recognit 9:1–5
10. Jones G, Willett P, Glen RC, Leach AR, Taylor R (1997) Development and validation of a genetic algorithm for flexible docking. J Mol Biol 267:727–748
11. Friesner RA, Banks JL, Murphy RB, Halgren TA, Klicic JJ, Mainz DT, Repasky MP, Knoll EH, Shelley M, Perry JK, Shaw DE, Francis P, Shenkin PS (2004) Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. J Med Chem 47:1739–1749
12. Trott O, Olson AJ (2010) AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J Comput Chem 31:455–461
13. Knox C, Law V, Jewison T, Liu P, Ly S, Frolkis A, Pon A, Banco K, Mak C, Neveu V, Djoumbou Y, Eisner R, Guo AC, Wishart DS (2011) DrugBank 3.0: a comprehensive resource for "omics" research on drugs. Nucleic Acids Res 39(Database issue):D1035–D1041
14. Wishart DS, Jewison T, Guo AC, Wilson M, Knox C, Liu Y, Djoumbou Y, Mandal R, Aziat F, Dong E, Bouatra S, Sinelnikov I, Arndt D, Xia J, Liu P, Yallou F, Bjorndahl T, Perez-Pineiro R, Eisner R, Allen F, Neveu V, Greiner R, Scalbert A (2013) HMDB 3.0 – the human metabolome database in 2013. Nucleic Acids Res 41(Database issue):D801–D807
15. Sweetman S (2004) Martindale: the complete drug reference, 34th edn. Pharmaceutical Press, New York, NY

16. Krogh A, Larsson B, von Heijne G, Sonnhammer EL (2001) Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. J Mol Biol 305:567–580
17. Yu NY, Wagner JR, Laird MR, Melli G, Rey S, Lo R, Dao P, Sahinalp SC, Ester M, Foster LJ, Brinkman FS (2010) PSORTb 3.0: improved protein subcellular localization prediction with refined localization subcategories and predictive capabilities for all prokaryotes. Bioinformatics 26:1608–1615
18. Zhang R, Lin Y (2009) DEG 5.0, a database of essential genes in both prokaryotes and eukaryotes. Nucleic Acids Res 37(Database issue):D455–D458
19. Langille MG, Brinkman FS (2009) IslandViewer: an integrated interface for computational identification and visualization of genomic islands. Bioinformatics 25:664–665
20. Wishart DS, Knox C, Guo AC, Shrivastava S, Hassanali M, Stothard P, Chang Z, Woolsey J (2006) DrugBank: a comprehensive resource for in silico drug discovery and exploration. Nucleic Acids Res 34(Database issue):D668–D672
21. Wishart DS, Knox C, Guo AC, Cheng D, Shrivastava S, Tzur D, Gautam B, Hassanali M (2008) DrugBank: a knowledgebase for drugs, drug actions and drug targets. Nucleic Acids Res 36(Database issue):D901–D906
22. Ihlenfeldt WD, Voigt JH, Bienfait B, Oellien F, Nicklaus MC (2002) Enhanced CACTVS browser of the Open NCI Database. J Chem Inf Comput Sci 42:46–57
23. Hamosh A, Scott AF, Amberger JS, Bocchini CA, McKusick VA (2005) Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic Acids Res 33(Database issue):D514–D517
24. Wagner MJ (2013) Rare-variant genome-wide association studies: a new frontier in genetic analysis of complex traits. Pharmacogenomics 14:413–424
25. Yang JO, Oh S, Ko G, Park SJ, Kim WY, Lee B, Lee S (2011) VnD: a structure-centric database of disease-related SNPs and drugs. Nucleic Acids Res 39(Database issue):D939–D944
26. Stelzer G, Dalah I, Stein TI, Satanower Y, Rosen N, Nativ N, Oz-Levi D, Olender T, Belinky F, Bahir I, Krug H, Perco P, Mayer B, Kolker E, Safran M, Lancet D (2011) In-silico human genomics with GeneCards. Hum Genomics 5:709–717
27. Hu Y, Hines LM, Weng H, Zuo D, Rivera M, Richardson A, LaBaer J (2003) Analysis of genomic and proteomic data using advanced literature mining. J Proteome Res 2:405–412
28. Cheng D, Knox C, Young N, Stothard P, Damaraju S, Wishart DS (2008) PolySearch: a web-based text mining system for extracting relationships between human diseases, genes, mutations, drugs and metabolites. Nucleic Acids Res 36(Web Server issue):W399–W405
29. Griffith M, Griffith OL, Coffman AC, Weible JV, McMichael JF, Spies NC et al (2013) DGIdb: mining the druggable genome. Nat Methods doi: 10.1038/nmeth.2689. [Epub ahead of print]
30. Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48:443–453
31. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25:3389–3402
32. Ullman JR (1976) An algorithm for sub-graph isomorphism. J ACM 23:31–42
33. Fiser A, Sali A (2003) Modeller: generation and refinement of homology-based protein structure models. Methods Enzymol 374:461–491
34. Montgomerie S, Cruz JA, Shrivastava S, Arndt D, Berjanskii M, Wishart DS (2008) PROTEUS2: a web server for comprehensive protein structure prediction and structure-based annotation. Nucleic Acids Res 36(Web Server issue):W202–W209
35. Barrett T, Wilhite SE, Ledoux P, Evangelista C, Kim IF, Tomashevsky M, Marshall KA, Phillippy KH, Sherman PM, Holko M, Yefanov A, Lee H, Zhang N, Robertson CL, Serova N, Davis S, Soboleva A (2013) NCBI GEO: archive for functional genomics data sets-update. Nucleic Acids Res 41(Database issue):D991–D995
36. Cheng F, Li W, Zhou Y, Shen J, Wu Z, Liu G, Lee PW, Tang Y (2012) admetSAR: a comprehensive source and free tool for assessment of chemical ADMET properties. J Chem Inf Model 52:3099–3105

Chapter 20

Molecular Docking to Flexible Targets

Jesper Sørensen, Özlem Demir, Robert V. Swift, Victoria A. Feher, and Rommie E. Amaro

Abstract

It is widely accepted that protein receptors exist as an ensemble of conformations in solution. How best to incorporate receptor flexibility into virtual screening protocols used for drug discovery remains a significant challenge. Here, stepwise methodologies are described to generate and select relevant protein conformations for virtual screening in the context of the relaxed complex scheme (RCS), to design small molecule libraries for docking, and to perform statistical analyses on the virtual screening results. Methods include equidistant spacing, RMSD-based clustering, and QR factorization protocols for ensemble generation and ROC analysis for ensemble selection.

Key words Relaxed complex scheme, Ligand filtering, Protein flexibility, QR factorization, RMSD-based clustering, ROC analysis

1 Introduction

It is widely accepted that proteins do not exist in solution as a single rigid structure but rather as an ensemble of conformations [1–3]. The atomic fluctuations that give rise to this ensemble range from small rotations of an individual amino acid methyl group to much larger fluctuations concerted between groups of residues and the protein backbone, loops, or domains. The necessity to consider alternate conformations, including subtle structural changes in a binding pocket, is highlighted by the difficulties reported for accurate ranking in cross-docking exercises [4–7]. Put another way, no single structure can represent the binding modes for all the competent inhibitors of a drug target. As computational chemists are often seeking new inhibitors that bind to receptor pockets, these fluctuations need to be accounted for in our computational methods [8–11]. Modeling these protein ensembles thus provide an opportunity and has demonstrated success in discovering novel and/or selective inhibitors that bind to subpockets or alternate conformations not obvious in the "snapshot" of a given crystal structure [12–15].

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4_20, © Springer Science+Business Media New York 2015

A variety of approaches have been used to incorporate protein flexibility into virtual screening (VS) methodologies and have been reviewed extensively elsewhere [16–19]. Briefly, one of the first methods (and now offered by nearly all docking software programs) is the ability to soften the van der Waals potential to allow for some receptor–ligand overlap, the so-called soft docking [20]. While this method does not increase the computational resources needed for virtual screening it has the less desirable characteristic of creating “softness” globally to the binding pocket despite the observation that proteins typically have regions with a range of relative flexibility. Alternatively, docking programs have been developed to include a side chain rotamer library search to introduce residue flexibility [21], a combined induced-fit docking minimization protocol [22, 23], docking followed by rescoring across alternate conformations [24], stochastic Monte Carlo search of side chain and backbone flexibility with a bound ligand [25–27] or cross-over mutations (genetic algorithm) of multiple side-chain and backbone conformations during docking [28]. Ultimately, the use of many representative explicit receptor conformations has the benefit of providing additional experimental or physical information despite the increased time to dock to multiple versions of the receptor structure.

Generation of the protein ensemble can be achieved through the use of experimental data, namely, multiple co-crystal structures, an NMR structure ensemble or through computational methods that explore a protein’s conformational energy landscape such as elastic network normal mode analysis [29, 30], Monte Carlo simulations [25], and molecular dynamic (MD) simulations [9, 31].

In this chapter we focus on the Relaxed Complex Scheme (RCS), which is a strategy that utilizes a physics-based molecular dynamics methodology to generate an ensemble of receptor structures and then exploits that predictive, simulation-based structural knowledge in the discovery and design of small molecule compounds [9, 32, 33]. This method has proven successful for the target, *Trypanosoma brucei* RNA-editing ligase 1 (TbREL1 [34]), investigated here, and other relevant drug targets [12, 14, 15, 35–40]. The advantage of this physics-based approach is that it allows for larger scale and concerted conformational changes to be captured, and it offers the potential to understand the ligand induced structural changes.

TbREL1 is part of the RNA editing complex for the parasite *Trypanosoma brucei*, a causative agent in African Sleeping Sickness. The role of TbREL1 is the catalytic ligation of two RNA molecules driven by the hydrolysis of ATP [34]. TbREL1 is critical for the survival of the parasite, which makes it a promising target.

2 Materials

Software used:
Classical and accelerated MD simulations were performed with NAMD 2.9 [41, 42], using the Amber99SB force field [43]. The docking programs used are Schrödinger Glide v. 6.0, 2013 with the SP scoring function (Schrödinger, LLC, New York, New York, www.schrodinger.com) [44, 45] and Autodock Vina version 1.1.2 (http://vina.scripps.edu) [46]. Statistics calculations were performed in MATLAB version R2011b 7.13.0.564 (MathWorks, Natick, MA, www.mathworks.com/matlab). Figures of the protein conformations were produced using VMD 1.9.1 [47].

Ensemble docking starting materials are

1. A crystal structure, an NMR ensemble, or a homology model used for MD simulation.
2. A set of ligand files formatted properly for the docking program used.
3. The docking program.

In this case, the single 1.20 Å resolution crystal structure for TbREL1 (PDB ID: 1XDN) [48] was used.

Multiple ligand files were compiled for docking; 121 and 40 known binders for TbREL1 from the Drug Discovery Unit (DDU) Diversity compounds and Kinase set [49], respectively, the ATP ligand extracted from the TbREL1 co-crystal structure [48] and additional known binders found in previous virtual screening efforts for TbREL1 [39, 40]. We have used these known actives to generate a set of nonbinders (decoys) using DUD-E [50].

3 Methods

Here, we outline the steps in selecting a representative receptor ensemble for docking from a dataset of molecular dynamics trajectories using the RCS, the preparation of the ligand files for docking, and the statistical analysis of the virtual screening results. Virtual screening efforts incorporating receptor flexibility have previously been reported for TbREL1 [39, 40, 51]. Discovery of inhibitors towards RNA editing enzymes in trypanosomatid pathogens has also been reviewed recently [52].

3.1 Generating a Conformational Ensemble

A set of receptor structure coordinates is a prerequisite to generating an ensemble and can be derived from X-ray crystallography or NMR spectroscopy. If no receptor structure is available for the target, a homology model based on a structure of a related protein can be used, preferably with a high sequence identity [53–56]. In the event that several crystal structures of the biomolecular target are available, these should be incorporated, as they will often represent different conformations of the protein. To generate a diverse conformational ensemble we used two sets of initial structure coordinates, a set where ATP and the magnesium ion are retained, and a set where they were deleted. Typically, using a crystal structure determined without a ligand is not advised unless there are no liganded examples [57]. We have used only the simulations containing ATP in the VS, as the simulations without ATP bound show partial occlusion of the binding pocket.

We performed both conventional and accelerated [58, 59] MD simulations using NAMD v2.9 [41] with the AMBER force fields [43] described in our recent study of the TbREL1–RNA complex [60]. Minor modifications to the force field employed a new magnesium ion model [61] to more accurately capture the dynamics around the ion (*see* **Note 1**) and a more recent water model, TIP4P-ew [62]. Detailed accounts of the required preparation prior to MD simulations have previously been described elsewhere for this system [63, 64]. Moreover, protein preparation in general has recently been reviewed [65]. Inclusion or exclusion of various crystallographic waters, ligands, and binding site ions or co-factors are among the details to be considered.

Simulation lengths will vary based on the biomolecular target and the level of conformational change is a factor to take into account. The same is true for whether to perform conventional or accelerated MD (*see* **Note 2**) since the latter can be used to enhance conformational sampling. Current typical simulation lengths vary from tens to thousands of nanoseconds, where snapshots are extracted at regularly spaced intervals (*see* **Note 3**).

3.2 Filtering the Conformational Ensemble into a Meaningful Subset

The MD simulations, depending on the simulation length, will result in tens to hundreds of thousands of snapshots of the biomolecule sampled in different conformations; in our example we have generated a total of 60,000 snapshots. It is not feasible to dock a (large) library of ligands into each and every one of these conformations, and recent studies where this has been attempted have shown that it is not necessarily an advantage to include a higher number of conformations of the biomolecular target [31, 66]. The goal then is to extract meaningful structures that will enrich the predictive power of the virtual screen.

This reduction can be achieved using a number of different algorithms, including QR factorization [39, 67], atomic or residue based root-mean-square-deviation (RMSD) clustering [68, 69], and active-site shape-based methods [70–74]. We outline here the steps for creating ensembles by: (1) equidistant frame samples, (2) RMSD-based clustering, and (3) QR factorization.

The simplest approach is to extract a predetermined number of structures with regular time intervals (equidistant spacing) between them from the MD simulations (*see* **Note 4**). In our example, the structures have been extracted using ptraj [75] in AmberTools 13 [76–78] with a spacing of every 10 ns. We executed ptraj with amber parameter file (prmtop) and a script that reads a number of commands for ptraj. We specified for ptraj to read in (using the *trajin* command) frames 1–10,000 (the number of frames in the simulated trajectory), but only to read every 1,000th frame and output these, resulting in ptraj outputting ten frames (using the *trajout* command), with an equidistant spacing of 10 ns. The advantage of *ptraj* is that we can specify it to align the protein structures to a reference structure, i.e., the crystal structure, using the *rms* command. In the input file below for ptraj, the file system.inpcrd contains the crystal structure conformation, which is loaded in and used as a reference structure for aligning. The output pdb files are used for docking.

```
ptraj system.prmtop < extract.script

      reference system.inpcrd

      trajin   trajectory.dcd 1 10000 1000

      rms reference

:7@CA,:9@CA,:35@CA,:36@CA,:41@CA,:60@CA,:108@CA,:158@CA,:159@CA,:232@CA

,:235@CA,:237@CA,:241@CA,:256@CA,:258@CA

      trajout extracted.pdb pdb
```

(Script 1)

RMSD-based clustering is fairly common [68, 69], yet has a number of variables to be determined by the user, which should be chosen based on the problem at hand. There are too many variations to outline here, instead we refer the reader to an excellent paper reviewing the possibilities [69] and provide a simple example of a commonly used method. When developing an ensemble for virtual screening, in which the goal is to capture the most diverse conformations of the active site, the RMSD calculation should be performed with a small set of atoms or residues that line the active site or are within a certain distance from the ligand (if one is bound in the protein). However, if one is interested in larger scale conformational changes as one may encounter with large loops near an active or allosteric ligand site, then the protein backbone atoms are most likely a better selection. Here we have chosen the former approach (*see* **Note 5** for the residue selection). The RMSD-based clustering was performed in ptraj [75], although several other programs are available for this task. As a first step, the trajectory snapshots were aligned to the crystal structure conformation based on the same residue selection, but only taking the backbone CA atoms into account. The remaining variables refer to the different variations of RMSD-based clustering, which have been described in great detail by Shao et al. [69]. In ptraj we have used the *cluster* command, employing the average-linkage algorithm, requesting ten clusters based on pairwise RMSD. The average-linkage clustering refers to merging of structures into clusters; initially each structure is assigned to its own cluster, the distance between each cluster is then calculated and the clusters with the shortest distance are merged iteratively until the target number of clusters is reached; when several protein conformations are part of a cluster it is then the "average" distance of each conformation in that cluster to other clusters that are used as the distance metric. We had ptraj print out the "average" and the most representative structures in pdb format for each cluster. For VS we have used the representative structure from each cluster. Note that we are only reading in every 5th frame from the simulation; RMSD-based clustering is time consuming and the time scales with N^2, with N representing the number of frames used.

```
ptraj system.prmtop < extract.script

      reference system.inpcrd

      trajin      trajectory.dcd 1 10000 5

      rms reference

:7@CA,:9@CA,:35@CA,:36@CA,:41@CA,:60@CA,:108@CA,:158@CA,:159@CA,:232@CA

,:235@CA,:237@CA,:241@CA,:256@CA,:258@CA

      cluster out Asitecluster representative pdb average pdb

averagelinkage clusters 10 rms

:7,9,35,36,41,60,108,158,159,232,235,237,241,256,258
```

(Script 2)

For the classical MD simulations, we have excluded clusters with populations lower than 5 % of the trajectory, although in some cases lowly populated states may also be viable for discovery. This results in seven clusters. For the accelerated MD simulations, we have chosen to keep all ten clusters, because we expect a much better conformational sampling, while not necessarily visiting the same conformation as often as in the case of conventional MD.

An alternative ensemble clustering methodology can also be performed using QR factorization which enables one to efficiently reduce the number of MD snapshots to a minimal set without compromising the loss of diversity in the geometric characteristics of the binding pocket [39]. QR factorization is a mathematical technique that performs repeated Householder transformations with column pivoting to reorder the ensemble of structures such that they are arranged with increasing linear dependence. The steps required for preprocessing the MD trajectory files for QR factorization are provided in a tutorial at the NBCR Web site listed below.

`http://nbcr.ucsd.edu/wiki/index.php/SI2011_track3_CADD_QR_factorization_tutorial`

The processed files can then be submitted to the publicly available server on the same Web site listed below.

`http://nbcr-222.ucsd.edu/opal2/CreateSubmissionForm.do?serviceURL=http://localhost:8080/opal2/services%2Ftrajqr_1.0`

Structure extraction techniques based on the shape and chemical properties of the active site are also emerging [70–74, 79, 80], but not used here (*see* **Note 6**). Furthermore, structural water molecules in the active site should be considered (*see* **Note 7**).

The final protein conformations used for VS were extracted using ptraj as detailed above in scripts 1 and 2 and output in the pdb format. The pdb files were then converted to the pdbqt format for Autodock Vina (*see* **Note 8**). The active site center was defined by X, Y, and Z coordinates using a fixed square box to enclose the active site (*see* **Note 9**). For Glide docking a receptor grid file was generated using the XGlide script provided by Schrödinger (*see* **Note 10**). The PDB files were used as input. In total 25 protein conformations were used for docking: eight different setups of the crystal structure, varying inclusion of structural water molecules, and 17 structures from ATP-bound simulations seven of which were extracted by RMS clustering of conventional MD trajectories and the remaining ten extracted from accelerated MD trajectories. We first explored which crystal structure setup was best able to discriminate binders from nonbinders and then added this one setup to the 17 clusters from MD. Thus, 18 protein conformations were included for ensemble statistics, resulting in the evaluation of 262,143 different ensembles.

3.3 Ligand Library Construction and Preparation for Docking

Assembly of a high-quality compound collection for virtual screening should be developed with the target of interest in mind. In the TbREL1 case, the DDU Kinase set and part of their Diversity set were utilized [49]. This collection of compounds was developed specifically for screening against a diverse set of novel parasitic enzyme targets and kinase homologues to human kinases [81], however, the protocol used incorporated many best practice criteria that are general to any target-focused library development [82–86]. A general workflow, loosely based on the DDU methodology is shown in Fig. 1. This workflow starts by culling compounds from commercial suppliers, followed by extensive filtering, clustering and visualization steps. Filtering first removes redundant compounds and subsequent steps can be added to remove unwanted compound functionalities, such as those with reactive groups, compounds with low functional complexity, compounds without lead-like chemical properties and known aggregators. These protocols can be accomplished using OpenEye's ToolKit as described in [81] or with alternate software (*see* **Note11**). However, a method for the filtering steps is provided by the OpenEye's Filter tool (Santa Fe, NM. www.eyesopen.com).

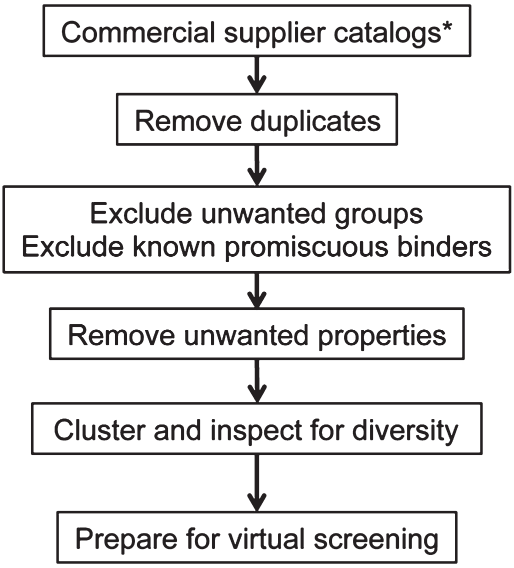

Commercial supplier catalogs*
Remove duplicates
Exclude unwanted groups
Exclude known promiscuous binders
Remove unwanted properties
Cluster and inspect for diversity
Prepare for virtual screening

Fig. 1 A general workflow to generate a compound library for virtual screening. *For compound sources *see* **Note 11**

This program provides a default filter file named "lead" containing many commonly accepted best practices for pre-filtering compounds used in a virtual screen or it can be modified by the user to use a custom set of rules specific to a dataset or target.

```
    >OpenEye/bin/filter –in inputfilename.sdf –filter
lead –prefix clean –fail failed –out outputfilename
```

Clustering of the compounds remaining after filtering can be performed to: (1) further assist in visualizing groups of your selections, (2) understand the relative diversity of the scaffold types and functionalization, and (3) easily select a set by retaining only cluster centroids or another selection criterion. In the DDU Dundee Diversity set, the Jarvis–Patrick algorithm within the Daylight cluster package (Daylight, Aliso Viejo) was used to give clusters containing at least nine neighbors, while removing singletons and other compounds within a cluster having a Tanimoto coefficient >0.9 to the centroid [81].

Decoy molecules were generated with DUD-E [50]. DUD-E decoys were pulled from the ZINC45 database [87] based on matching physicochemical properties to the known binders, such as molecular weight, estimated water–octanol partition coefficient (miLogP), rotatable bonds, number of hydrogen bond acceptors and donors, and net charge. Moreover, the decoys were selected to be topographically dissimilar. DUD-E aims to return 50 unique decoys per input known binder. The input for DUD-E are SMILES strings of the molecules, which we generated using Schrodinger's Maestro software.

2D depictions of the compounds in either .sdf or .mae format were subsequently converted into the 3D coordinates docking format with Schrodinger's LigPrep module and the following GUI settings:

```
Force field: OPLS 2005

Generate possible states at target pH: 7.0 +/- 2.0, Epik

Desalt: On

Generate Tautomers: On

Stereoisomers:

Retain specified chiralities (vary other chiralities)

Generate at most 2 per ligand

Generate at most 2 ring conformations per ligand
```

Tautomers with predicted probability <25 % were removed manually.

The Schrodinger LigPrep's output .mae file format was used directly as input for Schrodinger's Glide docking protocol, else exported as .mol2 format and converted to .pdbqt format for Autodock Vina (*see* **Note 12**).

3.4 Docking Protocols

Docking protocols typically do not differ significantly whether including receptor ensembles or a single receptor example, but they do usually extend the time-to-completion of the VS in real-time since the number of receptor structures to dock into increases. Modifications that enhance the speed or parallelization of the protocol are therefore broadly useful. Time-to-completion may be an important consideration when selecting a docking program for the VS. However, it is also known that not all docking programs perform well on all types of protein receptors [4, 88]. Therefore, selection of the docking program may depend more on the results of customary validation, namely, re-docking of the co-crystallized ligand to the original crystal structure. As many co-crystallized compounds as possible should be re-docked with an RMSD value below 2 Å, and a favorable interaction energy during the validation process.

In previous screens we used Autodock4 for docking, which successfully predicted compounds with inhibitory activity toward TbREL1 [39, 40]. However, for this study in which a larger number of protein receptor conformations and a larger library of ligands were employed, we have used more efficient docking programs, namely, Autodock Vina and Glide from Schrödinger (*see* **Note 13**). The re-docking was performed with Glide and the RSMD of the predicted pose of ATP compared to the crystal structure was 0.30 Å, and the binding energy was very favorable. Furthermore, poses of the inhibitors found in our previous studies with Autodock4 were reproduced by Glide (data not shown).

3.5 Generating Docking Statistics

When one has a set of known binders and known nonbinders for a target, statistical methods can be used to assist in the selection of ensemble conformations to include for the best virtual screening performance. Although there are various VS performance metrics available in the literature (*see* **Note 14**), the area under the curve (AUC) of the Receiver Operating Characteristic (ROC) plot is one of the most popular performance evaluation metrics, and is utilized in this study. AUC values are also used to compare docking protocols as well as different ensembles of receptor conformations that exhibit the best performance for a system of interest.

VS essentially reduces to a binary classification problem that tries to discriminate compounds as either binders or nonbinders. The predicted binding affinities or binding free energies generated for each compound in a VS experiment are used to rank the compounds, and the compounds with higher ranks are more likely to be experimentally assayed. In a VS experiment in which the binders and nonbinders are known, the probability distribution function (PDF) of predicted binding affinity of binders and nonbinders can be constructed separately (Fig. 2a). This allows us to choose a threshold docking score that we would use to pick compounds for testing. Integrating the PDF curve of binders from negative infinity to the chosen threshold value gives us the "true positive rate" (TPR), (*see* **Note 15**). Integrating the PDF curve for nonbinders from negative infinity to the chosen threshold value gives us the "false positive rate" (FPR) (*see* **Note 16**). Continuously evaluating TPR and FPR at each threshold value and plotting TPR versus FPR constructs a ROC plot (Fig. 2b). A docking protocol that yields a higher TPR value at the same FPR value compared to another protocol has a better chance of discovering larger number of hits using the same threshold value (*see* **Note 17**).

The area under the curve (AUC) of the ROC plot is a significant measure of performance, and it represents the probability that a randomly selected active will have a higher rank than a randomly selected inactive [89, 90]. The TPR and FPR can take on values

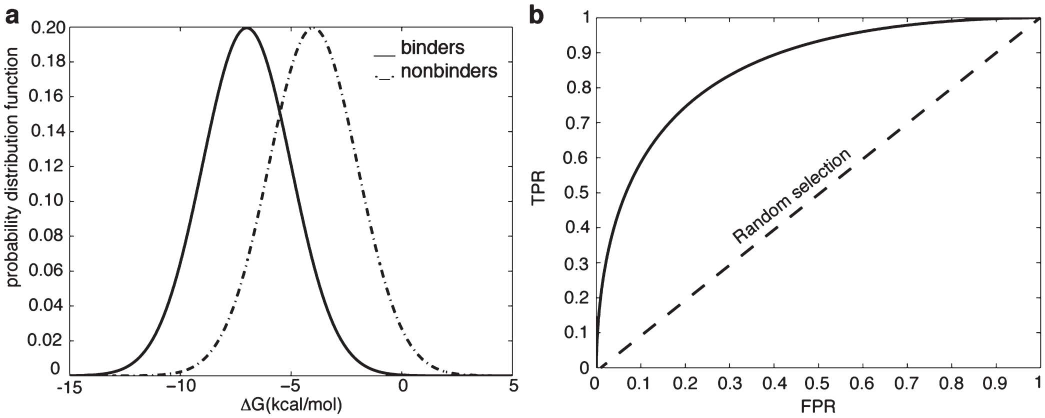


Fig. 2 (**a**) Probability distribution function of binding free energies of binders and nonbinders in a virtual screening experiment represented with the *solid* and *dashed lines*, respectively. (**b**) ROC plot corresponding to the virtual screening in panel **a**. ROC plot for a random selection is also depicted with a *dashed line* for comparison

ranging from 0 to 1. Thus, the maximum AUC value possible is 1 (*see* **Note 18**) corresponding to perfect discriminatory power of a VS protocol. The higher the AUC value, the better the VS performance. When reporting the AUC value for a VS experiment, a good practice is to report its 95 % confidence intervals as well. If virtual screens are performed repeatedly on different databases, the mean AUC value will be found in the range of the confidence intervals 95 % of the time. In practice, 95 % confidence intervals can be calculated using the formula below:

$$\mathrm{CI}_{95\%} = \sqrt{}\left(2 \times \mathrm{SE}_{\mathrm{calculated}} \times \mathrm{erfinv}\left(0.95\right)\right)$$

in which erfinv is the inverse error function and the standard error calculated using the formula [91, 92]:

$$\mathrm{SE}_{\mathrm{calculated}} = \sqrt{}\left(\left(\sigma^2_{\mathrm{NB}}\left(\mathrm{AUC}\right) / N_{\mathrm{NB}}\right) + \left(\sigma^2_{\mathrm{B}}\left(\mathrm{AUC}\right) / N_{\mathrm{B}}\right)\right)$$

in which N_{NB} and N_{B} are number of nonbinders and binders, respectively.

In the case of a random selection of compounds, the PDF curve for binders and nonbinders will be identical resulting in identical TPR and FPR values for any threshold value. In such a case, the plot of TPR versus FPR, or the ROC plot, will be a straight line bisecting the unit square (the dashed line in Fig. 2b), and the AUC will be equal to 0.5. A VS protocol should yield an AUC value better than 0.5 to demonstrate discriminatory power for binders and nonbinders. However, more effort is needed in order to determine whether a VS protocol performs better than random selection in a statistically significant way.

It is illustrative to consider an example. In the case when an AUC value of 0.7 and a standard error of 0.1 is obtained for a particular VS protocol, a PDF curve can be constructed as in Fig. 3a (solid-line). There is still a slight probability that this particular protocol performs randomly on average, but had a high performance for this instance. How large is this chance? To answer this question, first, we make use of the null-hypothesis, "our protocol performs randomly" and construct the null distribution, or the distribution of AUC values that corresponds to a protocol that performs randomly on average. Assuming the number of actives and inactives is large enough to justify use of the central limit theorem, the null distribution will be a Gaussian, centered on a mean AUC of 0.5 as depicted in the dashed-line curve in Fig. 3a according to the central limit theorem (*see* **Note 19**). Now, the probability of obtaining an AUC of 0.7, assuming the null hypothesis is true, is just the area under the null distribution for AUC values greater than or equal to 0.7. And this probability is called a *p*-value. The smaller the *p*-value, the less likely it is that the null-hypothesis is true. Generally, if the *p*-value is less than or equal to 5 % or 1 % [92], the null hypothesis is rejected and the VS protocol is deemed to perform statistically better than random. Once the null hypothesis

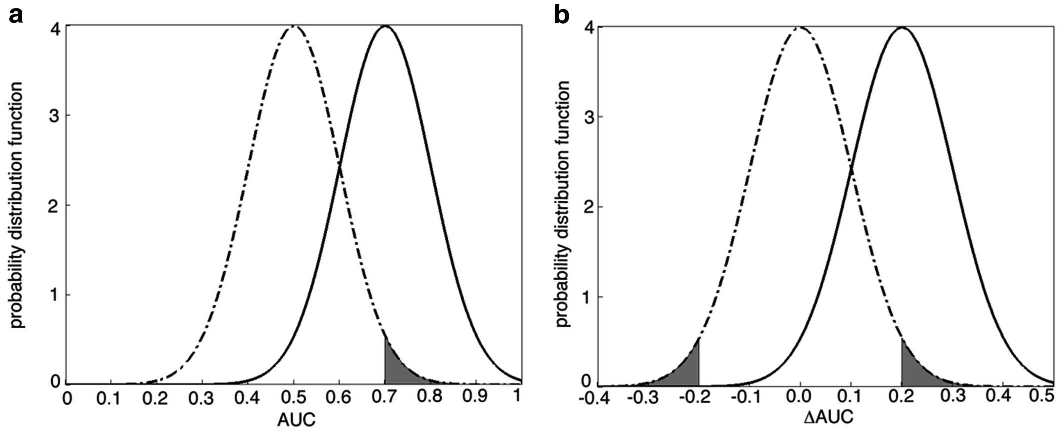


Fig. 3 (**a**) Probability distribution function of AUC for a real virtual screening protocol with a mean value of 0.7 and a standard deviation of 0.1 (depicted with a *solid-line* curve), and for a random selection with a mean value of 0.5 and a standard deviation of 0.1 (depicted with a *dashed-line* curve). The shaded area corresponds to the *p*-value to evaluate whether the real virtual screening protocol performs better than random. (**b**) Probability distribution function of ΔAUC for a set of two real virtual screening experiments with a mean value of 0.2 and a standard deviation of 0.1 (depicted with a solid-line curve), and for a set of two identical virtual screening experiments with a mean value of 0.0 and a standard deviation of 0.1 (depicted with a *dashed-line* curve). The shaded area corresponds to the *p*-value to evaluate whether the two real virtual screening experiments perform identically

is rejected, it can be replaced with the alternative hypothesis, which assumes that on average, the protocol does in fact perform better than random. Generally, the alternative hypothesis is centered on a mean identical to the observed performance and can now be used to estimate confidence intervals.

In order to determine whether one of the two VS protocols performs better in a statistically significant way, the null hypothesis of "the two VS methods perform identically" must be evaluated [91]. The PDF curve of ΔAUC for this null hypothesis (the dashed-line curve in Fig. 3b) will be centered at 0.0, and will have the same standard deviation as the alternative curve. Assuming the difference in the mean AUC values for the two VS protocols is 0.2, the probability distribution of the alternative curve (the solid-line curve in Fig. 3b) will be centered at 0.2 with a standard deviation calculated using the standard deviations of the two protocols (*see* **Note 20**). In this case, the *p*-value will be equal to the area under the curve of the null distribution for |ΔAUC|>=0.2 (the shaded areas in Fig. 3b). If the *p*-value is too small and rejected, then statistically one protocol is deemed to perform better than the other protocol.

In practice, one needs to evaluate all possible combinations of receptor conformations to distinguish which ensemble of receptor conformations among the N conformations predicts the true binders best. The following Matlab scripts can be utilized to monitor performance of all possible ensembles of conformations. The scripts require an input matrix "total" in which the first column has ligand identifiers, the second column has either 0 or 1 for nonbinders and binders, respectively, followed by columns containing the docking scores for each receptor conformation (also *see* **Note 21**).

```
run_best4allensembles.m
%Calls the function best.m for all possible ensemble sizes of k in
%range 0 < k <= N and 95% confidence intervals and outputs AUC values,
%confidence intervals, p-values and identity of receptor conformations
%in the current ensemble into files called AUCk.dat, CLk.dat, pk,dat
%and Ck.dat
load total.csv
   for k=1:N
   [AUC,CL,p,C]=best(total,k,0.05)
    filename1=sprintf(‘AUC%d.dat’,k);
    filename2=sprintf(‘C%d.dat’,k);
    filename3=sprintf(‘CL%d.dat’,k);
    filename4=sprintf(‘p%.dat’,k);
    save(filename1,’AUC’,’-ascii’);
    save(filename2,’C’,’-ascii’);
    save(filename3,’CL’,’-ascii’);
    save(filename4,’p’,’-ascii’);
    end
quit
 best.m:
%Returns the AUC, confidence intervals and 1-sided p-values of all
%unique ensembles of size k. Confidence intervals are calculated at the
%1-alpha level. Both confidence intervals and p-values are calculated
%assuming the validity of the central limit theorem using the auc.m
%script. Ensemble scores are calculated using the best score, i.e.
%min{scores}
function [AUC,CL,p]=best(total,k,alpha)
```

```
P=sum(total(:,2));      %P:number of positives
N=size(total,2)-2;      %N:number of receptor conformations
C=nchoosek(1:1:N,k);    %C:permutation matrix
for i=1:size(C,1)
    index=C(i,:)+2;
    scores=sort(total(:,index),2);
    data=sortrows(horzcat(total(:,1:2),scores(:,1)),3);
    [AUC(i),CL(i),p(i)]=auc(data,P,alpha);
end

auc.m:

%Calculates the AUC of data. Returns a confidence interval, CL, at the
%1-alpha level, and a one-sided p-value: using the central limit
%theorem for both. The rows of input data should be formatted as:
%compound_id  (0/1)  docking_score
%The second column is a 1 if the compound is a binder & 0 if it is a
%non-binder. The calculations follow Craig et al.[93]
function [AUC,CL,p]=auc(data,P,alpha)
Ncompounds=size(data,1);  %number of compounds
np=0;                     %number of positives at threshold
nn=0;                     %number of negatives at threshold
P=P;                      %total number of binders
N=Ncompounds-P;           %total number of nonbinders
tprTemp=0;                %initial true positive rate
fprTemp=0;                %initial false positive rate
alpa=alpha;     %confidence value(set to 0.05 for 95% level
%Calculate tpr(decoy) (Nx1) & fpr(active) (Px1)
j=1;k=1;
```

```
for i=1:Ncompounds
    if data(i,2)==1
        np=np+1;
        tprTemp=np/P;
        fpr(j)=fprTemp;
        j=j+1;
    elseif data(i,2)==0
        nn=nn+1;
        fprTemp=nn/N;
        tpr(k)=tprTemp;
        k=k+1;
    end
end
%AUC: eqn 2 from Ref.
AUC=mean(tpr);
%Variance calculation: eqns 4 & 5 from Ref.
varn=mean((tpr-mean(tpr)).^2);
varp=mean((fpr-mean(fpr)).^2);
%Standard error: eqn 6 from Ref.
SE=sqrt(varp/P+varn/N);
%Confidence intervals
erfinvIn=1-alpha;
CL=sqrt(2)*erfinv(erfinvIn)*SE;
%One-sided p-value
erfIN=(AUC-0.5)/(sqrt(2)*SE);
p=1-(0.5+0.5*erf(erfIN));
```

These analyses have been performed to determine the best ensemble for TbREL1 dataset described above. The results show only mild enrichment with the maximum AUC value reaching 0.58653. The result shows that three protein conformations are

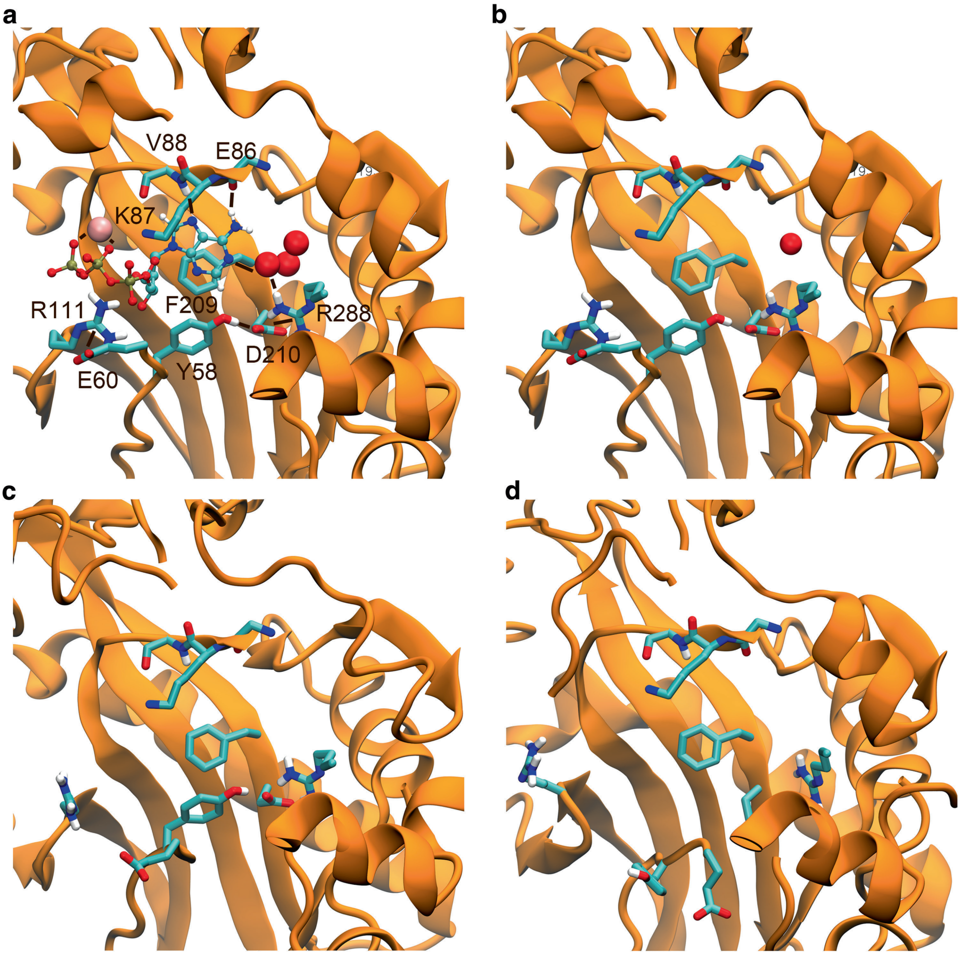


Fig. 4 (**a**) The crystal structure of TbREL1 with ATP bound (1XDN.pdb), also highlighting the magnesium ion and three water molecules bound deep in the protein that interact with ATP. Black markers highlight important interactions, the E60-R111 salt-bridge, Mg^{2+}-triphosphate tail of ATP, E86 and V88 backbone hydrogen bonds to ATP, Y58-D10 hydrogen bond, D210-R288 salt-bridge, R288-Water-N7 hydrogen bond, and stacking of F209 and the adenosine moeity. K87 is highlighted, as it is the catalytic residue that gets adenylated when attacking Pα in ATP. (**b**) a setup of the crystal structure with one specific water molecule at the deep end of the pocket has shown to improve the VS enrichment, (**c**) a representative structure from conventional MD, and (**d**) a representative structure from accelerated MD

performing the best, when it comes to discriminating binders from presumed nonbinders; the crystal structure, and a structural representative from both conventional and accelerated MD. These protein conformations are shown in Fig. 4, with comparison to the crystal structure conformation. Testing with different setups of the crystal structure showed that inclusion of one specific water molecule shown in Fig. 4b, improves the enrichment over other conformations. Further specifics about these receptor conformations and their ability to discriminate the binders from the nonbinders will appear in a separate publication. The mild enrichment demonstrated by this example could be a result of many factors, the most likely is the challenging example we posed to these docking protocols. Here we have a set of known binders with low affinity ($10\ \mu M < IC50 < 100\ \mu M$) and have asked these programs to distinguish them from a set of DUD-E ligands of similar physiochemical and topological properties, a task that continues to be an important area of research in computer-aided drug design.

4 Notes

1. The magnesium parameters can be downloaded from the Bryce group AMBER parameter database (http://www.pharmacy.manchester.ac.uk/bryce/amber) where we have contributed the parameter files with permission from the parameter developers [61].
2. We used dual-boost accelerated MD in NAMD [41, 42], which applies a boost to the entire potential energy, and also a boost to the dihedral potential. For each boosting term two parameters are set: the energy threshold (*E*) and a tuning parameter (*α*), which determines the depth of the potential energy well. To determine the boost energy a short (1–5 ns) classical MD simulation is performed, and the average of the POTENTIAL and DIHED terms in the NAMD output are calculated. The boost parameters are then determined according to the following formula. The factor "4" in the dihedral terms is not a fixed factor; values of 3.5–6 have been reported in the literature [58, 59, 94]. Since we have used the TIP4P-ew water model [62], we have counted the "extra" particle on the water model as an extra atom.

$$
\begin{aligned}
&E(\text{dihed}) = \text{DIHED}_{\text{NAMD}} + 4 * \#\,\text{residues}\\
&\alpha(\text{dihed}) = {}^{1}\!/_{5} + 4 * \#\,\text{residues}\\
&E(\text{total}) = \text{POTENTIAL}_{\text{NAMD}} + 0.16 * \#\,\text{atoms}\\
&\alpha(\text{total}) = 0.16 + \#\,\text{atoms}
\end{aligned}
$$

3. Here we have extracted conformations every 10 ps. This is specified in NAMD using the "DCDfreq" variable.
4. Equidistant means that there is an even spacing in time between the snapshots extracted from the simulations. The ideal is to select a number that will allow diverse conformations of the binding site; however, such a number is highly system specific. Another aspect to take into account is not to set this number too low, as this will lead to too many protein conformations, which will require more computational

resources for the VS, and adding too many protein conformations is not recommended [31, 66].

5. The RMS clustering included the following residues: Tyr58, Glu60, Glu86, Lys87, Asn92, Arg111, Asp159, Phe209, Asp210, Glu283, Val286, Arg288, Arg292, Lys307, and Arg309. These have all previously been highlighted in structural analyses as belonging to the active site [48, 64].
6. Osguthorpe et al. have created an ensemble based on shape diversity of the active site that had an enriching effect on their VS [70, 71]. The *MDpocket utility* found in *Fpocket al*gorithm will calculate the volume and specific chemical properties of a specified pocket from each snapshot of an MD simulation [72, 73]. Subsequent clustering can then be performed on these data to extract a representative set of structures describing variability in the active site. Alternatively, the FTMap algorithm floods the protein surface with a set of small organic molecules and calculates an interaction energy, thereby predicting druggable hotspots in the protein [79, 80]. This algorithm has recently been extended for the analysis of MD trajectories [95]. This method can also be useful in identifying, visualizing, and characterizing new subpockets of the target site.
7. There are three water molecules in the cavity wat1, wat2, and wat3, following the nomenclature of our previously published work [64]. Thus, we have made the following combinations: no waters, wat1, wat2, wat3, wat1 + wat2, wat1 + wat3, wat2 + wat3, wat1 + wat2 + wat3. In total, there are seven different receptor configurations. In recent years programs for the analysis of structurally resolved water molecules have been developed [16]. Schrödinger has developed the WaterMap framework to explore and exploit water molecules bound inside the ligand binding site in drug discovery [96, 97]. Molegro Virtual Docker [98] has developed a docking algorithm with attached water molecules that are then retained or displaced in the docked pose based on energy contributions [99].
8. This conversion was done with the utility *prepare_receptor4.py* in Autodock, which takes a pdb file as input and outputs a pdbqt file.
9. The center was specified as $x = 41.1100$, $y = 34.9382$, and $z = 35.8160$, based on the ATP binding site. The box size was defined as a square with the box length set to 25 Å. As all the structures used were previously aligned to the crystal structure the square box should encapsulate the active site in all the protein conformations.
10. The script is available on the Schrödinger Web site script center (http://www.schrodinger.com/scriptcenter/) and is already preinstalled in Maestro. The script can be used to easily

calculate a receptor grid for each protein conformation in an automated fashion. The script and its application are described in the Schrödinger knowledge base article ID 560 (http://www.schrodinger.com/kb/560). As mentioned the PDB files were all aligned to the crystal structure before docking, so a grid center and box dimensions could be specified on the command line. This is very helpful when you have a large number of protein receptor conformations and want to avoid going through the graphical user interface wizard for each and every conformation. The same grid center that was specified for Autodock Vina, was specified for XGlide. The inner and outer box dimensions were set to 10 and 20 Å, respectively.

11. Depending upon your goals for purchasing compounds, either the publicly available ZINC database (www.zincdocking.org) or other large compendiums can be used to explore an extensive chemical space or if you prefer to only screen compounds that are currently available or within a specific pricing range, contacting individual commercial companies may be preferred to reduce the size needing to be screened. These databases are available for download as sdf or SMILES format. Among filtering software alternatives to OpenEye's Filter are: Schrodinger's Canvas (Schrödinger, LLC, New York, NY, www.schrodinger.com), CCG MOE (Chemical Computing Group Inc., 1010 Sherbooke St. West, Suite #910, Montreal, QC, Canada, H3A 2R7, 2012; www.chemcomp.com), Molsoft (Molsoft, LLC, San Diego, CA; www.molsoft.com), or Accelrys Pipeline Pilot (Accelrys, San Diego, CA; http://accelrys.com).
12. The ligands can be converted using the *prepare_ligand4.py* utility in Autodock. This script takes either .pdb or .mol2 files as input. The current charges were predicted using LigPrep from Schrödinger, but Autodock Vina uses an internal charge scheme, so the input charges were ignored.
13. We have used the SP scoring function.
14. There are many methods available for performance analysis like enrichment factor (EF), Boltzmann-Enhanced Discrimination of ROC (BEDROC), and robust initial enhancement (RIE) methods [45, 100, 101].
15. True positives are the compounds that are predicted to be binders, and are actually binders.
16. False positives are the compounds that are predicted to be binders, but are actually not binders.
17. To compare VS enrichments of different protocols, it is recommended to compare TPR values at FPR values of 0.5 %, 1 %, 2 %, or 5 % [90].
18. The area of a unit square is 1.

19. The central limit theorem (CLT) states that if enough independent measurements of a property are performed on the same system, the average property will be distributed like a Gaussian. The center of the CLT curve will be the true mean, and its width will change with the variance value.
20. If methods A and B are combined, the standard deviation for this new method is $\sigma_{\mathrm{A+B}} = \sqrt{\left(\sigma_{\mathrm{A}}^2 + \sigma_{\mathrm{B}}^2\right)}$.
21. In this example, the best docking score that each ligand gets among the specified receptor conformations is picked. Alternatively, one could pick the average of the docking scores of each ligand for the specified receptor conformations. Or one could choose to compute a weighted-average of docking scores using the population percentages of each cluster if the receptor conformations are extracted by RMSD-based clustering.

Acknowledgements

This work was funded in part by through the NIH Director's New Innovator Award Program DP2-OD007237 and the National Science Foundation's XSEDE Supercomputer resources grant LRAC CHE060073N to R.E.A. Support from the National Biomedical Computation Resource (P41 GM103426), the Center for Theoretical Biophysics, and UCSD Drug Discovery Institute is gratefully acknowledged. J.S. thanks the Alfred Benzon Foundation for a generous postdoctoral fellowship.

References

1. Frauenfelder H, Sligar SG, Wolynes PG (1991) The energy landscapes and motions of proteins. Science 254(5038):1598–1603
2. Boehr DD, Nussinov R, Wright PE (2009) The role of dynamic conformational ensembles in biomolecular recognition. Nat Chem Biol 5(11):789–796. doi:10.1038/nchembio.232
3. Forman-Kay JD (1999) The "dynamics" in the thermodynamics of binding. Nat Struct Biol 6(12):1086–1087. doi:10.1038/70008
4. Cross JB, Thompson DC, Rai BK, Baber JC, Fan KY, Hu Y, Humblet C (2009) Comparison of several molecular docking programs: pose prediction and virtual screening accuracy. J Chem Inf Model 49(6):1455–1474. doi:10.1021/ci900056c
5. Cheng T, Li X, Li Y, Liu Z, Wang R (2009) Comparative assessment of scoring functions on a diverse test set. J Chem Inf Model 49(4):1079–1093. doi:10.1021/ci9000053
6. Armen RS, Chen J, Brooks CL 3rd (2009) An evaluation of explicit receptor flexibility in molecular docking using molecular dynamics and torsion angle molecular dynamics. J Chem Theory Comput 5(10):2909–2923. doi:10.1021/ct900262t
7. Sutherland JJ, Nandigam RK, Erickson JA, Vieth M (2007) Lessons in molecular recognition. 2. Assessing and improving cross-docking accuracy. J Chem Inf Model 47(6):2293–2302. doi:10.1021/ci700253h
8. Verkhivker GM, Bouzida D, Gehlhaar DK, Rejto PA, Freer ST, Rose PW (2002) Complexity and simplicity of ligand-macromolecule interactions: the energy landscape perspective. Curr Opin Struct Biol 12(2):197–203
9. Lin J-H, Perryman AL, Schames JR, McCammon JA (2002) Computational drug design accommodating receptor flexibility: the relaxed complex scheme. J Am Chem Soc 124(20):5632–5633. doi:10.1021/ja0260162
10. Teague SJ (2003) Implications of protein flexibility for drug discovery. Nat Rev Drug Discov 2(7):527–541. doi:10.1038/nrd1129

11. Cozzini P, Kellogg GE, Spyrakis F, Abraham DJ, Costantino G, Emerson A, Fanelli F, Gohlke H, Kuhn LA, Morris GM, Orozco M, Pertinhez TA, Rizzi M, Sotriffer CA (2008) Target flexibility: an emerging consideration in drug discovery and design. J Med Chem 51(20):6237–6255. doi:10.1021/jm800562d
12. Schames JR, Henchman RH, Siegel JS, Sotriffer CA, Ni H, McCammon JA (2004) Discovery of a novel binding trench in HIV integrase. J Med Chem 47(8):1879–1881. doi:10.1021/jm0341913
13. Gorfe AA, Caflisch A (2005) Functional plasticity in the substrate binding site of beta-secretase. Structure 13(10):1487–1498. doi:10.1016/j.str.2005.06.015
14. Cheng LS, Amaro RE, Xu D, Li WW, Arzberger PW, McCammon JA (2008) Ensemble-based virtual screening reveals potential novel antiviral compounds for avian influenza neuraminidase. J Med Chem 51(13):3878–3894. doi:10.1021/jm8001197
15. Baron R, McCammon JA (2007) Dynamics, hydration, and motional averaging of a loop-gated artificial protein cavity: the W191G mutant of cytochrome c peroxidase in water as revealed by molecular dynamics simulations. Biochemistry 46(37):10629–10642. doi:10.1021/bi700866x
16. Yuriev E, Agostino M, Ramsland PA (2011) Challenges and advances in computational docking: 2009 in review. J Mol Recognit 24(2):149–164. doi:10.1002/jmr.1077
17. Lavecchia A, Di Giovanni C (2013) Virtual screening strategies in drug discovery: a critical review. Curr Med Chem 20(23):2839–2860
18. B-Rao C, Subramanian J, Sharma SD (2009) Managing protein flexibility in docking and its applications. Drug Discov Today 14(7–8):394–400. doi:10.1016/j.drudis.2009.01.003
19. Sinko W, Lindert S, McCammon JA (2013) Accounting for receptor flexibility and enhanced sampling methods in computer-aided drug design. Chem Biol Drug Des 81(1):41–49. doi:10.1111/cbdd.12051
20. Jiang F, Kim SH (1991) “Soft docking”: matching of molecular surface cubes. J Mol Biol 219(1):79–102
21. Cerqueira NM, Bras NF, Fernandes PA, Ramos MJ (2009) MADAMM: a multistaged docking with an automated molecular modeling protocol. Proteins 74(1):192–206. doi:10.1002/prot.22146
22. Sherman W, Day T, Jacobson MP, Friesner RA, Farid R (2006) Novel procedure for modeling ligand/receptor induced fit effects. J Med Chem 49(2):534–553. doi:10.1021/jm050540c
23. Sherman W, Beard HS, Farid R (2006) Use of an induced fit receptor structure in virtual screening. Chem Biol Drug Des 67(1):83–84. doi:10.1111/j.1747-0285.2005.00327.x
24. Jain AN (2009) Effects of protein conformation in docking: improved pose prediction through protein pocket adaptation. J Comput Aided Mol Des 23(6):355–374. doi:10.1007/s10822-009-9266-3
25. Davis IW, Baker D (2009) RosettaLigand docking with full ligand and receptor flexibility. J Mol Biol 385(2):381–392. doi:10.1016/j.jmb.2008.11.010
26. Lemmon G, Meiler J (2012) Rosetta ligand docking with flexible XML protocols. Methods Mol Biol 819:143–155. doi:10.1007/978-1-61779-465-0_10
27. Abagyan R, Totrov M, Kuznetsov D (1994) ICM – a new method for protein modeling and design: applications to docking and structure prediction from the distorted native conformation. J Comput Chem 15(5):488–506. doi:10.1002/jcc.540150503
28. Corbeil CR, Moitessier N (2009) Docking ligands into flexible and solvated macromolecules. 3. Impact of input ligand conformation, protein flexibility, and water molecules on the accuracy of docking programs. J Chem Inf Model 49(4):997–1009. doi:10.1021/ci8004176
29. Cavasotto CN, Kovacs JA, Abagyan RA (2005) Representing receptor flexibility in ligand docking through relevant normal modes. J Am Chem Soc 127(26):9632–9640. doi:10.1021/ja042260c
30. Cavasotto CN (2012) Normal mode-based approaches in receptor ensemble docking. Methods Mol Biol 819:157–168. doi:10.1007/978-1-61779-465-0_11
31. Nichols SE, Baron R, Ivetac A, McCammon JA (2011) Predictive power of molecular dynamics receptor structures in virtual screening. J Chem Inf Model 51(6):1439–1446. doi:10.1021/ci200117n
32. Amaro RE, Baron R, McCammon JA (2008) An improved relaxed complex scheme for receptor flexibility in computer-aided drug design. J Comput Aided Mol Des 22(9):693–705. doi:10.1007/s10822-007-9159-2
33. Lin J-H, Perryman AL, Schames JR, McCammon JA (2003) The relaxed complex method: accommodating receptor flexibility for drug design with an improved scoring scheme. Biopolymers 68(1):47–62. doi:10.1002/bip.10218
34. Schnaufer A, Ernst NL, Palazzo SS, O'Rear J, Salavati R, Stuart K (2003) Separate insertion and deletion subcomplexes of the

Trypanosoma brucei RNA editing complex. Mol Cell 12(2):307–319

35. Landon MR, Amaro RE, Baron R, Ngan CH, Ozonoff D, Andrew McCammon J, Vajda S (2008) Novel druggable hot spots in avian influenza neuraminidase H5N1 revealed by computational solvent mapping of a reduced and representative receptor ensemble. Chem Biol Drug Des 71(2):106–116. doi:10.1111/j.1747-0285.2007.00614.x
36. Babakhani A, Talley TT, Taylor P, McCammon JA (2009) A virtual screening study of the acetylcholine binding protein using a relaxed-complex approach. Comput Biol Chem 33(2):160–170. doi:10.1016/j.compbiolchem.2008.12.002
37. Durrant JD, de Oliveira CAF, McCammon JA (2010) Including receptor flexibility and induced fit effects into the design of MMP-2 inhibitors. J Mol Recognit 23(2):173–182. doi:10.1002/jmr.989
38. Demir Ö, Baronio R, Salehi F, Wassman CD, Hall L, Hatfield GW, Chamberlin R, Kaiser P, Lathrop RH, Amaro RE (2011) Ensemble-based computational approach discriminates functional activity of p53 cancer and rescue mutants. PLoS Comput Biol 7(10):e1002238. doi:10.1371/journal.pcbi.1002238
39. Amaro RE, Schnaufer A, Interthal H, Hol W, Stuart KD, McCammon JA (2008) Discovery of drug-like inhibitors of an essential RNA-editing ligase in Trypanosoma brucei. Proc Natl Acad Sci U S A 105(45):17278–17283. doi:10.1073/pnas.0805820105
40. Durrant JD, Hall L, Swift RV, Landon M, Schnaufer A, Amaro RE (2010) Novel naphthalene-based inhibitors of Trypanosoma brucei RNA editing ligase 1. PLoS Negl Trop Dis 4(8):e803. doi:10.1371/journal.pntd.0000803
41. Phillips JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, Chipot C, Skeel RD, Kalé L, Schulten K (2005) Scalable molecular dynamics with NAMD. J Comput Chem 26: 1781–1802
42. Wang Y, Harrison CB, Schulten K, McCammon JA (2011) Implementation of accelerated molecular dynamics in NAMD. Comput Sci Discov 4(1):015002
43. Hornak V, Abel R, Okur A, Strockbine B, Roitberg A, Simmerling C (2006) Comparison of multiple Amber force fields and development of improved protein backbone parameters. Proteins 65(3):712–725. doi:10.1002/prot.21123
44. Halgren TA, Murphy RB, Friesner RA, Beard HS, Frye LL, Pollard WT, Banks JL (2004) Glide: a new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening. J Med Chem 47(7): 1750–1759. doi:10.1021/jm030644s
45. Friesner RA, Banks JL, Murphy RB, Halgren TA, Klicic JJ, Mainz DT, Repasky MP, Knoll EH, Shelley M, Perry JK, Shaw DE, Francis P, Shenkin PS (2004) Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. J Med Chem 47(7):1739–1749. doi:10.1021/jm0306430
46. Trott O, Olson AJ (2010) AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J Comput Chem 31(2):455–461. doi:10.1002/jcc.21334
47. Humphrey W, Dalke A, Schulten K (1996) VMD: visual molecular dynamics. J Mol Graph 14(1):33–38, 27–38
48. Deng J, Schnaufer A, Salavati R, Stuart KD, Hol WG (2004) High resolution crystal structure of a key editosome enzyme from Trypanosoma brucei: RNA editing ligase 1. J Mol Biol 343(3):601–613. doi:10.1016/j.jmb.2004.08.041
49. Drug Discovery Unit UoD DDU Library Collections (2013) http://www.drugdiscovery.dundee.ac.uk/libraries.html
50. Mysinger MM, Carchia M, Irwin JJ, Shoichet BK (2012) Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. J Med Chem 55(14):6582–6594. doi:10.1021/jm300687e
51. Moshiri H, Acoca S, Kala S, Najafabadi HS, Hogues H, Purisima E, Salavati R (2011) Naphthalene-based RNA editing inhibitor blocks RNA editing activities and editosome assembly in Trypanosoma brucei. J Biol Chem 286(16):14178–14189. doi:10.1074/jbc.M110.199646
52. Salavati R, Moshiri H, Kala S, Shateri Najafabadi H (2012) Inhibitors of RNA editing as potential chemotherapeutics against trypanosomatid pathogens. Int J Parasitol Drugs Drug Resist 2:36–46. doi:10.1016/j.ijpddr.2011.10.003
53. Sørensen J, Palmer DS, Qvist KB, Schiøtt B (2011) Initial stage of cheese production: a molecular modeling study of bovine and camel chymosin complexed with peptides from the chymosin-sensitive region of kappa-casein. J Agric Food Chem 59(10):5636–5647. doi:10.1021/jf104898w
54. Feher VA, Lawson JD (2009) Approaches to kinase homology modeling: successes and considerations for the structural kinome. In: Rongshi L, Stafford JA (eds) Kinase inhibitor drugs. Wiley, Hoboken, NJ, pp 433–460. doi: 10.1002/9780470524961.ch17
55. Marti-Renom MA, Stuart AC, Fiser A, Sanchez R, Melo F, Sali A (2000) Comparative protein structure modeling of genes and genomes.

Annu Rev Biophys Biomol Struct 29:291–325. doi:10.1146/annurev.biophys.29.1.291

56. Cavasotto CN, Phatak SS (2009) Homology modeling in drug discovery: current trends and applications. Drug Discov Today 14(13–14):676–683. doi:10.1016/j.drudis.2009.04.006

57. Damm-Ganamet KL, Smith RD, Dunbar JB Jr, Stuckey JA, Carlson HA (2013) CSAR benchmark exercise 2011–2012: evaluation of results from docking and relative ranking of blinded congeneric series. J Chem Inf Model. doi:10.1021/ci400025f

58. Hamelberg D, Mongan J, McCammon JA (2004) Accelerated molecular dynamics: a promising and efficient simulation method for biomolecules. J Chem Phys 120(24): 11919–11929. doi:10.1063/1.1755656

59. Pierce LCT, Salomon-Ferrer R, Augusto F, de Oliveira C, McCammon JA, Walker RC (2012) Routine access to millisecond time scale events with accelerated molecular dynamics. J Chem Theory Comput 8(9):2997–3002. doi:10.1021/ct300284c

60. Swift RV, Durrant J, Amaro RE, McCammon JA (2009) Toward understanding the conformational dynamics of RNA ligation. Biochemistry 48(4):709–719. doi:10.1021/bi8018114

61. Allnér O, Nilsson L, Villa A (2012) Magnesium ion–water coordination and exchange in biomolecular simulations. J Chem Theory Comput 8(4):1493–1502. doi:10.1021/ct3000734

62. Horn HW, Swope WC, Pitera JW, Madura JD, Dick TJ, Hura GL, Head-Gordon T (2004) Development of an improved four-site water model for biomolecular simulations: TIP4P-Ew. J Chem Phys 120(20):9665–9678. doi:10.1063/1.1683075

63. Demir O, Amaro RE (2013) Designing novel inhibitors of Trypanosoma brucei. In: Kortagere S (ed) Methods in molecular biology: in silico models for drug discovery, vol 993. Humana Press, Totowa, NJ, pp 231–243, doi: 10.1007/978-1-62703-342-8_15

64. Amaro RE, Swift RV, McCammon JA (2007) Functional and structural insights revealed by molecular dynamics simulations of an essential RNA editing ligase in Trypanosoma brucei. PLoS Negl Trop Dis 1(2):e68. doi:10.1371/journal.pntd.0000068

65. Shang Y, Simmerling C (2012) Molecular dynamics applied in drug discovery: the case of HIV-1 protease. Methods Mol Biol 819:527–549. doi:10.1007/978-1-61779-465-0_31

66. Nichols S, Baron R, McCammon JA (2012) On the use of molecular dynamics receptor conformations for virtual screening. In: Baron R (ed) Computational drug discovery and design, vol 819, Methods in molecular biology. Springer, New York, pp 93–103

67. O'Donoghue P, Luthey-Schulten Z (2003) On the evolution of structure in aminoacyl-tRNA synthetases. Microbiol Mol Biol Rev 67(4):550–573. doi:10.1128/MMBR.67.4.550-573.2003

68. Baron R, McCammon JA (2008) (Thermo) dynamic role of receptor flexibility, entropy, and motional correlation in protein–ligand binding. Chem Phys Chem 9(7):983–988. doi:10.1002/cphc.200700857

69. Shao J, Tanner SW, Thompson N, Cheatham TE (2007) Clustering molecular dynamics trajectories: 1. Characterizing the performance of different clustering algorithms. J Chem Theory Comput 3(6):2312–2334. doi:10.1021/ct700119m

70. Osguthorpe DJ, Sherman W, Hagler AT (2012) Generation of receptor structural ensembles for virtual screening using binding site shape analysis and clustering. Chem Biol Drug Des 80(2):182–193. doi:10.1111/j.1747-0285.2012.01396.x

71. Osguthorpe DJ, Sherman W, Hagler AT (2012) Exploring protein flexibility: incorporating structural ensembles from crystal structures and simulation into virtual screening protocols. J Phys Chem B 116(23):6952–6959. doi:10.1021/jp3003992

72. Schmidtke P, Barril X (2010) Understanding and predicting druggability. A high-throughput method for detection of drug binding sites. J Med Chem 53(15):5858–5867. doi:10.1021/jm100574m

73. Le Guilloux V, Schmidtke P, Tuffery P (2009) Fpocket: an open source platform for ligand pocket detection. BMC Bioinformatics 10:168. doi:10.1186/1471-2105-10-168

74. Durrant JD, de Oliveira CAF, McCammon JA (2011) POVME: an algorithm for measuring binding-pocket volumes. J Mol Graph Model 29(5):773–776. doi:10.1016/j.jmgm.2010.10.007

75. Roe DR, Cheatham TE (2013) PTRAJ and CPPTRAJ: software for processing and analysis of molecular dynamics trajectory data. J Chem Theory Comput 9(7):3084–3095. doi:10.1021/ct400341p

76. Salomon-Ferrer R, Case DA, Walker RC (2013) An overview of the Amber biomolecular simulation package. Wiley Interdiscip Rev Comput Mol Sci 3(2):198–210. doi:10.1002/wcms.1121

77. Case DA, Darden TA, Cheatham III TE, Simmerling CL, Wang J, Duke RE, Luo R, Walker RC, Zhang W, Merz KM, Roberts B, Hayik S, Roitberg A, Seabra G, Swails J, Goetz

AW, Kolossváry I, Wong KF, Paesani F, Vanicek J, Wolf RM, Liu J, Wu X, Brozell SR, Steinbrecher T, Gohlke H, Cai Q, Ye X, Wang J, Hsieh M-J, Cui G, Roe DR, Mathews DH, Seetin MG, Salomon-Ferrer R, Sagui C, Babin V, Luchko T, Gusarov S, Kovalenko A, Kollman PA (2012) AMBER. 12 edn. University of California, San Francisco, CA, USA

78. Case DA, Cheatham TE, Darden T, Gohlke H, Luo R, Merz KM, Onufriev A, Simmerling C, Wang B, Woods RJ (2005) The Amber biomolecular simulation programs. J Comput Chem 26(16):1668–1688. doi:10.1002/jcc.20290

79. Kozakov D, Hall DR, Chuang G-Y, Cencic R, Brenke R, Grove LE, Beglov D, Pelletier J, Whitty A, Vajda S (2011) Structural conservation of druggable hot spots in protein–protein interfaces. Proc Natl Acad Sci 108(33):13528–13533. doi:10.1073/pnas.1101835108

80. Brenke R, Kozakov D, Chuang G-Y, Beglov D, Hall D, Landon MR, Mattos C, Vajda S (2009) Fragment-based identification of druggable "hot spots" of proteins using Fourier domain correlation techniques. Bioinformatics 25(5):621–627. doi:10.1093/bioinformatics/btp036

81. Brenk R, Schipani A, James D, Krasowski A, Gilbert IH, Frearson J, Wyatt PG (2008) Lessons learnt from assembling screening libraries for drug discovery for neglected diseases. Chem Med Chem 3(3):435–444. doi:10.1002/cmdc.200700139

82. Rishton GM (2003) Nonleadlikeness and leadlikeness in biochemical screening. Drug Discov Today 8(2):86–96

83. Seidler J, McGovern SL, Doman TN, Shoichet BK (2003) Identification and prediction of promiscuous aggregating inhibitors among known drugs. J Med Chem 46(21):4477–4486. doi:10.1021/jm030191r

84. McGovern SL, Caselli E, Grigorieff N, Shoichet BK (2002) A common mechanism underlying promiscuous inhibitors from virtual and high-throughput screening. J Med Chem 45(8):1712–1722

85. Baell JB, Holloway GA (2010) New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. J Med Chem 53(7):2719–2740. doi:10.1021/jm901137j

86. Walters WP, Stahl MT, Murcko MA (1998) Virtual screening – an overview. Drug Discov Today 3(4):160–178

87. Irwin JJ, Sterling T, Mysinger MM, Bolstad ES, Coleman RG (2012) ZINC: a free tool to discover chemistry for biology. J Chem Inf Model 52(7):1757–1768. doi:10.1021/ci3001277

88. Warren GL, Andrews CW, Capelli AM, Clarke B, LaLonde J, Lambert MH, Lindvall M, Nevins N, Semus SF, Senger S, Tedesco G, Wall ID, Woolven JM, Peishoff CE, Head MS (2006) A critical assessment of docking programs and scoring functions. J Med Chem 49(20):5912–5931. doi:10.1021/jm050362n

89. Nicholls A (2011) What do we know?: simple statistical techniques that help. In: Bajorath J (ed) Chemoinformatics and computational chemical biology, vol 672, Methods in molecular biology. Humana Press, Totowa, NJ, pp 531–581. doi:10.1007/978-1-60761-839-3_22

90. Jain AN (2008) Bias, reporting, and sharing: computational evaluations of docking methods. J Comput Aided Mol Des 22(3–4):201–212. doi:10.1007/s10822-007-9151-x

91. Nichols SE, Swift RV, Amaro RE (2012) Rational prediction with molecular dynamics for hit identification. Curr Top Med Chem 12(18):2002–2012. doi:10.2174/156802612804910313

92. du Prel JB, Hommel G, Rohrig B, Blettner M (2009) Confidence interval or *p*-value?: part 4 of a series on evaluation of scientific publications. Dtsch Arztebl Int 106(19):335–339. doi: D – NLM: PMC2689604 OTO – NOTNLM

93. Craig IR, Essex JW, Spiegel K (2010) Ensemble docking into multiple crystallographically derived protein structures: an evaluation based on the statistical analysis of enrichments. J Chem Inf Model 50(4):511–524. doi:10.1021/ci900407c

94. Bucher D, Grant BJ, Markwick PR, McCammon JA (2011) Accessing a hidden conformation of the maltose binding protein using accelerated molecular dynamics. PLoS Comput Biol 7(4):e1002034. doi:10.1371/journal.pcbi.1002034

95. Votapka L, Amaro RE (2013) Multistructural hot spot characterization with FTProd. Bioinformatics 29(3):393–394. doi:10.1093/bioinformatics/bts689

96. Abel R, Young T, Farid R, Berne BJ, Friesner RA (2008) Role of the active-site solvent in the thermodynamics of factor Xa ligand binding. J Am Chem Soc 130(9):2817–2831. doi:10.1021/ja0771033

97. Beuming T, Farid R, Sherman W (2009) High-energy water sites determine peptide binding affinity and specificity of PDZ domains. Protein Sci 18(8):1609–1619. doi:10.1002/pro.177

98. Thomsen R, Christensen MH (2006) MolDock: a new technique for high-accuracy molecular docking. J Med Chem 49(11):3315–3321. doi:10.1021/jm051197e

99. Lie MA, Thomsen R, Pedersen CNS, Schiøtt B, Christensen MH (2011) Molecular docking with ligand attached water molecules. J Chem Inf Model 51(4):909–917. doi:10.1021/ci100510m

100. Truchon J-F, Bayly CI (2007) Evaluating virtual screening methods: good and bad metrics for the "early recognition" problem. J Chem Inf Model 47(2):488–508. doi:10.1021/ci600426e

101. Sheridan RP, Singh SB, Fluder EM, Kearsley SK (2001) Protocols for bridging the peptide to nonpeptide gap in topological similarity searches. J Chem Inf Comput Sci 41(5):1395–1406. doi:10.1021/ci0100144

Index

Andreas Kukol (ed.), *Molecular Modeling of Proteins*, Methods in Molecular Biology, vol. 1215,
DOI 10.1007/978-1-4939-1465-4, © Springer Science+Business Media New York 2015