

Spin Label Modeling Tutorial



This tutorial parallels the *Molecular Modeling of Spin Labels Appendix* in Biological Magnetic Resonance, eds. M. Hemminga, L. Berliner. The **tutorial** directory contains a sub-directory for each section within this tutorial. Each subdirectory contains the input and output files for the corresponding section. These files are often copied from the output of previous sections, and we note where these files existed before we copied them to the current section's directory. The output files (in **Output** directory) are for comparison when troubleshooting potential problems. Note that deviations between your output and our output will occur where a random variable seed is used.

1. System Requirements

The software used for simulation and analysis includes the commercial software CHARMM (v. 27). An academic license for CHARMM can be obtained from <http://www.charmm.org/> and requires a LINUX/UNIX operating system. Other programs include the freeware NAMD, PSFgen and VMD and are available at <http://www.ks.uiuc.edu/> for all popular operating systems. The tutorial assumes that all manipulations are executed in a LINUX/UNIX environment.

The scripts provided with this tutorial assume that CHARMM is on the system path and the following steps have been taken. The directory *sl_mod* (in the **1-ProgramScripts** directory of the tutorial) should be copied to the user's **home** directory. An environment variable **\$SL_MOD** pointing to the location of this directory needs to be

added to the user's shell configuration file. For *bash* shell users, add the following line to the *.bashrc* file:

```
export SL_MOD='~/sl_mod/'
```

2. Preparation of Structures

2-1. Protein Structure

During this tutorial we will study *Staphylococcal Nuclease* (PDB accession number: 1EY0, last digit is zero). The structure is available from the Protein Data Bank (<http://www.rcsb.org>), and can be found in the **2-1-Structure** directory of the tutorial.

2-2. Spin Label Mutagenesis

Our first task is to mutate one of the protein residues to a spin label adjunct using PSFgen. The spin label we will use is MTSSL, and the modified cysteine residue name is CMTS. The wild type protein structure, *1EY0.pdb*, and the VMD script *mutate.pgn* reside in the **2-2-Mutagenesis** directory. Within VMD use the Tk Console to move to the **2-2-Mutagenesis** directory. The mutagenesis script is executed by typing the following command into VMD's Tk Console:

```
source mutate.pgn
```

The mutated protein will be shown in the VMD graphics window and stored in the **2-2-Mutagenesis** directory as a *build.pdb* file. Note that the topology file used by PSFgen is based on the CHARMM-19 topology (files included in the **common** directory of the tutorial). A feature incompatible with PSFgen had to be removed from the original topology file.

2-3. Separation of Segments

Crystal structures often have missing amino acids. The polypeptide segments need to be separated to ensure that a covalent bond is not be created over the missing amino acid region. Launch VMD and move to the **2-3-Separation** directory. Execute the *separate.pgn* script using the Tk Console:

```
source separate.pgn
```

The PDB file will be broken into individual files, one for each segment. A file called *numsegs* records the total number of segments in the protein.

2-4. Building the Protein Structure File (PSF)

The molecular modeling requires the connectivity, mass, partial charges, etc. of each atom in addition to the coordinates provided by the PDB file. The additional information is stored in a Protein Structure File (PSF). The Monte Carlo simulation was developed using the CHARMM-19 topology, and a compatible PSF file must be generated using CHARMM.

Move to the **2-4-PSF** directory, which contains the results of section 2-3. Execute the building process with the following command:

```
$SL_MOD/programs/build
```

The script determines the number of segments (from the *numsegs* file) and reads the individual PDB files for each segment and recombines the segments into one PDB file

(*build.pdb*), CHARMM coordinate file (*build.crd*), an internal coordinates file (*build.ic*), and the protein structure file (*build.psf*).

2-5. Energy Minimization

The protein structure must be energy minimized to avoid steric conflicts of the mutated residue. We have already moved the CRD and PSF file generated in the previous step to a new directory, **2-5-Minimization**. The minimization process is accomplished with the following command:

```
$SL_MOD/programs/minimize
```

The energy minimized coordinates are stored in *build_min.crd* and *build_min.pdb*. The *build_min.out* in the *Output* directory contains a sample of the minimization output.

3. Simulations

3-1. Metropolis Monte Carlo Minimization (MMCM)

Search of the spin label rotamer structures is accomplished by random rotations of the spin label bonds followed by energy minimization. We have taken the *build_min.crd* from section 2-5 and renamed it *build.crd* in the **3-1-MMCM** directory. The *build.psf* and *build.ic* files from section 2-4 have also been copied to the **3-1-MMCM** directory. The *MMCM* script finds the CMTS residue (spin labeled cysteine) and performs a rotamer search until manually terminated. The process begins by executing:

```
$SL_MOD/programs/MMCM
```

The *frames* directory contains the coordinates of each minimized CMTS rotamer. Allow the MMCM program to proceed for a few minutes to generate approximately 15 rotamers and then manually terminate the program using *Ctrl+C*. The rotamer search in this time frame is not exhaustive, and the accepted global minima for future steps will be provided from previous, longer searches. Typical rotamer searches require 1000-2000 local minima (or frames). The energy of each rotamer is stored in the *mmcm.dat* file.

3-2. Compilation of MMCM Results

The results of the rotamer searches need to be compiled, energy sorted and the best structure stored as both a PDB and a CRD file. This will typically take place in the same directory as the MMCM simulation from step 3-1, but for this example we have moved the output of step 3-1 to the **3-2-Compilation** directory.

```
$SL_MOD/programs/MMCM_traj
```

The various spin label rotamers will be compiled into a single trajectory file, *mmcm.dcd*, analogous to the molecular dynamics trajectory files and can be displayed in the VMD. Each frame of *mmcm.dcd* corresponds to the local minima structure of a single iteration in the MMCM process and the potential energy and coordinates of each rotamer placed into *mmcm.dat* file. The file *mmcm_Vsorted.dat* contains the rotamer search results in ascending order of potential energy, with the MMCM iteration number as the first entry of each line. In our example, the 11th iteration yielded the lowest potential energy.

3-3. Selection of Global Minima

The compiled results of a longer MMCM run have been included in the **3-3-Selection** directory. A total of 2163 iterations were sorted, and the 656th iteration yielded the lowest potential energy. Loading *mmcm.dcd* into *build.pdb* within VMD allows for each iteration structure to be visualized. Inspection of the *mmcm_Vsorted.dat* file and visualization of the corresponding structure in VMD helps to identify non-unique rotamers: e.g. in our example the five lowest energy rotamers have nearly the same coordinates. The sixth lowest potential energy rotamer, MMCM iteration number 1805, belongs to a different rotamer family. To avoid selection of non-unique rotamers, make a habit of inspecting the *mmcm.dcd* and *mmcm_Vsorted.dat* files. As an exercise find the four lowest energy rotamer families in the provided example. (*Representative rotamers: 656, 1805, 193, and 615 listed in increasing energy.*)

3-4. Molecular Dynamics – Implicit Solvent

The lowest energy structure files, *frame656.pdb* and *frame656.crd*, from section 3-3 were copied to the **3-4-ImplicitMD** directory and renamed *build.pdb* and *build.crd*, respectively. The *build.psf* from section 2-4 was also copied to **3-4-ImplicitMD**. The heating/equilibration/production stages of the molecular dynamics are accomplished by executing:

```
$SL_MOD/program/MD_hep_sphere_300K
```

3-5. Explicit Solvation

The lowest energy structure, *frame656.pdb*, from section 3-3 was copied to the **3-5-Solvation** directory and renamed *build.pdb*. The PSF file used for CHARMM is no longer valid, since NAMD will use a modified CHARMM-27 topology file (CHARMM-19 was used in the previous examples). The generation of a new PSF file and the inclusion of explicit solvent have been combined into the *build_solvate.pgn* script. Change to the **3-5-Solvation** directory within VMD and execute the following command within the Tk Console:

```
source build_solvate.pgn
```

The resulting *build_wb.pdb* and *build_wb.psf* contain the solvated protein, and the *PBC.dat* file contains the dimensions of the solvation box for use in the simulation's periodic boundary conditions.

3-6. Molecular Dynamics – Explicit Solvent

The solvated protein PDB and PSF files from section 3-5 were copied to the **3-6-ExplicitMD** directory. The periodic boundary conditions in *PBC.dat* are used in the input file *heat.inp*. The heating process is executed with the following command:

```
namd2 heat.inp > heat.out &
```

Any simulation in explicit solvent takes a large amount of computational power and this example heating process will take several hours. A multiple CPU cluster is recommended for all work with explicit solvent. Details on restarting a manually terminated simulation can be found on the NAMD website (<http://www.ks.uiuc.edu/Research/namd/>).

4. Data Analysis

The **common** directory contains the initial structure *build.pdb* and the DCD trajectory file *build_wb.dcd* of 296 ps of molecular dynamics in explicit solvent. The CMTS spin label is in the 62nd residue position. For large trajectory files the *bigdcd.tcl* script (from the VMD website) allows loading a piece at a time. The following VMD analysis scripts are meant to highlight the capabilities of VMD, and can be altered or even combined to yield more specific analysis tools. Development and modification of the VMD scripts requires basic knowledge of the Tcl language and familiarity with the VMD command syntax. The VMD tutorial and manual at <http://www.ks.uiuc.edu/Research/vmd/> should be consulted for editing the scripts.

4-1. Dihedral Angle

The **4-1-Dihedral** directory contains the VMD analysis script *chi_analysis.tcl*, which analyses the bond angle of each spin label rotamer. Execute the script within VMD Tk Console using the following command:

```
source chi_analysis.tcl
```

The resulting *chi.dat* contains the first dihedral angle (χ -1) the 62nd side chain (dihedral angle formed by atoms: N, C α , C β and S γ). Editing the first several lines of the *chi_analysis.tcl* script changes the residue and the atoms forming the dihedral angle. For example, the second side-chain dihedral angle (χ -2) would be analyzed by changing atom1 to CA, atom2 to CB, atom3 to SG, and atom4 to S1.

4-2. Atomic Distance

The **4-2-Distance** directory contains the VMD script *distance_analysis.tcl*, which calculates distance between the selected atoms across the whole trajectory.

```
source distance_analysis.tcl
```

The resulting *distance.dat* contains the atomic distance between the CA (α -carbon) of the 23rd and 65th residues of the protein. The distance between any two atoms can be analyzed by altering the *res_id#* and *atom_name#* variables set in the first several lines of *distance_analysis.tcl*.

4-3. Atom Coordinates

The **4-3-Coordinates** directory contains the VMD script *coordinate_analysis.tcl* to determine the xyz coordinates of a specified atom.

```
source coordinate_analysis.tcl
```

The resulting *Nxyz.dat* will contain the atomic coordinates of the N of the 62nd residue. The coordinates of any atom can be analyzed by changing the *res_id* and *atom_name* variables set in *coordinate_analysis.tcl*.

4-4. Solvent Accessible Surface Area (SASA)

The **4-4-SASA** directory contains the VMD analysis script *SASA_analysis.tcl* to calculate solvent accessible surface area of a specified atom(s).

```
source SASA_analysis.tcl
```

The resulting *SASA.dat* contains the SASA of the 62nd residue using a 1.5 Å surface probe. The radius and residue are variables defined in the first few lines of *SASA_analysis.tcl* script.