Chapter 2
Application of Markov State Models to Simulate Long Timescale Dynamics of Biological Macromolecules

Lin-Tai Da*, Fu Kit Sheong*, Daniel-Adriano Silva*, and Xuhui Huang

Abstract Conformational changes of proteins are an essential part of many biological processes such as: protein folding, ligand binding, signal transduction, allostery, and enzymatic catalysis. Molecular dynamics (MD) simulations can describe the dynamics of molecules at atomic detail, therefore providing a much higher temporal and spatial resolution than most experimental techniques. Although MD simulations

---

*Author contributed equally with all other contributors.

L.-T. Da
Department of Chemistry, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California, San Diego, La Jolla, CA 92093, USA

F.K. Sheong
Department of Chemistry, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

D.-A. Silva
Department of Chemistry, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

Department of Biochemistry, University of Washington, Seattle, WA 98105, USA

X. Huang (✉)
Department of Chemistry, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

Division of Biomedical Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

Center of Systems Biology and Human Health, School of Science and Institute for Advance Study, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
e-mail: xuhuihuang@ust.hk

K. Han et al. (eds.), *Protein Conformational Dynamics*, Advances in Experimental Medicine and Biology 805, DOI 10.1007/978-3-319-02970-2_2,
© Springer International Publishing Switzerland 2014

have been widely applied to study protein dynamics, the timescales accessible by conventional MD methods are usually limited to timescales that are orders of magnitude shorter than the conformational changes relevant for most biological functions. During the past decades great effort has been devoted to the development of theoretical methods that may enhance the conformational sampling. In recent years, it has been shown that the statistical mechanics framework provided by discrete-state and -time Markov State Models (MSMs) can predict long timescale dynamics from a pool of short MD simulations. In this chapter we provide the readers an account of the basic theory and selected applications of MSMs. We will first introduce the general concepts behind MSMs, and then describe the existing procedures for the construction of MSMs. This will be followed by the discussions of the challenges of constructing and validating MSMs, Finally, we will employ two biologically-relevant systems, the RNA polymerase and the LAO-protein, to illustrate the application of Markov State Models to elucidate the molecular mechanisms of complex conformational changes at biologically relevant timescales.

Keywords Markov State Models • Molecular dynamics simulations • Free energy landscape • Molecular recognition • Biological macromolecules • Proteins • RNA polymerase

2.1 Introduction

Conformational changes are known to be critical in many biological processes such as protein folding, ligand binding, signal transduction, allostery, and enzymatic catalysis [1–4]. Due to its significance in biological field, a great amount of research attention has been drawn to investigate conformational changes in biological macromolecules. Throughout the decades of conformational studies in biomolecular systems, X-ray crystallography [5] has evolved and thus has already revolutionized our understanding on the atomic-level structural details as well as the functions of protein, DNA and RNA. More recently, the emergence of Cryo-electron microscopy [6] and small-angle X-ray scattering techniques [7] has further boosted the advance in the understanding of complex biomolecular structures. Despite their remarkable success in the field, these experimental methods can only provide static snapshots of the molecules under study but not the details of the conformational dynamics. To overcome this limitation, alternative experimental methods, including nuclear magnetic resonance spectroscopy (NMR) [8] and several different fluorescence spectroscopic techniques [9–13], are routinely used to study the dynamics of protein ensembles in real time. Even so, the atomic-level details of conformational changes in biological macromolecules are still hard to capture, mainly due to the fast-dynamics and microscopic nature of these systems.

Molecular dynamics (MD) simulations are a computational technique that can complement experiments and address the aforementioned issues. MD is a simulation technique based on Newton's equations of motion and in the recent years it has attracted great attention due to its ability to simulate dynamics of biological

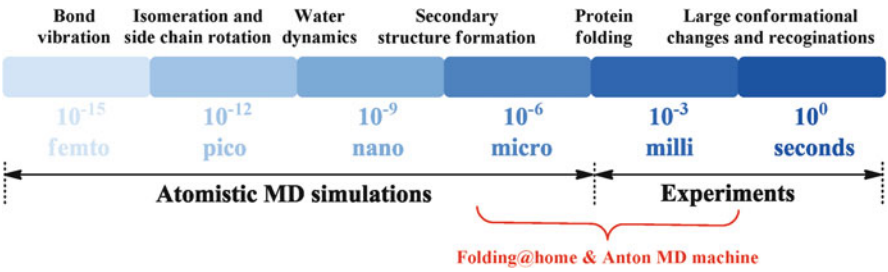


Fig. 2.1 The timescales gap between the conformational changes accessible by conventional MD simulations and the relevant biological functions observed experimentally for biomolecules. The picture illustrates the notion of timescale gap between the theory (MD simulations) and experiments, however, (*red color key*) the Folding@home project (using massive crowdsourcing computing) [22] and the specialized MD simulator machine Anton [23], have been able to MD simulate experimentally relevant timescales (μs-ms) by using its massive computational resources. However, these kinds of resources are not accessible to everyone and currently its capacity is indeed restricted to relatively small biomolecules

macromolecules [14, 15]. MD can describe the molecules dynamics in atomic detail, which is of a much higher spatial resolution than most experimental techniques. In the past decades great progress has been made in the development of force fields used for MD simulations of biologically relevant macromolecules, which had led to a more accurate description of the dynamics of protein, DNA and RNA. Furthermore, the exponential increase of computing power [16] and the development of crowd-sourcing computing [17], has allowed in recent years the simulation of biological macromolecules at timescales ranging from nanoseconds (ns) to microseconds (μs). In a limited number of cases, it has even been possible to simulate the millisecond timescale of small proteins with the aid of specialized MD computers [18], an unprecedented timescale [19, 20] that, for the first time, has permitted comparisons with experimental data and to elaborate hypothesis about the mechanisms of protein's function.

Although MD simulations have been widely applied to study protein dynamics at an atomic-level of detail, the timescales accessible by conventional MD methods are usually limited to the timescale that is orders of magnitude shorter than the conformational changes relevant to the biological function. In order to bridge this "timescale gap" [21] (see Fig. 2.1), many research efforts have been devoted to the development of theoretical methods that aim at faster sampling of the conformational space [24], some examples include: steered [25], targeted [26], accelerated [27, 28], replica exchange MD simulations [29] and metadynamics [30].

In contrast to these enhanced sampling techniques, a mathematical framework known as Markov State Model (MSM) has recently caught researchers' attention due to its potential to bridge the timescale gap. With the application of such framework, system dynamics at long timescales can be predicted by performing only short (time conventional) MD simulations [31–36]. The emergence of this promising method has thus opened the door of extracting at-equilibrium models of the complete energy landscape of biomolecules.

A number of successful examples of applying MSM have already been reported in the field of protein and RNA folding, protein-ligand binding mechanisms and the release of enzymatic reaction's sub-products. For further reference, we recommend the following notable recent examples: the Pande group has described the protein folding process of the Villin's headpiece [37], lambda repressor [38], NTL9 [20] and showed that for some protein the folded native-states are kinetic-hubs [39]; The Huang group used MSM to study the folding of small RNA hairpins [40], ligand binding mechanism of a periplasmic binding protein (PBP) [41] and the release of pyrophosphate ion from the active site of the yeast RNA polymerase II (Pol II) [42] and bacterial RNA polymerase [43]; the Noé group has used MSM to understand the folding mechanism of the PinWW protein [44]; while Bowman and Geissler have described a novel method to identify hidden allosteric sites in proteins based mainly on MSMs [45]. All these studies have demonstrated a good agreement with the available experimental observations.

In view of this widespread interest in the application of MSM to biomolecular studies, we will hereby give a general account of the theories as well as applications in the chapter. We will first introduce the basic concepts behind MSM and describe the detailed procedures for its construction, we will then illustrate the challenges of generating and validating a MSM, and finally we will employ two biological systems, the RNA polymerase (bacterial and eukaryotic) and the Lysine-, Arginine-, Ornithine-binding protein (LAO protein), as examples to explain the practical details of how MSM can be used to extract relevant kinetic and at-equilibrium information from an ensemble of short MD simulations.

2.2 Modeling the Dynamics of Biomolecules

Macromolecules, due to their high degrees of freedom and complicated molecular interaction, have numerous free energy minima in their conformational free energy landscape. In general, relatively low free energy barriers (within the order of several kcal/mol) separate these free energy minima. Because molecules are dynamic in nature at any temperature above absolute zero, and the amplitude of these motions increases with temperature, thermal fluctuation at biologically relevant temperatures are usually sufficient for the system to overcome these low conformational free energy barriers. Therefore, in most cases (if not all), at biologically relevant temperatures, what is called the native state of a protein is actually composed by a collection of protein conformations in dynamic equilibrium. To model the kinetics of such systems, a common approach is to divide the conformational space into a set of discrete states that are kinetically metastable in nature [31, 33], each corresponding to a free energy minima (or a grouped set of connected free energy minima). Therefore, the transitions between these metastable states can be approximated as the transitions between states in a kinetic scheme.

If the probability distribution of any future state $X(t + \Delta t)$ depends only on the present state $X(t)$ the transition process is known as Markovian, also sometimes dubbed as "memoryless". Such Markovian process in a kinetic scheme can be described by the memoryless Master equation:

$$\frac{dX(t)}{dt} = X(t)K \tag{2.1}$$

with $X(t)$ being an n-dimensional row vector describing the probability for the n-states to be occupied at time t. K is the rate matrix, where K_{ij} is the rate constant for the transition from state i to j. The diagonal elements of K are defined such that $K_{ii} = -\sum_{i \neq j} K_{ij}$ in order to have conservation of mass. This memoryless approximation is the underlying reason that allows modeling of macromolecular dynamics with MSM, which will be discussed in detail in the following section.

2.3 Markov Chain

To aid readers' understanding in the application of MSMs, some basic knowledge of Markov process will be first presented. A Markov Model (named in honor to Andrey Markov, who develop the theory of stochastic processes) defines mathematically a finite system (described by states) with transitions from one state to another. In this stochastic model, the fundamental assumption is that the population distribution $X(t)$ is sufficient to determine any later distribution $X(t + \Delta t)$ where $\Delta t > 0$. Under this model the states evolves over time in a probabilistic manner, and the distribution of states $X(t + \Delta t)$ after each Δt (namely propagation) depends only on its previous distribution $X(t)$, but not on any state before that. This is consistent with the "memoryless" approximation mentioned in the previous section, and thus MSM can be applied in the description of kinetics of macromolecular systems. Currently the prevailing type of Markov Model applied in macromolecular studies is known as Markov Chain, which considers an autonomous (no external contribution) process with fully observable states (occupancy of every states in the model are transparent to the observers). In a time-continuous Markov chain, the interval of propagation steps Δt is infinitesimally small such that the stochastic process can be represented as a continuous propagator. However, in the case of the applications of Markov Models for the analysis of data from MD simulations, due to the fact that MD simulations are intrinsically discrete in nature, the model most frequently employed is a discrete-time homogeneous Markov chain model in which the propagation only occurs as discrete steps. More details of this kind of MSM will be illustrated in detail in the following example.

2.4 The Transition Probability Matrix

Let us consider a simple case, suppose that a protein has three metastable states, namely: state 1, 2 and 3. If transitions only occur stochastically at some discrete time, the system can be modeled as a discrete-time Markov chain. In our example, we assume the propagation steps are equally spaced at an interval τ and the transition probabilities per propagation step are time-homogeneous (i.e. the transition probabilities depend only on Δt but not on t). Now, assume that the transition probabilities per propagation time step τ (also known as lag time) between any pair of these three states are known to be those listed in the following 2D matrix:

State	*1*	*2*	*3*
1	0.65	0.28	0.07
2	0.15	0.67	0.18
3	0.12	0.36	0.52

The previous matrix is known as the transition probability matrix (TPM). In a TPM we use the symbol p_{ij} to represent the transition probability from state i to state j. Hence, the probability of the transition from state 2 to 3 is represented as p_{23}. From the TPM above we know that $p_{23} = 0.18$. In terms of an MSM, the transition probability matrix ($\boldsymbol{P}$) is a row-normalized matrix ($\sum_j P_{ij} = 1, \forall\, i$), because the elements p_{ij} in each row represent the probability of a state i to transition to different states j and the summation $\sum_j P_{ij}$ is the probability to have a transition originating from state i to any state j. Please note that some literature uses a column-normalized matrix P^t instead of the row-normalized matrix, but in this chapter we will conform to the row-normalized matrix definition.

2.5 Propagation of the Markov Chain

As mentioned before, a Markov chain must meet the requirement that the probability of any state after the chain propagation is independent of all but the previous state. For the previous example, suppose that the initial distribution probabilities of the states follow the row vector $X(0) = [0.21,0.68,0.11]$ (i.e. 0.21 of the population is in state 1 and so forth). Because p_{ij} refers to the conditional probability of the transition from state i to j, the distribution after one chain propagation (Δt) can be calculated by: $X(0)P(\tau) \approx [0.25,0.55,0.19]$. In a similar way, the distribution after two chain propagations is determined as $\mathrm{X(0)P}(\tau)^2$. Therefore, we can write the distribution vector after the time $n\tau$ as:

$$X\,(n\tau) = X(0)[P\,(\tau)]^n \tag{2.2}$$

Given Eq. (2.2) and a vector of initial population distributions for the system states, it is possible to predict the evolution of the system on a longer timescale by the simple exponentiation of probability matrix. The Eq. (2.2) is actually equivalent to Eq. (2.1), but at discrete times, and they are related by the expression: $P(\tau) = e^{\tau \mathrm{K}}$ [32, 43].

2.6 Constructing a TPM from MD Simulations

TPM is the fundamental part of a discrete-time homogeneous MSM, because a vector of state probabilities can be propagated forward in time by simply multiplying it to the transition probability matrix. The construction of TPM is therefore the most important process in the construction of a discrete-time MSM. To construct the TPM in practice from MD trajectories, one has to first perform space discretization to group conformations in the trajectories together (details of the technique will be discussed in later sections of the chapter), because only if we consider a group of conformations as oppose to individual conformation we can empirically determine the observed conditional probability for a transition event to occur. With the conformational space properly discretized (in either microstate or macrostate level), a transition count matrix (TCM) N is then constructed by counting the total number of transitions (n_{ij}) from state i to j observed in all MD trajectories within a certain lag time τ. From the principle of detailed balance, the TCM obtained should be symmetric because all elementary transitions should be reversible under equilibrium condition. Yet due to the fact that equilibrium sampling is almost never reached in simulations (thus the need of MSM for equilibrium studies), the TCM is usually not strictly symmetric. In these cases, the TCM can be symmetrized by:

$$N^{symm} = \frac{N + N^{\mathrm{T}}}{2} \tag{2.3}$$

The TPM is then formulated by normalizing each row of the symmetrized TCM by:

$$\mathrm{P_{ij}} = \frac{\mathrm{N_{ij}^{symm}}}{\sum \left(\mathrm{N_{ij}^{symm}} \right)} \tag{2.4}$$

Simply symmetrizing the transition count matrix is the most trivial way to impose the detailed balance condition, but may introduce errors when the number of inter-state count is small or rather un-symmetric. Noé has introduced an algorithm to approximate the transition probability matrix induced by the observed count matrix. Under the framework of the Bayesian Inference, a distribution of transition probability matrices (posterior distributions) can be obtained using a Metropolis Monte Carlo scheme, subject to the constraint of the detailed balance with the observed transition count matrix as the maximum likelihood [32].

2.7 Considerations of TPM Construction

With a properly constructed TPM, it is possible to derive a MSM that can be used to understand the time-evolution of system. However, in spite of this attractive feature of MSM, it is necessary to understand that such property is founded on the Markovian assumption, but the assumption does not necessarily hold true for a kinetic scheme obtained from MD simulation. This is due to the fact that upon space discretization during the clustering step, one will introduce discretization error in the model [46]. This error is mainly produced by the existence of small internal free energy barriers inside of the discrete states, which will give rise to differences in the dynamics among the conformations existent within the state (due to the inertial effects of molecule dynamics at short timescales). In order to reduce such error, a finer discretization can be performed or a longer lag time could be chosen. None of these approaches is perfect, a finer discretization could reduce the differences in dynamics among conformations within a state, but at the same time the statistical significance of each cluster is reduced. On the other hand, by coarse-graining the simulations time into long time steps, the system can have more time to "lost its memory" so as to achieve better Markovianity. For example, if we consider the limit, theoretically any model will be Markovian at the infinity limit, despite the fact that this scenario is unfeasible in any time-dependent simulation. Yet if we could achieve perfect Markovianity under such condition, the dynamic information of the system will be completely lost. Therefore, in order to generate an insightful Markov model using MD data, apart from choosing a lag-time long enough to give a good approximation of the Markovian condition, it is equally necessary to choose a lag-time that is short enough to be useful (few ps-ns in most cases). In other words, when building an MSM from MD simulations, one always has to face the tradeoff among Markovianity, spatial resolution and preserving certain timescale-resolution of the dynamics. If we try to take into account these constraints as well as our aim of representing the energy landscape with a Markovian kinetic scheme, we can deduce one possible balance between these factors could be that the state defined in the model should be metastable (thus correspond the minima of the energy landscape), the intra-state relaxation times (the time a state takes for a conformation inside the state to transit to other conformations within the state and lost the memory) is minimal (thus all conformations can have similar kinetic behavior via fast interconversion) and the interstate transition times (the time that takes for a conformation to transit to a conformation in other state) is maximal (or in other words, high barriers lie between states), this implies the generation of metastable states without internal high free energy barriers [33, 34, 40].

If we examine closely the protocol presented above, one can discover that under the Markovian assumption the construction of a TPM does not require individual MD simulations to visit all the metastable regions in the free energy landscape [34, 44, 47]. Instead, only probabilities of local interstate transitions are necessary for constructing TPM and the corresponding MSM. This actually lessens the burden of computational cost to strive for a converged sampling with long trajectories, as

many short sampling-trajectories (just long enough to sample interstate transitions) are all that is required. Furthermore, apart from standard MD simulation, there are other sampling methods that can be applied in order to generating descriptions of the energy landscape that can be used in a MSM construction (e.g. Monte Carlo simulations), but those are beyond the scope of this chapter.

Now, the only remaining question for MSM construction is how to define the micro- and macro-states that are used to calculate the TPM.

2.8 Free Energy Landscapes of Biomolecules and Its Relation to Microstates, Macrostates and MSM

From the notion above, it can be understood that a proper partitioning of the conformational space such that the metastable states correspond to distinctive energy minima is necessary for generating a TPM that can give a valid MSM. Usually such partitioning is achieved in two stages, namely the microstate clustering and the macrostate lumping. The microstate clustering aims to generate clusters of conformations fulfilling the criteria that conformations within each cluster have similar kinetic behavior, while the macrostate lumping aims at putting the clusters generated in the previous stage together, in order to give place to larger groups, each composed of several microstates, such that the major energy barriers in the system lie between macrostates. In the case of the microstates, if we consider that an ensemble of converged MD simulations represents an "energy landscape" [44, 47], then we can assume that it is possible to generate a partition of the system just by grouping conformations that are related kinetically at short time-intervals. If the partition of the microstates is fine enough (see Fig. 2.2), a group of the microstates will correspond/minimize to the same energy-minimum basin [31, 33, 34].

Therefore, at the macrostate lumping stage (see Fig. 2.2), one can look at the transition/kinetics between the microstates in order to connect those microstates

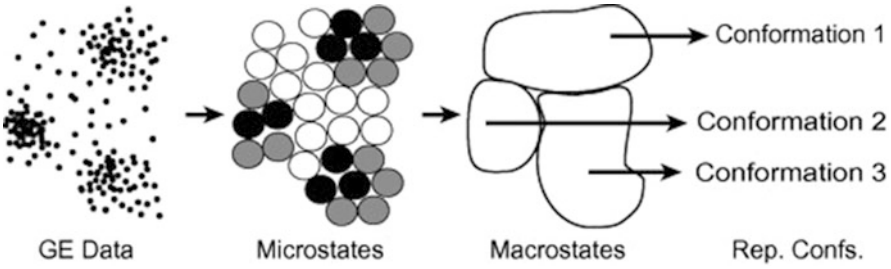


Fig. 2.2 The steps required when building a MSM. The conformations (GE data, represented by *points*) obtained from the MD simulations are firstly grouped into microstates; next, the structures are clustered in microstates based on its degree of geometric similarity. Next, the microstates are further lumped into several kinetically related macrostates (Figure adapted from reference [34])

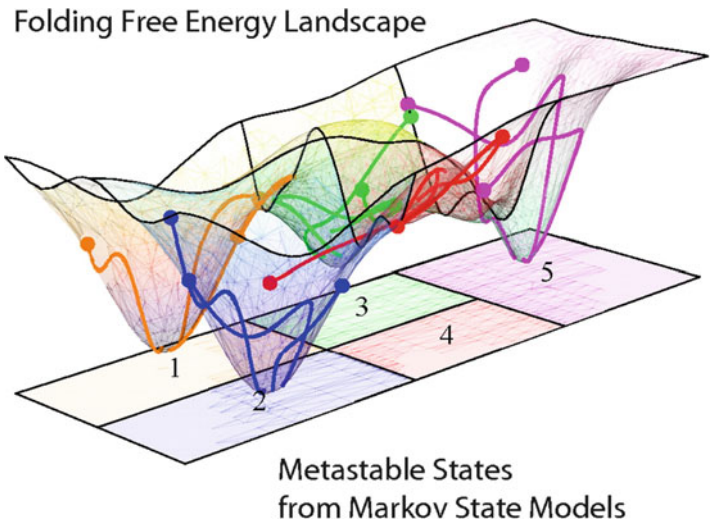


Fig. 2.3 Relationship between energy landscape, MD trajectories and metastable states. The schematic 3D free-energy landscape, comprised by 5 energy minimums, represents the conformational space of a certain protein. The different lines illustrate the behavior of 3 hypothetical MD trajectories started from different energy minima. It can be appreciated that some trajectories are able to overcome the energy barrier, escaping from its starting energy minimum to enter into another minimum, but none of the independent trajectories is able to visit the complete energy landscape (i.e. all the relevant protein conformations). Nevertheless, if considered as a conjunct the MD simulations had in fact visited all the energy landscape. The 2D projection in the bottom shows an idealized discretization of this energy landscape into the corresponding 5 metastable states (Figure adapted from reference [47])

separated by low-energy barriers (i.e. those with fast inter-microstates transitions) into a single metastable state (macrostate). In this way, by first partitioning the conformations into microstates and then lumping the microstates into macrostates the complete energy landscape can be partitioned into a small set of metastable states (see Fig. 2.3) [34], such macrostate division of the energy landscape is not only representing the underlying kinetics of the system, but also by reducing the number of states in the system (usually to less than 100) it is easier to analyze and in many cases it is also possible to extrapolate the MSM into a human-comprehensible fashion (e.g. a graphical representation showing the transitions between states). Finally, by calculating the transition probability matrix at the micro and macrostate level, we can construct and validate the MSM, which can be used to extract useful thermodynamic and kinetic properties of the dynamic process that we are interested. As explained before, any of the two models: micro- and macro-state level can be valid, however one should choice between the finer and the coarser model based. Nevertheless, usually the macrostate model is the common choice, since it is simpler, easier to analyze and its statistical certainty is intrinsically higher.

2.9 Microstate Clustering of MD Conformations

Although there is a lack of consensus about the best method to cluster kinetically related conformations, the most usual methods are based on some sort of geometrical clustering. The assumption behind structural clustering is that structures closely related in the geometrical space should also be closely related in the kinetic space, hence, grouping structures that are close in geometry will approximately give structures that are close in dynamics. Several structure-based clustering methods are already available, with the most fundamental and widely used are: K-centers, K-means and K-medoids clustering. The common goal for these three methods is to partition a set of n conformations into k mutually exclusive partitions C_1, $C_2, \ldots, C_k$. These k partitions are then used for macrostate lumping in the later stage.

K-centers clustering aims at find k "centers" (see Fig. 2.4A), which is defined by a subset S from the set of points V such that $|S| = k$ and minimizing the expression:

$$\max_{v \in V} \min_{s \in S} (v, s) \tag{2.5}$$

or, in simple words, find k points from the dataset such that the longest distance between any point to its closest corresponding center is minimized. The k partitions can then be obtained by assigning all points into their closest corresponding centers to form k mutually exclusive groups.

The k-centers problem is actually NP-hard, which implies that solving the exact solution is computationally expensive. In real practice though, k-centers clustering algorithm usually refers to an approximate algorithm shown below:

1. Randomly select one conformation as the center of the first microstate k_1.
2. Calculate the distance $d(x_i, k_1)$ between each of the conformations x_i in the dataset and k_1.
3. Choose the conformation with the largest $d(x_i, k_1)$ value as the second microstate center k_2.
4. Reassign the conformations in the dataset to the new cluster if the distance to the new cluster center is shorter than the distance to any other cluster centers (i.e. for a new cluster center k_2, conformation x_i is assigned to C_2 if $d(x_i, k_2)$ is shorter than $d(x_i, k_1)$.
5. Then choose the next cluster center that is furthest from the all previous centers and repeat step 4.
6. Repeat the same procedure until the desired number of microstates is obtained.

The k-centers clustering method can create clusters with an approximately equal geometric volume. Moreover, the clustering speed can be greatly improved by applying triangle inequality in the step of cluster assignment, which has been currently implemented in the MSMBuilder package [34, 50, 51].

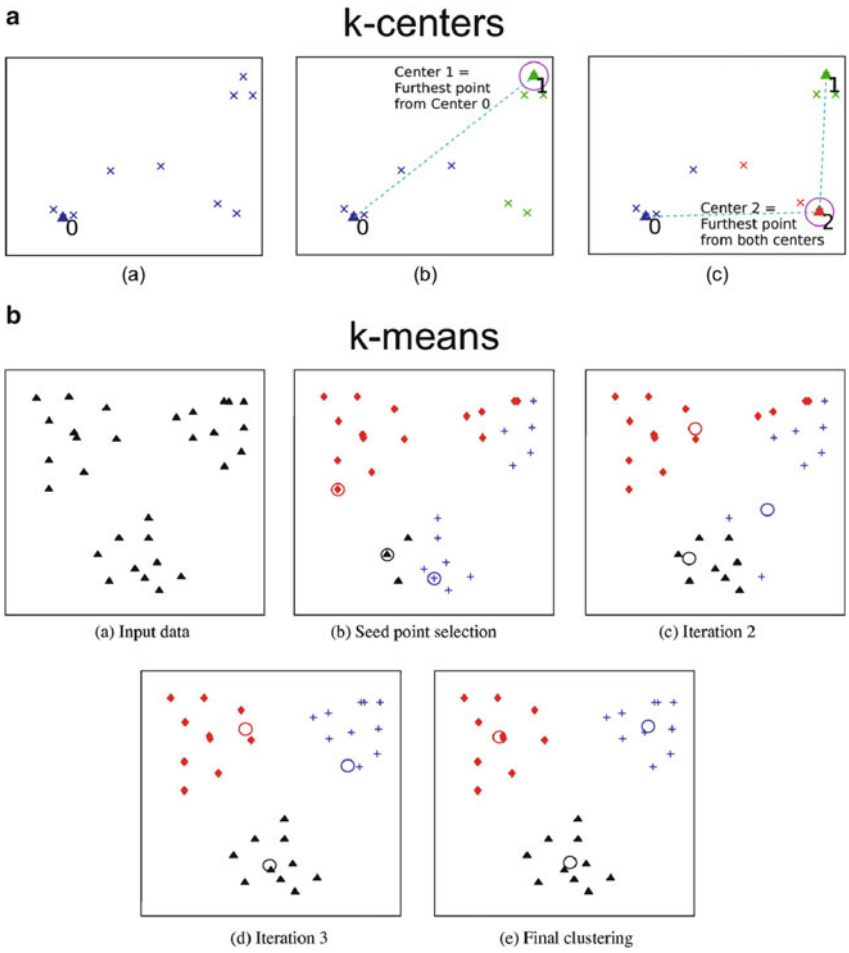


Fig. 2.4 (**A**) **Illustration of an approximate k-centers clustering algorithm.** The process of generating k geometric groups from a given dataset with the approximate k-centers algorithm is illustrated as follows: (from *left* to *right*) (*a*) From the given data points, choose a random point as the first cluster center. (*b*) Measure the distance of all points against the first center and choose the one with the furthest distance as the second cluster center. Assign all points to their closest cluster center such that all points are divided into two clusters ("partitions" in the mathematical sense), illustrated here with two different colors. (*c*) Measure the distance of all points against their assigned cluster centers, find the point with the maximum distance (i.e. furthest from all existing center) as the next cluster center. Re-assign all points to their closest centers into partitions. Repeat until the desired number of clusters k is obtained. The final partitioning is used as the geometrical grouping of the points (Figure adapted from reference [48]). (**B**) **Illustration of an approximate k-means clustering algorithm.** K-means algorithm attempts to divide the given dataset (*a*) into k geometric partitioning in the following way (From *left* to *right*, *top* to *bottom*). (*b*) From the data points, randomly choose k points as initial centers (*circled*). Assign all points to their closest corresponding centers into k partitions, shown here in different colors. (*c–d*) For each partition, take the "mean" position of the points within the group as the updated center position (*circles*). Re-assign all the points again to the new centers. (*e*) Repeat the process until no change in the cluster assignment is observed. The final cluster assignment is taken as the geometrical partitioning of the points (Figure adapted from reference [49])

K-means clustering refers to something very different from k-centers clustering (see Fig. 2.4B). Instead of aiming solely at points to be centers, the k-means clustering attempts to find k partitions so as to minimize:

$$\sum_{K=1}^{k} \sum_{x_i \in C_k} \sqrt{\left(x_i - \overline{k}_j\right)^T \left(x_i - \overline{k}_j\right)} \tag{2.6}$$

where $\overline{k}_j = \frac{1}{N_k} \sum_{x_i \in C_k} x_i$. In other words, the points are put into k partitions so that the sum of distances of all points to the partition average of their assigned partitions is minimized.

Just like k-centers, k-means is also an NP-hard problem, and so an approximation is also needed. A commonly used approximate k-means clustering protocol is illustrated here:

1. Instead of using one conformation as the first microstate center, k conformations are (randomly) chosen as the initial centers for the k microstates.
2. Calculate the Euclidean distance between every conformations in the dataset to each centers defined in step 1.
3. Assign the conformation to the microstate with the minimum distance.
4. Determine the mean vectors by averaging the distance vector for all the conformations within each microstate, and using the mean vector as the new center.
5. Repeat step 2, 3 and 4 until the clustering process is converged. That is, the new round of iteration does not change the assignments of any conformations from the previous iteration.

Despite the popularity of k-means cluster, this clustering technique is actually sensitive to the low density regions and tends to lump the points from the low density regions into the clusters from high density regions, which in the context of microstate clustering leads to the incorrect description of the some interesting states such as the transition states. A clustering algorithm that closely resembles k-means, which is known as k-medoids clustering, can overcome the above drawbacks of k-means by taking actual data points ("medoids") instead of the means of the partitions as centers. Unfortunately, k-medoids also has its own limitations, such as being inefficient for large data sets and offer a poor control of the cluster size.

2.10 Implied Timescales and Number of Macrostates

With the microstates generated from the first stage of clustering, we can construct the TPM based on the transitions between these states. Then if we attempt to do eigenvector decomposition of the TPM:

$$X_i P(\tau) = \mu_i X_i \tag{2.7}$$

where each eigenvector X_i actually corresponds to a certain state distribution that can give rise to a sustainable mode of transitions between groups of states, with the signed structure of the eigenvector indicating the two groups between which the transition occurs. The eigenvalue of each mode can be interpreted as reflecting the decay of the occupancy of the mode (N_i). If initially, the occupancy of the mode i is taken as:

$$N_i(0) = 1 \tag{2.8}$$

At $t = \tau$, the occupancy of the mode i then become:

$$N_i(\tau) = \mu_i \tag{2.9}$$

where: $\mu_i \leq 1$.

From the decaying property, the time dependence of the occupancy of each mode can be modeled as an exponential decay with decay rate constant $\frac{1}{\tau_i}$ (or τ_i as time constant), such that:

$$N_i(t) = e^{-\frac{t}{\tau_i}} \tag{2.10}$$

If we put $t = \tau$ in Eq. (2.10), and combine that with Eq. (2.9), we have:

$$N_i(\tau) = e^{-\frac{\tau}{\tau_i}} = \mu_i \tag{2.11}$$

We can then express the time constant τ_i of the decay of transition mode as:

$$\tau_i = \frac{-\tau}{\ln(\mu_i)} \tag{2.12}$$

This time constant τ_i is also known in the literature as the "implied timescale" of the transition mode. Due the fact that τ_i reflects the lifetime of a particular transition mode, it can also be used in the assessment of the timescale of the dynamics of the system and the identification of modes. A slow τ indicates a persistent transition mode, which can correspond to slow dynamics, and such modes are usually of particular interest in MSM construction. When the implied timescales are used in the determination of the Markovian time of the system, multiple transition matrices are built at different lag time τ and the corresponding sets of implied timescales τ_i are then determined (see Fig. 2.5). If all the microstates generated are ideally Markovian, all the implied timescales should remain constant regardless of the choice of lag time, but this is usually not the case in practice. It is instead expected that the implied time scale will first quickly rise and then flatten off. In such cases, the time at which all implied timescales have plateaued will be treated as the Markovian time of the system.

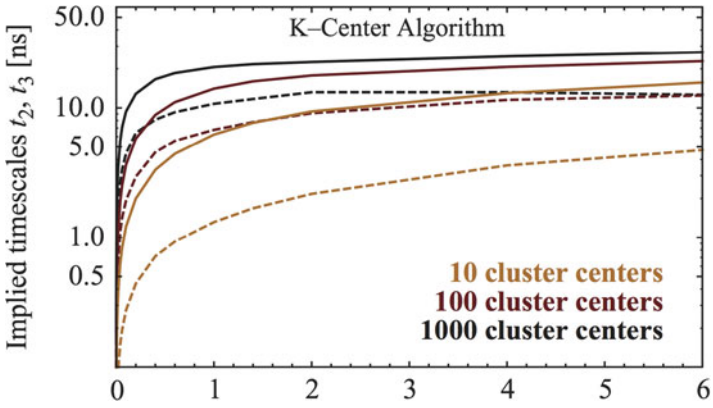


Fig. 2.5 Implied timescales convergence at different discretization levels. The plot shows the lag-time dependent implied timescales of the two slowest processes (t_2, *solid lines*) and (t_3 *dashed lines*), computed from different MSM of the MR121-GSGSW peptide. The different models (*different line colors*) correspond to k-centers clusterings of the same MD simulations data, but using different number of clusters (microstates). As it has been explained in the text, a model start to be Markovian at the shortest lag-time in which the slowest implied time scales converge, it is to say when the plot flattens. It can be seen that, as explained in the text, increasing the number of clusters (finer discretization) enhances a faster convergence of the implied timescales; hence, models with more microstates are Markovian at shorter lag-times (Figure adapted from reference [46])

2.11 Lumping Microstates into Macrostates

Under the protocol that we discussed, the second stage of MSM building involves lumping the microstates generated in the first stage into larger macrostates. The number of macrostates of the system can be chosen based on the major gap(s) between two consecutive modes in the implied timescales (see Figs. 2.5 and 2.19), as such gap indicates the slower modes and the faster modes have a significant separation in timescale (i.e. the slower modes will be significantly more sustainable than the faster ones). By choosing the number of macrostates in such way, theoretically the slow dynamics can all be properly preserved in the final model (i.e. no mixing of fast and slow dynamics in a single state), which in turn allows the model to fulfill the basic MSM requirements: (1) states are metastable, (2) intrastate transitions are fast, and (3) interstate transitions are distinct and slow.

The actual macrostate lumping can be performed using several methods, two of the most commonly used are: Perron cluster cluster analysis (PCCA), and its improved version (PCCA+). Basically, PCCA utilizes the properties of the eigenvectors and eigenvalues of the TPM to split the set of microstates into groups (see Fig. 2.6). As stated in the previous section, each eigenvector of the TPM corresponds to a certain mode of sustainable transitions between groups of states

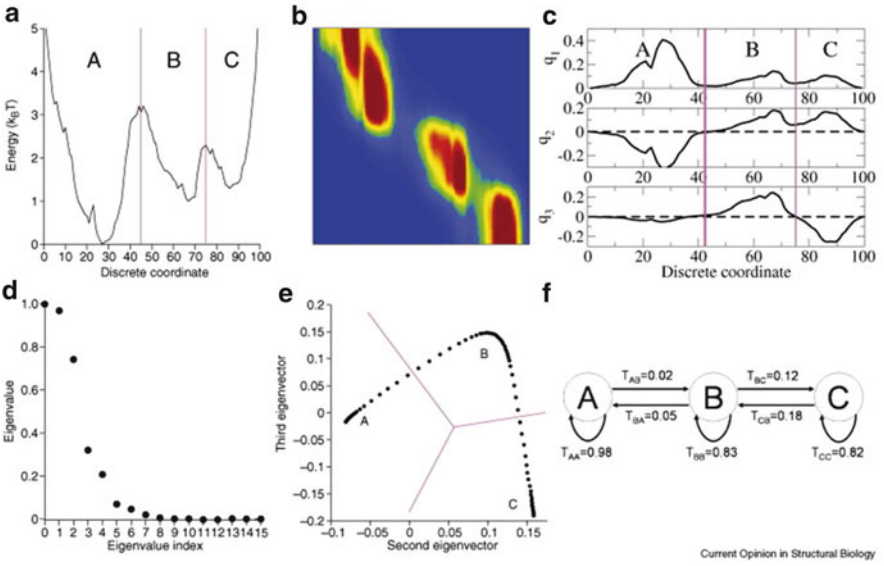


Fig. 2.6 PCCA lumping of microstates into macrostates. (**a**). Projection of the energy potential for 100 microstates onto one discrete coordinate, three energy bins were identified. (**b**). Transition matrix **T** for the 100 microstates. (**c**). Left eigenvectors of **T** indicating the transition information between different microstates. Each eigenvector corresponds to a certain mode of sustainable transitions between groups of states with the signed structure indicating the two groups. Except the first eigenvector provides the stationary distribution. (**d**). The eigenvalue spectrum of **T**. (**e**) Projections of the 100 microstates onto the second and third right eigenvectors of **T**. (**f**) Transition information for the macrostates *A*, *B* and *C* (Figure adapted from reference [33])

with the signed structure indicating the two groups. With the first eigenvector neglected (as it has an eigenvalue of 1 or implied timescale which represents the equilibrium), the set of microstates can thus be split into N groups by successively choosing the first N_i eigenvectors and partition the microstates into two groups according to their sign structure (see Fig. 2.6). After lumping we can recalculate the TCM on macrostate level, from which we can calculate the stationary distribution for the different metastable states. The next section treats a different algorithm that aims to generate multiple timescale-resolution MSMs at the macrostate level, which is based in the use of hierarchical clustering method.

2.12 Hierarchical Lumping of Microstates in Macrostates

In PCCA, microstates are lumped together based solely on the feature of TPM. Despite the mathematical correctness of the previous method, practically the method could suffer from sampling noise and cause errors in the resulted lumping. This is especially true because of the multi-resolution nature of energy landscapes [40] and

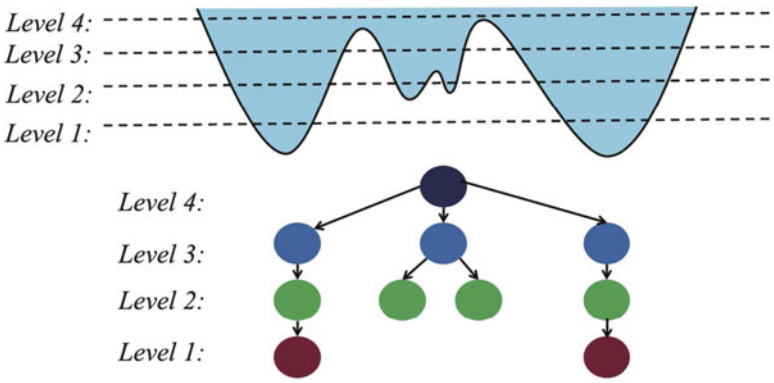


Fig. 2.7 Hierarchical clustering of microstates into macrostates. The kinetic clustering is done separately on different density levels. At first level (highest density), two separate states are identified, which are represented by the two nodes in *red*. At the next level, four separate states are identified. After the identification of states at all density levels, the connectivity between states at different level of hierarchy is identified, as shown in the figure below. The number of macrostate and their corresponding lumping can be identified from the leaf nodes of the graph (Figure adapted from reference [52])

also because the sampling quality of low populated microstates (e.g. near to high energy barriers) can be very different to the highly-populated microstates (i.e. those in the bottom of energy minima). The Super-density-level Hierarchical Clustering (SHC) introduced by Huang et al. [40] attempts to address this issue by treating the energy landscape in an hierarchical way instead of simply extracting features from the TPM (see Fig. 2.7). As stated in the previous discussions, to achieve a Markovian model the macrostates should be defined in a way that large internal free energy barriers are avoided and conformations within the same macrostate can interconvert quickly (within one lag time). Therefore, at smaller lag-times an MSM will require more macrostates to ensure that each state is small enough such that its dynamics per lag-time are memoryless. Intrinsically, shorter lag times result in higher resolution MSM that capture more free energy minima separated by small energy barriers. In the other hand, longer lag times result in a lower resolution MSM, with only a few macrostates separated by high-energy barriers. From other point of view, in a lower resolution MSM each macrostate is composed of multiple local free energy minima. SHC attempts to do lumping at different resolution by considering subsets of conformations with different densities successively, thus improves the accuracy of the kinetic lumping by treating poorly sampled states differently from states with better statistics [40]. Furthermore, for the issue that popular kinetic lumping algorithms such as PCCA and PCCA+ tend to identify poorly sampled states as being kinetically distinct from the others [40] and preserve them as metastable state in the resultant macrostate model, despite further investigations can easily show that they are likely just due to sampling noise rather than representing the true free energy minima, SHC can handle these states with very small populations separately and so the resultant model will not be skewed by these poorly sampled states.

The SHC algorithm clusters conformations hierarchically, by using super density level sets in a bottom-up fashion. It first divides the densest regions of phase space, which in a well-converged system may correspond to the free energy minima. Then, by allowing the user to fine-tune the super density level sets, this algorithm can generate multi-resolution models. From the best of the authors' knowledge, SHC is the first algorithm known to address the construction of MSM at different resolutions (see Fig. 2.7).

The SHC algorithm lump the microstate together in the following way:

1. Partition the conformations into a large number of microstates. K-centers clustering has been recommended by the authors, since it gives states with approximately uniform size, which has been proposed that this might result in a correlation between the population of each state and its density [40].
2. Split the microstates into n-density levels $L = \{L_1, \ldots L_n\}$.
3. Calculate the super density level sets $S_i = \mathrm{L}_1 \cup \mathrm{L}_2 \ldots \cup \mathrm{L}_{i-1} \cup \mathrm{L}_i$, and then each super density level also contains all previous levels $S_1 \subseteq S_2 \subseteq S_{i-1} \subseteq S_i$.
4. Perform spectral clustering, in each super density level, to group kinetically related microstates.
5. Build a graph of the states connectivity across super density levels. Then, generate a directed gradient flows along the edges of the graph from low to high-density levels. In SHC, it is denominated an attraction node (or attractive basin) where the gradient flow ends. Each attraction node is assigned to a new metastable state.
6. Assign every microstate not belonging to an attraction node, to the metastable state that it has the largest transition probability to (see Fig. 2.7).

In the SHC algorithm, the populations of microstates obtained from the K-centers clustering are used to approximate the conformation density, since K-centers algorithm can generate clusters with approximately equal radii in RMSD. However, it is extremely challenging to accurately estimate the conformational densities in high dimensional spaces, since small variances in the cluster RMSD radius may cause large differences in volume.

To address the above issue, Huang and coworkers have developed a new algorithm, which is based on the Nyström method and its multilevel extensions (HNEG) [52]. The HNEG algorithm allows us to approximate the transition probability matrix (P) with its dominant submatrix (A). Using the Nyström approximation, it can be shown that the leading eigenvectors of the submatrix A containing the most populated states (i.e. the entries in A are significantly larger than those in B and C) have the same sign structure as those of the original transition probability matrix P:

$$P = \begin{pmatrix} \mathrm{A} & \mathrm{B} \\ \mathrm{B}^{\mathrm{T}} & C \end{pmatrix} \tag{2.13}$$

Therefore, one can perform the kinetic lumping based on the eigenvector components of the submatrix A using either PCCA or PCCA+. In order to define the boundary between A and C, the same multi-level procedure as laid out in the SHC algorithm has been adopted.

In both SHC and HNEG algorithms, there are many possible choices of the super level sets (S) and each could result in different lumpings. The authors recommend trying many of them and then use Bayesian model comparison to choose the best model [52]. One additional advantage of SHC and HNEG is that they can automatically determine the number of macrostates, whereas many other methods (like PCCA and PCCA+) require the state number to be known in prior.

2.13 MSM Validation

With the microstates properly lumped into macrostates the TPM can be constructed at the macrostate level and the corresponding MSM can be built. Yet before using the MSM for any analysis, an assessment of the model accuracy is necessary. Apart from the aforementioned plateauing of the implied timescales, there also exist other tests for assessing the validity of the model. A notable method is known as Chapman-Kolmogorov Test [40, 46], which assesses the validity of the model based on the premise that the Markovian assumption:

$$P\,(t+s) = P(t)P(s) \tag{2.14}$$

must hold within the error margin (see Fig. 2.8). The actual testing procedures vary between implementations, but the general idea lies on testing the transition matrix propagated at the chosen Markovian time against the transitions counted from populations bootstrapped from simulations. Only if the transitions predicted by the MSM match with those observed in the simulation, the model is deemed valid. An implementation example of the test is shown in Fig. 2.8. Alternative approaches of model validation include application of Bayesian factors or information entropy [36, 53, 54], which will not be discussed in detail here. Apart from these pure theoretical validations, assessments can also be done via comparison with experimental data [19, 40, 41, 55, 56], which will be illustrated later in this chapter (see Sect. 2.18).

2.14 Mean First Passage Time

Having a properly built and validated MSM, kinetic information of the system can then be harvested from the model. Apart from simply propagating the TPM so as to obtain equilibrium populations, timescale information of the transition can also be

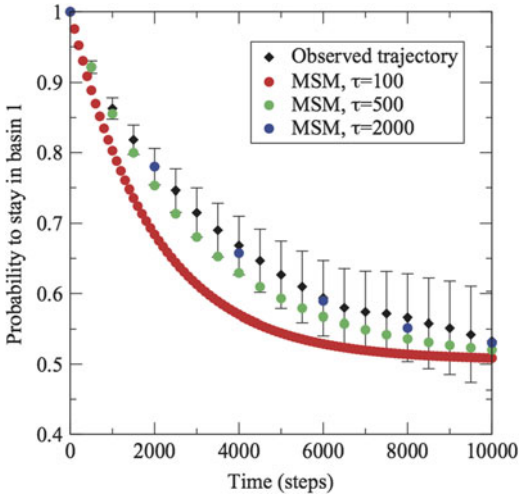


Fig. 2.8 Example of Chapman-Kolmogorov test. A representative implementation of Chapman-Kolmogorov test is illustrated here on one partitioned state. In this implementation the initial population is first solely populated on the state to be tested (i.e. $X_i(t=0)=1$, $X_{j\neq i}(t=0)=0$), the probability for this state to be occupied in the subsequent steps (which can be considered as the "self-transition" probability at the test step) are calculated via repeated propagation with TCM following Eq. (2.3) in the main text. The calculated self-transition probability is then compared against the actual occupation probability counted from MD trajectories for all propagated steps. In this example, at $\tau=100$ the self-transition probabilities propagated to different time steps clearly deviates from the ones directly observed from MD simulations, thus the MSM constructed at $\tau=100$ cannot represent the kinetics of the model properly. For the MSM constructed at $\tau=500$, the propagated self-transition probability marginally lies within the error bar, and thus can better reflect the dynamics of the state than the one constructed at $\tau=100$. The MSM constructed at $\tau=2{,}000$ gives self-transition probability that is close to the one observed from MD, and thus is considered to be the model that best conform to the Chapman-Kolmogorov equation under the test and should best represent the dynamics of the system (Figure adapted from reference [46])

obtained from the model. One of the commonly used timescale for interstate transition is known as mean first passage time (MFPT), which is defined as the average time taken for the transitions starting at state i until reaching state f for the first time, including both the direct transitions from states i to f and transitions through other intermediate states. MFPT of the transitions from i to f can be written as:

$$F_{if} = \sum_j P_{ij}\left(\tau + F_{jf}\right) \quad \text{or} \quad F_{if} = \tau + \sum_{j\neq f} P_{ij} F_{jf} \tag{2.15}$$

where τ is the lag time used to construct the transition matrix $P(\tau)$. Thus the MFPT for all transitions in the model can be determined by solving a set of linear equations defined by Eq. (2.11) with the boundary condition $F_{ff}=0$. Therefore, we

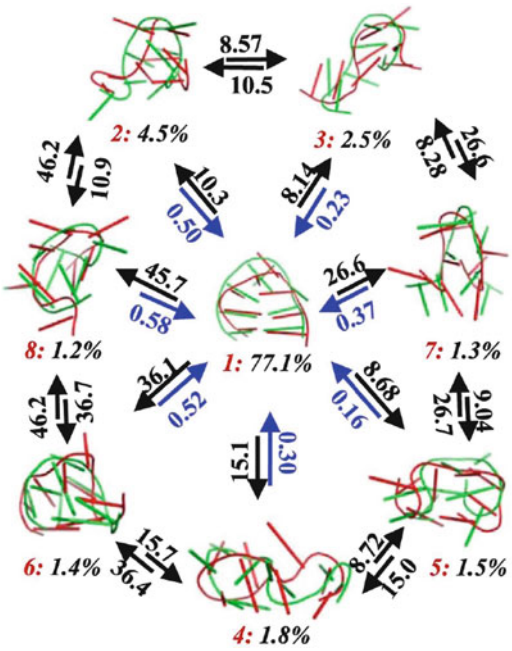


Fig. 2.9 Application of MFPT to the study of small-RNA folding mechanism. The figure shows the folding mechanism of a small RNA-hairpin tetraloop (5′-GCGGCAGC-3′). Next to the *arrows* are the corresponding Mean First Passage Times (MFPTs units: μs) between the eight most populated states in the MSM. States are labeled in *red* from 1 to 8 and the state populations are shown in *black* (Figure adapted from reference [40])

can understand that while the implied timescales describe the aggregated timescales for transitions between groups of states, the MFPT are average transition times between specific pairs or groups of metastable conformational states, and thus the MFPT calculation can provide detailed information about system's kinetics (see Fig. 2.9).

2.15 Transition Path Theory

Apart from the overall dynamical behaviors of the system, one might also be interested in some particular states or transitions. For example, in the case of a protein-ligand interaction, one might be interested more in the transitions from a unliganded protein state to a liganded protein state than the transitions between different unliganded states. Because multiple possible pathways between the two states are likely to coexist in an MSM, simple network analysis might not be adequate for such purpose. In this case, a framework known as transition path theory could be applied in order to study the relative likelihood of transitions between a particular state A to another state B [57]. Under this framework, by recursively solving for the interstate transitions between states, the pathway with highest flux, defined as highest number of transitions per unit time, can be identified. Such pathway can thus be understood as the "dominant pathway" for state A to B

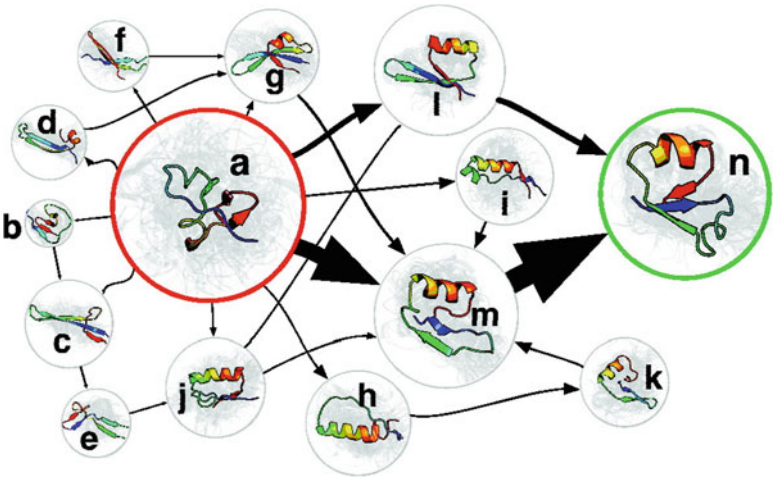


Fig. 2.10 Determination of the top 15 folding pathways of NTL9 using TPT. The transitions between the macrostates are connected with *sized arrows*. The pathway along the *larger arrows* indicate the dominant folding path (Figure adapted from reference [20])

transitions (see Fig. 2.10), and second or third pathways can thereby be identified by following this scheme. Detailed derivations are beyond the scope of this chapter and interested readers are advised to consult the relevant literature referenced for further information [33, 46, 54, 57, 58].

2.16 Visualization of MSM

As with many parts of research, on top of the quantitative analysis, visualization is also a very important part of understanding and appreciating a MSM. An intuitive approach of illustrating MSM would be to present them as graphs with macrostates as vertices and connectivity as weighted edges. Yet it is common to come across MSM with more than just a few macrostates, and the connectivity of these states may form a high dimensional network, which could make visualization difficult. Depending on the systems, solutions of visualization issue might vary considerably. Apart from plotting out the macrostate connectivity in whole or in part, if some specific representative geometric parameters exist in the system that can be used to describe the progress of the dynamics (e.g. a dihedral angle can be used to describe the rotation of a bond), conformations could be projected on such parameters and use them to describe the most prominent geometric differences between different macrostates. This is a commonly used approach for simple system as represented by the well-known system in the field: the terminally blocked alanine

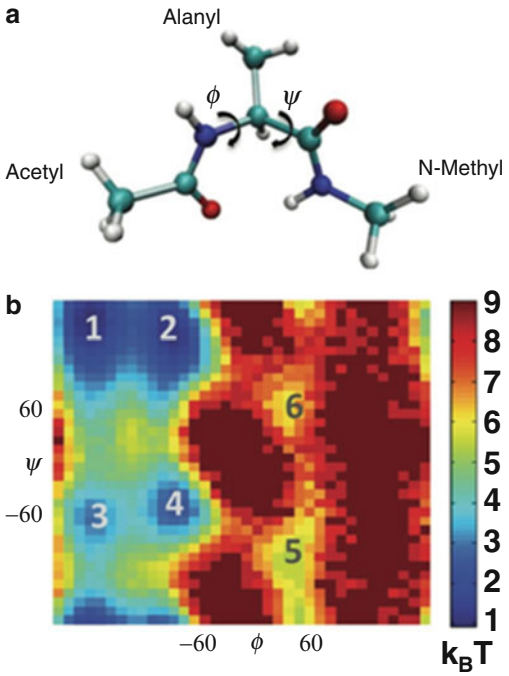


Fig. 2.11 Projections of the conformational space of the terminally blocked alanine peptide. (**a**) Illustration of the Φ and Ψ dihedral angles of the alanine in the terminally blocked alanine peptide. These two dihedral angles represent the major possible conformational changes in the system. (**b**) Energy landscape of the peptide projected onto the Φ-Ψ plane. The "energy" (shown here as the "potential of mean force") of the bins in the grid is determined from the density of the projected points in each bin. High density regions of the projection are shown as the minima of the energy landscape. With a proper choice of geometric parameters (in this case Φ and Ψ dihedral angle), geometric differences between different minima (which also correspond to six macrostates in this case) can become prominent and intuitive visualization of the states is thus possible (Figure adapted from reference [52])

peptide (NMe-Ala-Ace, also known as alanine dipeptide in some literature), which represents the peptide motion with the **Φ** and **Ψ** angle of alanine (see Fig. 2.11). If such parameters are not available, an alternative approach could be to project the conformations of interest on the first one or more principle components of the system in order to have a general understanding of the spatial distribution of the macrostates.

Recently a new program, the MSMExplorer [59], has become freely available. This Java suite allows interactive visualization and analysis of MSM built using the MSMBuilder package [34, 50, 51] (see Fig. 2.12). The representations of a MSM that it can generate includes graph of states connections, scatter plots of user definable data, visualization of hierarchical MSM and transition paths.

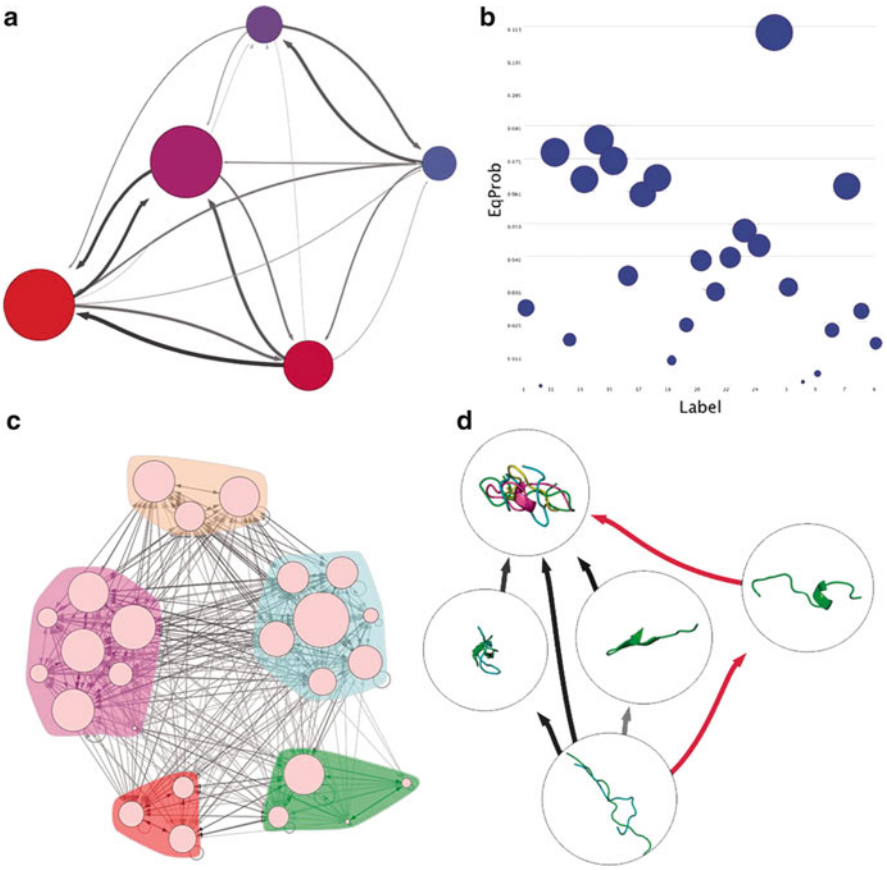


Fig. 2.12 Visualizations of MSM using the program MSMExplorer. The recently released MSMExplorer program allows to easily perform several visual analysis of MSM built using the MSMbuilder-2. For example: (**a**) Network representations of the states in the model and its connections to other states. (**b**) 2D scatter plots of arbitrary RAW data vs. state number, with visual representation of the state size. (**c**) Visualization of hierarchical MSM, in which the membership of a finer-grained model can be overlaid with a coarser-grained model, this allows the visualization of multiple resolutions of the MSM in a single plot. (**d**) TPT diagrams of the highest flux paths between two macrostates, with the advantage that images depicting conformations in each state can be overlaid on each node (Figure adapted from reference [59])

2.17 Mining Data from MSM

Constructing and validating an MSM can be challenging, but extracting relevant data from the model could be even more difficult. Some MSM construction packages have therefore offered an all-in-one solution for application of MSM in MD studies. For example, the MSMbuilder 2 package [50] offers several general tools coded in python for analysis of MSM, such as: (1) Extracting random conformations from

the micro and macrostates, which allows rapid visual identification of the structural properties of the system in each state. When one is dealing with large dataset, it is often convenient to measure physical properties (e.g. distances, SASA, RMSF, correlations, among others) in a reduced set of representative conformation of each state, which can be accomplished by randomly extracting an statistically significant number of random conformations; (2) Calculating cluster radii, which allows rapid assessment of the structural diversity among the clusters; (3) Calculating cluster RMSD to a reference structure, which can be used to identify known states of interest (e.g. folded and unfolded states, bound state or intermediates); (4) Calculating the aforementioned transition path between two states, and generate a plain text graph in a DOT file that can be visualized with a number of open source software widely available for several operating systems.

Nevertheless, as in any scientific research problem, each studied system can pose unique challenges and the users will frequently find themselves without the necessary tools to perform a particular analysis. In such scenario, in many cases it is possible to use simple Linux-like command line scripting (BASH, CSH, TCSH, etc.) to combine existing analysis tools in order to perform more complex analyses. However, the most powerful approach is to code custom analysis tools. Open-source packages that provide a framework to interact with MD trajectories are valuable aids for such customized programming, such as: MDAnalysis (in python language) [60] which can be advantageously combined with the open source NumPy and SciPy suites, VMD (Tcl/Tk) [61] and Gromacs (C) [62, 63]. Anyhow, there is no unique or simple answer of how to perform a certain analysis, we compel the readers to make an incursion in programming their own analysis tools. One can, in most cases, find that by just following the existent online literature and asking advice from more experienced programmers (in the many-existent Internet communities), it takes little time to get used to programming analysis tools.

2.18 Practical Examples of MSMs Construction

With all the basic theories and tools discussed above, readers should have already get hold of the essential techniques for applying MSM in practical biomolecular studies. In the following sections, two of our works are presented here as practical examples to illustrate how all these aforementioned techniques work in practice.

2.18.1 MSM Example #1. PP_i Release Mechanism in the Yeast RNA Polymerase II and Bacterial RNA Polymerases

RNA polymerase is a critical biological machine that is responsible for transferring the genetic information from the DNA template to the messenger RNA (mRNA) [64–66]. The nucleic addition cycle (NAC) of the RNA polymerase consists of

several steps: (1) The post-state of the polymerase contains an empty active site at register +1 site that can accommodate the incoming NTP. (2) The binding of the NTP that can form several important contacts with a critical domain of the polymerase named Trigger Loop (TL) and then fix it in a closed state. (3) The catalytic reaction which forms the phosphodiester bond. (4) The release of the produced pyrophosphate ion (PP_i) from the active site. (5) The opening of TL domain accompanied by the forward shifting template DNA by one register site, which creates a new active site and new NAC starts. Extensive experimental and theoretical studies have been devoted to understand the specific steps during the NAC, including NTP binding, TL motion, catalytic reactions and translocation.

The interplays between the PP_i release, TL opening motion and translocation have attracted extensive attentions [67, 68]. The crystallography studies indicated that the PP_i release in T7 RNA polymerase is the driving force to trigger the opening of the adjacent O-helix that allows the translocation [69]. However, the *E. coil* single molecular study did not observe the coupling between PP_i release and translocation [68]. Recent fluorescence studies suggest that translocation process proceeds shortly after or concurrently with the PP_i release in *E. coil* system [67]. Although these experimental studies shed light on the roles of the PP_i release on the translocation, the detailed mechanism of the PP_i release process as well as its role on the TL opening motion has been elusive. We have used MSM to address these questions [42, 43].

People have obtained the crystal structures of the RNA polymerase in both eukaryote and bacterial systems [70, 71]. Based on the NTP-bound RNA polymerase complexes in yeast and *T. thermophilus* (termed as Pol II and RNAP respectively afterwards), we build the PP_i-bound RNA polymerase complexes by directly cleaving the P_α-O bond to form the phosphodiester bond and the PP_i group (see Fig. 2.13). The comparison of the structures of these two complexes shows different features in the secondary channel and TL domains. These structural differences suggest that the PP_i release mechanism and its effects on the TL domains are likely to be different.

In order to obtain the initial release pathways of the PP_i group in both systems, we adopted steered MD (SMD) simulations to pull the PP_i group out of the active site. The pulling simulations were conducted along different directions with the aim of considering all the possible PP_i release pathways. Then, representative structures from the SMD simulations were chosen for the following unbiased MD simulations to erase the biases introduced by the SMD. These MD simulations were then used to build the MSM. At first, we divided all the conformations from the seeding MD simulations into hundreds of microstates by employing the K-center clustering algorithm. The distance between a pair of conformations was set to be the RMSD value of three PP_i atoms (the bridge oxygen and two phosphorus atoms). To compute RMSD, the structure was aligned to the modeled PP_i-bound RNAP complex according to the C_α atoms of the BH residues. The microstates are small, and the average RMSD values to its central conformation in each state are only $\sim$2 Å in both systems.

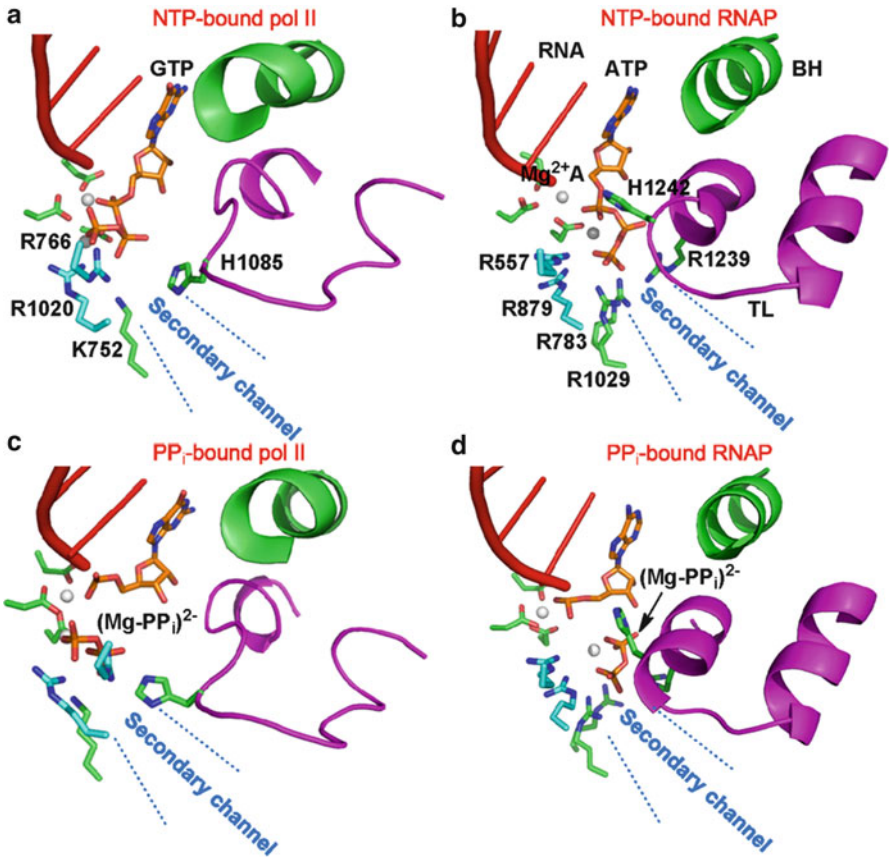


Fig. 2.13 Comparisons of the binding modes between the ligands and two RNA polymerases. (**a**) and (**b**) are the structures of the NTP-bound RNA Pol II and RNAP complexes respectively. (**c**) and (**d**) are the corresponding PP_i-bound models (Figure adapted from reference [43])

Next, we applied the Robust Perron Cluster Cluster Analysis (PCCA+) algorithm to lump these microstates obtained above into several macrostates. The number of the macrostates was determined from the major gap captured in the implied timescale plot on the microstates level. The macrostates number was finally determined to be 4 and 2 for the Pol II and RNAP system respectively.

Our results suggest that the PP_i release in the Pol II adopts a hopping mode in which four metastable states were well defined and several positively charged residues were observed to form favorable interactions with the PP_i group in each metastable state [42] (see Fig. 2.14). Furthermore, mutant MD simulations were individually performed to elucidate the specific roles of these residues on the PP_i release.

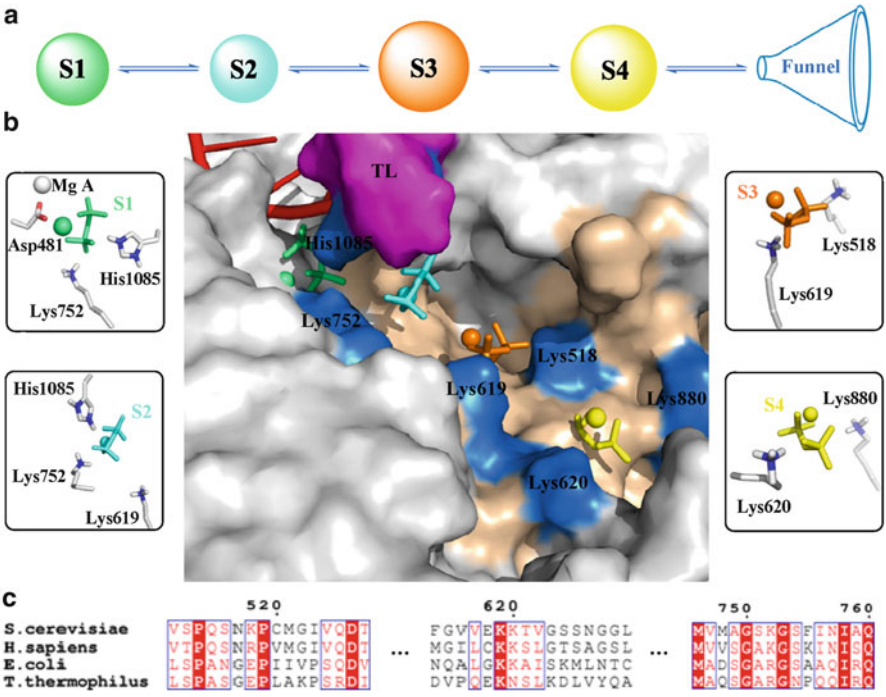


Fig. 2.14 PP_i release in Pol II adopts a hopping mode identified by MSM. (**a**). Four metastable states (S1–S4) on the releasing pathway are displayed in *sized circles* proportional to their equilibrium populations. (**b**). Key interactions between the PP_i group and the Pol II residues in each state. (**c**). Multiple sequence alignment of these positive residues in the secondary channel among different species (Figure adapted from reference [42])

However, a simpler two-state model was observed for the PP_i release in the RNAP [43] (see Fig. 2.15). We found that the difference in the number of metastable states in the release of the PPi between these systems is due to the different layout of the positive residues in the secondary channel. Specifically, in Pol II, the four residues, K619, K620, K518 and K880 are located at relatively separated sites. However, the positively charged residues in RNAP: K908, K912, K780 and K1369 are close to each other in a continuous region.

From the kinetic point of view, our MFPT calculation indicates that the PP_i release in bacterial RNAP is ~3 fold faster than that in Pol II (500 ns versus 1.5 μs), which is consistent with the faster elongation rates observed for RNAP. More strikingly, because of the higher stabilities of the TL domain in RNAP compared to that in Pol II, the PP_i release in RNAP cannot induce the backbone unfolding of the TL domain. Instead, the TL residue R1239 was observed to greatly facilitate the PP_i release in RNAP by rotating its long side chain (see Fig. 2.16). Further control MD simulations indicate that the TL domain must be exposed to the solvent before its secondary structures can be fully unfolded. And the full opening motion of the TL is likely to occur at a timescale longer than the timescale of PP_i release.

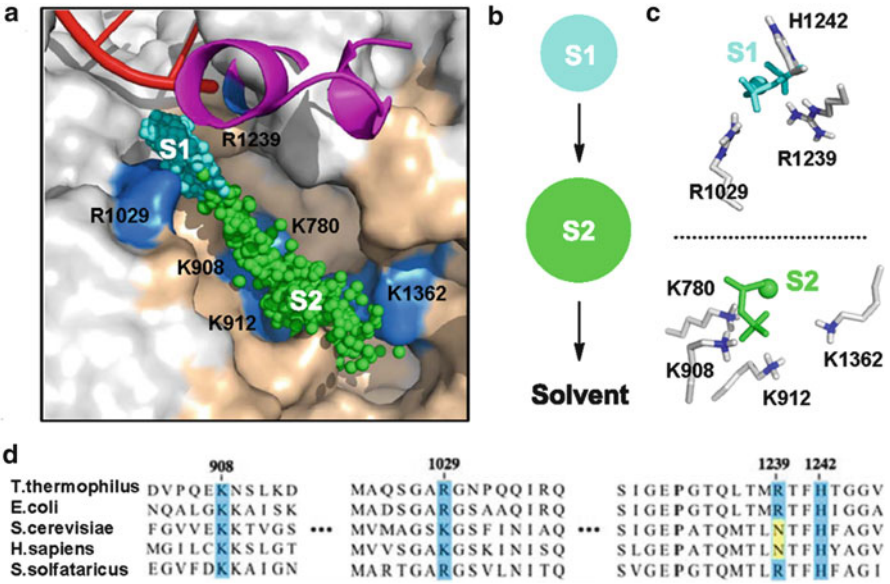


Fig. 2.15 Two-state model for the PP_i release in RNAP identified by MSM. (**a**). The distributions of the two macrostates. Each *sphere* represents the center of mass of the PP_i group. (**b**). *Sized circles* proportional to their equilibrium populations. (**c**). Key interactions between the PP_i group and the Pol II residues in each state. (**d**). Multiple sequence alignment of these positive residues in the secondary channel among different species (Figure adapted from reference [43])

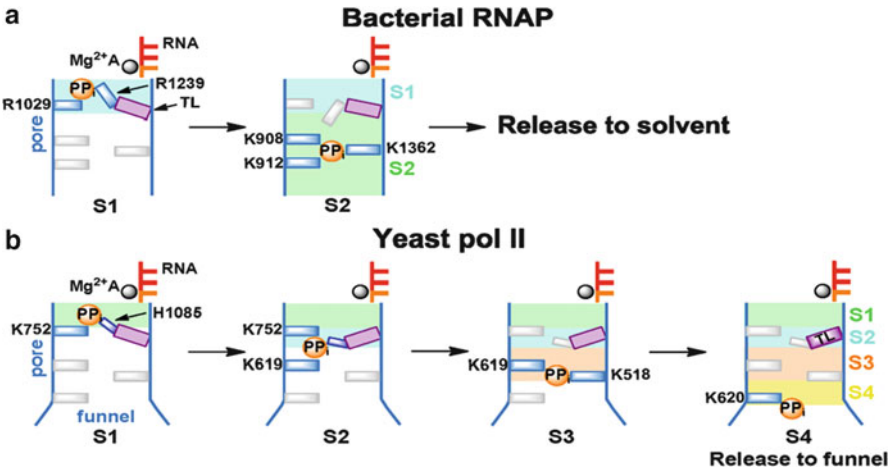


Fig. 2.16 Structural differences lead to distinct PP_i release mechanisms in RNAP (a) and Pol II (b) (Figure adapted from reference [43])

Taken together, we have built the MSM based on extensive MD simulations to investigate how the PP_i group releases from the active site in both Pol II and RNAP systems. By comparing the structural features of these two RNA polymerases, we have addressed how the structural differences influence the kinetics of the PP_i release from the active site, providing deeper insights on the structural basis underlying the transcription elongation process.

2.18.2 MSM Example 2. Ligand-Binding Mechanism in the LAO Protein

In this example, we used Markov State Models (MSM) to elucidate the mechanism by which the Lysine-, Arginine-, Ornithine-binding (LAO) periplasmic binding protein (PBP) binds to its ligand [41]. Two models of protein-ligand binding have been proposed for PBPs, the induced fit and conformational selection mechanisms, both of which attempt to explain how the protein could change from an unbound conformation to a bound conformation in complex with a ligand. In the induced fit model [72] the ligand first binds to the protein in its unbound conformation and this binding event induces the protein to go to the bound state. On the contrary, in the conformational selection model [73], the protein can access the protein-bound conformation even in the absence of the ligand, therefore the ligand can diffuse directly to the bound conformation and displace the equilibrium towards it. Using MSM, we directly monitored the mechanism of LAO binding to assess the role of conformational selection and induced fit.

We used the aforementioned MSMBuilder and SHC programs and algorithms to construct the state decomposition for our MSM of LAO's binding. We first performed 65 molecular dynamics simulations using the program GROMACS [62, 63], each 200 ns long, of the LAO protein from the organism *Salmonella typhimurium* and one of its ligands, L-arginine [74]. Ten simulations were started from the open protein conformation (PDB ID: 2LAO) with the ligand at more than 25 Å away from the binding site. The other 55 simulations were initialized from conformations randomly selected from those first ten simulations. To construct the microstate partition, we first used the k-centers algorithm in MSMBuilder to cluster our data into a large number of microstates. The objective of this clustering was to group together conformations that are so geometrically similar that one can reasonably assume (and later verify) that they are also kinetically similar. For the protein-based clustering, we created 50 clusters based on the Euclidean distance between a vector containing the protein opening and twisting angles (see Fig. 2.17). Then for the ligand-based clustering, we created 5,000 clusters using the Euclidean distance between all heavy-atoms of the ligand.

We then had to modify our clustering to account for the fact that the ligand dynamics fall into two different regimes (see Fig. 2.18): one where the ligand moves slowly due to interactions with the protein and one where the ligand is freely diffusing in solution.

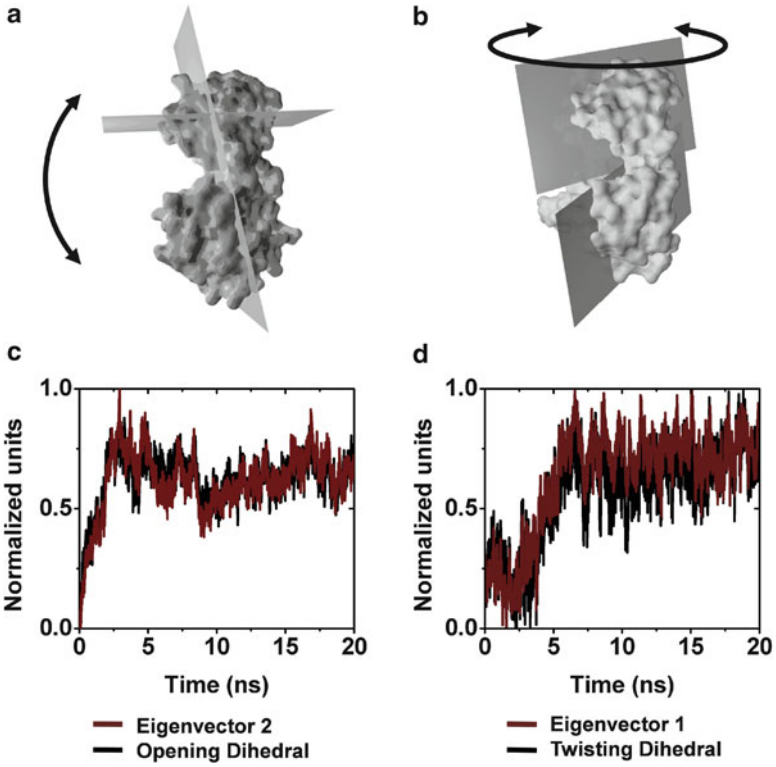


Fig. 2.17 Ad-hoc reaction coordinates used to describe the energy landscape of the LAO protein. (**a**) Opening and (**b**) twisting angles used to describe the motion of the protein. (**c**). The projection of conformations on the second eigenvector from Principle Component Analysis (PCA), and protein opening dihedral angle as a function of time are shown in *red* and *black* respectively. The 20 ns simulation is started from protein in the closed state (PDB ID: 1LAF), but ligand was not included in the simulation. (**d**) Same as (**c**) except that the projection of conformations on the first eigenvector from PCA and protein twisting dihedral angle are plotted. In this system, the twisting and opening angles are correlated well with the first and second eigenvectors from PCA (Figure adapted from reference [41])

The clusters described previously are adequate for describing the first regime, when the ligand interacts with the protein. However, when the ligand is freely diffusing (at more than ∼5 Å from the protein) the procedure outlined above results in a large number of clusters with poor statistics (less than ten transitions to other states). Better sampling of these states would be a waste of computational resources as there are analytical theories for diffusing molecules and a detailed MSM would provide little new insight. Instead, we chose to re-cluster these states using the Euclidean distance between the ligand's center of mass (as opposed to the Euclidean distance between all ligand heavy-atoms). At this stage, we created 10 new protein clusters and 100 new ligand clusters. After dropping empty clusters, this procedure yielded 3,730 microstates, of which 3,290 microstates came from the initial high

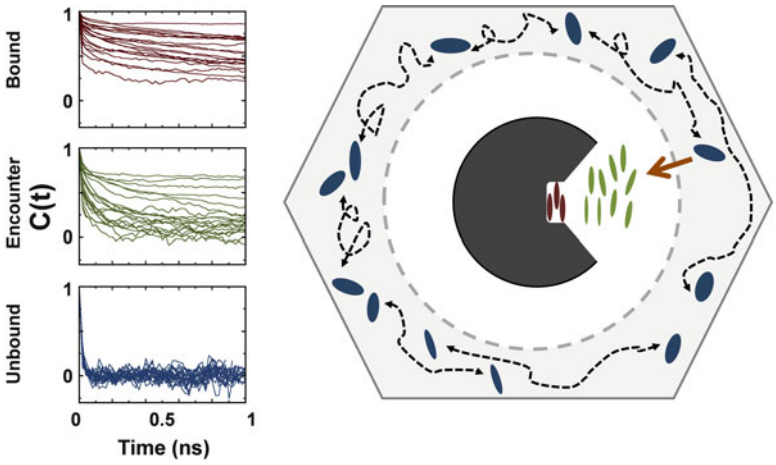


Fig. 2.18 In the presence of LAO the diffusion of the ligand L-arginine experiences two different relaxation-timescales. We found that the ligand of the LAO protein experience two very different timescales. As exemplified in the schematic figure, it can be seen that the ligand rotates quickly when it is far away from the protein but its rotation is restrained when it interacts with the protein. Thus, when constructing MSM, we only consider the ligand center of mass motion when the ligand does not have strong interactions with the protein (*blue color*) but we consider motion of all the ligand heavy atoms when the ligand is strongly interacting with the protein (*green* and *red color*). The graphs show the difference in the relaxation time of the ligand, which was assessed by analyzing its rotational autocorrelation in many independent MD trajectories, for: the unbound states (*blue*), the encounter complex state (*green*), and the bound state (*red*) (Figure adapted from reference [41])

resolution clustering and 440 came from the data that was clustered again at low resolution. To verify that the final microstate model is valid (i.e. Markovian) we plotted the implied timescales and found that they level off at a lag time between 2 and 6 ns (see Fig. 2.19), implying that the model is Markovian for lag times in this range.

We then lumped kinetically related microstates into macrostates using the SHC algorithm with density levels $L_{high} = [0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 0.99]$ and $L_{low} = [0.4, 0.95]$, for the high and low-density regions respectively. The low and high resolution states were lumped separately because the states in each set have different sizes, so it is difficult to compare their densities. We then combined these two sets of macrostates to construct an MSM with 54 macrostates. Once again, we used the implied timescales test to verify that the model is Markovian and found that a 6 ns lag time yields Markovian behavior (see Fig. 2.19).

To generate the transition matrix using the above state decomposition, we have used a sliding window of the lag time on each 200 ns trajectory with a 20 ps interval between stored conformations (i.e. each trajectory contains 10,000 conformations)

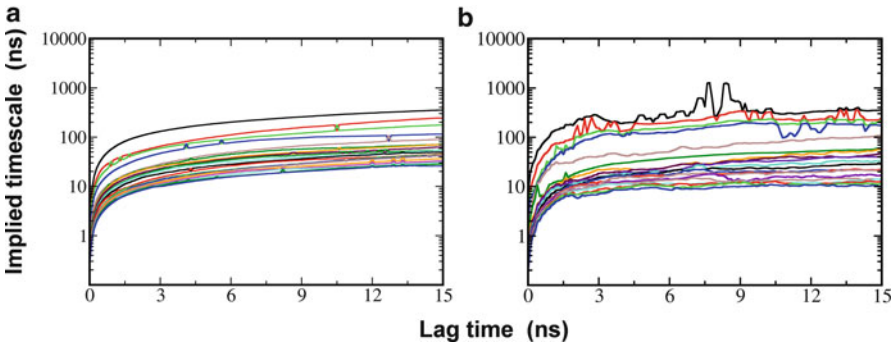


Fig. 2.19 Validating the MSM by analyzing the micro- and macro-states implied timescales. To validate the Markovianity of our model, we examined the 20 slowest implied timescales as a function of the lag time computed from: (**a**) MSM containing 3,730 microstates and (**b**) MSM containing 54 macrostates. It can be appreciated that both plots level off at a lag time of ∼4 ns, hence from this point the model can be consider Markovian. Thus we choose a lag-time of 6 ns to construct our MSM. Furthermore, it can be seen that the implied timescales in the micro and macrostate models has good correspondence, meaning that both models (micro and macro) give a similar representation of the system (Figure adapted from reference [41])

to count the transitions. Because we used a hard cutoff between states, simulations at the top of the barriers between states can quickly oscillate from one state to the other, leading to an over-estimate of the transition rate between such states. To mitigate the effect of these recrossing events, we only counted the transitions from state x to state y if the protein remained in state y for at least 300 ps before transitioning to a new state. To generate the transition probability matrix we normalized each row of the transition count matrix.

To further assess the validity of our model we also verified that the system could reproduce known experimental observables. First we confirmed that the state with the largest population closely resembled the bound conformation observed in crystals (see Fig. 2.20); we also confirmed that the model is also in reasonable agreement with the experimentally measured binding free energy and association rates. From the MFPT from all unbound states to the bound state, our model predicts an association timescale of 0.258 ± 0.045 μs, in reasonable agreement with the experimental value of ∼2.0 μs found in the highly homologous HisJ protein. Also, by using the algorithm introduced by van Gunsteren and co-workers [75] in conjunction with the equilibrium populations derived from our model, we estimate a binding free energy of −8.46 kcal/mol, in reasonable agreement with the experimental value of −9.95 kcal/mol. Together, this agreement between theory and experiment suggests that our model is in good reflection of reality.

Our final model suggests that three dominant-states need to be considered to adequately describe LAO's binding mechanism and that both: conformational selection and induced fit, play important roles in the transitions between these states (see Fig. 2.21). The third dominant state in our model—besides the previously

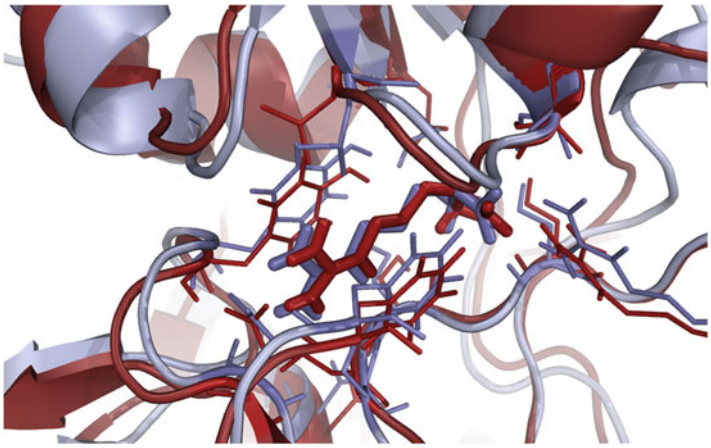


Fig. 2.20 Validating the MSM by structural comparison of the bound state. In our MSM, as in the reality, the bound state is the most populated state in the system, (in our model having an equilibrium population of 74.9 %). We used a snapshot from our simulations (structure in *red color*), to verify that the bound state is equivalent to the known crystal structure. We found that a snapshot in our model achieves a Cα-atoms RMSD of 1.2 Å (within 8 Å of the ligand C.O.M.) to the crystal structure of the bound state (*blue structure*, PDB ID: 1LAF), which confirms that the structures contained in our model are in good agreement with the experimental information (Figure adapted from reference [41])

known open and closed states—is only partially closed and weakly bound to the ligand, thus representing an encounter complex state. The ligand can induce the protein to have transition from the open state to the encounter complex (induced fit); however, the ligand-free protein can also go directly to the encounter complex state, indicating also an important role for the conformational selection mechanism (see Fig. 2.21).

2.19 Remarks and Future Perspectives

In this chapter we have reviewed the fundamental theories underlying the construction and applications of MSM, we have also highlighted that the main advantage of this method is to access timescales that are usually unreachable through conventional MD simulations. Finally, we presented two applications to illustrate the ability of MSM to investigate protein dynamics (at biologically relevant timescales) and to extract information about biological mechanisms. In conjunction with the increasing computing power, MSMs hold a great potential to address many more important problems related to the dynamics of complex biological macromolecules,

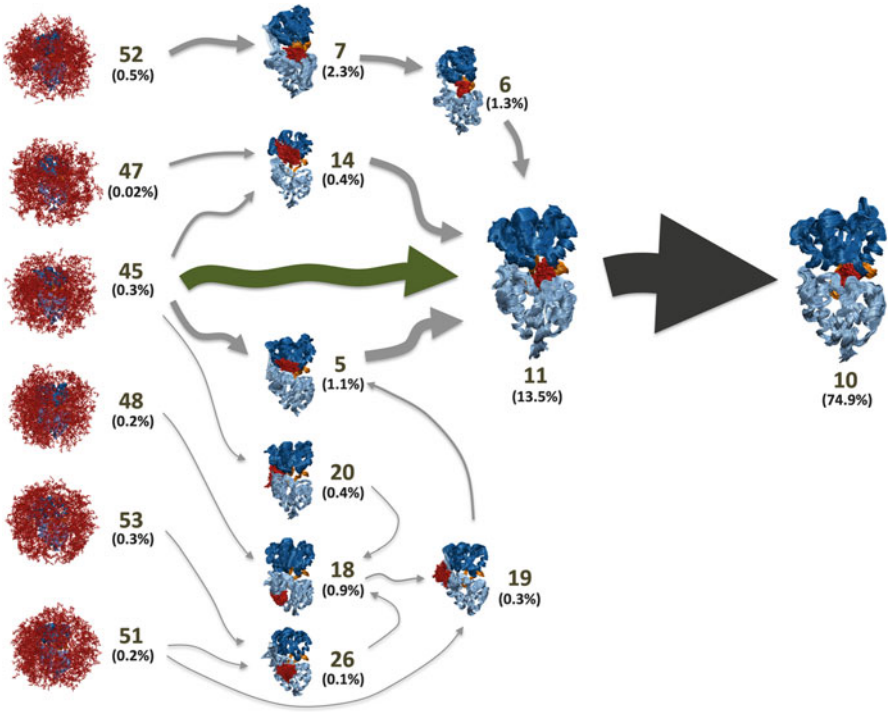


Fig. 2.21 The mechanism of LAO's binding revealed by TPT. The figure shows the superposition of the 10 highest flux pathways from the unbound macrostates to the bound macrostate. These pathways account for 35 % of the total flux from unbound states to the bound state. The conformational selection and induced fit pathways from the unbound states to the encounter complex state is shown in *green* and *grey arrows* respectively; it can be seen that the two mechanisms coexist. The arrow sizes are proportional to the interstate flux. State numbers and their equilibrium population calculated from MSM are also shown. The flux was calculated using a greedy backtracking algorithm applied to our 54-states MSM (Figure adapted from reference [41])

including problems that were impossible to attack just few years ago, mainly due to their prohibitive computational cost and the intrinsic complexity of analyzing complete free energy landscapes from a MD trajectory perspective. We envision that MSMs will be widely applied to elucidate molecular mechanisms of functional conformational changes in the near future.

Acknowledgements XH acknowledges support from the National Basic Research Program of China (973 Program 2013CB834703), National Science Foundation of China: 21273188, and Hong Kong Research Grants Council GRF 661011 and HKUST2/CRF/10. DAS acknowledges support from the PEW Charitable Trusts as postdoctoral fellow in the Biomedical Sciences. FKS acknowledges support from Hong Kong PhD Fellowship Scheme (2012/13).

References

1. Parak FG (2003) Proteins in action: the physics of structural fluctuations and conformational changes. Curr Opin Struct Biol 13:552
2. Reichmann D, Rahat O, Cohen M, Neuvirth H, Schreiber G (2007) The molecular architecture of protein-protein binding sites. Curr Opin Struct Biol 17:67
3. Mackerell AD Jr, Nilsson L (2008) Molecular dynamics simulations of nucleic acid-protein complexes. Curr Opin Struct Biol 18:194
4. Warshel A et al (2006) Electrostatic basis for enzyme catalysis. Chem Rev 106:3210
5. Kendrew JC et al (1958) A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. Nature 181:662
6. Frank J et al (1995) A model of protein synthesis based on cryo-electron microscopy of the E. coli ribosome. Nature 376:441
7. Svergun DI (1999) Restoring low resolution structure of biological macromolecules from solution scattering using simulated annealing. Biophys J 76:2879
8. Wuthrich K (1986) NMR of proteins and nucleic acids. Wiley, New York
9. Callender R, Dyer RB (2002) Probing protein dynamics using temperature jump relaxation spectroscopy. Curr Opin Struct Biol 12:628
10. Ha T et al (1999) Single-molecule fluorescence spectroscopy of enzyme conformational dynamics and cleavage mechanism. Proc Natl Acad Sci USA 96:893
11. Lippincott-Schwartz J, Snapp E, Kenworthy A (2001) Studying protein dynamics in living cells. Nat Rev Mol Cell Biol 2:444
12. Michalet X, Weiss S, Jager M (2006) Single-molecule fluorescence studies of protein folding and conformational dynamics. Chem Rev 106:1785
13. Misteli T (2001) Protein dynamics: implications for nuclear architecture and gene expression. Science 291:843
14. Levitt M (1983) Protein folding by restrained energy minimization and molecular dynamics. J Mol Biol 170:723
15. Karplus M, McCammon JA (2002) Molecular dynamics simulations of biomolecules. Nat Struct Biol 9:646
16. Schaller RR (1997) Moore's law: past, present and future. Spectr IEEE 34:52
17. Larson SM, Snow CD, Shirts M (2002) Folding@ Home and Genome@ Home: using distributed computing to tackle previously intractable problems in computational biology
18. Shaw DE et al (2007) Anton, a special-purpose machine for molecular dynamics simulation. SIGARCH Comput Archit News 35:1
19. Snow CD, Nguyen H, Pande VS, Gruebele M (2002) Absolute comparison of simulated and experimental protein-folding dynamics. Nature 420:102
20. 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
21. Schlick T, Barth E, Mandziuk M (1997) Biomolecular dynamics at long timesteps: bridging the timescale gap between simulation and experimentation. Annu Rev Biophys Biomol Struct 26:181
22. Beberg AL, Ensign DL, Jayachandran G, Khaliq S, Pande VS (2009) IEEE international symposium on Parallel & Distributed Processing, 2009 (IPDPS 2009), Italy, pp 1–8
23. Shaw DE et al (2008) Anton, a special-purpose machine for molecular dynamics simulation. Commun ACM 51:91
24. Voter AF (1997) A method for accelerating the molecular dynamics simulation of infrequent events. J Chem Phys 106:4665
25. Isralewitz B, Gao M, Schulten K (2001) Steered molecular dynamics and mechanical functions of proteins. Curr Opin Struct Biol 11:224
26. Schlitter J, Engels M, Kruger P (1994) Targeted molecular dynamics: a new approach for searching pathways of conformational transitions. J Mol Graph 12:84

27. Hamelberg D, Mongan J, McCammon JA (2004) Accelerated molecular dynamics: a promising and efficient simulation method for biomolecules. J Chem Phys 120:11919
28. Voter AF (1997) Hyperdynamics: accelerated molecular dynamics of infrequent events. Phys Rev Lett 78:3908
29. Zhou R (2007) Replica exchange molecular dynamics method for protein folding simulation. Methods Mol Biol 350:205
30. Laio A, Parrinello M (2002) Escaping free-energy minima. Proc Natl Acad Sci USA 99:12562
31. Chodera JD, Singhal N, Pande VS, Dill KA, Swope WC (2007) Automatic discovery of metastable states for the construction of Markov models of macromolecular conformational dynamics. J Chem Phys 126:155101
32. Noe F (2008) Probability distributions of molecular observables computed from Markov models. J Chem Phys 128:244103
33. Noe F, Fischer S (2008) Transition networks for modeling the kinetics of conformational change in macromolecules. Curr Opin Struct Biol 18:154
34. Bowman GR, Huang X, Pande VS (2009) Using generalized ensemble simulations and Markov state models to identify conformational states. Methods 49:197
35. Singhal N, Snow CD, Pande VS (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
36. Park S, Pande VS (2006) Validation of Markov state models using Shannon's entropy. J Chem Phys 124:054118
37. Ensign DL, Kasson PM, Pande VS (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
38. Bowman GR, Voelz VA, Pande VS (2011) Atomistic folding simulations of the five-helix bundle protein lambda(6–85). J Am Chem Soc 133:664
39. Bowman GR, Pande VS (2010) Protein folded states are kinetic hubs. Proc Natl Acad Sci USA 107:10890
40. Huang X et al (2010) Constructing multi-resolution Markov State Models (MSMs) to elucidate RNA hairpin folding mechanisms. Pac Symp Biocomput 15:228
41. Silva DA, Bowman GR, Sosa-Peinado A, Huang X (2011) A role for both conformational selection and induced fit in ligand binding by the LAO protein. PLoS Comput Biol 7:e1002054
42. Da LT, Wang D, Huang X (2012) Dynamics of pyrophosphate ion release and its coupled trigger loop motion from closed to open state in RNA polymerase II. J Am Chem Soc 134:2399
43. Da LT, Pardo Avila F, Wang D, Huang X (2013) A two-state model for the dynamics of the pyrophosphate ion release in bacterial RNA polymerase. PLoS Comput Biol 9:e1003020
44. Noe F, Schutte C, Vanden-Eijnden E, Reich L, Weikl TR (2009) Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations. Proc Natl Acad Sci USA 106:19011
45. Bowman GR, Geissler PL (2012) Equilibrium fluctuations of a single folded protein reveal a multitude of potential cryptic allosteric sites. Proc Natl Acad Sci USA 109:11681
46. Prinz JH et al (2011) Markov models of molecular kinetics: generation and validation. J Chem Phys 134:174105
47. Huang X, Bowman GR, Bacallado S, Pande VS (2009) Rapid equilibrium sampling initiated from nonequilibrium data. Proc Natl Acad Sci USA 106:19765
48. Zhao Y, Sheong FK, Sun J, Sander P, Huang X (2013) A fast parallel clustering algorithm for molecular simulation trajectories. J Comput Chem 34:95
49. Jain AK (2010) Data clustering: 50 years beyond K-means. Pattern Recognit Lett 31:651
50. Beauchamp KA et al (2011) MSMBuilder2: modeling conformational dynamics at the picosecond to millisecond scale. J Chem Theory Comput 7:3412
51. Bowman GR, Beauchamp KA, Boxer G, Pande VS (2009) Progress and challenges in the automated construction of Markov state models for full protein systems. J Chem Phys 131:124101

52. Yao Y et al (2013) Hierarchical Nystrom methods for constructing Markov state models for conformational dynamics. J Chem Phys 138:174106
53. Bacallado S, Chodera JD, Pande V (2009) Bayesian comparison of Markov models of molecular dynamics with detailed balance constraint. J Chem Phys 131:045106
54. Pande VS, Beauchamp K, Bowman GR (2010) Everything you wanted to know about Markov State Models but were afraid to ask. Methods 52:99
55. Beauchamp KA, Ensign DL, Das R, Pande VS (2011) Quantitative comparison of villin headpiece subdomain simulations and triplet-triplet energy transfer experiments. Proc Natl Acad Sci USA 108:12734
56. Zhuang W, Cui RZ, Silva DA, Huang X (2011) Simulating the T-jump-triggered unfolding dynamics of trpzip2 peptide and its time-resolved IR and two-dimensional IR signals using the Markov state model approach. J Phys Chem 115:5415
57. Weinan E, Vanden-Eijnden E (2010) Transition-path theory and path-finding algorithms for the study of rare events. Annu Rev Phys Chem 61:391
58. Bowman GR, Voelz VA, Pande VS (2011) Taming the complexity of protein folding. Curr Opin Struct Biol 21:4
59. Cronkite-Ratcliff B, Pande V (2013) MSMExplorer: visualizing Markov state models for biomolecule folding simulations. Bioinformatics 29:950
60. Michaud-Agrawal N, Denning EJ, Woolf TB, Beckstein O (2011) MDAnalysis: a toolkit for the analysis of molecular dynamics simulations. J Comput Chem 32:2319–2327
61. Humphrey W, Dalke A, Schulten K (1996) VMD: visual molecular dynamics. J Mol Graph 14:33
62. 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
63. 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
64. Kornberg R (2007) The molecular basis of eukaryotic transcription (Nobel Lecture). Angew Chem 46:6956
65. Lee TI, Young RA (2000) Transcription of eukaryotic protein-coding genes. Annu Rev Genet 34:77
66. Shilatifard A, Conaway RC, Conaway JW (2003) The RNA polymerase II elongation complex. Annu Rev Biochem 72:693
67. Malinen AM et al (2012) Active site opening and closure control translocation of multisubunit RNA polymerase. Nucleic Acids Res 40:7442
68. Abbondanzieri EA, Greenleaf WJ, Shaevitz JW, Landick R, Block SM (2005) Direct observation of base-pair stepping by RNA polymerase. Nature 438:460
69. Yin YW, Steitz TA (2004) The structural mechanism of translocation and helicase activity in T7 RNA polymerase. Cell 116:393
70. Wang D, Bushnell DA, Westover KD, Kaplan CD, Kornberg RD (2006) Structural basis of transcription: role of the trigger loop in substrate specificity and catalysis. Cell 127:941
71. Vassylyev DG et al (2007) Structural basis for substrate loading in bacterial RNA polymerase. Nature 448:163
72. Koshland DE (1958) Application of a theory of enzyme specificity to protein synthesis. Proc Natl Acad Sci USA 44:98
73. Tsai CJ, Kumar S, Ma B, Nussinov R (1999) Folding funnels, binding funnels, and protein function. Protein Sci Publ Protein Soc 8:1181
74. Oh BH et al (1993) Three-dimensional structures of the periplasmic lysine/arginine/ornithine-binding protein with and without a ligand. J Biol Chem 268:11348
75. Hünenberger PH et al (1997) Experimental and theoretical approach to hydrogen-bonded diastereomeric interactions in a model complex. J Am Chem Soc 119:7533

http://www.springer.com/978-3-319-02969-6

Protein Conformational Dynamics
Han, K.-l.; Zhang, X.; Yang, M.-j. (Eds.)
2014, XII, 488 p. 123 illus., 102 illus. in color.,
Hardcover
ISBN: 978-3-319-02969-6