MARKOV DYNAMIC MODELS FOR LONG-TIMESCALE PROTEIN MOTION

CHIANG TSUNG-HAN

B. Comp. (Hons.), NUS

A THESIS
SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE

2011

To my loving parents.

Acknowledgments

Looking back, the level of understanding I gained of dynamics is truly unexpected. As I strive out into the "*real*" world and embrace the fascinating opportunities before me, I want to thank the people who made all these possible.

I would like to thank David Hsu and Jean-Claude Latombe, for without your supervision and guidance, this thesis will certainly be impossible. I would like to thank Nina Hinrichs and people at the Folding@home project, for without your generosity in sharing invaluable data, the experiments will be impossible. I would like to thank my examiners, for without your insightful feedback, the broader potential of this thesis may remain obscured.

I would also like to thank the friends I met on this journey. To Anshul, Amit and Wu Dan who came before me, for shining a light for me to tumble along after you, precariously. To Harish, Ashwin, Difeng, Hugo and Liu Bing who went through it all with me, I am glad we found each other on this side, beautifully. To Ah Fu, Benjamin, Hufeng and Sucheendra who followed me, may you finish up nicely and expeditiously. To Deepak, Zakaria and Naveed who came a tangent to me, may the passion we shared help us all find future success, however you define it, satisfying. To those I have not mentioned specifically, my thoughts are certainly with you, affectionately.

Most importantly, I want to thank my loving family for your unwavering support over the years, the world is meaningless without any *one* of you.

Table of Contents

Summary

Molecular Dynamics (MD) simulation is a well-established method used for studying protein motion at the atomic scale. However, it is computationally intensive and generates massive amounts of data. One way of addressing the dual challenges of computation efficiency and data analysis is to construct simplified models of long-timescale protein motion from MD simulation data.

This thesis proposes the use of Markov Dynamic Models (MDMs) for the modeling of long-timescale protein motion. In a MDM, each state represents a probabilistic distribution of a protein's 3-D structure, and the transitions between states represent the change of conformation over time, *i.e.* motion. Therefore, the dynamics of protein motion can be intuitively analyzed from the explicit graphical representation of a MDM.

A principled criterion is also proposed for evaluating the quality of a model by its ability to predict simulation trajectories. This allows the most suitable model complexity to be determined, and addresses a main shortcoming of existing methods. In addition, equations are derived to compute ensemble properties of protein motion. This crucially allows MDMs to be validated against wet lab experiments.

Experimental results on the alanine dipeptide and the villin headpiece proteins are consistent with current biological knowledge, and demonstrate the usefulness of MDMs in practical use.

List of Tables

List of Figures

Chapter 1

Introduction

Proteins are essential molecules responsible for carrying out vital functions necessary for life. From enzymes promoting reactions, to hormones carrying signals from one cell to another, proteins are not only essential to the living and breathing of human beings, but also critical to all known forms of life.

Proteins' wide range of functions is due to their dynamic, yet specific, interactions with other molecules. Stabilized by strong covalent bonds and weak forces of attraction, each protein molecule is not only rigid enough to maintain a 3-D structure conducive for specific functions, but is also flexible enough to be folded from simple linear chains.

The biological importance of proteins makes the understanding of their motion dynamics crucial to furthering science. However, an intuitive abstraction of the complex dynamics is needed for human comprehension. This thesis proposes using Markov Dynamic Models (MDMs) to model protein motion as a probabilistic distribution of 3-D structures changing over time [27]. By unveiling graphically a protein's biologically significant changes at experimentally inaccessible timescales, MDMs beneficially offer scientists an opportunity to gain a deeper understanding of protein dynamics.

1.1 Protein Motion and Function

Proteins are one of the most abundant biological molecules in the cell. Critical proteins include hormones such as insulin, oxygen carriers such as hemoglobin in blood cells, the DNA replicating polymerase ... *etc.* [2, 71, 77]. The key to proteins' broad range of functions is their structural flexibility and chemical diversity. Therefore, understanding how proteins interact with other molecules, and consequently, perform their cellular functions, is critical to the molecular basis of biology.

1.1.1 Protein structure and organization

A protein molecule consists of one or more chains of polypeptides and its overall 3-D structure is known as its ***conformation***, see Fig. 1.1. Each polypeptide is a linear, unbranched chain of amino acids joined together via peptide bonds. There are many types of amino acids, and when combined into chains of different lengths, can create an infinite variety of polypeptides with distinct structural and chemical properties. The precise sequence of amino acids in a polypeptide (***primary*** structure) is determined by genetic information encoded in the DeoxyriboNucleic Acid (DNA) [19].

A polypeptide is flexible and extensively foldable due to freedoms of rotation along its backbone. It is structurally organized according to the range of interactions involved: ***secondary*** structures only involve amino acids not too far apart along the same polypeptide, ***tertiary*** structures involve farther interactions across the same polypeptide, while ***quaternary*** structures involve interactions between different chains of polypeptides. The different levels of structural organization result in a highly compact molecule that is both biologically functional and energetically stable.

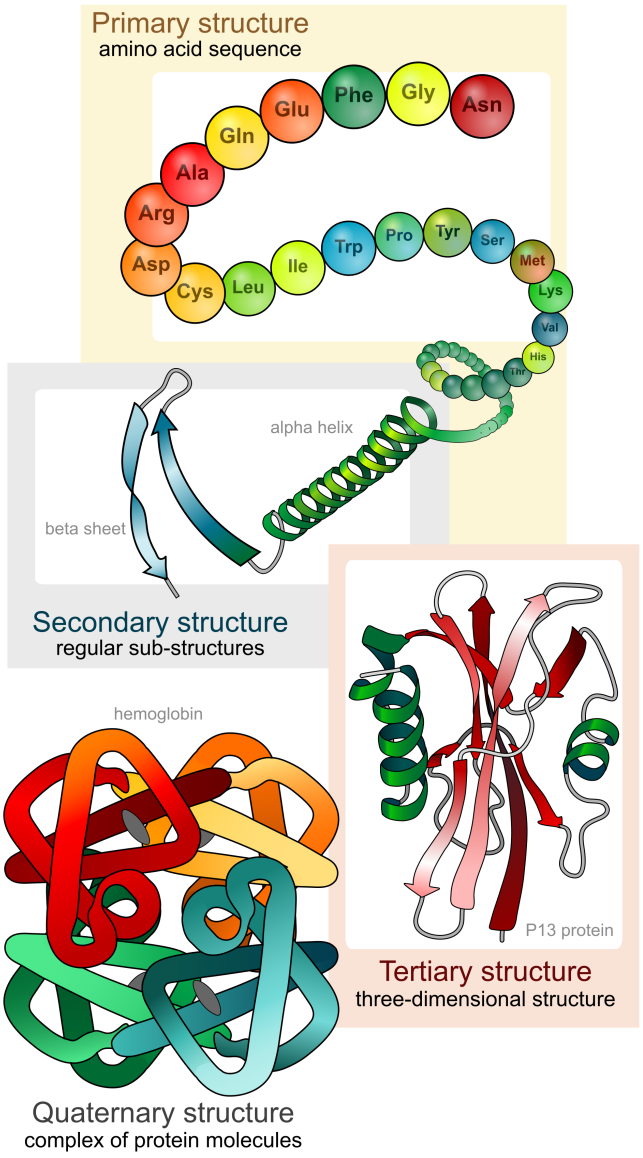


Figure 1.1: A protein's structural organization. Alanine (Ala), glycine (Gly), phenylalanine (Phe) ... *etc.* are names of different amino acids with distinct structural and chemical properties. ***Primary*** structure is the precise sequence of amino acids along a bonded chain. ***Secondary*** structures α-helix and β-sheet only involve amino acids not too far apart along the same polypeptide. ***Tertiary*** structure involves interactions between secondary structures across the same polypeptide. ***Quaternary*** structure involves interactions between different chains of polypeptides. [1]

1.1.2 Protein motion and function

Motion is critical for a protein to achieve its function. The long-range motion of folding a linear polypeptide into a compact conformation is a critical step towards cellular function. For proteins serving as enzymes, the 3-D structure of the functional or ***native*** conformation places catalytic agents at positions conducive for reactions to take place. Whereas for structural proteins, complementary 3-D structures allow multiple molecules to bind together and form larger tissues. The consistent folding of a polypeptide into a native conformation unique to its amino acid sequence remains one of the great unsolved mysteries of biology [8, 68].

However, the long range folding process is *not* the only motion. A protein in its native conformation is still structurally flexible because many of the stabilizing forces are reversible non-covalent bonds. Therefore, even "*folded*" proteins undergo constant structural rearrangements, and the native conformation is actually a set of closely related conformations [110]. For example, certain segments of a protein may slide or shear against each other locally, or open and close as if connected by a hinge. These localized motions collectively affect the way a protein interacts with other molecules. They have also led to mechanisms such as the induced fit model of enzyme action, in which a protein has to reshape itself in order to bind to a substrate and catalyze the reaction [18, 32, 41].

More importantly, it is the unique combination of different motions that allows a protein to perform its life critical function. Any mutation that changes the structural or chemical properties of a protein can potentially affect the way it folds or interacts with other molecules, and lead to debilitating illnesses such as mad cow, Huntington's, Alzheimer's and Parkinson's diseases [28, 85, 92].

1.2 Trends in Structural Biology

Structural biology is concerned with the structural basis of molecular function and is at the forefront of biology today. The goal is to understand how molecules, such as proteins, acquire their 3-D structure, and how changes in their structure affect their biological function. The trend over the past decade has been towards the adoption of ever more precise experimental techniques in order to obtain better resolution of structural changes.

1.2.1 Wet lab approaches

Ever since James Watson and Francis Crick unraveled the double helix structure of DNA in 1953 [112], scientists have striven to unravel the 3-D structure of biological molecules. Over the years, the number of 3-D molecular structures that have been confirmed has exploded (Fig. 1.2). The main reason behind this phenomenal success is the improvement in X-ray crystallography and Nuclear Magnetic Resonance (NMR) spectroscopy techniques for the imaging of proteins at atomic resolutions [78, 79].

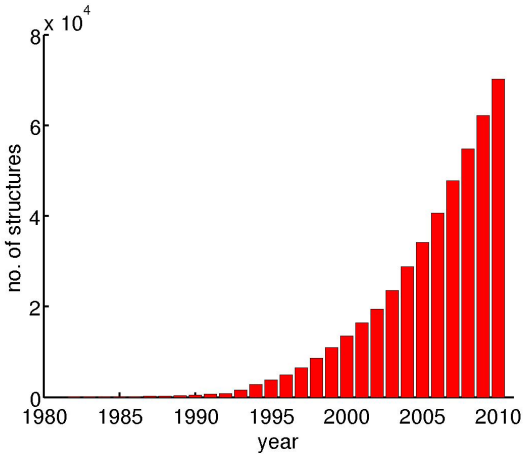


Figure 1.2: Growth in the number of 3-D molecular structures in Protein Data Bank (PDB) [15].

Both X-ray crystallography and NMR spectroscopy can pinpoint the positions of atoms relative to each other to the nanometer scale [23, 73, 91]. By reconstructing the overall 3-D geometry of a protein based on the atomic positions, scientists can understand how the relative placement of different parts of a protein can facilitate, or inhibit, its cellular function [52, 99]. Structures of mutated proteins can also be compared to investigate the effects of mutation on structure, and by extension, the folding process [88, 89].

The 3-D geometry of protein molecules is invaluable to scientists. Unfortunately, X-ray crystallography and NMR spectroscopy are severely limited by lengthy sample preparation times [78, 79]. For example, X-ray crystallography relies on the lattice structure of crystallized proteins to scatter X-ray in a reconstructible diffraction pattern. However, purifying and crystallizing proteins can take months, or even years for difficult cases. Although NMR spectroscopy does not use crystallized proteins, the resource intensive process of culturing and purifying proteins is still unavoidable.

More importantly, it is difficult to *directly* observe protein motion in 3-D. Since X-ray crystallography relies on crystallized proteins, it only provides a static view of fixed structures. Although NMR spectroscopy handles proteins in solution, the information derived is rather indirect. A typical wet lab approach relies on exposed parts of a protein to uptake deuterium isotopes from the solvent faster than other parts hidden within the protein's structure [78, 79]. By stopping the reaction at various times and measuring the difference in deuterium uptake with NMR spectroscopy, the folding process can be *inferred*. However, this approach is far from a comprehensive view of proteins in motion.

1.2.2 Computational approaches

Fortunately, advances in computer hardware and algorithms are making computational methods increasingly feasible for studying molecular motions. Early successes include investigations into short range motions of molecular binding [57, 96], and the flexibility of native conformations [95, 108]. The wealth of structural information in Protein Data Bank has also enabled scientists to deduce the structure of mutated proteins by comparing sequence similarity to known structures [62, 83, 113].

However, great potential still exists in Molecular Dynamics (MD), which is the computational simulation of molecular motions based on statistical mechanics [36, 44, 66]. MD simulation computes successive changes to all atoms in a molecular system by integrating Newtonian physics at the femtosecond timescale (10^{-15} s), *i.e.* $F = -\nabla V$, where V is the potential energy of a conformation, and F is the resultant force acting on it. The resulting trajectory is a temporal sequence of the positions, velocities, and even higher order derivatives of all atoms in the simulated system.

MD simulation not only allows scientists to directly visualize the precise motion of a protein molecule as it folds or binds with a substrate. More importantly, the wealth of information available from MD simulation is impossible to obtain with existing wet lab techniques.

With today's petaFLOPS computers, thousands of atoms can be accurately simulated for up to a millisecond (10^{-3} s) in real time [14, 94]. Although sufficient to study proteins with 30 amino acids, there are plenty of more complex molecules to be investigated. Fortunately, scaling up MD simulation is an actively pursued research area, notable projects including IBM's Blue Gene [3], and the distributed computing Folding@home project [14].

1.3 Challenges in Modeling Protein Motion Dynamics

The dynamics of a protein's motion is about its ***change of conformation over time***. More specifically, this includes both the *direction* and *magnitude* of the change, as well as the *time* of the change. In addition, scientists want to understand what makes a protein change its conformation. Therefore, capturing the precise *sequence* of events is important. A better understanding of the underlying factors that determine protein motion will allow novel molecules and better drugs to be designed and engineered *de novo*.

Like any scientific pursuit, gaining a better understanding requires a continuous cycle of making observations, formulating hypothesis, and testing predictions. Modeling is an integral part of this process, and a good approach should allow scientists to formalize theories into understandable representations, for the validation and prediction of future outcome.

1.3.1 Massively distributed MD simulation

However, the molecular nature of a protein's structural changes make direct observations in the wet lab difficult. Therefore, MD simulation at the atomic resolution is a very attractive experimental alternative.

In order to accurately simulate protein motion, MD simulation has to be carried out at the femtosecond timescale (10^{-15} s), and sustained up till the biologically interesting milliseconds (10^{-3} s), or even seconds [56, 64]. Moreover, for a realistic simulation, a large number of protein molecules has to be simulated to better represent the diverse motion of individual protein molecules in actual solution. Due to these considerations, large-scale MD simulation is usually required, and gathering sufficient data for modeling is a significant challenge in itself [3, 14, 39, 94].

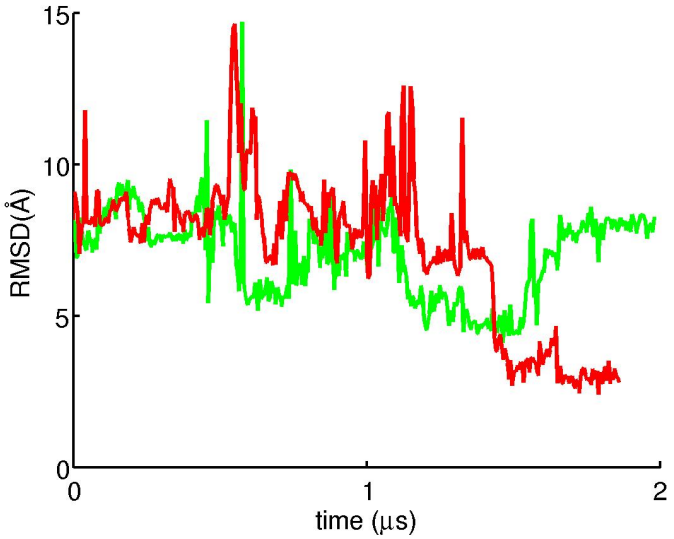
1.3.2 Abstraction for a better understanding

Unfortunately, gaining a conceptual understanding by direct data analysis of MD trajectories is not very effective, and considering the massive amounts of data, can be humanly impossible.

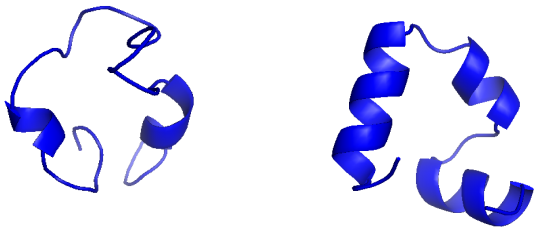
For example, Fig. 1.3 shows two MD trajectories of villin headpiece protein that started from the *same* initial I_0 conformation. However, at around 1.5 µs, we can see that one trajectory achieves the native conformation, while the other came close temporarily, before deviating significantly again. Scientists want to know: "***Why?***"

Traditional direct data analysis is rather tedious. To know the difference between the trajectories in Fig. 1.3, it is necessary to visually inspect how the 3-D structures change at 1.5 µs. However, there can be thousands of trajectories to compare. Furthermore, due to the stochastic nature of molecular motion, similar events can occur at different times for different trajectories. It is even more difficult to understand the *sequence* of events. The RMSD in Fig. 1.3 is only with reference to the native conformation. In order to discover *intermediate* conformations along the folding process, it is necessary to include other reference structures for comparison. This either requires *prior* knowledge of the protein, or a brute force comparison against all possible intermediates. More crucially, theories of mechanisms have to generalize over individual MD trajectories, and yet, be applicable for all protein molecules with the same sequence, under the same conditions.

Consequently, it is crucial to construct an accurate model of protein dynamics that abstracts away unnecessary details, and reveal the biologically interesting events in an *easily comprehensible* representation. Without which, the MD trajectories painstakingly obtained from large-scale simulations will be of rather limited use.



a) RMSD of all heavy atoms to the native conformation.



I_0 PDB: 2F4K

b) Initial (I_0) and native (2F4K) conformations.

Figure 1.3: MD trajectories of villin headpiece protein from the Folding@home project [14, 40]. *a*) Two trajectories were started from the same initial I_0 conformation. Between 1.3 µs and 1.5 µs, the red trajectory quickly achieved the native conformation with a RMSD ≈ 3 Å. While at the same time, the green trajectory also came close to the native conformation, but quickly deviated afterwards. **RMSD** is the *root mean square deviation* between the Cartesian coordinates of corresponding atoms in two conformations. For two conformations q and r with n atoms each, $RMSD(q,r) = \min_T \sqrt{\frac{1}{n}\sum ||q_i - Tr_i||^2}$, where T is a rigid body transform that minimizes the deviation between the two sets of atomic coordinates [51]. Due to atomic fluctuations, an *exact* match with RMSD = 0 Å is difficult to observe in practice.

1.3.3 Model selection

A key question that arises when constructing a model is:

What is the most suitable model?

This is an important consideration because it is possible to construct different models from the same set of data. Although a model with a greater number of parameters has the ability to better fit data, an over-complex model can also fail to generalize over training data and lose its predictive accuracy on unseen data. On the other hand, although a simpler model may be easier to interpret, a model can be too simplistic to provide any useful information. Since each model offers a different interpretation of dynamics, it is crucial to have an appropriate criterion to compare between different models so that the most suitable model can be identified.

1.3.4 Experimental validation

The computational modeling of biology is only possible due to the culmination of scientific advancement over the centuries. From biology to biophysical theories, then from MD simulation to models of dynamics, an important question is whether the resulting MDMs are still biologically accurate.

The ultimate test of accuracy is a direct validation of computational results against wet lab experiments. However, the molecular nature of protein motion makes it difficult to observe directly. This means that only ensemble properties (*e.g.* a protein's average folding time) that are measurable in the wet lab, are usable for comparison and validation.

Computationally, this requires a model to generalize over the individual trajectories used for its construction, and accurately capture a protein's ensemble dynamical properties. More specifically, experimental validation

requires computable equations that can provide numerical quantities for comparison against corresponding values measurable in the wet lab. In addition, the way the quantities are computed has to adhere closely to scientific theories explaining the dynamical property being compared. It is only with such experimental validations that computational models can be relied upon for gaining scientific understanding.

1.3.5 Computational efficiency

The *space* and *time* efficiency of modeling protein motion dynamics are significant challenges. MD trajectories are huge datasets, and building a compact model that can summarize only the essential details is critical. Although compactness suggests a simple model, simplicity alone is insufficient. To understand protein motion, we need compact models that can identify both the biologically significant *conformational changes*, as well as the *time* of the corresponding change.

More importantly, to be truly useful, a modeling approach must model a protein with minimal prior knowledge of its motion. This requires an efficient search for the most suitable model and the interesting timescales. Consequently, the efficiency of the *overall* modeling process significantly outweighs the time it takes to construct a *single* model at *one* timescale. In addition, the choice of a suitable initialization that allows model parameters to be efficiently optimized is going to be crucial to the success of the modeling approach.

1.4 Contributions and Thesis Overview

1.4.1 Contributions

The main contributions are:

- The Markov Dynamic Model (**MDM**) proposed here accurately models long-timescale protein motion as a graphical model that intuitively identifies both the interesting motions, and the relevant timescales for analysis.
- A principled criterion is proposed for evaluating the quality of a model based on its likelihood on MD trajectories. This allows the most suitable model complexity to be determined, and addresses a main shortcoming of existing methods.
- Equations are derived to compute ensemble properties of protein motion. This crucially allows MDMs to be validated against wet lab experiments.

1.4.2 Overview of Thesis

This dissertation is organized as follows:

- **Chapter 2** covers the background of the thesis, including a brief outline of the historical developments and techniques relevant to the study of protein motion dynamics.
- In **Chapter 3**, Markov Dynamic Models (**MDMs**) is proposed for the modeling of long-timescale protein motion. Motivation for modeling the dynamics of an energy basin as a *hidden* state is discussed. Model construction procedure is given. Results on the widely studied alanine dipeptide protein demonstrate the key contribution towards gaining biological understanding.

- In **Chapter 4**, a hierarchical model of protein motion dynamics is proposed to scale up the modeling approach. Reasons for the hierarchy, the relevance to biology, as well as the gain in space and time efficiency will be discussed. Model construction procedure is given. Results on the larger villin headpiece protein demonstrate the usefulness of MDMs for practical scientific research.

- In **Chapter 5**, equations to compute ensemble properties are derived. Ensemble quantities such as mean first passage time are measurable from wet lab experiments. The equations here allow MDMs to be directly validated against wet lab experiments. Models of alanine dipeptide and villin headpiece are validated here.

- Finally, **Chapter 6** concludes with a summary of the thesis and discusses areas with potential for future development.

Chapter 2

Background

Many attempts have been made in the past to model protein motion dynamics. Initially, simple approximations have often sufficed because data with accurate dynamics is scarce. Since long simulations are harder to obtain, the range of motion that can be studied is also limited. However, with rapid improvements in MD simulation, data is becoming more readily available. Consequently, the need for better analysis is becoming increasingly urgent.

In this chapter, various approaches will be discussed. In Section 2.1, the class of *graphical models* is highlighted due to their many desirable properties. In particular, the pictorial representation of graphical models is extremely beneficial for analysis. By representing the global relationship across a system as local connections between individual components, graphical models allow complex interactions to be intuitively presented and easily comprehended.

In Section 2.2, other approaches are also discussed due to their applications in specific areas. Although these techniques are more limited, and some do *not* have an explicit model, they can still be helpful as a pre-processing step, or when the range of motion is constrained.

2.1 Graphical Models of Protein Motion

This thesis proceeds from a series of developments that started with adapting motion planning algorithms from robotics to model molecular motion [59, 65]. The relevance of robotics to biology is due to the similarity between a robot's *configuration* and a protein's *conformation*. The ***configuration*** of an articulated robot is its overall shape, and is usually encoded as orientation angles of segments of a robot with respect to each other. A protein's ***conformation*** can be similarly encoded as (ϕ, ψ) rotation angles along its polypeptide backbone [19]. The similarity in their representations makes motion planning algorithms adaptable for protein motion.

In Section 2.1.1, the probabilistic roadmap models are the very first adaptation from robot motion planning. However, without timing information, it is actually *not* a model of *dynamics*.

In Section 2.1.2, initial Markov Dynamic Models (MDMs) show how time can be incorporated, but are unspecific in how the states should be defined.

In Section 2.1.3, the *point*-based MDMs attempt to model each conformation (without velocity) as a state. However, this violates the Markovian property because velocity is dependent on history.

In Section 2.1.4, the *cell*-based MDMs attempt to correct the problem of point-based MDMs by modeling a *region* of conformation space as a state. However, without a systematic criterion for evaluating the model quality, it is difficult to determine the most suitable model *without* prior knowledge of the protein, *i.e.* number of states.

Consequently, existing graphical models have limited use in practice. This is because without being able to determine the number of biologically significant states a protein has, it is difficult to apply existing methods to investigate new proteins with less well understood dynamics.

2.1.1 Probabilistic RoadMap models (PRMs)

PRMs are originally used to control the motion of complex robots [59, 65]. The goal is to compute a continuous motion that changes a robot from a *starting* configuration to a *destination* configuration, without collisions. More precisely, a PRM for a robot is an *undirected* graph. Each node q in the graph represents a feasible configuration, and an edge between two nodes q and q' represents a *reversible*, collision-free motion that connects them. By creating a graph with nodes broadly sampled from the space of all feasible robot configurations, a PRM can be constructed to control and move a robot safely to anywhere within its range of possible motions.

The PRM approach was first adapted to model the motion of a flexible ligand binding with a protein [96]. The modified roadmap is a *directed* graph. Each node in the graph represents a sampled ligand conformation, and each directed edge represents the change from one conformation to another. Additionally, a *heuristic weight* is assigned to each directed edge to reflect the energetic preference for changes that lead to a lower potential energy. The different paths in the constructed graph represents the different ways a ligand can move and bind with a protein. By searching the graph for paths of least resistance (*e.g.* Dijkstra's algorithm), PRM has successfully been used to predict the active binding sites of proteins [96], and the dominant order of secondary structure formation in protein folding [6].

The main contribution of PRM is in opening up a new class of algorithms for modeling protein motion. However, the heuristics based PRM is actually *not* a model of *dynamics*. This is because the heuristic weight is only an indication of the *preference* for change based on the difference in potential energy, while the *timing* of the corresponding change critical to the actual dynamics is left unspecified.

2.1.2 Markov Dynamic Models (MDMs)

In order to incorporate time, a PRM can be transformed into a Markov Dynamic Model (MDM) with *stochastic transitions*. Instead of heuristic weights, each edge of the graph now represents a probabilistic *transition* that occurs over a certain unit of time. Each graph node becomes a *state* with "*clocked*" transitions. In this way, the motion dynamics can then be modeled as the state-to-state transitions taking place over time.

However, the inclusion of time requires the issue of history to be considered in the modeling process. More specifically, the length of history to take into account for each transition has to be well defined. Consequently, the Markov assumption is enforced to explicitly bound the temporal dependency [16, 61]. A *first-order* Markov chain is simply this:

> Given the *current* state of the system s_t at time t, the *future* outcome of the system s_{t+1} is *independent* of its *past* $(s_0, \ldots, s_{t-2}, s_{t-1})$

$$p(s_{t+1}|s_0, \ldots, s_{t-2}, s_{t-1}, s_t) = p(s_{t+1}|s_t). \qquad (2.1)$$

In many applications, the common practice is to approximate the dynamics by discrete transitions uniformly spaced in time, and a set of conditional transition probabilities invariant with time.

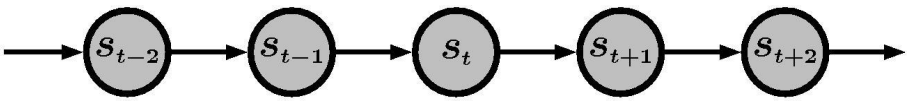


Figure 2.1: A first-order Markov chain. The probability of transitioning from state s_t at time t, to state s_{t+1} at time $t+1$, is independent of the past $(s_0, \ldots, s_{t-2}, s_{t-1})$.

2.1.3 From PRMs to point-based MDMs

The first MDM applied to the analysis of molecular motion treated each node in a PRM as a Markov state, and assigned each edge (q, q') a *transition probability* derived from the energetic difference between the conformations corresponding to q and q' [9]. The transformation to MDM is crucial in allowing a protein's conformational changes to be *temporally* correlated with the time-step of individual transitions. Additionally, the probabilistic transitions embody the stochasticity of molecular motion. Since each state represents a *single* conformation, we call this model a *point-based* MDM.

The point-based MDM was used to efficiently compute a protein's probability of folding (p-fold) [9]. The p-fold value measures the progress of folding on a scale between 0 to 1, with 0 indicating a protein is totally *unfolded*, and 1 being totally *folded* [38]. P-fold is calculated based on the ensemble of *all* possible motion pathways a protein can follow, and is a significant improvement over the graph search algorithms previously used in PRMs. Crucially, p-fold enables the dominant energy barrier that limits the rate of folding to be characterized, and then used to computationally predict wet lab experimental measures of folding kinetics, such as folding rates and ϕ-values [25, 26].

By now, what is becoming evident is that a good sampling of conformations and an accurate measure of time are both necessary to model dynamics. Consequently, an improved sampling method made use of MD trajectories to create the states of a MDM, and thus obtained better coverage of biologically relevant parts of the conformation space [97]. It is also apparent that the intuitive analysis made possible by graphical models is useful for modeling molecular motion.

2.1.4 From point-based to cell-based MDMs

The transformation from a *point-based* MDM to a *cell-based* MDM is an attempt at correcting a number of problems.

In a ***point-based*** MDM, a state represents a single conformation *without* velocity. However, a single conformation rarely contains sufficient information to guarantee the Markovian property fundamental to MDMs. The reason is that a protein's motion is determined by both its momentum and the instantaneous forces it experiences. In the absence of *explicit* velocity, history in the form of consecutive conformations can also serve as a good proxy. Consequently, without velocity or history, a single conformation is hardly adequate to determine the future motion of a protein.

Additionally, a ***point-based*** MDM needs to create a tremendous number of states in order to achieve sufficient coverage of the conformation space. However, not only is a comprehensive sampling of the high-dimensional conformation space impossible, but analyzing thousands or more states for biological understanding is also humanly inconceivable.

The ***cell-based*** MDMs attempt to correct these problems by defining a state as a *region* (a cell) of the protein's conformation space that roughly matches an energy basin [29, 53, 81]. The idea is that a protein will interconvert rapidly among different conformations within a basin s before it overcomes the energy barrier and transits to another basin s'. The assumption is that after many interconversions within s, the protein will "*forget*" its history as it gradually loses the initial momentum that brought it into s. Therefore, when the protein eventually emerges from s, it will transit to s' with a probability dependent only on s, and is thus Markov. The much fewer states based on regions is also more amenable for analysis.

In order to construct a ***cell-based*** MDM with K states, MD trajectories are first used to create a large number of *microstates*. The microstates are then clustered into a small number of K states in a way that maximizes the sum of self-transition probabilities over the K states [29]. The process of creating the microstates and clustering them is iterated to adjust the boundaries between the cells. Ideally, each cell of the final model will outline a biologically significant energy basin and capture its dynamics.

However, clustering microstates into the K states of a *cell-based* MDM is only applicable when the actual number of energy basins is precisely K. This requires *prior* knowledge of the protein. If K is wrong, the resulting *cell-based* MDM will falsely identify biologically inaccurate regions as distinct energy basins. This raises doubts about the generality and accuracy of *cell-based* MDMs.

In addition, energy basins may not be well separated enough to be precisely partitioned into individual cells. Instead of lumping a number of closely connected energy basins into a single cell, identifying their tightly coupled dynamics is potentially much more interesting.

More importantly, the optimization based on self-transition probabilities is unable to determine the most suitable value of K. This is because although a trivial *one*-state model has the *optimal* self-transition probability of 1, it is rather uninformative. Therefore, without a systematic criterion for evaluating model quality, it is difficult to determine the actual number of energy basins. This significantly limits the usefulness of cell-based MDMs in the investigations of new proteins with unknown dynamics.

2.2 Other Approaches

The key to modeling protein dynamics is to capture the change of conformation with respect to time. However, due to the structural complexity of biological molecules and their broad range of motion timescales, each of the following approaches only addresses a specific area of concern, and comes with various limitations.

Gaussian network models (Section 2.2.1) is only applicable to motion near the native conformation. This is due to its approximation of motion according to harmonic oscillations. Although this greatly simplifies the complexity of motion, it is an unsuitable approximation for the long-range motion of folding.

The reaction coordinate (Section 2.2.2) measures the progress of a protein's change in conformation, *e.g.* folding motion. Although reaction coordinate is theoretically applicable to the whole range of protein motion, it is difficult to compute in practice. More crucially, knowing the extent of conformational change alone is *insufficient* for a model of *dynamics*. The reason is that the change needs to be correlated with *time* in order to predict dynamics.

Dimension reduction (Section 2.2.3) is useful for identifying major conformational changes in the high-dimensional MD data. Unfortunately, linear techniques are only appropriate for local motions near the native conformation. Although non-linear techniques are also available, dimension reduction usually only captures the range of motion, but *not* time. Consequently, the result is not a model of dynamics that can predict the change of conformation over time.

2.2.1 Gaussian network models

Gaussian network models are used to understand a protein's motion near its native conformation. A Gaussian network model represents a protein molecule as a mass-spring system, and approximates its motion as fluctuations about an equilibrium [13, 45]. The model is constructed by first assuming the native conformation to be the equilibrium position. Then, each atom or amino acid is represented as a node, and each node is connected to other nodes within a cutoff distance r_c by elastic springs to form an elastic network. The protein's motion about its native conformation is then mimicked by the harmonic oscillations of the mass-spring system, and the approximated fluctuations are Gaussian distributed.

More specifically, the network of nodes and springs representing the protein's structure is encoded as the Kirchhoff or connectivity matrix Γ. Each element Γ_{ij} is based on the distance between the ith and jth nodes:

$$\Gamma_{ij} = \left\{ \begin{array}{rl} -1 & \text{if } i \neq j \text{ and } R_{ij} \leq r_c \\ 0 & \text{if } i \neq j \text{ and } R_{ij} > r_c \\ -\sum_{i,i\neq j}^{N} \Gamma_{ij} & \text{if } i = j \end{array} \right\}, \tag{2.2}$$

where for N nodes, R_{ij} is the distance between the ith and jth nodes, and r_c is the cutoff distance between interacting nodes, usually around 7Å.

In order to analyze the motion of the protein molecule, it is necessary to first decompose Γ^{-1}:

$$\begin{aligned}\Gamma^{-1} &= \mathrm{U}\ (\Lambda^{-1})\ \mathrm{U}^T \\ &= \sum_{i=2}^{N} \lambda_i^{-1} u_i u_i^T, \end{aligned} \tag{2.3}$$

where the columns of the matrix U are the eigenvectors u_i of Γ, and the elements of the diagonal matrix Λ are the corresponding eigenvalues λ_i [45]. The first eigenvalue of Γ is zero and corresponds to the zero net translation of the molecule, therefore, it is not included in the summation of Eq. 2.3.

Therefore, the motion of a protein molecule can be seen as the sum of different modes of motion. Each eigenvector indicates a mode of motion based on a particular contributing combination of network nodes, and the associated eigenvalue indicates its relative significance to the overall motion. The key benefit of this analysis is that the motion of the protein molecule can be reconstructed and animated by using individual modes, or a set of modes for a more general understanding. More importantly, the correlated fluctuations of the network nodes can be validated against X-ray measured experimental quantities known as β-factors [11, 12].

The main disadvantage of Gaussian network models and related methods based on elastic networks [10, 49, 109, 114] or normal mode analysis [31, 70] is that they are only applicable to short range motions near an equilibrium. Additionally, the structure of the elastic network deviates from the actual network of bond interactions that is maintaining a protein's conformation. The dissimilar strengths of different chemical bonds are also unaccounted for. These shortcomings can potentially distort the analysis of concerted motions between different parts of a molecule, or the binding between molecules.

2.2.2 Reaction coordinate

The purpose of finding the reaction coordinate is to better understand significant rate limiting events by mapping them out along a principal axis. Traditionally, the reaction coordinate of a chemical reaction is the path of minimum energy resistance from the initial to the final states of the reaction [36]. Similarly, protein folding can also be described as a reaction occurring along a reaction coordinate, or the path of least resistance. If scientists can understand the order of events along the reaction coordinate and identify the reasons that prevent a protein from folding according to the desired rate or form, better molecules may be designed and engineered.

However, to analytically choose a reaction coordinate for protein folding requires *a priori* understanding of the detailed protein motion trajectory. Moreover, due to the high degrees of flexibility, not all proteins can have their motion described and understood along a single pathway. To address this, Du *et al.* introduced the notion of probability of folding (***p-fold***) [38]. In a folding process, the p-fold value of a conformation q is defined as the probability of a protein to reach the native conformation before reaching an unfolded conformation, taking into account all possible pathways starting from conformation q. Therefore, p-fold measures the kinetic distance between conformation q and the native conformation, and allows the sequence of structural formation of the folding process to be identified.

The use of p-fold as a reaction coordinate is advantageous because it takes into account all possible pathways. However, calculating p-fold is nontrivial because it requires the simulation of infinite trajectories. Fortunately, a technique called Stochastic Roadmap Simulation (SRS) was developed to compute p-fold efficiently [9], and has allowed experimental quantities such as folding rates and ϕ-values to be predicted [25, 26].

2.2.3 Dimensionality reduction

Instead of building simplified dynamic models, one may also analyze MD simulation data directly through dimensionality reduction methods. However, dimensionality reduction does not provide a predictive model that generalizes the original data. Additionally, without the time component, dimensionality reduction is not able to provide a dynamically accurate representation of the original data.

Linear dimensionality reduction

Principal Component Analysis (PCA) is a technique commonly used in data analysis to reduce the dimensionality of data, while retaining as much of the variance in the data as possible [58]. PCA makes use of orthogonal linear transformations to convert data points from the original observation space into data points in a new vector space. The transformation is done such that the first vector, the principal component, contains the greatest variance among the data points. Subsequent vectors constitute decreasing amounts of variance in the data. By retaining the most significant vectors, a substantial portion of the total variance can be preserved within a reduced dimension space.

PCA is commonly used to analyze near equilibrium motions such as the fluctuations about a protein's native conformation [5, 69, 105, 106]. Due to the short range motion of such fluctuations, linear dimensionality reduction can often extract the major modes of motion while removing much of the noisy, high-frequency vibrations. The obvious downside is that for conformational changes involved in the folding process, motion is likely to be nonlinear, and linear dimensionality reduction techniques are likely to introduce artificial distortions.

Nonlinear dimensionality reduction

Nonlinear dimensionality reduction methods attempt to alleviate the limitations of linear techniques. A commonly used technique involves making use of a nearest-neighbor graph to embed relationships in the original data into a low dimensional nonlinear space [33, 37, 84, 90].

The key to the embedding is to preserve the geodesic, or shortest path, distance between the original data points [104]. For neighboring points, direct distance in the original space well approximates the geodesic distance. For two faraway points, the geodesic distance can be approximated by a sequence of shortest paths that connects them via some intermediate points. Therefore, the shortest paths in a nearest-neighbor graph can be used to provide a good approximation of the geodesic distance in the original data.

In order to embed the original data, a geodesic distance matrix is created using the shortest-path distance between all pair-wise data points in the nearest-neighbor graph. Multidimensional scaling via eigen-decomposition of the geodesic distance matrix can then be applied to obtain a nonlinear embedding. The resulting embedding minimizes the difference in geodesic distances between the original space and the embedded space [104].

However, the major drawback of dimensionality reduction techniques is that they do not provide a predictive model of the motion dynamics. Even though dimensionally reduced data can approximately exhibit the same range of motion, and if timestamped according to MD simulation, can exhibit similar motion dynamics, but it is only a simplified version of the original data with little predictive power of the phenomenon in general. Consequently, although tremendously useful, dimensionality reduction is better suited as a pre-processing step in the modeling of protein motion dynamics.

Chapter 3

Modeling Motion Dynamics with Hidden States

The limitation of wet lab techniques to directly observe proteins in motion frustrates scientists who seek a more fundamental understanding of molecular biology. However, with the progress of computer science and MD simulation, the mathematical modeling of protein motion is becoming increasingly feasible (Section 1.2).

A model is both a representation as well as a tool to assist understanding. For biologists, the usefulness of a model is measured by the amount of biological insight it can help to unravel. Therefore, a useful model of protein motion has to accurately identify biologically significant events, and discard noise, from among a tremendous number of MD trajectories. Moreover, in order to assist understanding, a useful model has to capture the biologically significant motions in a simple and easily comprehensible representation. These requirements lead us to the challenge of creating a model that can automatically and efficiently summarize MD trajectories in a compact representation that reveals scientific insight.

3.1 Protein Motion and Dynamics

The dynamics of a protein's motion is about its ***change of conformation over time***. This includes both the *direction* and *magnitude* of change, as well as the *timing* of change. More importantly, scientists are keen to find out what makes a protein change its conformation, this requires knowing the *sequence* of conformational change.

3.1.1 Simulating change of conformation over time

The motion of a protein molecule is the aggregate result of complex interactions among a protein's atoms, as well as between the protein and other molecules in its environment. MD simulates protein motion by taking into account forces acting on atoms of the protein molecule, and atoms in its environment, *e.g.* water:

$$V = V_{covalent} + \underbrace{V_{electrostatic} + V_{waals}}_{non-bonded\ interactions}, \tag{3.1}$$

where V is the total potential energy of the system. The covalent interactions $V_{covalent}$ are due to the arrangement of bonded atoms with respect to each other in the same molecule, which is fewer in number. The non-bonded interactions are due to the interaction between *all* atoms in the system, and is much more numerous [44, 66].

Since each term in Eq. 3.1 is dependent on the distance between atoms, not only do different protein conformations experience different forces, even in the same conformation, a protein can experience different forces due to differences in its environment. Consequently, the same folding event can occur at different times for different molecules of the same protein, as we saw earlier in the villin headpiece trajectories in Fig. 1.3 (page 22).

3.1.2 A Markovian abstraction of dynamics

Unfortunately, MD simulation generates massive amounts of data and is difficult to analyze. More importantly, among the numerous trajectories, what is truly biologically interesting is the ability of *different* molecules of the same protein, to attain the *same* native conformation. Therefore, it is the *ensemble* behavior of protein molecules that is biologically important.

However, a protein's motion is also dependent on the *velocities* of atoms in the system. The dependence on velocity is the reason that MD *simulation* is required to study protein motion, because velocity is the result of a molecule's *past* interactions with its environment. Without simulation, it is difficult to predict how a particular conformation will change over time. Without a large number of trajectories, it is difficult to know the ensemble behavior of protein molecules in solution.

This dependency on velocity is also a main source of difficulty faced by previous attempts at modeling protein motion dynamics. This is because the need to analyze MD data collectively requires an abstraction that generalizes over individual trajectories. Previous attempts at constructing MDMs have tried to satisfy the Markovian property. However, the important issue of identifying the number of biologically significant states remains unresolved.

More importantly, the abstraction of the velocity space is *nontrivial*. Predicting a protein's motion *without* knowing its velocity involves significant assumptions about the uncertainty in its dynamics. It requires finding a compatible match between the underlying phenomenon and the Markovian model assumptions, as well as mutually verifiable observables. This crucial abstraction is essentially what we are searching for.

3.2 Markov Dynamic Models with Hidden States

A Markov Dynamic Model (MDM) Θ of a protein can be represented as a weighted directed graph. A node s of Θ represents a state of the protein, and a directed edge (s, s') from node s to s' represents a transition between the corresponding states. Each edge (s, s') is assigned a weight $a_{ss'}$ representing the probability that the protein in state s transitions to state s' in a time step of fixed duration $\triangle t$. The probabilities associated with the outgoing edges from any node s must sum up to 1. The duration $\triangle t$ is the *time resolution* of the model.

A MDM describes how the state of the protein changes stochastically over time. Given an initial state s_0 of the protein at time 0, a MDM can be used to predict a sequence of future states $\{s_1, s_2, \ldots, etc.\}$, where s_t is the state of the protein at time $t \times \triangle t$ for $t = \{1, 2, \ldots, etc.\}$. If $s_t = s$, then the next state s_{t+1} can be predicted by choosing an outgoing edge (s, s') from s with probability $a_{ss'}$ and setting $s_{t+1} = s'$. The simple and explicit structure of MDMs allows such predictions to be computed efficiently.

In a point-based MDM, a state represents a single conformation, whereas in a cell-based MDM, a state represents a set of conformations (Section 2.1). The definition of states is crucial. The choice of a single conformation as a state is more precise than the choice of a set of conformations. However, choosing a single conformation as a state causes serious violation of the Markovian property and consequently reduces the predictive power of the point-based MDM. Therefore, the question is:

What *should* be a state?

3.2.1 Why hidden states?

By defining a state as a subset of the protein conformation space, rather than a single conformation, cell-based MDMs achieve the dual objectives of better satisfying the Markovian assumption and reducing the number of states (Section 2.1). Even though this is a major step forward, cell-based MDMs still violate the Markovian assumption in a subtle way.

Consider a protein at a conformation q near the boundary of a cell. The future state of the protein is not only dependent on q, but is also highly dependent on the protein's velocity, in other words, on the past history of how the protein reached q. Therefore, by requiring each conformation to belong to a *single* state, cell-based MDMs violate the Markovian assumption. This is especially so near the cell boundaries, where a slight difference in conformation can drastically affect the state assignment. Similar violations also occur in cells corresponding to shallow energy basins, where a conformation's momentum will carry it further before it is overshadowed by the force of the relatively flat energy surface.

One way of avoiding such violations is to define more refined states using both conformation and conformational velocity. However, the additional velocity dimension necessarily increases the number of states, thus partially reversing a key advantage of cell-based MDMs. Furthermore, a much larger dataset is needed for model construction in order to capture the detailed transition probabilities among the refined states.

In contrast, in order to satisfy the Markov assumption, we propose to assign *every* conformation to *multiple* states and use probability to capture the uncertainty of state assignment. This leads us to a MDM with *hidden* states, formally, a Hidden Markov Model (HMM).

3.2.2 Hidden Markov Models (HMMs)

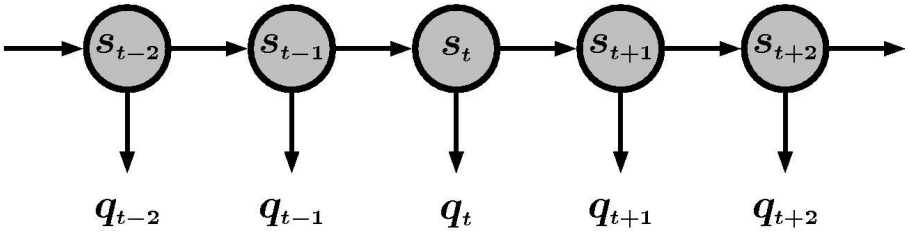


Figure 3.1: A Hidden Markov Model (HMM). s_t is the hidden state at time t. q_t is the observed conformation at time t and is dependent on state s_t. The probability of transitioning from state s_t at time t, to state s_{t+1} at time $t+1$, is independent of the past $(s_0, \ldots, s_{t-2}, s_{t-1})$, and is also independent of q_t, for $t = \{0, 1, \ldots\}$.

In an HMM, the state is not directly visible, while the observed data is dependent on the state (Fig. 3.1).

Our HMM MDM for protein dynamics is defined as $\Theta = (\mathcal{C}, \mathcal{S}, A, \Pi, E)$:

- The conformation space $\mathcal{C}$ of a protein.
- The set of states $\mathcal{S} = \{i \mid i = 1, 2, \ldots, K\}$.
- $A = \{a_{ij} \mid i, j = 1, 2, \ldots, K\}$, where $a_{ij} = p(s_{t+1} = j | s_t = i)$ is the probability of transitioning from state $i \in \mathcal{S}$ to state $j \in \mathcal{S}$ in a single time step of duration $\triangle t$.
- $\Pi = \{\pi_i \mid i = 1, 2, \ldots, K\}$, where π_i is the prior probability that the protein is in state $i \in \mathcal{S}$ at time $t = 0$.
- $E = \{e_i(q) \mid i = 1, 2, \ldots, K,\ q \in \mathcal{C}\}$, where $e_i(q) = p(q | s_t = i)$ is the *emission probability* of observing conformation q when the protein is in state $i \in \mathcal{S}$, and is invariant for any time t.

The state space $\mathcal{S}$ is discrete, while the conformation space $\mathcal{C}$ is continuous. Intuitively each hidden state $i \in \mathcal{S}$ loosely matches a basin in the protein's energy landscape, and the corresponding emission probability $e_i(q) = p(q|s_t = i)$ connects each hidden state with the observed conformations by modeling the distribution of protein conformations influenced by the basin.

In an HMM MDM, we cannot assign a conformation q to a unique state. Instead we calculate $p(s_t = i|q)$, the probability that q belongs to a state i. The uncertainty in state assignment arises because at the conformation q, the protein may have different velocities, as well as other differences that we choose not to model or do not know about. For instance, if conformation q is on a relatively *flat* energy landscape and we do not know its velocity, we cannot be certain how q will change. We model the uncertainty due to this lack of information with the emission probability distributions. Therefore, we estimate the probability that q belongs to a particular state, and by extension future states, based on the proportional outcome of the trajectories that have crossed q in the past.

In contrast, a cell-based MDM in Section 2.1 partitions the conformation space into disjoint regions $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots\}$, and each state i represents a particular region $\mathcal{C}_i$. As a result, a conformation q can be assigned to a unique state. If we define the HMM MDM's emission probability e_i as a step function such that $e_i(q)$ is a strictly positive constant for $q \in \mathcal{C}_i$, and 0 otherwise, then the states are no longer hidden, and our model degenerates into a cell-based MDM. Therefore, our distribution-based models provide a more general abstraction than cell-based MDMs.

Even though hidden states has been used to model protein structure [50], the goal there was to capture the variations in an ensemble of *static* protein structures from NMR experiments, rather than the motion *dynamics*.

3.2.3 What is a good model?

Another difficulty with cell-based MDMs in Section 2.1 is the lack of a principled criterion for evaluating model quality. Cell-based MDMs are constructed to maximize the self-transition probabilities for the states [29]. This criterion, however, results in the paradoxical conclusion that a trivial *one*-state model encompassing the *whole* conformation space is perfect. This is because all transitions are self-transitions of the only state, and as such, can be predicted with absolute certainty. Since simple models are usually preferred, how can we decide that a simple model, such as the trivial one-state model, is not as good as a more complex one?

Originally, the purpose of building a model Θ from a dataset $\mathcal{D}$ of MD trajectories is to better understand a protein's motion. Additionally, for experimental validation and scientific understanding, we want to use model Θ to predict a protein's kinetic and dynamic properties, *e.g.* mean first-passage times [66], p-fold [38], transition state ensembles [66], ... *etc.*. Ideally, we want to compare the predictive power of alternative models on all such properties, and then pick the most accurate model for scientific study. Unfortunately, we may not have comparable values of all known properties in advance, and we also do not know what other properties might be useful in the future.

Fortunately, since the kinetic and dynamic properties of a protein are determined by its motion and molecular interactions, we can check instead the ability of the model Θ to predict the MD trajectories. In our HMM MDM framework, we do this by calculating the likelihood $p(\mathcal{D}|\Theta)$, which is the probability that a dataset $\mathcal{D}$ of MD trajectories will occur under the model Θ. The likelihood $p(\mathcal{D}|\Theta)$ measures the quality of Θ.

Specifically, let $\mathcal{D} = \{\mathcal{D}_i \mid i = 1, 2, \ldots\}$ be a dataset of MD trajectories. Each trajectory $\mathcal{D}_i$ is a sequence of protein conformations $(q_0, q_1, \ldots, q_T)$, where q_t is the protein conformation at time $t \times \triangle t$. The likelihood of model Θ for a trajectory $\mathcal{D}_i$ is:

$$p(\mathcal{D}_i|\Theta) = \sum_{Q \in \mathcal{S}^T} \Big(p(s_0) \prod_{t=1}^{T} p(s_t|s_{t-1}) \prod_{t=0}^{T} p(q_t|s_t) \Big), \tag{3.2}$$

where s_t is the state of the protein at time $t \times \triangle t$, while $p(s_0)$, $p(s_t|s_{t-1})$, and $p(q_t|s_t)$ are given by the parameters Π, A, and E of Θ, respectively [16]. The summation $\sum_Q$ is performed over all possible state assignments $Q = (s_0, s_1, \ldots, s_T) \in \mathcal{S}^T$ to the conformations $(q_0, q_1, \ldots, q_T)$ in trajectory $\mathcal{D}_i$. The likelihood of Θ for the entire dataset $\mathcal{D}$ is:

$$p(\mathcal{D}|\Theta) = \textstyle\prod_i p(\mathcal{D}_i|\Theta). \tag{3.3}$$

Therefore, in order for a model Θ to score well in Eq. 3.3, it has to accurately predict the change of conformation over time for all MD trajectories.

In contrast to cell-based MDMs in Section 2.1, the likelihood $p(\mathcal{D}|\Theta)$ provides a quantitative measure of model quality that enables us to compare models with different number of states. This is possible because our model makes use of emission probabilities $e_i(q) = p(q|s_t = i)$ to connect states with conformations. Whereas a cell-based MDM does not even attempt to predict the conformations. In fact, the likelihood criterion shows that a trivial one-state MDM is bad. For a one-state MDM, although the transition probability $p(s_t|s_{t-1}) = 1$ is perfect for all t, emission probabilities $p(q_t|s_t)$ are small because the model relies on a single state to capture variability over the entire conformation space. Hence, the overall likelihood $p(\mathcal{D}|\Theta)$ of a one-state MDM is bad.

3.2.4 Benefits and limitations

The key benefit of modeling an energy basin as a hidden state is that it allows us to build a MDM that satisfies the Markovian property. When a protein converts rapidly between closely related conformations within an energy basin, it loses the initial momentum it had when it first entered the energy basin. This allows the protein's transition to the next energy basin to be independent of history, and is therefore Markov. Consequently, an HMM MDM accurately represents the change of a protein's conformation between energy basins as Markovian transitions between states.

The likelihood score in Eq. 3.3 plays the critical role of allowing us to quantitatively measure the ability of a model to predict the dynamics of MD trajectories. By comparing the likelihood scores of different models on the same dataset $\mathcal{D}$, we can objectively determine the suitable number of states, and identify the biologically significant energy basins of the protein. The successful characterization of the energy basins, and the accurate prediction of transitions among them, provide a compact abstraction of the original data useful for understanding a protein's dynamics.

Additionally, modeling the conformations of each energy basin as a probabilistic distribution over the entire conformation space is beneficial. The emission probabilities $p(q_t|s_t)$ allow us to capture the inherent uncertainty of conformations, especially those mid-way between energy basins, by assigning them to multiple states. More importantly, probabilistic distributions allow states to overlap each other. When an excess state overlaps directly on top of another state, it contributes little to the likelihood score (Section 3.3.5). Therefore, for K energy basins, the likelihood score is expected to plateau when the number of states exceed K. This is the primary reason that we are able to determine if a model is *sufficiently* complex.

Furthermore, one goal of modeling is to predict a protein's kinetic and dynamic properties. Since our model is constructed from MD trajectories, a basic question is: "How can the model provide better predictions than the MD trajectories themselves?" The answer is that the model generalizes the data under the Markovian property and thus contains a lot more trajectories than the data used to construct the model. Consider, for example, a dataset containing two trajectories with state sequences (s_0, s_1, s_2) and (s'_0, s_1, s'_2). Using the Markovian property, the model assumes that two additional state sequences (s_0, s_1, s'_2) and (s'_0, s_1, s_2) are also valid. By combining the trajectories, the model generates exponentially more trajectories than the dataset contains. If the assumption of the Markovian property is valid, then the resulting model is a more accurate approximation of the underlying protein dynamics and can better predict kinetic and dynamic properties.

A related question is: "With MD trajectories at the nanosecond scale, how can the model predict events at the microsecond or millisecond scale?" Again, using the Markovian property, the model concatenates short simulation trajectories into much longer ones [29, 30], and uses them to predict long-timescale kinetic and dynamic properties. This approach can succeed even for large proteins, if the transitions between stable energy basins are relatively fast enough to be simulated, they can be reliably estimated and effectively concatenated [47].

At the same time, our model cannot have state transitions not implied by the original simulation trajectories and thus does not address the question of how to sample the conformation space comprehensively and efficiently. This is a difficult problem, but it has also seen rapid progress in recent years [87, 97]. Advances in sampling methods will provide better simulation data and improve the quality of the resulting models.

3.3 Model Construction

The process of modeling protein motion dynamics is a search for the most suitable model across *both* space and time. In terms of *space*, we are interested in discovering the number and structural characteristics of biologically significant conformations, *i.e.* states. In terms of *time*, we are interested in characterizing the different timescales of a protein's motion. Most crucially, it is through the transitions between states over time, that we can better understand a protein's motion and function.

We will begin by first assuming that the timescale of interest is known and search for models across the number of states. The reason is that a protein's folding time is easier to ascertain through wet lab experiments [78, 79], as compared to the precise sequence of folding events. In Chapter 4, we will relax our assumption on timescale, and build hierarchical models that identify the interesting timescales of dynamics.

Under the likelihood criterion, we want to construct each K-state model Θ such that it maximizes $p(D|\Theta)$ for a given dataset D of MD trajectories. Expectation Maximization (EM) is a standard algorithm for such optimization problems. However, EM is computationally intensive. It may also get stuck in local optima and fails to find the model with maximum likelihood.

In order to alleviate the difficulties of EM, we take advantage of the density of conformations near energy basins and attempt numerous initializations *before* utilizing EM. We proceed as follows:

- We pre-process the MD trajectories to remove "noise", *i.e.* motions at timescales much quicker than that of interest. The data is also divided into separate training $\mathcal{D}_{train}$ and test $\mathcal{D}_{test}$ sets.

- We use K-medoids algorithm to identify compact clusters of conformations. Intuitively, each cluster represents a possible energy basin, and will serve as the basis of a state when creating the initial MDM Θ_0. Since clustering is much faster than EM, we can afford to run the clustering algorithm numerous times and choose the best result. This reduces the chance of ending up with a bad local optimum.

- We use the clustering information to create an initial MDM Θ_0. A state is created based on the distribution of a corresponding cluster. In addition, each trajectory is labeled according to the sequence of clusters it traverses. The labeling provides an approximation of dynamics that allows us to estimate the parameters of Θ_0.

- We initialize EM with Θ_0 and optimize for the K-state model Θ with maximum $p(\mathcal{D}_{train}|\Theta)$. Due to the use of K-medoids, parameters of Θ_0 are already well estimated, and only a few iterations of EM is needed.

- Finally, we score the model Θ on the test dataset $\mathcal{D}_{test}$, compare the score to other models with a different number of states, and choose the *most suitable* model Θ_K.

3.3.1 Data preparation

We first consider the sub-sampling of MD trajectories according to the timescale of interest. The temporal resolution $\triangle t$ of a model is determined by the time between successive conformations along trajectories used for training. If $\triangle t$ is too slow, the model may miss biologically interesting events. If $\triangle t$ is too fast, the model will try to capture uninterestingly fine details and become unnecessarily complex. Also, the model's accuracy may suffer due to the influence of non-Markovian high-frequency fluctuations.

In our experiments, h is typically set to be 1/100 to 1/10 of the timescale of interest. We then apply standard signal processing techniques [69, 80, 93] to smooth and sub-sample every MD trajectory by taking successive conformations spaced $\triangle t$ apart. If smoothing is incorporated into MD to direct the simulation in real-time, it will also be beneficial for the quick exploration of biologically relevant parts of the conformation space [69, 82].

3.3.2 K-medoids clustering

The ideal modeling algorithm should *not* require prior knowledge of the protein's dynamics. Although knowing a protein's biologically significant conformations can allow us to directly initialize the states of a model, relying on such information will also limit the usefulness of the algorithm. However, this broadens the search for the most suitable model.

In order to create an *efficient* modeling algorithm, we take advantage of the density of conformations to attempt different initializations. This is possible because the states in our model correspond to energy basins. Within an energy basin, a protein interconverts rapidly, this allows inter-state protein motions to satisfy the Markovian property. The rapid interconversion results in a high-density cluster of conformations roughly centered at each energy basin. Intuitively, if we can first locate the clusters of conformations, we can then locate the energy basins, and subsequently build a model of dynamics. Therefore, the idea is to first identify the clusters of conformations, and then make use of each cluster to initialize a hidden state in our MDM.

More importantly, the use of K-medoids clustering allows us to identify the most significant energy basins first. This is important because energy basins are not necessarily well-separated enough to be considered distinct entities. Since the K-medoids algorithm minimizes the sum of intra-cluster

distances [16], it favors the initial identification of large clusters of densely packed conformations. These initial clusters are likely to be broad, and due to the insufficient model complexity, are also likely to encompass smaller clusters internally. Only when the number of states increases, can the smaller clusters be individually characterized. Since each cluster is used to initialize a state, K-medoids beneficially allows us to initially model tightly coupled energy basins collectively as one, and when the number of states is sufficient, to characterize them individually.

We proceed by treating the training data $\mathcal{D}_{train}$ as a set of conformations, but ***without time information***. We then use K-medoids algorithm to group the conformations into K clusters, for a particular K. Although K-medoids clustering does *not* create a model of dynamics, each resulting cluster will be used later to initialize a hidden state of a MDM. More importantly, since clustering is relatively cheap, we can afford to run K-medoids multiple times with random restarts, and choose the best clustering result. This helps us to efficiently avoid creating models stuck in local optima [46, 102].

The resulting K clusters of conformations *implicitly* define a K-partition of the conformation space, with the boundary between partitions delineated by the separation between clusters. However, this K-partition may not be an accurate demarcation of the energy basins. If K is too small, two energy basins may be erroneously lumped together as one partition, with the center of the partition far away from the actual bottoms of the energy basins. If K is too large, an energy basin may be artificially partitioned into two. Consequently, the clustering only provides an indication of the locations of *possible* energy basins, and we do *not* have a model of dynamics at this stage yet.

3.3.3 Initialization

There are two aspects of an HMM MDM that need to be initialized. First, the emission probabilities that characterize the ***states*** need to be initialized. Since a state loosely correspond to an energy basin, the location and dimension of its distribution of conformations is modeled as a temporally static property in our MDM. Consequently, we assume that each cluster obtained from K-medoids represents the distribution of conformations influenced by a hidden energy basin. This allows us to quickly initialize the emission probabilities directly from the results of K-medoids clustering.

However, in order to capture dynamics, we need to incorporate time and parameterize the ***transitions*** that capture the *change* of conformation. The key to this is the K partitions implicitly defined by the K clusters. The K partitions *reveal* the "*hidden*" nature of states and allow us to assign each conformation *unambiguously* to a single state. This allows us to label each trajectory $\mathcal{D}_i = (q_0, q_1, \ldots, q_T)$ by an *unambiguous* sequence of states $Q_i = (s_0, s_1, \ldots, s_T)$, where s_t is the state of the conformation q_t at time t. The labeling provides an approximation of dynamics that allows us to estimate the transition probabilities. Details are as follows.

Emission probability distributions $E_0 = \{e_i(q)\}$

The emission probability e_i models the distribution of protein conformations of state i. Consequently, the complexity of e_i determines the precision in which an energy basin can be characterized. We have two main considerations in choosing e_i. First, it should match the distribution of conformations influenced by an energy basin. Second, it should be simple enough to be learned effectively with our limited amount of data.

Under the theoretical framework of energy minima acting as basins of attraction [36, 44, 66, 86], we approximate e_i with a Gaussian distribution:

$$e_i(q) = \mathcal{N}(q|\mu_i, \sigma_i^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\big(-\frac{1}{2\sigma^2}\ d^2(q, \mu_i)\big), \tag{3.4}$$

where q is a conformation, μ_i is the conformation at the center of state i, σ_i^2 is the variance of state i, and $d(q, \mu_i)$ denotes a suitable distance measure between conformations q and μ_i. Other distributions are also possible.

Here, we assume a one dimensional Gaussian distribution, because RMSD or the graph based distance we used for the large villin protein are one dimensional (Chapter 4). However, higher dimensional distributions can also be used, as is the case for our synthetic examples (Section 3.4.1) and alanine protein (Section 3.4.2). To estimate $e_i(q)$, we only need to consider conformations within state i:

$$\mu_i = \arg\min_q \sum_{D_i \in D_{train}} \sum_{t=0}^{T} \{d(q, q_t) \mid s_t = i\}, \tag{3.5}$$

$$\sigma_i^2 = \frac{\sum_{D_i \in D_{train}} \sum_{t=0}^{T} \{d^2(\mu_i, q_t) \mid s_t = i\}}{\sum_{D_i \in D_{train}} \sum_{t=0}^{T} \delta(s_t = i)}, \tag{3.6}$$

where q is any conformation within state i, q_t is the conformation of a particular trajectory $\mathcal{D}_i$ at time t, s_t is the state of q_t, and $d(q_i, q_j)$ is the distance between conformations q_i and q_j.

The high degree of conformation flexibility of a protein entails that the energy landscape is highly convoluted when the protein is compact and hard collisions between atoms are frequent. As a result, the distribution of conformations near the native state can be incredibly complicated. Therefore, if sufficient data is available, it is also possible to use a simpler distribution early in the training process for efficiency reasons, and then

switch to a more complex distribution after the number of states has been determined, to further characterize the energy basins.

Prior probability $\Pi_0 = \{\pi_i\}$

The prior $\pi_i = p(s_0 = i)$ is the probability that the protein is in state i at time $t = 0$. The dependency on time means that we need to make use of the sequence of states $Q_i = (s_0, s_1, \ldots, s_T)$ when estimating the parameters. However, we only need to count the first state s_0 traversed by each trajectory $\mathcal{D}_i = (q_0, q_1, \ldots, q_T)$:

$$\pi_i = \frac{\sum_{D_i \in D_{train}} \delta(s_0 = i)}{N}, \tag{3.7}$$

where $\delta(\cdot)$ evaluates to 1 if the condition is true, and $\sum_{D_i \in D_{train}}$ sums over all N trajectories in $\mathcal{D}_{train}$.

Transition probability $A_0 = \{a_{ij}\}$

The transition probability $a_{ij} = p(s_{t+1} = j | s_t = i)$ is the probability of transitioning from state i to state j in a single time step. Therefore, we make use of the sequence of states $Q_i = (s_0, s_1, \ldots, s_T)$, and count the successive states traversed by each trajectory $\mathcal{D}_i = (q_0, q_1, \ldots, q_T)$:

$$a_{ij} = \frac{\sum_{D_i \in D_{train}} \sum_{t=0}^{T-1} \delta(s_t = i, s_{t+1} = j)}{\sum_{D_i \in D_{train}} \sum_{t=0}^{T-1} \delta(s_t = i)}, \tag{3.8}$$

where T is the last time step of each trajectory, and the denominator is the sum of all the transitions from state i.

Therefore, instead of the more expensive EM algorithm $O(K^2T)$ (Section 3.3.4), the unambiguous sequence of states Q_i allows us to efficiently initialized the transition probabilities in $O(T)$ time.

3.3.4 Optimization

The model Θ_0 initialized with K-medoids clustering is only an initial step. In particular, Θ_0 is estimated from the clustering criterion of minimizing the sum of intra-cluster distances between conformations. This is different from the likelihood criterion based on the accuracy of predicting a temporal sequence of conformations (Eq. 3.2). Therefore, although Θ_0 is a model of dynamics, it has not been optimized for its purpose.

For optimization, we initialize the EM algorithm with Θ_0 and search for a K-state HMM MDM Θ that maximizes the likelihood $p(D_{train}|\Theta)$. EM iterates over two steps, expectation (E) and maximization (M), and improves the current model until no further improvement is possible. The distinction between the optimization via EM and the initialization via K-medoids is the probabilistic distribution over the hidden states. For a particular conformation q_t at time t, instead of an absolute assignment to one state, we now use a probabilistic assignment over all states (E-step). Instead of simple counting to estimate the model parameters $\Theta = (\Pi, A, E)$, we need to use a weighted average (M-step).

Inspection of Eq. 3.2 shows that the main difficulty is the summation of all possible state assignments $Q = (s_0, s_1, \ldots, s_T) \in \mathcal{S}^T$ to the conformations $(q_0, q_1, \ldots, q_T)$ along a trajectory $\mathcal{D}_i$. Performing this summation by brute force takes time $O(K^T)$, which is exponential in the length T of the trajectory. EM overcomes this difficulty through dynamic programming in $O(K^2T)$ time [4, 16, 61]. In practice, the length T of a trajectory is usually orders of magnitude larger than K.

Expectation (E) step

The purpose of E-step is to calculate a probability distribution of the missing state labels using the *current estimate* of the model parameters, so that we can *re-estimate* the parameters in the M-step [4, 16, 61].

Given a trajectory $\mathcal{D}_i = (q_0, q_1, \ldots, q_T)$ of length T and model Θ, we proceed in two phases. First, we compute variables $\alpha_t(i)$ and $\beta_t(i)$ that estimate the hidden states by accounting for only a *partial* trajectory. Then, we combine $\alpha_t(i)$ and $\beta_t(i)$, and compute variables $\xi_t(i,j)$ and $\gamma_t(i)$ that estimate the hidden states by accounting for the *whole* trajectory.

First, the *forward* variable $\alpha_t(i) = p(q_0, q_1, \ldots, q_t, s_t = i)$ is the probability of observing the partial sequence $(q_0, q_1, \ldots, q_t)$ up till time t and being in state i at time t. $\alpha_t(i)$ can be calculated recursively:

- Initialization:

$$\begin{aligned} \alpha_0(i) &= p(q_0, s_0 = i) \\ &= p(q_0 \mid s_0 = i)p(s_0 = i) \\ &= \pi_i e_i(q_0), \end{aligned} \tag{3.9}$$

- Recursion:

$$\begin{aligned} \alpha_{t+1}(j) &= p(q_0, q_1, \ldots, q_{t+1}, s_{t+1} = j) \\ &= \left[\sum_{i=1}^{K} \alpha_t(i) a_{ij}\right] e_j(q_{t+1}), \end{aligned} \tag{3.10}$$

where π_i is the prior probability that the protein started in state i, and $e_i(q_0)$ is the probability of observing conformation q_0 when in state i.

In Eq. 3.10, $\alpha_t(i)$ is the probability of observing the conformations up till time t and ending in state i at time t. By following a transition a_{ij} from state i to state j, we can account for the next conformation q_{t+1} at time $t+1$. Therefore, by summing over all states at time t, $\sum_{i=1}^{K}$ in Eq. 3.10 accurately accounts for all possible sequences that lead to state j at time $t+1$.

However, $\alpha_t(i)$ only accounts for the partial trajectory up till time t, and we need to compute a similar variable, in reverse temporal sequence. We define the *backward* variable $\beta_t(i) = p(q_{t+1}, q_{t+2}, \ldots, q_T \mid s_t = i)$ as the probability of being in state i at time t and observing the partial sequence $(q_{t+1}, q_{t+2}, \ldots, q_T)$ from time $t+1$ onwards, computed similarly:

- Initialization:

$$\beta_T(i) = 1, \tag{3.11}$$

- Recursion:

$$\begin{aligned} \beta_t(i) &= p(q_{t+1}, q_{t+2}, \ldots, q_T \mid s_t = i) \\ &= \sum_{j=1}^{K} a_{ij} e_j(q_{t+1}) \beta_{t+1}(j), \end{aligned} \tag{3.12}$$

where in Eq. 3.12, the reasoning is that we can transit from state i to K possible states via a_{ij}, then observe the conformation q_{t+1} at $t+1$, before continuing on via $\beta_{t+1}(j)$.

Now, we can combine the variables $\alpha_t(i)$ and $\beta_t(i)$, and estimate the hidden states with the *whole* trajectory. We define $\xi_t(i,j)$ as the probability of being in state i at time t and in state j at time $t+1$:

$$\begin{aligned}\xi_t(i,j) &= p(s_t = i, s_{t+1} = j) \\ &= \frac{p(q_0,\ldots,q_t, s_t = i)p(s_{t+1} = j|s_t = i)p(q_{t+1}|s_{t+1} = j)p(q_{t+2},\ldots,q_T)}{p(q_0,q_1,\ldots,q_T)} \\ &= \frac{\alpha_t(i)a_{ij}e_j(q_{t+1})\beta_{t+1}(j)}{\sum_{m=1}^{K}\sum_{n=1}^{K}\alpha_t(m)a_{mn}e_n(q_{t+1})\beta_{t+1}(n)},\end{aligned} \tag{3.13}$$

where $\alpha_t(i)$ explains the first t conformations while ending in state i at time t, this is followed by a transition to state j via a_{ij}, observation of conformation q_{t+1} at time $t+1$, before continuing on from state j for the rest of the trajectory, *i.e.* $\beta_{t+1}(j)$. The normalization is over all possible pairs of states that can be visited at time t and time $t+1$.

Lastly, we can calculate the probability of being in state i at time t by marginalizing over all possible next states in $\xi_t(i,j)$:

$$\gamma_t(i) = \sum_{j=1}^{K}\xi_t(i,j). \tag{3.14}$$

With the estimated state labels from variables $\xi_t(i,j)$ and $\gamma_t(i)$, we are ready to re-estimate the model parameters.

Maximization (M) step

The purpose of M-step is to calculate the parameters of a *new* model based on weighted averages of state occupancies estimated from the *current* model.

The prior $\pi_i = p(s_0 = i)$ is the probability that the protein is in state i at time $t = 0$, therefore, we only consider the first state s_0 of every trajectory $\mathcal{D}_i$:

$$\pi_i = \frac{\sum_{D_i \in D_{train}} \gamma_0(i)}{N}, \tag{3.15}$$

where $\sum_{D_i}$ sums over all N trajectories in $\mathcal{D}_{train}$.

The transition probability $a_{ij} = p(s_{t+1} = j | s_t = i)$ is the probability of transitioning from state i to state j in a single time step, therefore, we need to consider the states of successive conformations along every trajectory $\mathcal{D}_i$:

$$a_{ij} = \frac{\sum_{D_i \in D_{train}} \sum_{t=0}^{T-1} \xi_t(i,j)}{\sum_{D_i \in D_{train}} \sum_{t=0}^{T-1} \gamma_t(i)}, \tag{3.16}$$

where T is the last time step of each trajectory, and the denominator is the sum of all the transitions from state i.

The emission probability $e_i(q)$ of state i can be estimated accordingly:

$$\mu_i = \arg\min_q \sum_{D_i \in D_{train}} \sum_{t=0}^{T-1} \gamma_t(i)\ d(q, q_t), \tag{3.17}$$

$$\sigma_i^2 = \frac{\sum_{D_i \in D_{train}} \sum_{t=0}^{T-1} \gamma_t(i)\ d^2(\mu_i, q_t)}{\sum_{D_i \in D_{train}} \sum_{t=0}^{T-1} \gamma_t(i)}, \tag{3.18}$$

where q is any conformation in the dataset $\mathcal{D}_{train}$ (see next page), q_t is the conformation of a particular trajectory $\mathcal{D}_i$ at time t, and $d(q_i, q_j)$ is the distance between conformations q_i and q_j.

The estimation of the emission probabilities $e_i(q)$ is a potential bottleneck of optimization. The reason is because for analysis reasons, we want to use a *feasible* conformation to be the center of a state, as opposed to a simple *averaged* conformation. The consequence is that $\arg\min_q$ requires a pair-wise comparison of all conformations in $O(T^2)$ time. Although it is possible to limit the search for q in Eq. 3.17 to only conformations near the current μ_i, this may potentially increase the number of EM iterations required. Later in Chapter 4, for the larger villin headpiece protein, we make use of a distance graph to reduce the cost of this search down to $O(M^2)$, where M is a much smaller number of representative "*microstates*".

More importantly, based on our experiments, the likelihood of the model Θ_0 created in initialization is relatively close to optimum, and is sufficient to significantly narrow the range of K-state models to compare. We simply perform a few iterations of EM to ensure optimality before we determine the number of states.

3.3.5 Determining the number of states

Until now, we have constructed models with different number of states without knowing which is the most suitable model to use for analysis. In choosing the most suitable model, our primary concern is the *accuracy* of a model in predicting new data, with respect to the complexity of the model, *i.e.* K. In principle, a complex model with many states is able to fit the data better. However, an excessively complex model may over-fit, and fails to generalize over individual trajectories. On the other hand, although a simple model is easier to analyze, an overly simple model may not be accurate enough. Therefore, the most desired model should be the one that is both *simple* and *accurate*.

In order to choose the most suitable value of K, we need to calculate the likelihood of each model Θ on an unseen *test* dataset $\mathcal{D}_{test}$, and compare the likelihood score to models with different number of states. We make use of the $\alpha_t(i)$ variable in Eq. 3.10 to calculate the likelihood $p(\mathcal{D}_{test}|\Theta)$:

$$\begin{aligned} p(\mathcal{D}_{test}|\Theta) &= \prod_{D_j \in D_{test}} p(D_j|\Theta) \\ &= \prod_{D_j \in D_{test}} \left[\sum_{Q \in \mathcal{S}^T} \left(p(s_0) \prod_{t=1}^{T} p(s_t|s_{t-1}) \prod_{t=0}^{T} p(q_t|s_t) \right) \right] \\ &= \prod_{D_j \in D_{test}} \left[\sum_{i=1}^{K} \alpha_T(i) \right], \end{aligned} \tag{3.19}$$

where $\alpha_T(i)$ is the probability of observing the whole trajectory $\mathcal{D}_j$ and being in state i at time T. $\sum_{i=1}^{K}$ marginalizes $\alpha_T(i)$ over all states at time T to obtain $p(D_j|\Theta)$, for $D_j \in D_{test}$.

If the actual number of energy basins is $\mathbf{K}$, we expect:

- $K \ll \mathbf{K}$. We expect the improvement in likelihood to be steep. Each additional state accurately models the dynamics of an additional energy basin, and the model gains an increase in predictive power.
- $K \approx \mathbf{K}$. We expect the improvement in likelihood to start leveling off. Due to the density based K-medoids clustering, the last few energy basins to be characterized are less important to the overall dynamics.
- $K \gg \mathbf{K}$. We expect the likelihood to remain relatively constant. When multiple states are trying to model the same energy basin, they overlap and contribute little additional information.

Therefore, the most suitable value for K is the number of states when the likelihood score *first plateaus out.*

We can perform such a search over model complexity because the likelihood criterion enables us to compare *different* models based on their accuracy in predicting the *same* test set of trajectories. Consequently, each state in the most suitable model Θ_K corresponds to an energy basin crucial to understanding a protein's motion dynamics. This addresses a main shortcoming of existing methods, which are unable to determine the number of states without prior knowledge of the protein, *e.g.* cell-based MDMs.

More importantly, the EM optimization process iteratively improves the model parameters in order to better predict the *correct conformation* at the *correct time*. Simultaneously, the model is also predicting the *absence* of the wrong conformation at the wrong time. The objectivity of the likelihood criterion is an important issue to investigate. In Chapter 4, we will make use of false data to address this issue and illustrate why we do not consider models with $K \gg \mathbf{K}$ to be suitable, despite their similarly good scores.

3.4 Results

We apply our modeling approach in two experiments to illustrate the applicability of our method. So far, we have assumed that a particular timescale of interest is known in advance. However, biologists are actually interested in events occurring across a range of timescale, *e.g.* 10^{-6} s to 10^{-3} s. This necessitates the building of different models at different timescales, either from the beginning, or by raising the transition matrix A^t to create a model at $t \times \triangle t$. We will relax this requirement later in Chapter 4.

In Section 3.4.1, we first make use of synthetic energy landscapes to better understand how our modeling process will perform under different scenarios. More importantly, we wish to demonstrate that we are able to identify the most suitable number of energy basins that influence the motion dynamics. This is important because the energy landscape of real proteins is usually *not* explicitly available. Moreover, energy basins are not necessarily well-separated enough to be considered distinct entities. Therefore, it is crucial to first understand what are the characteristics that can be captured with an increase in model complexity.

In Section 3.4.2, we apply our approach on alanine dipeptide, a protein with two amino acids. This is a well studied protein in biology, and has served as a model system due to its ability to exhibit torsion angles observed in α-helix and β-strands of proteins [20, 98]. This is our first test on a real protein, and we produced interesting results on the number of states required to predict its dynamics.

3.4.1 Synthetic energy landscapes

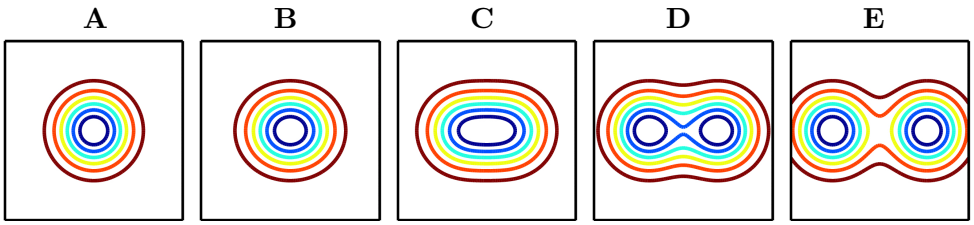
Synthetic energy landscapes are useful for testing our algorithms in controlled settings where the ground truth is known. In particular, we want to examine whether our likelihood criterion and model construction algorithm can identify simple models with strong predictive power.

We created a series of five energy landscapes in two dimensions (Fig. 3.2). Each dimension corresponds to a degree of freedom in an artificial molecule, and the XY-space corresponds to the space of all possible conformations $\mathcal{C}$. The main difference in these landscapes is the gradual change from one energy basin to two distinct energy basins. Landscapes A and B each contains one energy basin, but B's basin is slightly more elongated. Landscapes C, D, and E each contains two basins with varying amount of separation. The corresponding energy barrier between the basins is an important factor in determining the ease of transitions between them. This is a crucial scenario to investigate because basins in a real protein's energy landscape are not necessarily well separated. Therefore, we are interested to understand how our models will characterize the dynamics.

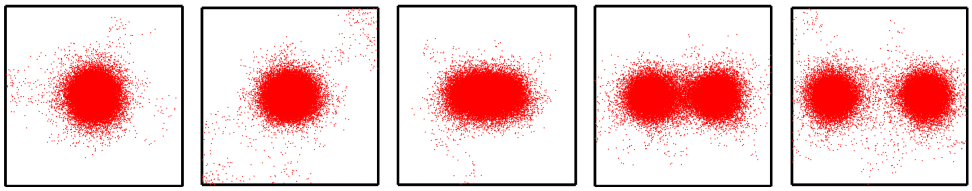
Each landscape is constructed by parameterizing the potential function:

$$V = \sum_{i=1}^{K} a_i \exp\left(-\left(\frac{x - x_i}{b_i}\right)^2 - \left(\frac{y - y_i}{c_i}\right)^2\right), \tag{3.20}$$

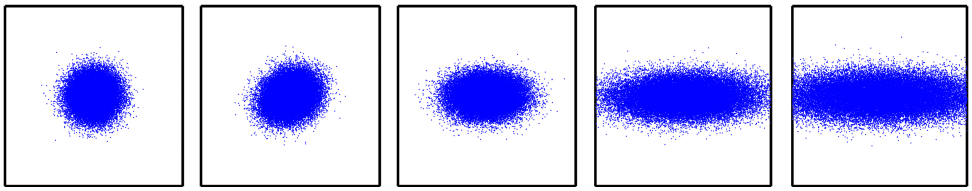
where for K energy basins, (x_i, y_i) is the center, and a_i, b_i, c_i are constants of each basin. For each landscape, we used Langevin dynamics to generate 1000 trajectories of 200 time steps each [44, 66]. We set aside half of the trajectories as the training dataset $\mathcal{D}_{train}$ for model construction, and the other half as the test dataset $\mathcal{D}_{test}$ for checking the quality of models constructed.



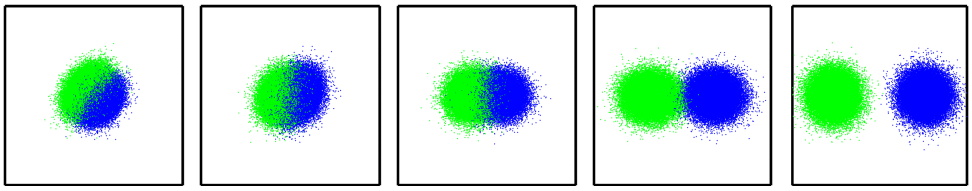
Contour plots of synthetic energy landscapes.



Training dataset generated from Langevin dynamics.



Conformations generated from 1-state HMM MDMs.



Conformations generated from 2-state HMM MDMs.

Figure 3.2: Five synthetic energy landscapes and the corresponding HMM MDMs. The energy landscapes are labeled A, B, C, D and E, with a gradual change from one energy basin to two distinct energy basins. Each square box represents the XY conformation space $\mathcal{C}$ of an artificial protein (axis not labeled).

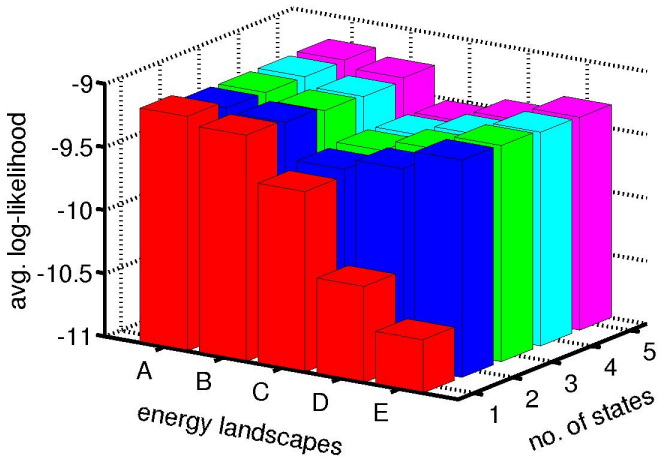


Figure 3.3: Average log-likelihood scores of HMM MDMs for the synthetic energy landscapes.

For each landscape, we built models with increasing K number of states at $\triangle t = 10$ simulation time steps. The distance measure d used in defining the emission probabilities E is the Euclidean distance in the plane.

Fig. 3.3 plots the scores of all the models. The score is the average log-likelihood of a model for a single transition step along a trajectory. It is computed by dividing the log-likelihood of a model given $\mathcal{D}_{test}$ by the total number of conformations in $\mathcal{D}_{test}$. For each K number of states:

- $\boldsymbol{K = 1}$. We can see that for landscape A, which contains only 1 energy basin, the 1-state model is slightly better than the other models. However, as we move from landscape A to E, the predictive power of the 1-state model degrades. This is because when there are actually 2 energy basins, the 1-state model generalizes over *both* basins (Fig. 3.2). Consequently, it is less accurate in predicting trajectories as compared to models with a greater number of states.

- $\boldsymbol{K = 2}$. The 2-state model performs fairly well on all five energy landscapes. This is because when there is only one energy basin, the emission probabilities E of the states begin to overlap almost directly on top of each other (Fig. 3.2). Although this brings no additional improvement in likelihood score, it allows the 2-state model to perform well on all five energy landscapes.

- $\boldsymbol{K \geq 3}$. Due to the overlapping of emission probabilities E, increasing the number of states further has negligible benefit. Although excess states are not reflected in the likelihood scores, we penalize them by choosing simpler models when the likelihood scores are similar.

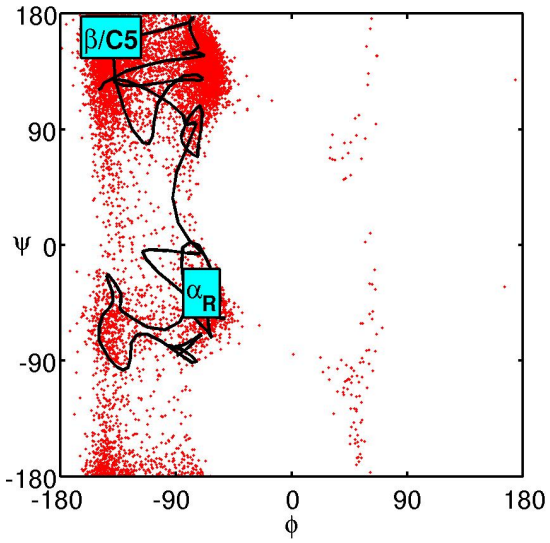
In summary, although these results are not surprising, they highlight the importance of a principled criterion for evaluating the model quality. In particular, we consider additional model complexity to be beneficial only when it results in a more accurate prediction of MD trajectories. Therefore, when modeling a protein with unknown motion dynamics, a search across the number of states is required. More specifically, the most suitable K-state HMM MDM Θ_K can only be determined by comparing the predictive accuracy of models with different number of states.

3.4.2 Alanine dipeptide

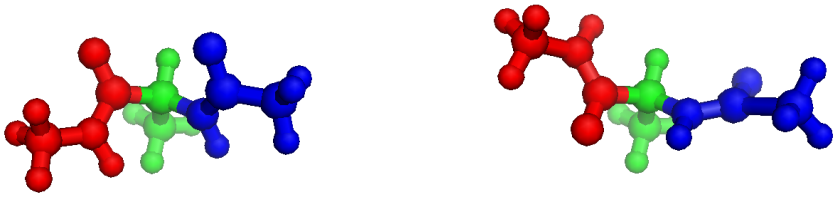
Alanine dipeptide (Ace-Ala-Nme) is a protein with two amino acids widely used for studying biomolecular motion [20, 98]. This is due to its simple structure and its ability to exhibit the wide range of torsion angles observed in α-helix and β-strands of proteins (Fig. 3.4). We use the same dataset as that from a previous study [29]. It consists of 1000 MD simulation trajectories, each roughly 20 ps in duration. Again, we divide them equally into training $\mathcal{D}_{train}$ and test $\mathcal{D}_{test}$ datasets.

We built models with up to 7 states. They are named $K1$ to $K7$. As alanine dipeptide is relatively small, its motion is fast. So the time resolution $\triangle t$ of the models is set to 1.0 ps. A conformation of alanine dipeptide is specified by three backbone torsional angles (ϕ, ψ, ω), and the distance between two conformations is defined as the root sum squared angular differences between the corresponding torsional angles.

The conformation space of alanine dipeptide has also been manually decomposed into 6 disjoint regions, each corresponding to a meta-stable state. This well-accepted decomposition has led to several dynamic models of alanine dipeptide [29, 30]. For comparison, we built an additional 6-state model $M6$ based on the same manual decomposition. During the model construction, instead of applying K-medoids, we group conformations into a cluster if they belong to the same disjoint region of the manual decomposition. Other steps of the construction algorithm remain the same.



a) Data



b) α_R conformation (α for short)

c) $\beta/C5$ conformation (β for short)

Figure 3.4: MD trajectories and structures of alanine dipeptide. In *a*), the ϕ and ψ angles have the greatest freedom of rotation and are projected here, while the ω angle is much more rigid (180$\pm$15 degrees). The black line traces a trajectory going from the $\beta/C5$ conformation to the α_R conformation. Red dots are a sample of conformations from the rest of the dataset. In *b*) and *c*), the main difference between the two conformations is the rotation of the red portion of the molecule about the polypeptide backbone. This corresponds to the large change in the ψ angle. In a long polypeptide, such a rotation can result in a large change of conformation further down the chain. Since the change in ϕ angle is small, rotation of the blue portion of the molecule is correspondingly small.

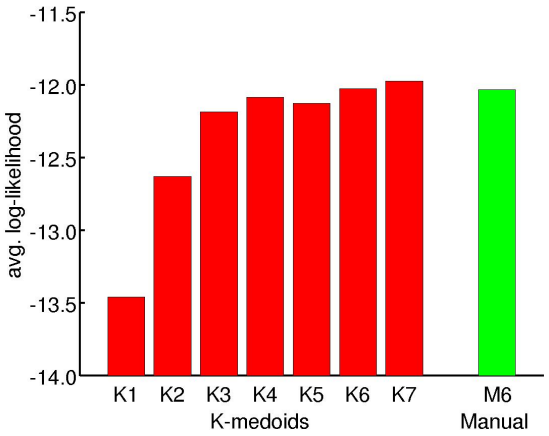


Figure 3.5: Average log-likelihood scores of alanine dipeptide HMM MDMs. $K1$ to $K7$ are initialized via K-medoids clustering, while $M6$ is initialized manually based on a 6-state partition according to Chodera *et al.* [29].

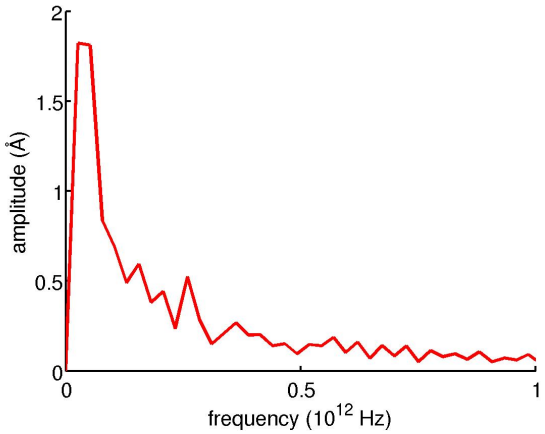
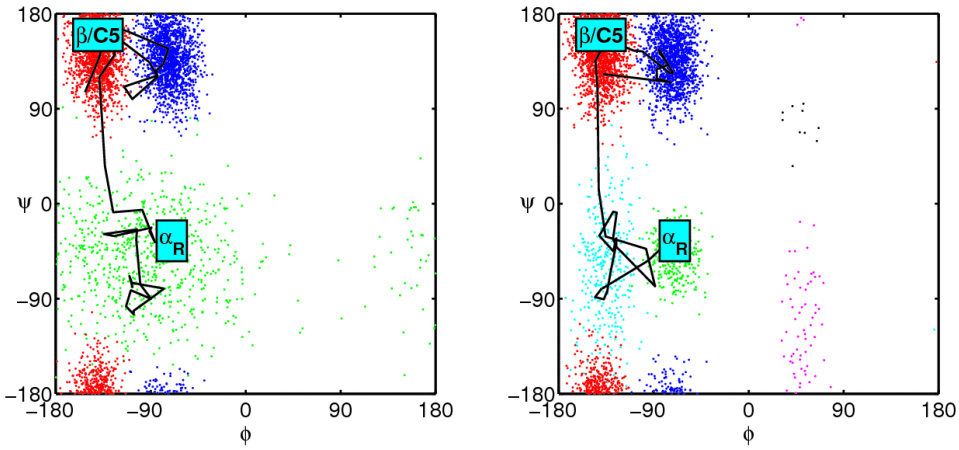


Figure 3.6: Frequency analysis of smoothed alanine dipeptide trajectory. Trajectories are sampled at 1.0×10^{12} Hz, and most of signal is within the Nyquist frequency of 0.5×10^{12} Hz. Amplitude is the average fluctuation of heavy atoms. Filtering is done using Gromacs [48], reducing fluctuations with period 0.4 ps by 85%, with period 0.8 ps by 50%, and with period 1.2 ps by 17%.

Fig. 3.5 plots the average log-likelihood scores of the models for a single transition step along a trajectory. Models $K3$ to $K7$ all achieve scores comparable to that of $M6$. The interesting finding is that although the score jumps substantially as we move from $K1$ to $K3$, the score of $K3$ is almost as good as those of $K6$ and $M6$. This indicates that for predicting the motion of alanine dipeptide, the simpler 3-state model $K3$ is almost as good as the 6-state model $M6$, which is derived from the well-accepted manual decomposition of the alanine dipeptide conformation space!

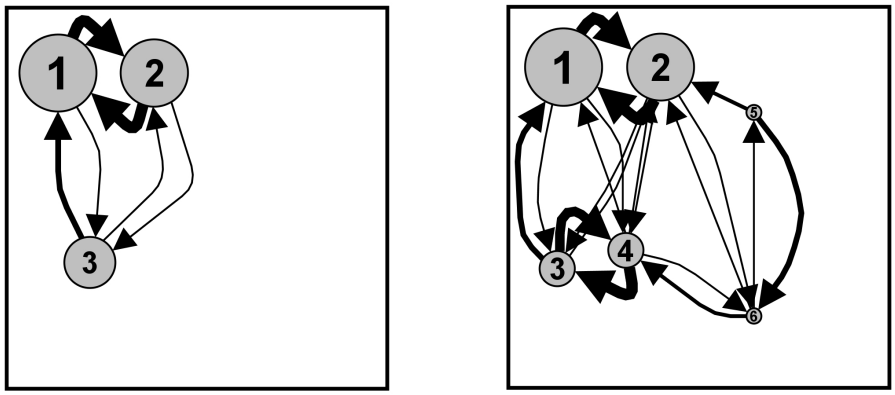
Fig. 3.7 shows the difference between $K3$ and $M6$. Both models accurately capture the frequently visited regions of the conformation space, shown in red and blue in Fig. 3.7*a*. These densely sampled regions correspond to energy basins that dominate the long-timescale dynamics, and the accurate modeling of these regions is crucial. For $K3$, the conformations shown in green capture a large, but less frequented region of the conformation space. Although $M6$ models the same region as two closely spaced clusters of conformations, the overall density and the location of the conformations are similar in both models. $M6$ also models the rarely visited region between $0 < \phi < 90$. Due to the transient nature of the protein in these conformations, the additional model complexity contributes little to the observable long-term dynamical phenomena. Therefore, the average log-likelihood score levels off when the number of states surpasses 3.

More importantly, dynamics is about the *change* of conformation over *time*, which corresponds to the transitions between states shown in Fig. 3.7*b*. The size of a node intuitively indicates the relative importance of a state, and the weight of an edge indicates the ease of transition between two states. We can see that the major transitions are between states 1 and 2 in both models. This corresponds to fast equilibration between the two regions.



$K3$ $M6$

a) Generated conformations.



$K3$ $M6$

b) Transitions between hidden states.

Figure 3.7: 3-state $K3$ versus 6-state $M6$ HMM MDMs of alanine dipeptide. In *a*), the black line traces a trajectory going from the α_R conformation to the $\beta/C5$ conformation. For the dots, each color represents conformations generated by a particular state. In *b*), the size of the node corresponds to the state's stationary distribution probability. The weight of an edge corresponds to the transition probability. The self-transitions are *not* shown.

What is interesting to note is that in $M6$, transitions between states 3 and 4 also occur with high probability. However in $K3$, the dynamics in the same region is modeled as the self-transitions of state 3 (edge not shown), but generalized over a *broader* distribution. Despite this difference, the likelihood scores between the models indicate that the additional detail of $M6$ does not predict trajectories more accurately.

More specifically, $M6$ attempts to provide a more precise prediction of the *conformation* by modeling this region with two individually tighter distributions of conformations. On the other hand, $K3$ models the same region with a single state, via a broader distribution of conformations.

However, dynamics is not just about the conformation, it is also about the *transition*. Consider a trajectory traversing *within* the region mentioned above. For $M6$, predicting the next transition requires accurately predicting whether a self-transition to the same state, or a transition to the neighboring state has occurred. Whereas $K3$ predicts the next transition perfectly via the self-transition of the only state in the region.

Consequently, the similar likelihood scores of the 3-state $K3$, versus the 6-state $M6$, suggests that there is a compensating trade-off between the increased precision in the predicted *conformation*, versus the loss of accuracy in the predicted *transition*. This is reasonable because when a basin of attraction is shallow, it is *unlikely* to trap a trajectory within a localized region for long. Therefore, trajectories have *insufficient* time to equilibrate internally, before a transition to a different region occurs. Consequently, the dynamics is insufficiently Markov for a more complex model to achieve a better accuracy.

More importantly, we are concerned about analyzing larger proteins with more complex motions. In particular, as the complex network of edges in the 6-state $M6$ already suggests, analyzing the K^2 transitions between many states is going to be a significant challenge. Although smaller states and edges can be pruned, such simplifications is undesirable. For example, it is important to know that although the transition out of states 1 and 2 are difficult, but it is *not* impossible. This is because such a transition corresponds to a change from a β-strand, to an α-helix. This can significantly affect a protein's overall structure, and therefore, biological function. Such critical information would be lost if edges with small transition probabilities are pruned before analysis. Therefore, in order to study larger proteins, we need to scale up our modeling approach.

Chapter 4

Hierarchical Model of Protein Motion Dynamics

We model a protein's dynamics to better understand how it achieves its biological function. With the HMM MDMs, we are able to abstract away unnecessary details, and model a protein's motion as transitions among energetically stable conformations. In addition, the graphical nature of the MDM allows biologically significant events to be analyzed intuitively.

However, as we scale up to larger proteins, the dynamics becomes significantly more complex. For example, since secondary structures involve mainly local interactions, they can be formed within a shorter, and different timescale than the overall folding process. Even in the native conformation, different parts of a protein can move at different frequencies depending on the extent individual parts are constrained. Therefore, to thoroughly study a larger protein, it is necessary to have a *collective* understanding of its different *types* of motion, over a *range* of timescales. Instead of constructing multiple models at different timescales, we want to build a single model for a combined analysis of a protein's complex dynamics.

4.1 Complex Dynamics of Large Proteins

Our goal in modeling protein dynamics is to provide a way to better understand how a protein achieves its biological function. Ideally, we would like to intuitively identify a protein's different types of motion, and the corresponding timescales, without prior knowledge of the protein. This is a significant challenge for larger proteins with complex dynamics.

For a large protein, the additional degrees of rotational freedom in a longer polypeptide permit a greater range and variety of motion. Multiple polypeptides can also come together to form the quaternary structure of the final functional protein. The structural organization into primary, secondary, tertiary and quaternary levels creates many opportunities for different parts of the molecule to interact and interlock with each other. As a result, the folding of a large protein is much more complicated.

Even in the native conformation, different parts of the same protein molecule can exhibit different dynamics. For example, an unconstrained loop on the exterior of a protein is likely to exhibit a wider range of motion at a higher frequency, as compared to a tightly packed portion in the interior of a folded molecule. At the same time, the sliding or shearing of different parts of a protein against each other, or the concerted opening and closing of the native conformation influence how a protein will interact or bind with other molecules [18, 32, 41].

More importantly, it is the collective contribution of different types of motion, over the whole range of timescales, that will determine how a protein can interact with its surrounding, and therefore, perform its function. Consequently, identifying the different structural changes (spatial), and the corresponding timescales at which they occur (temporal), is crucial for gaining biological understanding.

4.1.1 Dynamics over a range of timescales

Ideally, we would like to have a *single* model for analysis. In addition, we would like this model to be able to characterize the fastest biologically interesting phenomenon. More importantly, this model has to intuitively reveal *how* the fast conformational changes are related to longer timescale dynamics, *e.g.* the overall folding process. Characterizing the relationship between dynamics across different timescales is crucial to understanding because it is the accumulation of the high frequency motions that will ultimately determine the long term biological function.

In particular, we want to characterize dynamics according to these timescales:

a) $\triangle t$ timescale of ***fastest*** biological phenomenon, *e.g.* α-helix formation.

b) time after ***stationary distribution*** is attained, *e.g.* protein has folded.

c) the ***intermediate*** time frame *between* (a) and (b).

a) ***Fastest*** $\triangle t$ timescale. This is the timescale at which the fastest biologically interesting events occur, *e.g.* the formation of α-helices and β-strands. Compared to the femtosecond time step of MD simulation, $\triangle t$ is *already* long-timescale. $\triangle t$ is also the timescale at which MD trajectories should be sub-sampled to construct MDMs. Since the conformational change over $\triangle t$ time is already explicitly represented as probabilistic transitions in an HMM MDM, dynamics at this timescale can be directly analyzed.

b) ***Stationary distribution***. The dynamics beyond the stationary distribution is also relatively straightforward to analyze. The stationary distribution Π_s is a probability distribution of occupancy over the states that is *invariant* with time, *i.e.* $\Pi_s = \Pi_s A$, where A is the transition matrix. Consequently, the importance of stationary distribution is the relative *occupancy* of the states in Π_s. In particular, a state with a high stationary

probability will remain dominant indefinitely. As a result, the associated conformations will persist over time, and contribute substantially to a protein's biological function. Since Π_s can be obtained by calculating the eigenvector of A associated with the eigenvalue of 1, the dynamics beyond stationary distribution can also be directly analyzed in an HMM MDM.

c) ***Intermediate*** time frame. Most interesting is the change of conformation *beyond* the $\triangle t$ time of a single transition, but *before* stationary distribution has been reached. This intermediate time frame is particularly interesting because this is the time in which we can observe *how* a protein actually folds into the native conformation, *i.e.* the equilibrating *process*.

Unfortunately, the intermediate time frame is also the most difficult to investigate because biologically interesting events can occur anywhere within the broad range of timescales. Although it is possible to simulate the dynamics, this is not necessarily helpful in explaining the *process* of conformational change. For example, knowing the folded conformation does not explain *how* the protein actually folds, nor the constraints that limit the rate of folding. Similarly, even though it is possible to calculate the state occupancies Π_t after t time-steps by multiplying the transition matrix A, *i.e.* $\Pi_t = \Pi_0 A^t$. However, knowing Π_t does not explain *how* Π_t is reached.

Therefore, with dynamics varying over a range of timescales, it is impractical to attempt constructing multiple models at different timescales. Even if all the interesting timescales are known a priori, combining the analysis of multiple models will be difficult because different models may *not* be directly and easily comparable. This makes the complex dynamics of large proteins difficult to investigate, and there is a need to better characterize a protein's conformational *change*, over a *range* of timescales, all within a *single* model.

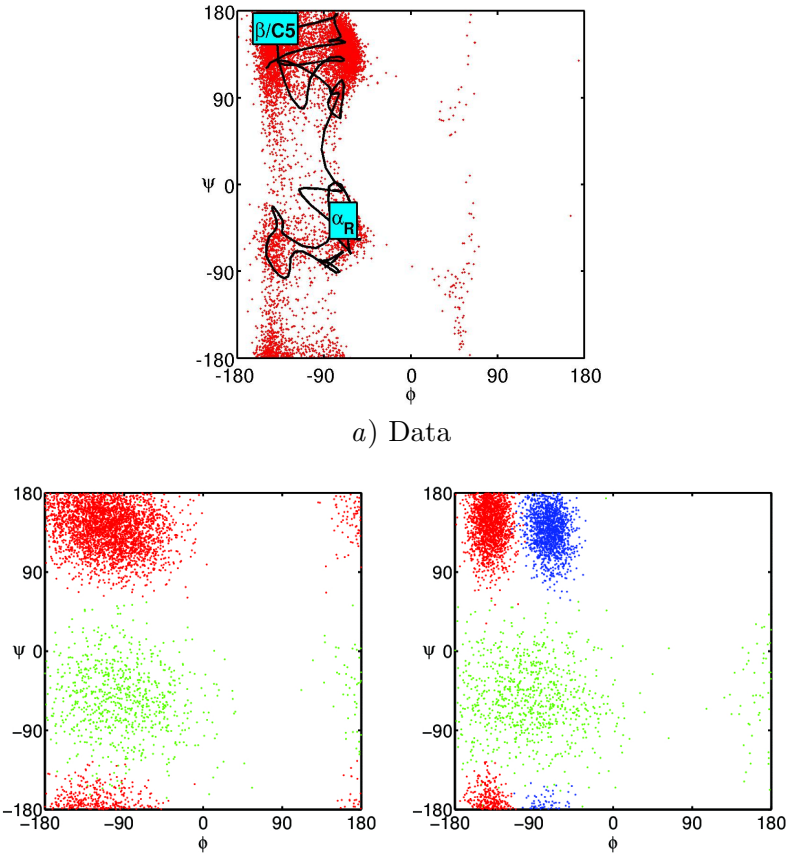
4.2 Hierarchical Model of Markovian Dynamics

For a large protein, due to the complexity of its dynamics, there can be a huge number of states in a *single* MDM, let alone *multiple* MDMs. Some of these states will be biologically important because they represent the protein's native conformation and will persist over time. On the other hand, some states may only be transient in nature and represent structural hurdles that a protein will eventually overcome, *e.g.* temporary misfolds. Furthermore, it is also possible for some states to be transient, yet crucial to biology because they represent a *necessary* stage along a *critical* pathway.

However, recognizing the biological significance of individual states is difficult because a protein molecule can follow a myriad of motion pathways. For example, although individual helices can form independently and simultaneously, their formation is also influenced by environmental conditions. As a result, helices of different molecules of the same protein may be formed at different times, following different pathways, corresponding to different order of formation. On the other hand, secondary structures tend to form earlier than the overall tertiary or quaternary structures. Therefore, there can be a global order of events embedded within a complex network of short-timescale transitions.

Consequently, identifying the biologically interesting dynamics requires a simultaneous analysis of *multiple* pathways, over *multiple* time steps of a MDM. Although graphical algorithms exist, identifying the best or top few pathways is insufficient for understanding because it is the *collective* contribution of all possible pathways that will determine a protein's function. Without an intuitive way to analyze the myriad pathways over a huge number of states, it is very difficult to identify limitations of a protein's structural transformation, and pinpoint interesting aspects of its dynamics.

4.2.1 Hierarchical clustering of dynamically similar states



a) Data

b) 2-state HMM MDM $K2$. *c*) 3-state HMM MDM $K3$.

Figure 4.1: 2-state vs 3-state HMM MDMs of alanine dipeptide. *a*) is the *original* data from MD simulation. *b*) and *c*) show conformations *generated* by HMM MDMs.

Earlier in Section 3.4.2, we investigated alanine dipeptide and have determined the 3-state model $K3$ as the most suitable HMM MDM. Fig. 4.1 shows the distribution of conformations generated by $K3$, and the 2-state model $K2$. The difference between these two models is in how the two energy basins in the $\beta/C5$ region are modeled (Fig. 4.1*a*). In $K3$, these energy basins are modeled as individual states and transitions between them occur with high probability. While $K2$ models these as a single state.

The difference between $K2$ and $K3$ highlights an interesting opportunity for an abstraction that will allow us to analyze the complex dynamics of a large protein. In particular, although $K3$ has a higher likelihood score (Fig. 3.5 on page 73) and better fits the original data, the simpler $K2$ does intuitively capture the dynamical similarity between the two energy basins in the $\beta/C5$ region. More specifically, due to the ease of transition between the two energy basins in $\beta/C5$, trajectories initiated in either basin are likely to equilibrate rapidly in $\beta/C5$, before transitioning to the α_R region. In other words, molecules with $\beta/C5$ conformations are likely to change similarly over time, *i.e.* dynamically similar. Therefore, although $K2$ is less precise within $\beta/C5$, it does provide a simpler abstraction for an intuitive understanding of the longer timescale dynamics between $\beta/C5$ and α_R.

Ideally, we want to use similar clustering to simplify the analysis of multiple pathways over numerous states, but *without* the loss of accuracy in predicting data. Fortunately, the accuracy of $K2$ is mainly affected by the *merging* of states in $\beta/C5$, and is not an inherent consequence of clustering. Since only a single distribution of conformations is used, $K2$ is naturally less able to distinguish which energy basin a trajectory is traversing.

There is a better way to capture the dynamical similarity between states. In many domains, there is a natural multiplicity of lengths or timescales, including handwriting [42], robot navigation [107], and surveillance [21]. In these applications, multi-level hierarchical models have been successful in learning the dependencies across different timescales. For example, the length of a sequence of characters is dependent on the word or phrase. Therefore, a hierarchical model that is more precise at the shorter timescale of phonemes, while generalizing over the longer timescale of phrases, can often provide a better recognition of the actual words spoken.

Similarly, a protein's motion trajectory is also a temporal sequence. Instead of characters, a protein has conformations. Instead of words, a protein has short sequences of *similar* conformations as a trajectory traverses through each energy basin. The rapid transitions within an *individual* energy basin results in the equilibration of a protein's motion before it escapes, and is the Markovian property we relied upon in Chapter 3.

More importantly, the dynamics of a *cluster* of energy basins can be modeled hierarchically. Conceptually, the dynamics of a cluster is similar to the rapid equilibration inside a *single* energy basin, but on a *broader* scale. In particular, when a trajectory eventually escapes the cluster, its future is likely to be independent of the initial energy basin it was in, and is therefore also Markov at the timescale it takes to escape the cluster. Satisfying the Markovian assumption will be crucial in allowing tools applicable to analyzing HMM MDMs to be applied to hierarchical MDMs.

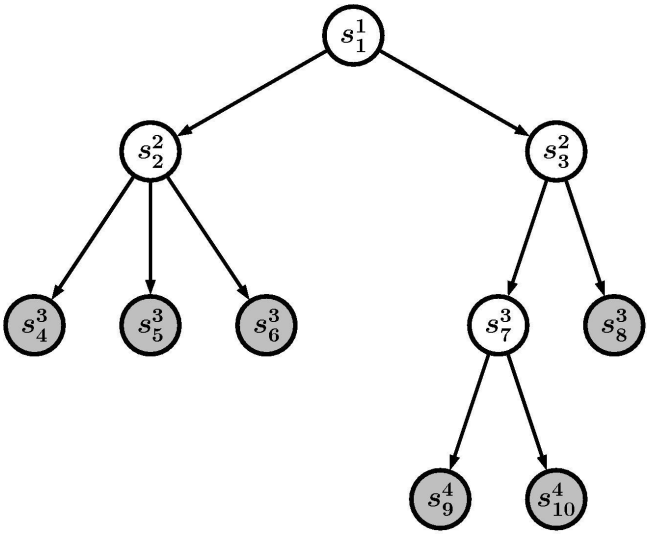
This suggests that we can make use of the clustering of energy basins to provide a multi-level hierarchical abstraction of dynamics. At the bottom of the hierarchy, we can preserve individual states and accurate short-timescale dynamics of *individual* energy basins. While higher up the hierarchy, we model the longer timescale dynamics of transitions between *clusters* of energy basins. In this way, a hierarchical model can provide a simpler abstraction of the long-timescale dynamics, without sacrificing the accuracy of short-timescale transitions.

4.2.2 Hierarchical Hidden Markov Model ($\mathcal{H}$HMM)

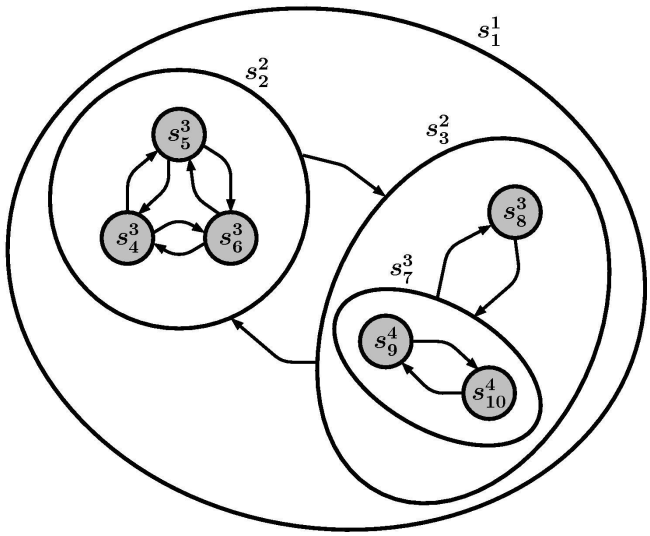
We propose to use the clustering of dynamically similar energy basins to construct a hierarchical model of protein dynamics. We use the hierarchy to recursively define protein dynamics according to parent-child relationships. In particular, *given* a cluster of energy basins, the cluster is represented as a subtree in the hierarchy, with the energy basins as the children, and a parent node to represent the whole cluster. Intuitively, energy basins in the same subtree are dynamically more similar to each other, as compared to energy basins in a different branch of the hierarchy, see Fig. 4.2.

More specifically, the hierarchy in an $\mathcal{H}$HMM MDM is a tree where the leaf nodes are ***basin-states*** corresponding to energy basins. Basin-states are equivalent to the hidden states in an HMM MDM, and have emission probabilities E that predict the observation of conformations. Nodes with children are ***macro-states*** that represent a (possibly nested) clustering of its descendants. A macro-state does *not* directly predict conformations, but is capable of generating a sequence of conformations by *recursively activating* its descendants. As such, the root s_1^1 at the top of hierarchy in Fig. 4.2*a* represents a nested clustering of all basins in the energy landscape.

More importantly, the multi-level structure of the hierarchy represents a multi-level separation of dynamics according to timescales. The states at the bottom of hierarchy capture transitions between energy basins at the $\triangle t$ timescale of the fastest biological phenomenon. While slower conformational changes are represented as transitions between parents at the top of hierarchy. As such, the explicit organization of transitions according to the hierarchy intuitively identifies the difficulty of a protein's various conformational changes, and relate its dynamics across different timescales.



a) Hierarchical organization of energy basins.



b) Nested clustering of energy basins.

Figure 4.2: An $\mathcal{H}$HMM MDM with general hierarchy. Each subtree in *a*) corresponds to a clustering of children nodes shown in *b*). Shaded nodes are ***basin-states*** representing energy basins. Unshaded nodes, and ellipses in *b*), are ***macro-states*** representing a clustering of its descendants. For state s_i^d, d is its level in the hierarchy, and i is its index number. Edges are different between sub-figures. Edges in *a*) are ***downward*** transitions representing parent-child dependencies. Edges in *b*) are ***horizontal*** transitions capturing dynamics within a cluster. The multi-level hierarchy represents a multi-level separation of dynamics according to timescales. Fast equilibration within a cluster occurs at the bottom of hierarchy, while slower conformational changes across clusters occur near the top.

Model parameters

We define an $\boldsymbol{\mathcal{H}}$HMM MDM by the tuple $\Theta = (\mathcal{C}, \mathcal{S}, \mathcal{H}, \Pi, B, A, E)$.

- The conformation space $\mathcal{C}$ of a protein.
- The set of states $\mathcal{S} = \{s_i^d \mid d \in \{1, 2, \ldots, D\},\ i = 1, 2, \ldots, |\mathcal{S}|\}$, where d is the level of a state in the hierarchy, D is the maximum depth of the hierarchy, and i is the state index (subscript may be omitted for clarity). A state s_i^d can be one of three types:
  - ***Basin-state*** loosely corresponds to an energy basin. Basin-states are the leaves of the hierarchy and are the only states that directly predict conformations.
  - ***Macro-state*** represents the clustering of its children. Macro-states do *not* directly predict conformations, and *cannot* be the leaves of the hierarchy.
  - ***Exit-state*** represents termination of transitions within the cluster represented by its *parent*. To avoid clutter, exit-states may *not* be shown (*e.g.* Fig. 4.2*a*), or *explicitly* shown (*e.g.* Fig. 4.3).
- The hierarchy $\mathcal{H}$ is a tree represented by a set of parent-child dependencies between states in $\mathcal{S}$. Each *subtree* in $\mathcal{H}$ represents a (possibly nested) clustering of energy basins. A parent can have many children, but a child can only have one parent. At the top of the hierarchy $\mathcal{H}$ is the singleton root s_1^1 that encompasses all states of the hierarchy. Each path in the hierarchy $\mathcal{H}$ from the root to a leaf node represents the activation of a basin-state, *i.e.* $(s_1^1, s_i^2, \ldots s_k^D)$, from root s_1^1 to a leaf s_k^D. In general, $\mathcal{H}$ needs *not* be a full tree and leaf nodes are allowed at $d < D$.

Transitions in an $\mathcal{H}$HMM MDM capture the *change* in a protein's conformation. However, transitions occur in a particular order. ***Downward*** transitions occur *before* the conformation q_t at time t has been predicted. While ***horizontal*** transitions occur *after* the conformation q_t at time t has been predicted. The ***termination*** of transitions within a cluster only occurs via a *horizontal* transition to an exit-state, *after* the conformation q_t at time t has been predicted. The parameters Π, B and A are as follows:

- For each ***macro-state*** s^d (subscript omitted), the prior probability over its L children at ***time*** $\boldsymbol{t = 0}$ is the vector $\Pi^{s^d} = \{\pi_i^{s^d} \mid i = 1, 2, \ldots, L\}$, where $\pi_i^{s^d} = p(s_i^{d+1}|s^d)$ is the probability of a ***downward*** transition from the parent s^d to its child s_i^{d+1}. (Here, L children *exclude* exit-state.)

- For each ***macro-state*** s^d, the ***downward*** transitions to its L children for ***time*** $\boldsymbol{t > 0}$ is the vector $B^{s^d} = \{b_i^{s^d} \mid i = 1, 2, \ldots, L\}$, where $b_i^{s^d} = p(s_i^{d+1}|s^d)$ is the probability of transiting from the parent s^d to its child s_i^{d+1}. (Here, L children also *exclude* exit-state.)

- For each ***macro-state*** s^d, the probability to transit among its L children for all time t is the matrix $A^{s^d} = \{a_{ij}^{s^d} \mid i, j = 1, 2, \ldots, L\}$, where $a_{ij}^{s^d} = p(s_j^{d+1}|s_i^{d+1})$ is the probability of a ***horizontal*** transition from child s_i^{d+1} to child s_j^{d+1}. For the ***exit-state*** in L, instead of transiting to its siblings, it transfers control to the parent s^d with probability of 1.

Finally, the emission probabilities E:

- For the basin-states at the leaves, the emission probabilities are denoted by $E = \{e_i^d(q) \mid d \in \{1, 2, \ldots, D\},\ i \in \{1, 2, \ldots, |\mathcal{S}|\},\ q \in \mathcal{C}\}$, where $e_i^d(q) = p(q|s_i^d)$ is the probability of observing conformation q when in basin-state $s_i^d \in \mathcal{S}$. Each $e_i^d(q)$ is defined over the *entire* conformation space $\mathcal{C}$.

More importantly, each subtree of an $\boldsymbol{\mathcal{H}}$HMM MDM is itself a probabilistic model capable of generating a sequence of conformations by *recursively activating* its descendants. If a macro-state is activated, it activates one of its children. The recursive activation continues down the hierarchy until a basin-state is activated. Then the activated basin-state predicts (or generates) a conformation through its emission probabilities. This is followed by the return of control to the top of the hierarchy via exit-states from the bottom up. The return of control up the hierarchy occurs until a horizontal transition to a macro or basin-state is made, at which point, the cycle of recursive activation continues.

Consequently, the state of the *whole* $\boldsymbol{\mathcal{H}}$HMM MDM that predicts a conformation at each $\triangle t$ time step is encoded by the vector $s_t = (s_1^1, s_i^2, \ldots, s_k^D)$ of states from the root s_1^1 to a leaf node s_k^D at level D in the hierarchy. More specifically, $(s_1^1, s_i^2, \ldots, s_j^{D-1})$ are macro-states, s_k^D is the only basin-state, and exit-states are excluded from s_t because they do not result in a conformation. Therefore, an $\boldsymbol{\mathcal{H}}$HMM MDM simulates the dynamics by changing from one branch of the hierarchy to another. In general, the hierarchy $\mathcal{H}$ needs *not* be a full tree and leaf nodes are allowed at $d < D$, where D is the maximum depth of the hierarchy.

4.2.3 $\mathcal{H}$HMM versus HMM MDMs

The key distinction between an $\mathcal{H}$HMM and an HMM is that an $\mathcal{H}$HMM models dynamics according to a *separation* of timescales, while an HMM sticks to *one* particular timescale. However, despite the different treatment of timescales, there is a correspondence between the two types of MDMs. In particular, an $\mathcal{H}$HMM is a more general version of an HMM, and a K-state HMM is equivalent to a "*flat*" $\mathcal{H}$HMM with K basin-states and a "*dummy*" root macro-state.

More importantly, we can transform between an $\mathcal{H}$HMM and an HMM and speed up our model construction algorithm (Section 4.3). To better understand why this is possible, we need to first discuss the similarity in how the dynamics ***within a cluster*** of energy basins is modeled. Then, we will discuss the different modeling of dynamics ***between clusters*** of energy basins. Finally, we will show how the transformation can be accomplished.

Similarity: dynamics *within* a cluster

Fig. 4.3 shows how an $\mathcal{H}$HMM models the dynamics within a cluster of energy basins. Since the change from state s_4^3 at time $t \times \triangle t$, to state s_5^3 at time $(t+1) \times \triangle t$ only involves changes in the leaf nodes of the hierarchy, a *direct* transition between basin-states is taken. Consequently, both an $\mathcal{H}$HMM and an HMM model the dynamics within a cluster explicitly as one-to-one transitions between states representing energy basins.

This is beneficial because the dynamics within a cluster involves frequent transitions that quickly equilibrate among the energy basins within several $\triangle t$ time steps. By modeling this dynamics directly as individual transitions, both $\mathcal{H}$HMM and HMM can precisely capture the *process* of equilibration within the cluster, *before* a transition exiting the cluster occurs.

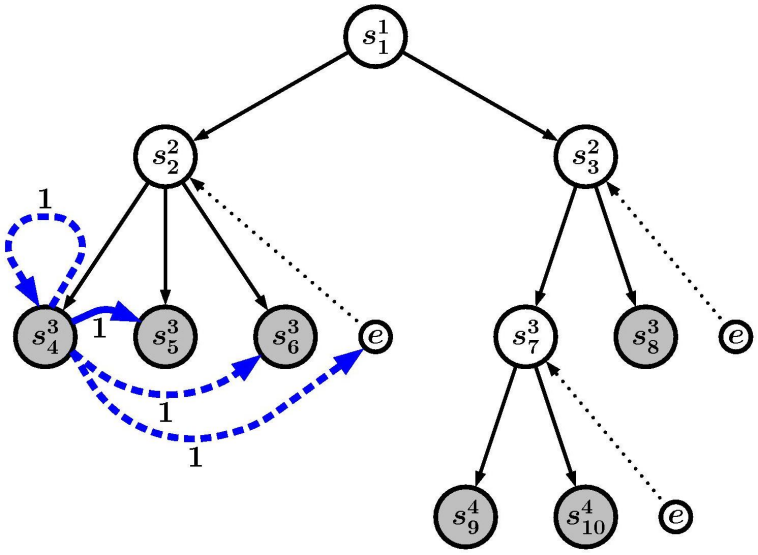


Figure 4.3: An $\mathcal{H}$HMM MDM illustrating transitions *within* a cluster. The transition shown is from state $s_t = (s_1^1, s_2^2, s_4^3)$ at time $t \times \triangle t$, to state $s_{t+1} = (s_1^1, s_2^2, s_5^3)$ at time $(t+1) \times \triangle t$. Since the transition only involves a change in the leaf node of the hierarchy, only the *solid* edge numbered **1** is taken. *Dashed* edges also numbered **1** are other possible transitions from basin-state s_4^3. Transitions of all edges numbered **1** sum up to a probability of 1. Additionally, due to the Markovian dynamics of an energy basin, the self-transition probability of a basin-state is usually significantly higher than the probability to transit to a different basin-state. Shaded nodes at the leaves are basin-states that emit conformations. Conformations are not shown in the diagram. Unshaded nodes are macro-states that represent a clustering of its children. Unshaded nodes labeled e are exit-states. A transition to an exit-state represents an exit from the cluster represented by the parent. Exits involve further transitions better explained in Fig. 4.4.

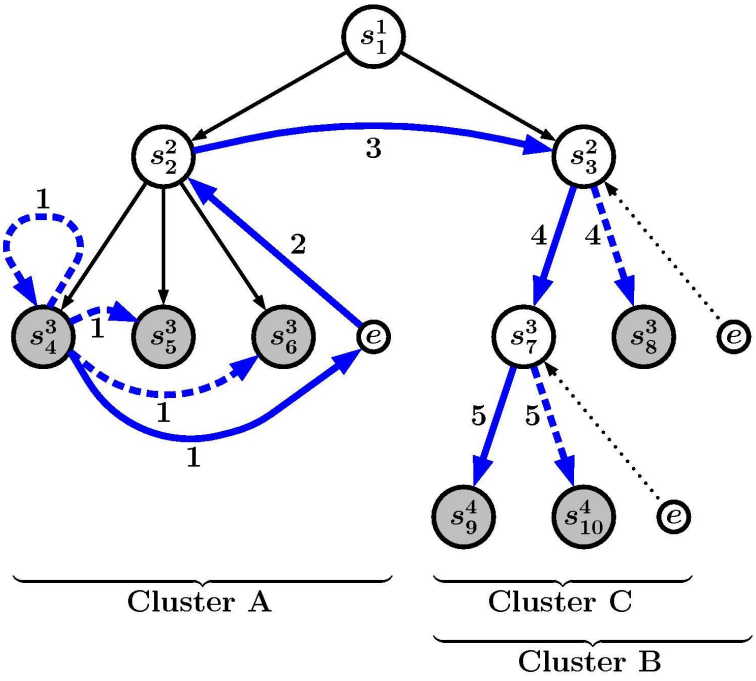


Figure 4.4: An $\mathcal{H}$HMM MDM illustrating transitions *between* clusters. The transition shown is from state $s_t = (s_1^1, s_2^2, s_4^3)$ at time $t \times \triangle t$, to state $s_{t+1} = (s_1^1, s_3^2, s_7^3, s_9^4)$ at time $(t+1) \times \triangle t$. This involves a sequence of 5 internal changes via *solid* edges numbered accordingly. *Dashed* edges that are also numbered indicate other possible transitions at each step. All possible transitions at each step sum up to a probability of 1. Shaded nodes at the leaves are basin-states that emit conformations. Conformations are not shown in the diagram. Unshaded nodes are macro-states that represent a clustering of its children. Unshaded nodes labeled e are exit-states. A transition to an exit-state represents an exit from the cluster represented by the parent. Since a MD trajectory can be simulated infinitely, there is no exit-state at level 2 in the hierarchy to terminate a trajectory. Additionally, since an $\mathcal{H}$HMM MDM *explicitly* simulates the dynamics within a cluster of energy basins, a macro-state representing a cluster only models transitions to other clusters and does *not* have a self-transition. A detailed discussion follows on page 94.

Difference: dynamics *between* clusters

The dynamics between clusters of energy basins is modeled differently by an $\boldsymbol{\mathcal{H}}$HMM and an HMM. Since an HMM does not distinguish which cluster an energy basin is in, transitions between all energy basins are directly modeled. More specifically, for a K-state HMM, a single K^2 matrix explicitly parameterized all the state-to-state transition probabilities.

However, an $\boldsymbol{\mathcal{H}}$HMM models the dynamics between clusters as *collective* transitions between the energy basins. The assumption is that transitions between clusters are likely to occur *after* multiple $\triangle t$ time steps. When a transition *actually* occurs, the trajectory is likely to have substantially equilibrated within the original cluster, such that its future is independent of its history in the original cluster, *i.e.* is Markov. Consequently, the dynamics between clusters can be approximated by collective transitions involving a sequence of internal changes.

Consider the example in Fig. 4.4, where the $\boldsymbol{\mathcal{H}}$HMM transits from basin-state s_4^3 in cluster A at time $t \times \triangle t$, to basin-state s_9^4 in cluster C at time $(t+1) \times \triangle t$. This involves a change of the whole $\boldsymbol{\mathcal{H}}$HMM from $s_t = (s_1^1, s_2^2, s_4^3)$ to $s_{t+1} = (s_1^1, s_3^2, s_7^3, s_9^4)$ via a sequence of 5 internal changes: ***exit*** from cluster A (**1** and **2**), ***transit*** to cluster B (**3**), and ***descent*** into an energy basin in cluster C (**4** and **5**).

- **Exiting** a cluster is equivalent to a transition to any energy basin outside the cluster. Due to the difficulty of exiting the cluster, a transition out of the cluster is relatively *less* likely, compared to a transition within the cluster. In addition, the equilibrating dynamics within the cluster means that when a trajectory actually exits the cluster, its future can be estimated via Markovian cluster-to-cluster transitions higher up the hierarchy.

Consequently in Fig. 4.4, a single *solid* edge numbered **1** represents the sum of all probabilities to exit cluster A. The upward edge numbered **2** transfers the control back to the parent s_2^2, which occurs with probability of 1 by definition. The sole purpose of an exit-state is to transfer control back to the parent so that a transition can be made to switch to a different cluster.

- **Transiting** between clusters of energy basins is the distinguishing feature of $\boldsymbol{\mathcal{H}}$HMM MDMs. The hierarchical structure enforces the assumption that dynamics within a cluster has sufficiently equilibrated, so that when a transition across clusters is *actually* taken, the future of an exiting trajectory is independent of its history in the original cluster, *i.e.* is Markov. Consequently, a transition *across* clusters is estimated based on the *collective* transitions from all energy basins in the source cluster, to all energy basins in the destination cluster. In Fig. 4.4, the solid edge numbered **3** represents the transition from cluster A to cluster B.

- **Descending** into basin-state s_9^4 within cluster C is modeled by downward transitions represented by the solid edges numbered **4** and **5** in Fig. 4.4. Due to the hierarchical dependencies, the descent relies on the equilibration of dynamics *outside* the destination cluster, in *other* clusters. Consequently, the downward transitions are estimated based on all trajectories entering the destination cluster. Once inside basin-state s_9^4, the MDM can generate the conformation q_{t+1} at time $(t+1) \times \triangle t$, and subsequently starts transiting among states within the new cluster, or exit.

Therefore, an $\boldsymbol{\mathcal{H}}$HMM MDM models the dynamics of a protein's motion via a sequence of transitions in the hierarchy over time. Although transitions between clusters involve additional changes, they beneficially reduce model complexity without losing accuracy in predicting dynamics (see Results later in Section 4.4). More importantly, each descent into a subtree corresponds to an entry into a cluster of energy basins. This results in a sequence of conformations to be generated, with a multiplicity in lengths due to the characteristics of the energy basins.

Although the clustering of states according to a hierarchy corresponds to a partition of the *state* space $\mathcal{S}$, it is *not* a partition of the *conformation* space $\mathcal{C}$. This is because the emission probabilities $e_i^d(q) = p(q|s_i^d)$ are still defined over the *entire* conformation space $q \in \mathcal{C}$. This follows from our assumption that the actual energy basins are *not* observed, and this uncertainty is reflected in the "*hidden*" nature of not just the basin-states, but the whole hierarchy.

Transformation between an $\boldsymbol{\mathcal{H}}$HMM and an HMM MDM

Despite the difference in how an $\boldsymbol{\mathcal{H}}$HMM and an HMM models the dynamics between clusters, it is possible to transform from one MDM to another. This is because both types of MDMs model protein dynamics as transitions between energy basins, and predicts a conformation at every $\triangle t$ time step. The difference is in *how* the predictions are made, which affects the information that can be preserved through the transformation.

For an $\boldsymbol{\mathcal{H}}$HMM transition *across* clusters from $s_t = (s_1^1, \ldots, s_i^{d^*}, \ldots, s_j^{D-1}, s_k^D)$, to $s_{t+1} = (s_1^1, \ldots, s_l^{d^*}, \ldots, s_m^{D-1}, s_n^D)$, where s_k^D and s_n^D are basin-states, while $s_i^{d^*}$ and $s_l^{d^*}$ share the same parent. This transition involves:

$$\hat{p}(s_{t+1}|s_t) = \prod_{\{s_j^d,\ s_k^{d+1}\} \in s_t,\ d \geq d^*} p(s_e^{d+1}|s_k^{d+1}) p(s_j^d|s_e^{d+1}) \ \times \quad \text{(4.1a)}$$

$$p(s_l^{d^*}|s_i^{d^*}) \ \times \quad \text{(4.1b)}$$

$$\prod_{\{s_m^d,\ s_n^{d+1}\} \in s_{t+1},\ d \geq d^*} p(s_n^{d+1}|s_m^d), \quad \text{(4.1c)}$$

where $p(s_e^{d+1}|s_k^{d+1})$ is a transition to a sibling exit-state, and $p(s_j^d|s_e^{d+1}) = 1$ is the transfer of control to parent. This corresponds to transitions in Fig. 4.4, where Eq. 4.1a is the *exit* from s_t, Eq. 4.1b is the *transit* across clusters, and Eq. 4.1c is the *descent* into s_{t+1}.

More importantly, $\hat{p}(s_{t+1}|s_t)$ is also a transition from one energy basin to another, similar to a transition in an HMM MDM. This leads us to the following possible transformations:

- **From $\boldsymbol{\mathcal{H}}$HMM to HMM**. Due to the hierarchical dependencies, $\hat{p}(s_{t+1}|s_t)$ is a product of transitions across the $\boldsymbol{\mathcal{H}}$HMM. Therefore, given an $\boldsymbol{\mathcal{H}}$HMM, it is possible to directly calculate $\hat{p}(s_{t+1}|s_t)$, and use that to construct the K^2 transition matrix of a K-state HMM. In addition, if the basin-states from the $\boldsymbol{\mathcal{H}}$HMM are used as the hidden states in the HMM, then both MDMs characterize the conformation space via the same set of emission probabilities. Consequently, although the resulting HMM loses the spatial and temporal organization of dynamics in the original $\boldsymbol{\mathcal{H}}$HMM, both MDMs will propagate and predict dynamics in the same way. No information is lost.

- **From HMM to $\mathcal{H}$HMM**. Given an HMM and a hierarchy $\mathcal{H}$, the transformation to an $\mathcal{H}$HMM *cannot* be accomplished directly. The reason is because in an $\mathcal{H}$HMM, the transitions *between* clusters of energy basins are *collective* transitions, and cannot be directly calculated from the HMM's transition matrix. The difficulty is due to the need for an appropriate normalization. Intuitively, a transition across clusters combines the transitions from multiple source basin-states, to multiple destination basin-states (*e.g.* Eq. 4.1b). Although it might seem natural to simply sum up the HMM probabilities and average over the number of states, this is wrong because the *occupancy* of the source states are unequal. Therefore, inference over data is needed to obtain an appropriately *weighted* average, and *direct* transformation from an HMM to an $\mathcal{H}$HMM is impossible.

  More importantly, due to dependencies introduced by the hierarchy, the resulting $\mathcal{H}$HMM may make different predictions of dynamics than the original HMM. For example in Fig. 4.4 (page 93), the horizontal cluster-to-cluster transition **3** is conditioned on the *whole* of cluster A. Therefore, through **3**, the transition from s_4^3 to s_9^4 becomes dependent on the collective outcome of all trajectories exiting cluster A. This effectively combines individual transitions between K^2 basins into a fewer number of cluster-to-cluster transitions. By modeling the dynamics between clusters collectively, the transformation to an $\mathcal{H}$HMM can lose detailed information of individual transitions.

  However, the error due to the collective transitions in predicting MD trajectories should be minimal for a suitably constructed hierarchy, and the benefit of an $\mathcal{H}$HMM in providing a more intuitive representation of complex dynamics is much more valuable to biological understanding.

4.2.4 What is a good $\boldsymbol{\mathcal{H}}$HMM MDM?

The purpose of modeling the dynamics of protein motion is to better understand their motion at the molecular level. Therefore, an ideal model should reveal a protein's change of conformation *from* the $\triangle t$ timescale of the fastest biologically significant phenomenon, till *after* equilibration in a stationary distribution. Since an $\boldsymbol{\mathcal{H}}$HMM MDM relies on the Markov property to propagate dynamics beyond the timescale it is trained at, a good $\boldsymbol{\mathcal{H}}$HMM MDM should be constructed with data sub-sampled at the $\triangle t$ timescale, while making use of the hierarchical structure to model dynamics longer than the $\triangle t$ timescale.

Additionally, a good model of protein motion dynamics should be able to accurately predict the *change* of a protein's *conformation* over *time*. Therefore, we still compare models in terms of their ability to predict MD trajectories and calculate the likelihood $p(\mathcal{D}|\Theta)$, which is the probability that a dataset $\mathcal{D}$ of MD trajectories will occur under the model Θ.

Specifically, given a set of MD trajectories $\mathcal{D} = \{\mathcal{D}_i \mid i = 1, 2, \ldots\}$, where trajectory $\mathcal{D}_i$ is a sequence of conformations $\mathcal{D}_i = (q_0, q_1, \ldots, q_T)$, and q_t is conformation at time $t \times \triangle t$. The likelihood of model Θ for trajectory $\mathcal{D}_i$ is:

$$p(\mathcal{D}_i|\Theta) = \sum_{Q \in \mathcal{S}^T} \Big(\hat{p}(s_0) \prod_{t=1}^{T} \hat{p}(s_t|s_{t-1}) \prod_{t=0}^{T} \hat{p}(q_t|s_t) \Big), \tag{4.2}$$

where $s_t = (s_1^1, s_i^2, \ldots, s_j^D)$ is the state of the $\boldsymbol{\mathcal{H}}$HMM MDM at time $t \times \triangle t$, this D is height of hierarchy, and $\sum_Q$ is a sum over all possible sequences of state assignments. Due to the hierarchical structure, $\hat{p}(s_0)$, $\hat{p}(s_t|s_{t-1})$, and $\hat{p}(q_t|s_t)$ now requires resolving dependencies from root s_1^1 to basin-state s_j^D using the parameters Π, B, A and E [42]. Especially state transitions $\hat{p}(s_t|s_{t-1})$, which now require $O(D)$ internal changes (Eq. 4.1).

The likelihood is a suitable measure of a model's accuracy in predicting dynamics because at any moment in time, an $\mathcal{H}$HMM MDM has an expectation of a protein's conformation based on a probabilistic distribution over states of the system. Furthermore, through emission probabilities E of the basin-states, this expectation extends over the entire conformation space $\mathcal{C}$. The likelihood will only be high if a conformation is *actually* observed where the model *expects* it to be observed. With each subsequent time step, the model propagates its internal dynamics, updates the probability distributions, and makes a new prediction. Consequently, if a model Θ consistantly predicts the *right* conformation at the *right* time, its likelihood over the the entire dataset $\mathcal{D}$ will be high:

$$p(\mathcal{D}|\Theta) = \prod_i p(\mathcal{D}_i|\Theta). \tag{4.3}$$

Conversely, if a model Θ persistently predicts the *wrong* conformation at the *wrong* time, its likelihood score will be low. In order to demonstrate the reliability of using likelihood for model selection, we will make use of the example later in Section 4.4.1, which contrasts the difference in a model's likelihood in predicting *true* data, versus *false* data.

4.2.5 Benefits of $\mathcal{H}$HMM MDM

Our $\mathcal{H}$HMM MDM offers a multi-level, nested clustering organization of a protein's motion dynamics. Each subtree in the hierarchy corresponds to a separation of fast equilibration within a cluster of energy basins, versus slower transitions with the outside. This allows scientists to *intuitively* identify a protein's major conformational changes. For example, a highly nested cluster of energy basins is likely to be more stable and harder to escape. Consequently, a conformational change *out* of the cluster is more biologically significant than a change *within* the cluster. More interestingly, by characterizing the conformation structure of the states *before* a difficult transition, scientists can further understand the reasons that *limit* the corresponding change, *e.g.* structural factors that *prevent* folding.

Furthermore, our $\mathcal{H}$HMM MDM identifies the *timescale* of conformational change. More specifically, the expected number of $\triangle t$ transitions to escape from an energy basin, or a cluster of energy basins can be calculated as: $\tau_i = \frac{1}{1-a_{ii}}$, where a_{ii} is self-transition probability of a macro or basin-state (Eq. 4.5 on page 109). Therefore, by making use of the hierarchical structure, it is also possible to estimate the *timescale* beyond which a particular conformational change is expected to occur.

Also advantageous is the reduction in model parameters with minimal loss in accuracy. This is possible because fast equilibration *within* clusters of energy basins allow transitions *between* clusters to be collectively estimated. As a result, the K^2 transition matrix of K energy basins can potentially be reduced to a much smaller $\boldsymbol{k}^2$ transitions between $\boldsymbol{k}$ clusters, and a sub-matrix for each cluster. Therefore, for a good hierarchy with suitably clustered energy basins, the impact on the accuracy in predicting MD trajectories should be minimal.

More importantly, transitions *between* clusters of energy basins correspond to energetically difficult, but often *necessary*, conformational changes a protein has to undergo. During MD simulation, these conformational changes are only observable after long simulations. Consequently, there is relatively less data to estimate transitions *across* clusters of energy basins, as compared to transitions *within* clusters. Therefore, by combining trajectories traversing across clusters into a collective transition probability value, it is more likely to obtain a more reliable estimate.

4.3 Model Construction

A model of protein dynamics needs to capture a protein's change of conformation over time. This requires a search for the most suitable model across *both* space and time. In terms of ***space***, not only are we interested in identifying the number of biologically significant states, we also want to discover the clustering relationship between them. In terms of ***time***, we are interested in conformational changes from the $\triangle t$ timescale of the fastest biologically significant phenomenon, till after equilibration in the stationary distribution. By combining the spatial and temporal organization of a protein's dynamics in a *single* $\boldsymbol{\mathcal{H}}$HMM MDM, we hope to provide a way for scientists to intuitively understand *how* a protein achieves its function.

However, constructing an $\boldsymbol{\mathcal{H}}$HMM is complicated by the need to *discover* the hierarchy. Given a particular timescale, we can identify the number of biologically significant states by searching for the most suitable HMM. However, it is impossible to simultaneously search *both* the number of states and all possible hierarchies for the most suitable $\boldsymbol{\mathcal{H}}$HMM. It is also uncertain how many different timescales should be sampled to construct models. Too detailed, and numerous models will result. Too coarse, and biologically interesting phenomenon may be missed.

More crucially, even if we construct multiple HMMs at different timescales, it is *insufficient* because the value of an $\boldsymbol{\mathcal{H}}$HMM is in the *collective* characterization of dynamics across different timescales. Since different HMMs are constructed from data sub-sampled at different timescales, the states in different HMMs may not be directly comparable, nor easily combinable into a *single* MDM. Consequently, we need a different approach to determine the hierarchical dependencies and construct an $\boldsymbol{\mathcal{H}}$HMM MDM.

We propose a three stage process to construct the most suitable $\boldsymbol{\mathcal{H}}$HMM $\Theta_{\mathcal{H}}$.

- **HMM construction**. We focus on characterizing the spatial complexity of dynamics, and search for the most suitable K-state HMM Θ_K at the $\triangle t$ timescale (Section 4.3.1).
- **Hierarchy construction**. We focus on the temporal organization of dynamics, and attempt to identify hierarchical dependencies $\mathcal{H}$ embedded within the K-state HMM Θ_K (Section 4.3.2).
- **$\boldsymbol{\mathcal{H}}$HMM construction**. We construct the most suitable $\boldsymbol{\mathcal{H}}$HMM $\Theta_{\mathcal{H}}$ using *both* HMM Θ_K and the hierarchy $\mathcal{H}$. This includes estimating an initial $\boldsymbol{\mathcal{H}}$HMM (Section 4.3.3), optimization (Section 4.3.4), and determining the most suitable $\boldsymbol{\mathcal{H}}$HMM $\Theta_{\mathcal{H}}$ (Section 4.3.5).

The reason we use the K-state HMM Θ_K as the basis of hierarchical construction is because Θ_K is *already* a model of a protein's dynamics. As a MDM constructed at $\triangle t$ timescale, Θ_K has also abstracted away noisy atomic vibrations at the femtosecond timescale. More crucially, Θ_K has *already* determined the suitable K number of energy basins, and as a result, significantly reduced the search space. Under the Markovian property, Θ_K is also capable of simulating dynamics in place of raw MD data, and will allow us to efficiently search for the hierarchy across different timescales. Furthermore, Θ_K beneficially allows us to take advantage of mathematical techniques to further our analysis.

Before we begin construction, we divide the data into 3 separate sets. The first is a training set $\mathcal{D}_{train}$ used to train multiple HMMs. The second is a test set $\mathcal{D}_{test1}$ used to determine the most suitable K-state HMM Θ_K. Both $\mathcal{D}_{train}$ and $\mathcal{D}_{test1}$ are then used to train multiple $\boldsymbol{\mathcal{H}}$HMMs. The third is also a test set $\mathcal{D}_{test2}$, and is used to determine the most suitable $\boldsymbol{\mathcal{H}}$HMM $\Theta_{\mathcal{H}}$.

4.3.1 Constructing the most suitable K-state HMM Θ_K

We follow steps in Chapter 3 to construct the most suitable K-state HMM Θ_K. To ***recap***, the $\triangle t$ timescale of the fastest biological phenomenon is usually obtainable from wet lab experiments. This is the timescale at which the fastest biologically interesting events occur, *e.g.* the formation of α-helices and β-strands. Therefore, by constructing the most suitable K-state HMM Θ_K at the $\triangle t$ timescale, we abstract away noisy atomic vibrations, while preserving the biologically interesting dynamics longer than $\triangle t$.

The steps to construct the most suitable K-state HMM Θ_K are:

- **Data preparation**. MD trajectories are smoothed and sub-sampled at the $\triangle t$ timescale. In addition, due to the high dimensionality of large proteins, a distance graph that better captures the kinetic distance between conformations than RMSD is needed, see Section 4.4.2.

- **K-medoids clustering**. We use K-medoids algorithm to identify compact clusters of conformations. Each cluster represents a potential energy basin, and serves as the basis of a basin-state.

- **HMM initialization**. We use the clustering information to create an initial HMM Θ_0. Θ_0 is already a model of dynamics, with states representing energy basins, and transitions corresponding to a protein's change of conformation.

- **HMM optimization**. We initialize EM algorithm with Θ_0 and optimize for the model Θ with maximum $p(\mathcal{D}_{train}|\Theta)$.

- **HMM selection**. We score each model Θ on the first test dataset $\mathcal{D}_{test1}$, and choose the most suitable K-state HMM Θ_K.

4.3.2 Constructing the hierarchy $\mathcal{H}$

The K-state HMM Θ_K constructed earlier in Section 4.3.1 is *already* of a model of a protein's dynamics. However, with a complex network of K^2 transitions, it can be difficult to visually analyze Θ_K, and gain an intuitive understanding of how a protein achieves its function.

Despite the difficulty of directly using Θ_K for biological understanding, we constructed Θ_K first because Θ_K captures the protein's dynamics from the $\triangle t$ timescale onwards. More importantly, Θ_K allows us to apply mathematical tools to analyze the dynamics, *without* relying on data.

In particular, we want to identify the equilibration timescales of a protein's dynamics using Θ_K. Intuitively, if a clustering of energy basins exists, a trajectory is likely to transition rapidly *within* the cluster, before eventually transitioning *out* of the cluster. Consequently, there is likely a *separation* of timescales between the dynamics *within* the cluster, and *between* clusters. Therefore, the idea is this: if we can identify a *separation* of equilibration timescales between different parts of Θ_K, we can identify the corresponding clustering of energy basins, and then build a hierarchical model on top of it.

We use a *combination* of spectral clustering and hierarchical constraints on timescales to construct the hierarchy. The reason is because traditionally, spectral clustering only provides a bi-partition of the system, and the equilibration timescale between the two partitions. Although informative, this does *not* fully satisfy our need for a hierarchical clustering, which requires a *separation* of timescales between *various* parts of the system.

In the following pages, we will first discuss the details of spectral clustering and the hierarchical constraints separately. Then, we will combine them into our hierarchy construction algorithm given on page 110.

Spectral clustering

We use the eigen-decomposition of a transition matrix to guide us in constructing a hierarchy. For ergodic systems where every state can be reached from any other state, there exists a stationary distribution Π_s over states of the system. For protein dynamics, this corresponds to the equilibration into a protein's native conformation. More importantly, the transition matrix of such a system can be decomposed into a set of eigenvalues $\{\lambda_i\}$ and eigenvectors $\{u_i\}$ that can be used to represent the probability distribution of the system Π_t at time $t \times \triangle t$:

$$\Pi_t = \Pi_0 A^t = \sum_i c_i \lambda_i^t u_i, \tag{4.4}$$

where Π_0 is the initial distribution of the system at time $t = 0$, and c_i are coefficients determined with initial condition Π_0, *i.e.* $c_i = \sum_j \frac{u_{i,j}\Pi_{0,j}}{\Pi_{s,j}}$, where the second subscript refers to a particular element in the vector [34, 67].

Therefore, the behavior of the system can be described as a sum of *modes* decaying over time. In particular, the eigenvector u_1 with an eigenvalue of 1 does *not* decay ($\lambda_1^t = 1^t$), and corresponds to the stationary distribution Π_s. Other eigenvectors u_i have eigenvalues $\lambda_i < 1$, and decays exponentially with an implied timescale of $-\frac{\triangle t}{\ln \lambda_i}$ [60]. More specifically, each eigenvector u_i with $\lambda_i < 1$, represents aggregate transitions between states with *positive* values in u_i, and those with *negative* values [34, 67].

The clustering is done by using each eigenvector to partition the system into *two*. For each eigenvector u_i with $\lambda_i < 1$, states with positive values in u_i are assigned to one partition, while those with negative values are assigned to another. This creates a *two*-way partition with a timescale determined by the corresponding eigenvalue λ_i. A *multi*-way partition can be done with

the combined used of k eigenvectors to create possibly 2^k partitions [35, 100].

Particularly interesting are the slow modes with eigenvalues *nearly* 1. These eigenvectors represent rate limiting processes that takes relatively long to equilibrate. By using only eigenvectors with λ_i *nearly* 1, clusters with slow *inter*-cluster transitions can be identified. This has been successfully used to identify *meta-stable* conformations of proteins that undergo relatively slow, but biologically important conformational changes [29, 54, 101].

Hierarchical constraints on timescales

Although we will make use of spectral clustering, we have additional requirements on the hierarchy. Namely, we want each level to correspond to a particular timescale, and the number of levels to reflect the granularity of temporal separation in a protein's dynamics. This requires a measure of timescale based on *individual* clusters after *multiple* partitioning, as opposed to the eigenvalue λ_i timescale of a particular *mode* of the system.

In addition, since we measure the goodness of an $\boldsymbol{\mathcal{H}}$HMM according to accuracy in predicting MD trajectories. This requires the hierarchy to have significant *separation* of timescales for transitions *within* each cluster, versus transitions *between* clusters, so that the collective transitions across the hierarchy can be accurately approximated.

However, we do *not* have a proper $\boldsymbol{\mathcal{H}}$HMM at this stage to verify the accuracy of a hierarchy in predicting MD trajectories. Calculating transition times is also rather involved (see Chapter 5), and making pair-wise comparisons between states will further complicate the construction.

Instead, we check the *escape time* of each cluster, and enforce the additional constraint that a parent state has to have longer escape time than any of its children. The idea is that for a cluster to have a significantly longer

escape time compared to its internal energy basins, trajectories that exited an internal energy basin has to go to somewhere else *within* the cluster. Therefore, a cluster with a good separation of escape times between the parent and child is likely to have fast equilibrating dynamics within it.

This is where the use of Θ_K is beneficial because the escape time can be calculated from the expected number of transitions τ needed to escape from a state in Θ_K. In particular, we condition on the probability of transiting to the same state a_{ii}, or to a different state $(1 - a_{ii})$ in one step:

$$\begin{aligned} \tau_i &= 1 + a_{ii}\tau_i + (1 - a_{ii})0 \\ &= \frac{1}{1 - a_{ii}}, \end{aligned} \tag{4.5}$$

where a_{ii} is the self-transition probability of state i in Θ_K. Escape time of a *cluster* can be similarly defined based on the collective transitions within a *set* of states in Θ_K (Eq. 4.8 on page 113).

By combining spectral clustering with the hierarchical constraints in Algo. 1, we can make use of Θ_K to identify a hierarchical clustering of the underlying energy basins. However, since the hierarchy has *not* been verified against MD data, we still need to construct an $\mathcal{H}$HMM MDM later.

Algorithm 1 Constructing a hierarchy. Explanations are on page 111.

Require: $\Theta_K, \mathbb{E}$

1: $\mathcal{G} = \{\mathcal{S}, \mathcal{H}\} \leftarrow \Theta_K$; //initialization
2:
3: **for** $g_i \in \mathcal{G}$ **do**
4:
5: $A^{sub} = matrix\ (g_i, \mathbb{E})$; //sub-matrix construction
6: $\{\{\lambda_i\}, \{u_i\}\} = eigen\ (A^{sub})$; //eigen-decomposition
7:
8: $g_{new} = g_i$; //creates a temporary copy
9:
10: //multi-way partition
11: **for** $j = 2 \ldots dim\ (A^{sub})$ **do**
12:
13: $g_{tmp} = partition\ (g_i, \{u_2, u_3, \ldots, u_j\})$;
14: $g_{tmp} = constrain\_escape\_time\ (g_{tmp}, \mathbb{E})$;
15:
16: **if** $j == 2$ **then**
17: $threshold = set\_escape\_time\_threshold\ (g_{tmp})$;
18: **end if**
19:
20: **if** $g_{new} == g_{tmp}\ ||\ min\_cluster\_escape\_time\ (g_{tmp}) < threshold$ **then**
21: go to line 28;
22: **else**
23: $g_{new} = g_{tmp}$;
24: **end if**
25:
26: **end for**
27:
28: **if** $g_i ! = g_{new}$ **then**
29: $\mathcal{G} \leftarrow replace\ (g_i, g_{new})$;
30: save $\mathcal{G}$;
31: **else**
32: mark g_i to be skipped in future iterations;
33: **end if**
34:
35: **end for**

Hierarchy construction algorithm

We recursively construct the hierarchy from the *top down*, by identifying clusters with the *longest* escape time *first*. We require expectations $\xi_t(m,n) = p(s_t = m, s_{t+1} = n)$ from Eq. 3.13 (page 61), where m, n are states in the K-state HMM Θ_K, and are treated as basin-states in an $\boldsymbol{\mathcal{H}}$HMM. More specifically, we require the expectations summed over all training data $\mathcal{D}_i \in \mathcal{D}_{train}$ inferred using Θ_K:

$$\mathbb{E}(m,n) = \sum_{\mathcal{D}_i \in \mathcal{D}_{train}} \sum_{t=0}^{T-1} \xi_t(m,n), \tag{4.6}$$

Details of the construction algorithm Algo. 1 (page 110):

Line 1: **Initialization**. We create a simple tree $\mathcal{G} = \{\mathcal{S}, \mathcal{H}\}$ by putting all states from the K-state HMM Θ_K as children of the root in $\mathcal{G}$. This creates a "*flat*" hierarchy with *one* cluster of K basin-states.

Line $3-35$: **Iteration**. We iteratively partition $\mathcal{G} = \{\mathcal{S}, \mathcal{H}\}$ at the leaves to create clusters deeper in the hierarchy, at increasingly shorter timescales. Specifically, each candidate subtree $g_i \in \mathcal{G}$ is a subtree with 2 levels, with one parent on top, and all children are basin-states. At the end of each iteration, either g_i will be partitioned into sub-clusters and subtrees will be created, or if unsuccessful, g_i is marked and skipped.

Line 5: **Sub-matrix construction**. We assume g_i is an *enclosed* system, and create a transition matrix A^{sub} among the basin-states in g_i, $A^{sub} = \{a^{sub}_{mn} \mid m,n \in basin(g_i)\}$:

$$a^{sub}_{mn} = \frac{\mathbb{E}(m,n)}{\sum_{n \in basin(g_i)} \mathbb{E}(m,n)}, \tag{4.7}$$

where the normalization is over all basin-states in g_i.

Line 6: **Eigen-decomposition**. The eigenvalues $\{\lambda_i\}$ and eigenvectors $\{u_i\}$ of A^{sub} provide the different modes of transitions *within* g_i. We sort the eigenvectors in descending values of the eigenvalues, *i.e.* $\lambda_1 > \lambda_2 > \lambda_3 \ldots$ *etc.*. Specifically, the eigenvector u_1 with $\lambda_1 = 1$ corresponds to the stationary distribution of the *enclosed* system g_i. We are interested in using the slower modes (u_i with λ_i nearly 1) to create partitions corresponding to clusters *within* the enclosed system g_i.

Lines $11 - 26$: **Multi-way partition**. We attempt to create multiple subtrees within g_i by using the partition information provided by multiple eigenvectors. Since each eigenvector u_i indicates a mode of transitions that decays over time, a two-way partition can be created by assigning states with positive values in u_i to one partition, and states with negative values in u_i to a second partition. More importantly, if we *overlap* the partition information from a total of j eigenvectors, a maximum of 2^j partitions can be created *simultaneously* [34, 67].

However, due to the *multi-way* partitioning, the timescales of the resulting partitions are *different* from the timescales indicated by the eigenvalues $\{\lambda_i\}$ of the *two-way* modes $\{u_i\}$.

Lines 13: **Partition**. For each new partition within g_i, a macro-state is inserted into the hierarchy to create a new tree g_{tmp}. g_{tmp} has 1 root, a maximum of 2^j macro-states as children of the root, and basin-states at the leaves. Since *zero* basin-state is added, the *total* number of basin-states in $\mathcal{G}$ remains at K.

Lines 14: **Constraint on escape time**. We enforce the constraint that each *parent* must have a *higher* self-transition probability (which corresponds to escape time) than its *children*. The self-transition

probability of a state s_i^d (which can represent a cluster) is:

$$a_{ii}^d = \frac{\sum_{m \in basin(s_i^d)} \sum_{n \in basin(s_i^d)} \mathbb{E}(m,n)}{\sum_{m \in basin(s_i^d)} \sum_{n \in basin(\mathcal{S})} \mathbb{E}(m,n)}, \tag{4.8}$$

which is estimated from transitions originating from s_i^d. Since we are interested in the timescale of individual clusters with respect to the *whole* system, the normalization is with respect to *all* transitions originating from s_i^d, to all basin-states in $\mathcal{S}$.

If the constraint is not satisfied, the parent is eliminated, and its children are re-attached to the grandparent. The function $basin(\cdot)$ returns basin-states in the subtree rooted at state s_i^d, or in $\mathcal{S}$.

Lines $16-18$: **Threshold**. The threshold affects the separation between different levels in the hierarchy, *i.e.* separation in timescales. In signal processing techniques such as wavelet transform, the frequency spectrum is usually separated in equal halves with each iteration [22, 72]. However, interesting dynamics may be clustered within narrow range of timescales, and a generic rule such as halving the bandwidth may produce a hierarchy that *fails* to provide sufficient detail at the crucial timescales. Therefore, we threshold each additional level of the hierarchy on the escape time of the slowest dynamics within g_i, *i.e.* two-way partition of g_i. Additional partitions with timescales too different from the slowest dynamics within g_i are reserved for *future* levels deeper in the hierarchy.

Lines 20: **Terminating multi-way partition**. We terminate when an additional eigenvector fails to increase the partitioning in g_i (*i.e.* $g_{new} == g_{tmp}$). Or results in clusters with escapes times too fast compared to the slowest dynamics in g_i (*i.e.* $min\_cluster\_escape\_time\ (g_{tmp}) < threshold$).

In summary, each iteration through Algo. 1 creates a new level deeper in the hierarchy, where transitions occur at an increasingly shorter timescale. Although we used constraints to avoid creating hierarchies that deviate from the Markovian dynamics of clusters of energy basins, we do not know at this stage which hierarchy will lead to the most suitable MDM. Therefore, we save the hierarchy at each iteration to construct a candidate $\boldsymbol{\mathcal{H}}$HMM in the next stage (Lines $29-30$). Lastly, for each macro-state s_i^d (except the root s_1^1), an exit state s_e^{d+1} is added to its children to complete the hierarchy.

4.3.3 Estimating $\mathcal{H}$HMM parameters

With the K-state HMM Θ_K and the hierarchy $\mathcal{H}$, we are now ready to construct an $\mathcal{H}$HMM MDM. Constructing an $\mathcal{H}$HMM is still necessary because the hierarchy $\mathcal{H}$ has been derived from Θ_K, and has *not* been directly verified against MD data. Therefore, although a hierarchy with multiple levels is more informative, it might violate the Markovian property and fail to predict MD trajectories accurately enough. Consequently, despite the care we have taken in constructing $\mathcal{H}$, we do not yet know which iteration of the hierarchy is the *most suitable* for biological understanding.

The tree $\mathcal{G} = \{\mathcal{S}, \mathcal{H}\}$ constructed in Section 4.3.2 is a hierarchy with all necessary states $\mathcal{S}$ and parent-child dependencies $\mathcal{H}$. In particular, there are K basin-states at the leaves, and for each macro-state s_i^d (except the root s_1^1), there is one exit-state s_e^{d+1} in its children. In order to build a full $\mathcal{H}$HMM MDM $\Theta = (\mathcal{C}, \mathcal{S}, \mathcal{H}, \Pi, B, A, E)$, we need to estimate the remaining parameters, namely, the ***vertical*** transitions $\{\Pi, B\}$, the ***horizontal*** transitions A, and the ***emission*** probabilities E.

We estimate the parameters by combining each tree $\mathcal{G} = \{\mathcal{S}, \mathcal{H}\}$ with the K-state HMM Θ_K. We use HMM Θ_K to infer the expectations $\xi_t(m,n) = p(s_t = m, s_{t+1} = n)$ (Eq. 3.13 on page 61), and their sums $\mathbb{E}(m,n) = \sum_{\mathcal{D}_{train}} \sum_t \xi_t(m,n)$ (Eq. 4.6 on page 111). Based on these expectations, we estimate parameters of an $\mathcal{H}$HMM Θ based on the normalization between *sets* of basin-states. This is possible because both $\mathcal{H}$HMM Θ and HMM Θ_K model the same K number of energy basins, and a branch in the hierarchy of $\mathcal{H}$HMM Θ corresponds to a state in HMM Θ_K. Therefore, by using HMM Θ_K as initialization, we avoided the costly search for *both* the number of energy basins, and their clustering at the same time.

We use $basin(s_i^d)$ to indicate the set of basin-states under the subtree rooted at s_i^d, and $basin(\mathcal{S})$ indicates *all* basin-states in $\mathcal{S}$. Subscripts of a *parent* state s^d will be omitted for clarity. Fig. 4.4 (page 93) will be a useful reference for the rest of this section.

Vertical transitions $\{\Pi, B\}$

A vertical transition can be either upwards, or downwards. An ***upward*** transition $p(s^d|s_e^{d+1}) = 1$ is the transfer of control from an exit-state s_e^{d+1} back to its parent s^d (subscript omitted), and is 1 by definition.

Downward transitions are distinguished according to the time at which they occur. More specifically, the prior probabilities Π occur *before* the conformation q_0 is observed at time $t = 0$. While downward transitions B occur *after* time $t > 0$.

A downward transition corresponds to an entry into one of the underlying energy basins, *before* conformation q_t at time t is observed. Consequently, for each ***parent*** state s^d, a downward transition probability $p(s_i^{d+1}|s^d)$ is the fraction of expected transitions into basin-states under the ***child*** state s_i^{d+1}, over all basin-states under the parent s^d.

First, the prior $\Pi^{s^d} = \{\pi_i^{s^d} \mid i = 1, 2, \ldots, L\}$, where L is the number of children under the parent s^d, *excluding* the exit-state because a conformation has *yet* to be generated. The prior $\pi_i^{s^d} = p(s_i^{d+1}|s^d)$ probability of transiting from the parent s^d to basin-states under its child s_i^{d+1} is:

$$\begin{aligned}
\pi_i^{s^d} &= p(s_i^{d+1}|s^d) \\
&= p\left(basin(s_i^{d+1}), t = 0\right) \\
&= \frac{\sum_{m \in basin(s_i^{d+1})} \left[\sum_{\mathcal{D}_i \in \mathcal{D}_{train}} \sum_n^K \xi_0(m, n)\right]}{\sum_{m \in basin(s^d)} \left[\sum_{\mathcal{D}_i \in \mathcal{D}_{train}} \sum_n^K \xi_0(m, n)\right]} \\
&= \frac{\sum_{m \in basin(s_i^{d+1})} \pi'_m}{\sum_{m \in basin(s^d)} \pi'_m},
\end{aligned} \tag{4.9}$$

where the normalization $\sum_{m \in basin(s^d)}$ is over all basin-states in the subtree rooted at the parent s^d. In addition, π'_m is the prior probability to be in state m in HMM Θ_K, and the last line is the result of multiplication by $\frac{N}{N}$, where N is the number of trajectories in $\mathcal{D}_{train}$, see Eq. 3.7 on page 57.

Next, the downward transitions for time $t > 0$, $B^{s^d} = \{b_i^{s^d} \mid i = 1, 2, \ldots, L\}$, where L is the number of children under the parent s^d, and also *excludes* the exit-state. The downward transition $b_i^{s^d} = p(s_i^{d+1}|s^d)$ is the probability of transiting from the parent s^d to basin-states under its child s_i^{d+1}:

$$\begin{aligned}
b_i^{s^d} &= p(s_i^{d+1}|s^d) \\
&= p\left(basin(s_i^{d+1}) \mid basin(\mathcal{S}) - basin(s^d)\right) \\
&= \frac{\sum_{m \in \{basin(\mathcal{S}) - basin(s^d)\}} \sum_{n \in basin(s_i^{d+1})} \mathbb{E}(m, n)}{\sum_{m \in \{basin(\mathcal{S}) - basin(s^d)\}} \sum_{n \in basin(s^d)} \mathbb{E}(m, n)},
\end{aligned} \tag{4.10}$$

which is the expected transitions from outside the subtree rooted at the parent ($m \in \{basin(\mathcal{S}) - basin(s^d)\}$), into basin-states under its child ($n \in basin(s^{d+1})$), normalized over all basin-states under the parent ($n \in basin(s^d)$).

Horizontal transitions A

Horizontal transitions A are more complicated because it can either be a direct transition between energy basins within a cluster, or lead to further transitions across the hierarchy. We need to consider 4 cases, differentiated by the type of state (macro or basin), and the presence or absence of an exit-state.

The ***type of state*** determines the presence of self-transitions. This is an important distinction because the dynamics within a cluster of energy basins involves *rapid* transitions between states inside the cluster. If we model these transitions *collectively* as a repeating self-transition at the cluster level, we lose the ability to distinguish the change of conformation at the fast $\triangle t$ timescale. Consequently, we model the dynamics within a cluster as direct transitions at the leaves of the hierarchy, and self-transitions are only allowed for basin-states (which can only be at the leaves).

Exits correspond to the termination of a sequence of transitions. Since we do not wish to predict the *length* of trajectories, there is *no* exit-state at the top of the hierarchy to terminate the global sequence, *i.e.* $d \leq 2$. However, for $d > 2$, an exit-state exists among children of the same parent.

Horizontal transitions are also estimated based on *sets* of basin-states. For each parent state s^d, the horizontal transitions among its L ***children*** is $A^{s^d} = \{a_{ij}^{s^d} \mid i, j = 1, 2, \ldots, L\}$, where $a_{ij}^{s^d} = p(s_j^{d+1}|s_i^{d+1})$ is the probability of transiting from child s_i^{d+1} to child s_j^{d+1}. Consequently, for each ***source*** child s_i^{d+1}, a horizontal transition probability $p(s_j^{d+1}|s_i^{d+1})$ is the fraction of expected transitions into basin-states under the ***destination*** child s_j^{d+1}, over basin-states under *all* possible destinations.

Case 1: Macro-state *without* exit. We first consider a macro-state s_i^2 at $d = 2$, directly under the parent root s^1. Since there is *no* exit-state among its siblings, there is only one kind of transition, which is a transition to a *different* sibling, *i.e.* $p(s_j^2|s_i^2)$ where $i \neq j$:

$$\begin{aligned} a_{ij}^{s^1} &= p(s_j^2|s_i^2) \\ &= p\left(basin(s_j^2) \mid basin(s_i^2)\right) \\ &= \frac{\sum_{m \in basin(s_i^2)} \sum_{n \in basin(s_j^2)} \mathbb{E}(m,n)}{\sum_{m \in basin(s_i^2)} \sum_{n \in \{basin(\mathcal{S}) - basin(s_i^2)\}} \mathbb{E}(m,n)}, \end{aligned} \tag{4.11}$$

which is the expected transitions from within the source state ($m \in basin(s_i^2)$), into basins within the destination state ($n \in basin(s_j^2)$). Since all children of the root s^1 are siblings of s_i^2, and s_i^2 has *no* self-transition, the only *inaccessible* basins are those within s_i^2, *i.e.* $basin(s_i^2)$. Therefore, the normalization is over all basins *outside* the source child ($n \in \{basin(\mathcal{S}) - basin(s_i^2)\}$).

Case 2: Macro-state *with* exit. When a macro-state is lower down in the hierarchy, we need to consider the case of a transition to an exit-state. More specifically, for a parent state s^d at $d > 1$, the horizontal transitions for its child macro-state s_i^{d+1} is:

$$\begin{aligned} a_{ij}^{s^d} &= p(s_j^{d+1}|s_i^{d+1}) \\ &= p\left(basin(s_j^{d+1}) \mid basin(s_i^{d+1})\right) \\ &= \frac{\sum_{m \in basin(s_i^{d+1})} \sum_{n \in basin(s_j^{d+1})} \mathbb{E}(m,n)}{\sum_{m \in basin(s_i^{d+1})} \sum_{n \in \{basin(\mathcal{S}) - basin(s_i^{d+1})\}} \mathbb{E}(m,n)}, \end{aligned} \tag{4.12a}$$

$$\begin{aligned} a_{ie}^{s^d} &= p(s_e^{d+1}|s_i^{d+1}) \\ &= p\left(basin(\mathcal{S}) - basin(s^d) \mid basin(s_i^{d+1})\right) \\ &= \frac{\sum_{m \in basin(s_i^{d+1})} \sum_{n \in \{basin(\mathcal{S}) - basin(s^d)\}} \mathbb{E}(m,n)}{\sum_{m \in basin(s_i^{d+1})} \sum_{n \in \{basin(\mathcal{S}) - basin(s_i^{d+1})\}} \mathbb{E}(m,n)}, \end{aligned} \tag{4.12b}$$

where $i \neq j$ since there is *no* self-transition, and s_e^{d+1} is the exit-state whose parent is s^d. Eq. 4.12a is similar to **Case 1**. The difference is in the transition to the exit-state s_e^{d+1} in Eq. 4.12b. A transition to an exit-state terminates transitions within the subtree rooted at the parent s^d, and lead to the transfer of control back to s^d with probability $p(s^d|s_e^{d+1}) = 1$ by definition. This will then be followed by further transitions to other parts of the hierarchy (see Fig. 4.4 on page 93). Consequently, a transition to an exit-state is equivalent to a transition to *any* basin-state *outside* the subtree rooted at the parent ($n \in \{basin(\mathcal{S}) - basin(s^d)\}$).

Case 3: Basin-state *without* exit. Basin-states can be at level $d = 2$, directly under the parent root s^1. For example, the K-state HMM MDM Θ_K is equivalent to a *flat* $\boldsymbol{\mathcal{H}}$HMM MDM with K basin-states at $d = 2$, a "*dummy*" root, and *no* exit-states. The difference from **Case 1** is that *basin-states* are *allowed* self-transitions, and *all* subtrees are now *accessible*. Therefore, the normalization is over *all* basin-states in $\mathcal{S}$.

More specifically, for a basin-state s_i^2 under the parent root s^1, the horizontal transition is:

$$\begin{aligned} a_{ij}^{s^1} &= p(s_j^2 | s_i^2) \\ &= p\left(basin(s_j^2) \mid basin(s_i^2)\right) \\ &= \frac{\sum_{m \in basin(s_i^2)} \sum_{n \in basin(s_j^2)} \mathbb{E}(m,n)}{\sum_{m \in basin(s_i^2)} \sum_{n \in basin(\mathcal{S})} \mathbb{E}(m,n)}, \end{aligned} \tag{4.13}$$

which is the expected transitions from within the source state ($m \in basin(s_i^2)$), into basins within the destination state ($n \in basin(s_j^2)$), normalized over transitions to *all* basin-states ($n \in basin(\mathcal{S})$). Self-transitions $a_{ii}^{s^1}$, for $i = j$, are allowed.

Case 4: Basin-state *with* exit. For basin-states lower down in the hierarchy, the difference from **Case 2** is that *basin-states* are *allowed* self-transitions, and *all* subtrees are now *accessible*. Therefore, the normalization is over *all* basin-states in $\mathcal{S}$.

For a basin-state s_i^{d+1} under the parent s^d, the horizontal transitions:

$$\begin{aligned} a_{ij}^{s^d} &= p(s_j^{d+1}|s_i^{d+1}) \\ &= p\left(basin(s_j^{d+1}) \mid basin(s_i^{d+1})\right) \\ &= \frac{\sum_{m\in basin(s_i^{d+1})}\sum_{n\in basin(s_j^{d+1})}\mathbb{E}(m,n)}{\sum_{m\in basin(s_i^{d+1})}\sum_{n\in basin(\mathcal{S})}\mathbb{E}(m,n)}, \end{aligned} \tag{4.14}$$

$$\begin{aligned} a_{ie}^{s^d} &= p(s_e^{d+1}|s_i^{d+1}) \\ &= p\left(basin(\mathcal{S}) - basin(s^d) \mid basin(s_i^{d+1})\right) \\ &= \frac{\sum_{m\in basin(s_i^{d+1})}\sum_{n\in\{basin(\mathcal{S})-basin(s^d)\}}\mathbb{E}(m,n)}{\sum_{m\in basin(s_i^{d+1})}\sum_{n\in basin(\mathcal{S})}\mathbb{E}(m,n)}, \end{aligned} \tag{4.15}$$

where self-transitions $a_{ii}^{s^d}$, for $i = j$, are allowed, and s_e^{d+1} is the exit-state whose parent is s^d. The transition to the exit-state s_e^{d+1} is equivalent to a transition to *any* basin-state *outside* the subtree rooted at the parent ($n \in \{basin(\mathcal{S}) - basin(s^d)\}$).

Emission probabilities E

Since each basin-state corresponds to an energy basin, the emission probabilities of observing conformations for the basin-states are similar to those of HMM MDMs in Chapter 3. Therefore, during the first initialization of an $\boldsymbol{\mathcal{H}}$HMM MDM, the emission probabilities from the K-state HMM Θ_K can be directly used by mapping the energy basins. Subsequent optimizations rely on the same estimation equations as Eq. 3.17 and Eq. 3.18 on page 62, which are reproduced here for convenience:

$$\mu_i = \arg\min_q \sum_{\mathcal{D}_i \in \mathcal{D}_{train}} \sum_{t=0}^{T-1} \gamma_t(i)\ d(q, q_t),$$
$$\sigma_i^2 = \frac{\sum_{\mathcal{D}_i \in \mathcal{D}_{train}} \sum_{t=0}^{T-1} \gamma_t(i)\ d^2(\mu_i, q_t)}{\sum_{\mathcal{D}_i \in \mathcal{D}_{train}} \sum_{t=0}^{T-1} \gamma_t(i)},$$

where the emission probability $e_i^d(q) = \mathcal{N}(q|\mu_i, \sigma_i^2)$ is a Gaussian distribution of conformations $q \in \mathcal{C}$ for basin-state s_i^d, and $\gamma_t(i)$ is the probability of being in state s_i^d at time t (Eq. 3.14 on page 61).

In summary, by first constructing the K-state HMM MDM Θ_K, we avoided the costly search for *both* the number of energy basins, and their clustering at the *same time*. More specifically, HMM Θ_K allows us to calculate inference on MD trajectories *once*, and initialize *multiple* candidate $\boldsymbol{\mathcal{H}}$HMM MDMs with different hierarchies of K basin-states. Although $\boldsymbol{\mathcal{H}}$HMM transitions involve additional internal changes, they correspond to transitions among a set of *smaller* transition matrices, and the use of *fewer* parameters. In addition, when there is a lack of long trajectories, the collective transitions between clusters of energy basins can potentially provide a more reliable estimate of longer timescale dynamics.

4.3.4 Optimizing $\mathcal{H}$HMM parameters

The $\mathcal{H}$HMM Θ we have constructed from the K-state HMM Θ_K and the hierarchy $\mathcal{H}$ can be further optimized. The reason is because the hierarchy $\mathcal{H}$ enforces additional constraints on the $\mathcal{H}$HMM Θ. Therefore, even if the basin-states are exact, the averaging nature of cluster-to-cluster dynamics can cause transitions in $\mathcal{H}$HMM Θ to be different from the original HMM Θ_K. Consequently, $\mathcal{H}$HMM Θ has yet to be *directly* optimized.

There are a few ways to further optimize the initial $\mathcal{H}$HMM Θ. The $\mathcal{H}$HMM framework was originally proposed by Fine *et al.* in [42]. The inference algorithm and parameter estimations run in $O(M^D T^3)$ time, where M is the maximum number of states at each level in the hierarchy, and D is the depth of the hierarchy. This runtime is undesirable because the temporal length T of MD trajectories is orders of magnitude larger than the M number of states, and D tends to be small.

However, since HMMs is the basis of our construction, the most direct approach for optimization is to resolve hierarchical dependencies in Eq. 4.1 (page 97), and make use of the forward-backward algorithm for HMM inference in $O(K^2 T)$ time, where K is the number of basin-states. With the expectations, the $\mathcal{H}$HMM parameters can be re-estimated according to Section 4.3.3.

More specifically, since the $\mathcal{H}$HMM is the more constrained model, when we use Eq. 4.1 (page 97) to convert an $\mathcal{H}$HMM to an HMM, the resulting basin-to-basin transitions between the two MDMs are exact. Consequently, if we also use the exact emission probabilities in the HMM, the inferred expectations obtained by the two MDMs will be the same. With these expectations, we can *re*-estimate the $\mathcal{H}$HMM parameters according to Section 4.3.3.

Although we can also re-estimate the HMM, it is *not* the model we are trying to optimize. In addition, due to the lack of hierarchy, the re-estimated HMM may be different from the re-estimated $\boldsymbol{\mathcal{H}}$HMM.

More importantly, what we gain through this transformation and re-estimation is speed. Instead of resolving the $\boldsymbol{\mathcal{H}}$HMM's hierarchical dependencies at every time step of every trajectory during inference, only a *single* conversion costing $O(K^2D)$ is needed for *each iteration* of the EM algorithm. This is because for each of the K basin-states in the $\boldsymbol{\mathcal{H}}$HMM, it is necessary to propagate the dynamics up and down the hierarchy of depth D, where at most K states is present at each level. However, since each macro-state can have many children, the depth D of the hierarchy with K leaf basin-states should be relatively small.

What we compromise is the use of more parameters in an HMM. In some other applications, converting a hierarchy to an HMM may be unsuitable. For example, in natural language processing, substructures for phonemes may be shared in the hierarchy, and conversion will require the duplication of shared substructures with a tremendous increase in number of parameters. However, since there are no shared substructures in our hierarchy, and we began the construction with the search for the most suitable K-state HMM Θ_K, space considerations do not constrain us.

There are also alternative ways to optimize an $\boldsymbol{\mathcal{H}}$HMM. Murphy *et al.* has proposed an inference algorithm that also runs in linear time [74, 76]. They showed that $\boldsymbol{\mathcal{H}}$HMMs are a type of dynamic Bayesian network (DBN), and inference can be done in $O(M^{2D}T)$ time by applying the junction tree algorithm. Approximate DBN inference using factored frontier algorithm can also be applied in $O(M^DT)$ time [75].

4.3.5 Determining the most suitable $\mathcal{H}$HMM $\Theta_{\mathcal{H}}$

Our motivation of building an $\mathcal{H}$HMM MDM is to better characterize dependencies in a protein's motion dynamics, in order to identify major conformational changes beyond the $\triangle t$ timescale the model was trained in. However, our recursive construction process leads us to a number of possible hierarchical models, each enforcing different assumptions on transitions based on different clustering of energy basins. Although we are interested in the model with the most detailed separation of dynamics, we are also concerned about the impact of the hierarchy on the accuracy in predicting a protein's motion. Therefore, we compare the likelihood of candidate models on another dataset of MD trajectories $\mathcal{D}_{test2}$ (Eq. 4.3 on page 100).

In addition, we also wish to compare the likelihood of candidate models with the K-state HMM Θ_K. The reason is because the K-state Θ_K is the model from which all the hierarchical models have been initialized with. Therefore, all candidate hierarchies have the same K *number* of basin-states. Furthermore, the K-state Θ_K corresponds to an $\mathcal{H}$HMM with the simplest "*flat*" hierarchy, where all basin-states are children of a "*dummy*" root. Also, Θ_K has the *largest* (K^2), and least constrained transition matrix. Therefore, comparison with Θ_K will give us an indication of the impact of hierarchical dependencies on the accuracy in predicting MD trajectories.

As we compare the likelihood on $\mathcal{D}_{test2}$, we expect these scenarios:

a) ***Comparable*** to HMM Θ_K.

b) ***Better*** than HMM Θ_K.

c) ***Worse*** than HMM Θ_K.

a) ***Comparable*** to HMM Θ_K. We expect the most suitable $\boldsymbol{\mathcal{H}}$HMM $\Theta_{\mathcal{H}}$ to predict MD trajectories as well as Θ_K. This is because for hierarchical dependencies to exist, there must be fast equilibration within the underlying clusters of energy basins that allows transitions between clusters to be accurately estimated *collectively* in an $\boldsymbol{\mathcal{H}}$HMM MDM. However, assuming sufficient data, transitions can also be individually estimated. Consequently, the most suitable $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$ should score as well as Θ_K.

Despite comparable likelihood, we favor an $\boldsymbol{\mathcal{H}}$HMM MDM because the hierarchy offers potentially interesting biological insight, and is also easier to comprehend than Θ_K. In addition, an $\boldsymbol{\mathcal{H}}$HMM MDM with K basin-states will likely have achieved the same accuracy with fewer parameters than Θ_K.

b) ***Better*** than HMM Θ_K. Although a better likelihood indicates better accuracy in predicting dynamics, a noticeable improvement is *undesirable*. A better likelihood indicates that transitions between energy basins in *different* clusters are *better* estimated *collectively* via *fewer* parameters in the hierarchy. This is a potential problem because it means the *larger* K^2 transition matrix of Θ_K could have been further optimized if *more* data is available. As a result, this raises doubts as to whether the reference HMM MDM Θ_K we relied upon to initialize hierarchical models is actually the *most* suitable HMM MDM.

c) ***Worse*** than HMM Θ_K. A worse likelihood is certainly bad. This indicates hierarchical dependencies violated the fast internal equilibration property and erroneously grouped energy basins with different dynamics together. As a result, with *insufficient* equilibration within the enforced clusters, the collective transitions between clusters fail to accurately predict MD trajectories. Since our construction began with the simplest hierarchy, this case indicates an over-complex hierarchy with too much dependencies.

4.4 Results

We demonstrate the applicability of our approach by applying the full modeling process in two experiments. In Section 4.4.1, we first make use of a synthetic energy landscape to illustrate scenarios where our modeling approach is beneficial for understanding motion dynamics. This is a crucial demonstration because the long-timescale MD simulation of proteins with complex motions is still difficult to obtain today. Additionally, a large number of independent MD trajectories is required to capture the stochastic nature of molecular motion. More importantly, the synthetic landscape allows us to test our approach and construct interesting hierarchies where the ground truth is known.

In Section 4.4.2, we apply our modeling approach on the 35 amino acids villin headpiece protein (HP-35 NleNle). The villin headpiece trajectories was simulated by the Folding@home project, and is one of the largest MD datasets publicly available [14, 40]. The villin headpiece was a major motivation behind our development of the hierarchical model, and our search for hierarchical dependencies has beneficially allowed us to gain a better understanding of its motion dynamics. Our results on villin headpiece demonstrate the applicability of our approach on a practical scale.

We begin the modeling by first searching for the most suitable K-state HMM MDM Θ_K. This is a crucial phase of construction because Θ_K identifies the energy basins and predicts MD trajectories at the resolution of the fastest $\triangle t$ timescale of interest. With Θ_K, we can then efficiently search for dependencies at longer timescales by constructing $\boldsymbol{\mathcal{H}}$HMM MDMs with different hierarchies of K energy basins. Finally, the most suitable $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$ will be chosen for analysis based on its accuracy in predicting trajectories, with respect to model complexity.

4.4.1 Synthetic energy landscape

We created a 2-D synthetic energy landscape with 11 energy basins by parameterizing the potential function in Eq. 3.20 (page 67). Each dimension corresponds to a degree of freedom in an artificial molecule, and the XY-space corresponds to its conformation space $\mathcal{C}$. Langevin dynamics is used to generate 1000 trajectories of 1000 time steps each [44, 66]. An equal number of trajectories were initiated from each energy basin, and the distance measure used is the Euclidean distance in the plane.

The interesting aspect about this landscape is *how* the energy basins are connected, see Fig. 4.5. In particular, there are 4 main clusters:

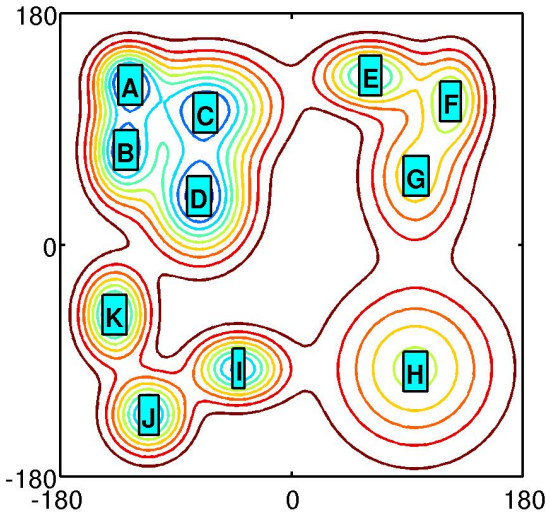
- **Top Left: A, B, C, D.** This cluster is the most interesting because the energy basins are differently connected to each other. Although basin A is the *deepest* in the landscape, it is *not* the most significant state under stationary distribution. This is due to the steep energy barriers surrounding A and B, which favor transitions towards C. Consequently, trajectories eventually equilibrate among C and D, which are akin to the native conformation of a protein.

- **Top Right: E, F, G.** This cluster has a cascade of energy basins, with G at the shallow end, F is slightly deeper, and E is the deepest. Connectivity between the basins means that the transition "*downhill*" from G→F→E is relatively easy compared to the reverse direction. As such, entry into any basin here will eventually lead to E.

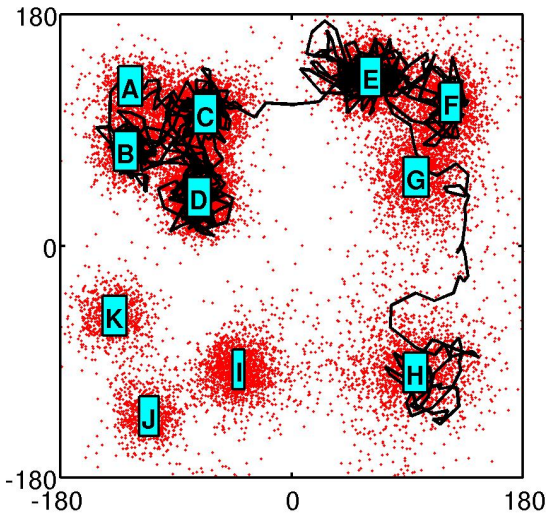
- **Bottom Right: H** is a broad and shallow energy basin. This corresponds to denatured conformations where the protein is relatively stretched out. Although motion is relatively unhindered in this region, conformations are unstable and transient in the long term.

- **Bottom Left: I, J, K.** Each energy basin is narrow and deep. The relatively high energy barrier between the basins results in a clear separation between them. Consequently, transitions between the basins here occur at longer timescales compared to those in clusters {A, B, C, D} and {E, F, G}. These correspond to stable conformations that a protein may temporarily get trapped in.

The combination of 11 energy basins with individual characteristics means that trajectories transit across the landscape at different timescales. For example, a trajectory traversing from basin H to basin D is a lot easier and faster than a trajectory going in the opposite direction. This is because although basins A, B, C and D are easy to escape *individually*, but *collectively*, they are the most significant region of attraction in the entire landscape. Therefore, the difficulty of traversing through the landscape is not only due to the difference in the depth or width of individual energy basins, but is also the result of the *connectivity* between basins. More importantly, it is the *indirect* relationship between energy basins that will determine the motion dynamics. Consequently, *without* actually seeing the energy landscape, it is very difficult to figure out the reasons that cause different trajectories to traverse distinctly over the landscape.



a) Energy landscape



b) Data

Figure 4.5: A synthetic landscape with 11 energy basins in the XY-plane. The four main clusters are {A, B, C, D}, {E, F, G}, {H}, and {I, J, K}. The black line in *b*) traces a sample trajectory that started in basin H.

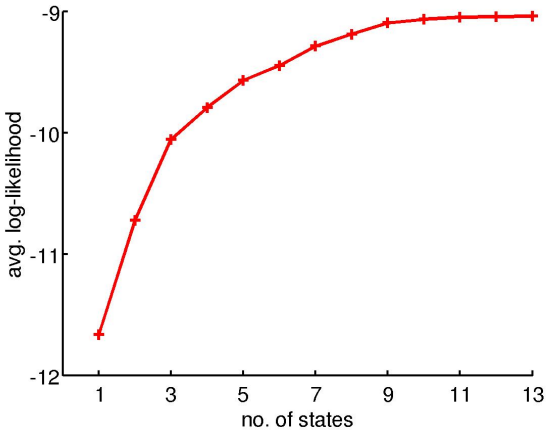
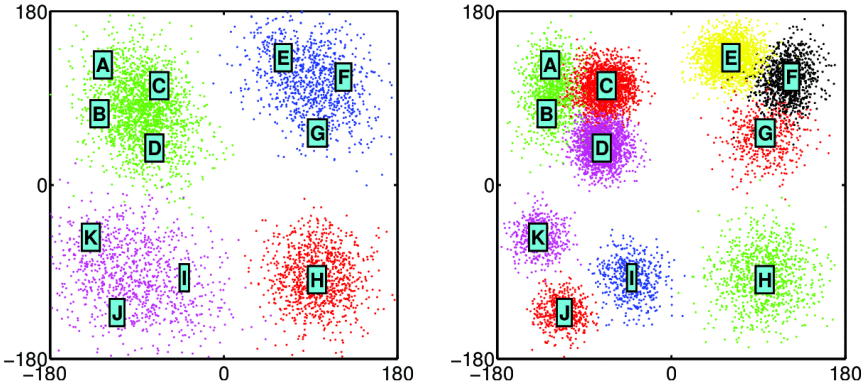


Figure 4.6: Average log-likelihood scores of HMM MDMs on the first test dataset $\mathcal{D}_{test1}$. We choose $K = 11$ as the most suitable HMM MDM Θ_K.

Search for the most suitable number of states

In order to model the dynamics, we first search for the most suitable K-state HMM MDM Θ_K. We follow procedures outlined in Chapter 3 to construct HMMs with different number of states. 50% of trajectories are used as the training dataset $\mathcal{D}_{train}$, at $\triangle t = 10$ simulation time steps. Another 25% of trajectories are used as the first test dataset $\mathcal{D}_{test1}$ to identify the most suitable K-state HMM MDM Θ_K. We save the remaining 25% of trajectories as the *second* test dataset $\mathcal{D}_{test2}$ for the hierarchy.

Fig. 4.6 shows the likelihood scores of different HMM MDMs on $\mathcal{D}_{test1}$. The plateauing of likelihood as K increases from 1 to $K = 11$ is just as we expected. With each increase in the number of states, the energy basins become increasingly better characterized, and the MDM is able to predict MD trajectories with higher accuracy.



a) 4-state HMM. *b*) 10-state HMM.

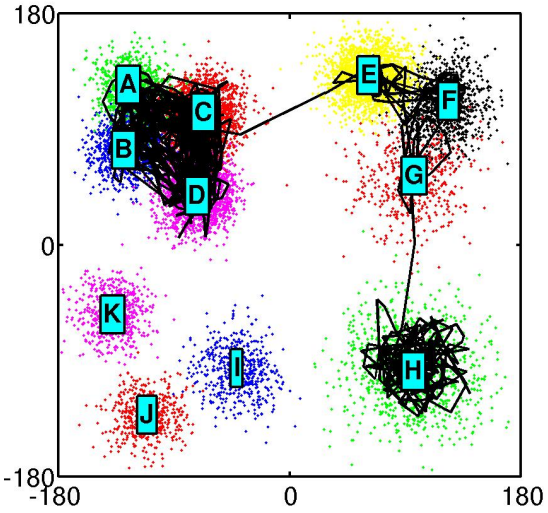
c) 11-state HMM. *d*) 12-state HMM.

Figure 4.7: Distribution of conformations generated by simulating HMM MDMs. *a*) When the number of states is insufficient, only the *major* clusters can be characterized. *b*) The 10-state HMM characterizes energy basins A and B together as one, while the rest of the energy basins are accurately identified. *c*) The 11-state HMM is the most suitable HMM. *d*) When the number of states exceeds the actual number of energy basins, the states start to overlap on top of each other. For the 12-state HMM, two states overlap the energy basin H.

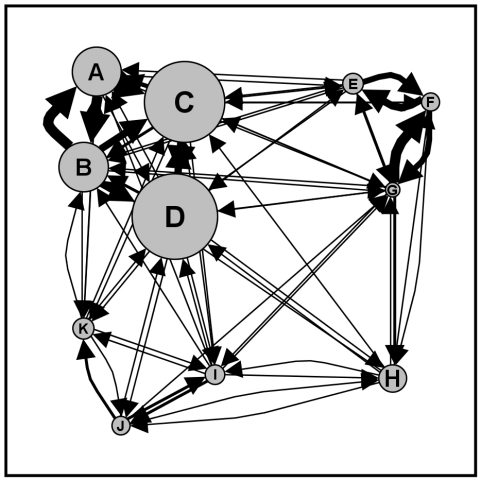
Fig. 4.7 illustrates the distribution of conformations generated by HMM MDMs with 4, 10, 11 and 12 states. When the number of states is less than the actual number of energy basins, only the *major* clusters can be characterized, *i.e.* 4-state HMM. This is due to the use of K-medoids clustering for initialization, which favors the *initial* identification of regions with a high-density of conformations. Only when the number of states increases sufficiently, can energy basins be individually characterized. This is beneficial because energy basins are not necessary distinct, and can be difficult to distinguish.

More specifically, in the 10-state HMM, we can see that energy basins A and B are modeled as a single state. Although these two energy basins can be better distinguished, the distribution of conformations predicted by the 10-state HMM is still roughly accurate. Therefore, the dynamics modeled by the 10-state HMM is mostly accurate, and the improvement in likelihood score from 10 to 11 states is slight.

When the number of states increases further to 12, we see an overlapping of two states over the energy basin H. This is similar to the over-complexity scenario we encountered earlier in alanine dipeptide (Section 3.4.2 on page 71), and is actually difficult to recognize *without* using the likelihood score. Just as before, there is a compensatory trade-off between an increase in spatial resolution and a decrease in the accuracy of predicted transitions. The result is no further improvement in likelihood score for HMM MDMs with *more* than 11 states. Consequently, it is reasonable to choose the 11-state HMM as the most suitable MDM Θ_K.



a) Distribution of conformations.



b) Transitions between states.

Figure 4.8: 11-state HMM MDM Θ_K of the 11-basin synthetic landscape. In *a*), the black line traces a generated trajectory that started in basin H. In *b*), the size of a node corresponds to the stationary distribution, the weight of an edge corresponds to the probability of transition. Self-transition probabilities are not shown.

Fig. 4.8 illustrates the parameters of the 11-state HMM MDM Θ_K. In Fig. 4.8*a*, we can see that Θ_K accurately identifies the locations of the 11 energy basins. The dimensions of energy basins are also well characterized by the variance σ^2 of the emission probabilities. In terms of dynamics, from the node sizes in Fig. 4.8*b*, we can see the dominance of energy basins C and D under the stationary distribution. This is despite the fact that A and B are actually the *deeper* energy basins. This also illustrates the crucial difference between an analysis based on the energy landscape, versus analysis based on predicting dynamics. More specifically, the dynamics of a protein's motion is also influenced by its momentum, and can be accurately predicted if *simulated*. Consequently, even if the high-dimensional energy surface of real proteins are explicitly available, an analysis *without* simulation is not absolutely reliable.

Also in Fig. 4.8*b*, although we can identify the more important transitions from the thickness of edges, there are also many thinner edges which can be rather confusing. More importantly, so far our analysis is limited to the two extreme timescales, the $\triangle t$ timescale of individual transitions, and the stationary distribution. The interesting dynamics during the intermediate time frame is rather difficult to obtain from Fig. 4.8.

For example, although it is tempting to infer that the sequence of transitions from H to {A, B, C, D} is via the intermediate clusters {E, F, G} and {I, J, K}, we want to point out that the 2-D *embedding* plays a crucial role in this interpretation. *Without* an accurate embedding and prior knowledge of the energy landscape, it can be rather difficult to distinguish the presence of *two intermediate clusters* from the network of K^2 connections. Additionally, understanding the intermediate dynamics requires the analysis of the *ensemble* of pathways between states across

multiple transitions. Due to the difficulty of escaping from basins, small differences in transition probabilities can further complicate analysis.

The difficulty in gaining an understanding arises because it is necessary to analyze a particular change at the timescale at which it occurs. For example, the high probability transitions *within* {E, F, G} are easily taken at each $\triangle t$ time step (thick edges in Fig. 4.8*b*). However, over a longer timescale, trajectories from {E, F, G} are more likely to have transited to {A, B, C, D}. This information is hardly apparent in the edges representing $\triangle t$ time transitions. However, by checking the size of states in Fig. 4.8*b*, we can clearly identify that transiting to {A, B, C, D} is much more likely than any other alternatives. This is because the size of states corresponds to the stationary distribution, which is the suitable timescale for analyzing the *longer* timescale dynamics. The problem is, "What about the *intermediate* time frame?"

Although it is possible to construct different models at different timescales, we do not necessarily know all the interesting timescales of a protein's motion. Correlating between the different models will also be difficult. Therefore, we need to better characterize the dynamics, and search for the most suitable $\boldsymbol{\mathcal{H}}$HMM MDM.

Search for the most suitable $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$

We proceed by constructing $\boldsymbol{\mathcal{H}}$HMM MDMs with different hierarchies of K energy basins. We make use of the 11-state HMM Θ_K and start with a simple hierarchy with 11 basin-states and 1 root. At each iteration, we grow the hierarchy downwards by clustering sibling basin-states into newly created subtrees deeper in hierarchy. We combine the hierarchy from each iteration with Θ_K to construct a candidate $\boldsymbol{\mathcal{H}}$HMM.

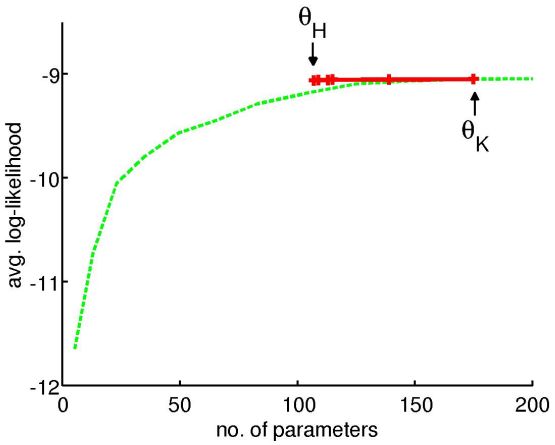


Figure 4.9: Average log-likelihood scores of $\mathcal{H}$HMM MDMs with different hierarchies on the second test dataset $\mathcal{D}_{test2}$ (red solid line). As we construct the hierarchy with each iteration, we create $\mathcal{H}$HMM MDMs with an *increasing* hierarchy, but a *decreasing* number of parameters. Since all the candidate models have similar score as compared to Θ_K, we choose the model with the least number of parameters as the most suitable $\mathcal{H}$HMM MDM $\Theta_{\mathcal{H}}$. The scores of HMM MDMs with different number of parameters are also shown for reference (green dashed line). Due to the relatively few 11 energy basins, the better scores of $\mathcal{H}$HMM MDMs is less visible here, as compared to results on villin headpiece in Fig. 4.17.

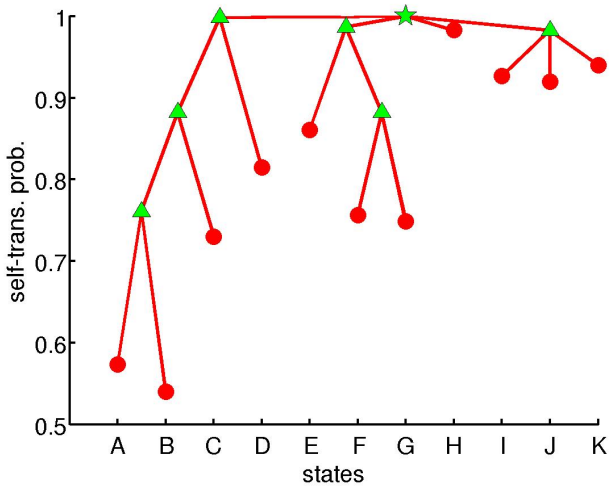
Trajectories from $\mathcal{D}_{train}$ and $\mathcal{D}_{test1}$ are used to optimize parameters of the candidate $\mathcal{H}$HMMs. Trajectories from the remaining 25% of trajectories are used as the second test dataset $\mathcal{D}_{test2}$ to identify the most suitable $\mathcal{H}$HMM MDM $\Theta_{\mathcal{H}}$.

Fig. 4.9 shows the likelihood scores of $\mathcal{H}$HMMs with different hierarchies of 11 energy basins. Due to the constraints imposed on parents having longer escape time than their children, all candidate $\mathcal{H}$HMMs have clusters with fast internal transitions. This allows them to accurately predict trajectories despite the collective estimation of transitions across clusters. Consequently, the candidate $\mathcal{H}$HMMs score similarly well as compared to the 11-state HMM Θ_K. An unsuitable hierarchy will fail to predict MD trajectories and lead to a deterioration in the likelihood scores with respect to Θ_K.

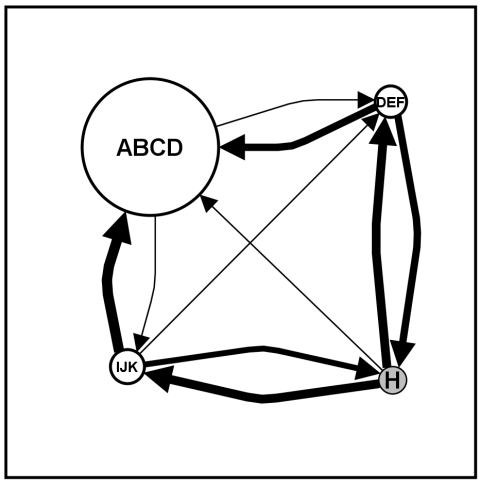
More importantly, although each $\boldsymbol{\mathcal{H}}$HMM models 11 energy basins with 11 basin-states, due to the different hierarchies, the number of *parameters* is different. In addition, the $\boldsymbol{\mathcal{H}}$HMMs score *better* than HMMs with the same number of parameters. This demonstrates the benefit of combining the estimation of transitions *between* clusters of energy basins across the hierarchy. Since the likelihood scores among candidate $\boldsymbol{\mathcal{H}}$HMMs are similar, we choose the model with the least number of parameters as the most suitable $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$.

Fig. 4.10 shows the most suitable $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$ of 11 energy basins. The interesting part of $\Theta_{\mathcal{H}}$ is the clustering of states previously unavailable. At the top of Fig. 4.10*a* are 4 subtrees corresponding to clusters {A, B, C, D}, {E, F, G}, {H} and {I, J, K}. This corresponds to our understanding of the energy landscape in Fig. 4.5 on page 131. More importantly, Fig. 4.10*b* shows that in the long-timescale near the top of the hierarchy, transitions *out* of {A, B, C, D} are significantly more difficult than transitions *into* it. Consequently, {A, B, C, D} are the major states under stationary distribution. In addition, to change from H to {A, B, C, D}, there are *two* dominant pathways via intermediate clusters {E, F, G} and {I, J, K}.

Contrast this with Fig. 4.8*b* on page 135, which shows major transitions at the fast $\triangle t$ timescale are those *within* {A, B, C, D}. This is *not* wrong, it is just that the change we are interested in, from H to {A, B, C, D}, hardly occurs at the fast $\triangle t$ timescale. If pruning is applied to remove smaller edges or nodes in Fig. 4.8*b*, it will likely result in a wrong understanding of dynamics *without* intermediates, and separate models at different timescales will be required for analysis. As such, $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$ is not only beneficial in clustering the states, it also identifies the crucial timescales to interpret a protein's dynamics.



a) Hierarchical dependencies.



b) Transitions between clusters.

Figure 4.10: Hierarchy and inter-cluster transitions of the most suitable $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$ with 11 basin-states. In *a*), the x-axis is labeled according to the basin-states (red dots). The green star at the top is the root state representing the whole MDM. Directly connected to the root are the 3 macro-states (green triangles) representing the clusters {A, B, C, D}, {E, F, G}, {I, J, K}, and the basin-state H. Clustering can be nested. For example, {A, B, C, D} is in fact nested according to {{{A, B}, C}, D}. Exit-states are not shown to avoid clutter. In *b*), the size of a node corresponds to the stationary distribution, the weight of an edge corresponds to the probability of transition. Self-transitions within each cluster, and H, are not shown. H is a basin-state, and is shaded.

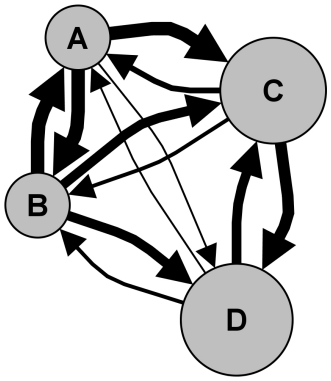
More specifically, this is our interpretation of dynamics in each cluster:

- **A, B, C, D.** With the highest *collective* self-transition probability nearly 1 (Fig. 4.10*a*), this is the *most* difficult to escape cluster. The large separation in escape times between the cluster as a whole and states within it indicates frequent transitions among energy basins. From Fig. 4.11, if a trajectory exits from {A, B}, it is likely to *first* go to C or D, because escaping the whole cluster occurs at a much longer timescale. If a trajectory exits from C, it is likely to go to either {A, B} or D. If transit is to {A, B}, then a subsequent return is rather likely. Only D offers greater stability as an individual energy basin. Interestingly, {A, B, C} collectively offers greater stability than D, and is likely the reason that C has a high stationary probability.

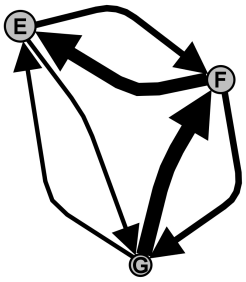
- **E, F, G.** From Fig. 4.11, we can clearly recognize the favored sequence of transitions from G to F, and then to E. This corresponds to our intuition that the cascade of energy basins results in equilibration in E.

- **H** has a surprisingly high self-transition probability (Fig. 4.10*a*). Due to its broadness, although a trajectory can cover large distances across the region, it takes a long time to escape from the energy basin.

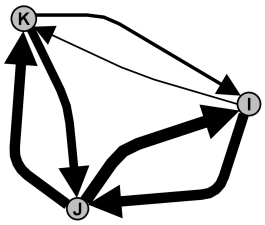
- **I, J, K.** Each basin is relatively difficult to escape, and is especially interesting when compared to {E, F, G} in Fig. 4.10*a*. The separation in escape times between the cluster as a whole and states within is relatively small for {I, J, K}, as compared to {E, F, G}. This suggests there is relatively less transitions *between* states in {I, J, K}, and corresponds intuitively to the isolated nature of each basin. In addition, Fig. 4.11 shows there is *no* favored sequence of transitions through {I, J, K}, as opposed to the cascade in {E, F, G}.



a) Cluster {A, B, C, D}.



b) Cluster {E, F, G}.



c) Cluster {I, J, K}.

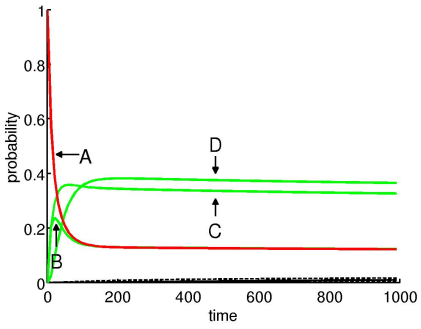
Figure 4.11: Intra-cluster transitions of the most suitable $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$ with 11 basin-states. Each graph shows the internal transitions within each major cluster. The size of a node corresponds to the stationary distribution, the weight of an edge corresponds to the probability of transition. Self-transitions are not shown to avoid clutter.

Equilibration within clusters of energy basins

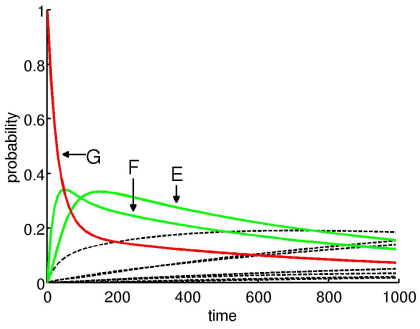
Due to the connectivity within a cluster, rapid transitions between energy basins result in the equilibration of trajectories before they exit. Upon exit, the future of a trajectory is independent of how it entered the cluster in the past, *i.e.* is Markov. Consequently, a good hierarchy corresponds to a suitable clustering such that regardless of the initial state within the same cluster, the system converges to a *similar* probability distribution over the states *quickly*. Conversely, states in different clusters have *distinct* probability distributions *before* the global stationary distribution is achieved.

Since the equilibration within a cluster occurs over *multiple* $\triangle t$ time steps, this *process* can be observed by simulating the $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$. Fig. 4.12 shows the probability distribution over states simulated with $\Theta_{\mathcal{H}}$ across the equivalent 1000 time steps of the trajectories. The interesting thing to note is that states within the same cluster converges to a certain ratio of probabilities quickly, while different clusters exhibit different "*profiles*" of probability distributions.

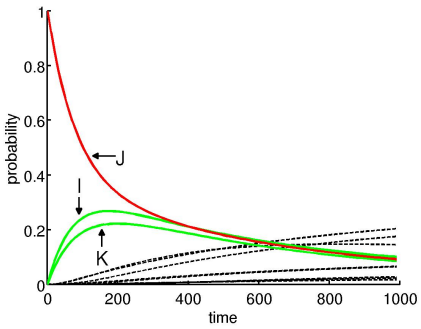
Additionally, the global equilibration to the stationary distribution occurs over a timescale much *longer* than the 1000 simulated time steps. In Fig. 4.12, this is reflected in the *continued* changing probability distributions in {E, F, G}, {H} and {I, J, K}. Whereas the probabilities in {A, B, C, D} has very much stabilized to the stationary distribution. In fact, for all trajectories in our dataset, only 40% of trajectories initiated from H managed to reach the native cluster {A, B, C, D}. Without the combined use of trajectories initiated from different energy basins, significantly more trajectory data will be required for model construction.



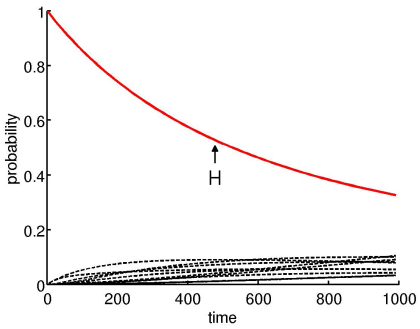
a) Initialized with A in {A, B, C, D}.



b) Initialized with G in {E, F, G}.



c) Initialized with J in {I, J, K}.



d) Initialized with H.

Figure 4.12: Dynamics simulated using the most suitable $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$ with 11 basin-states. In each simulation, one branch in the hierarchy is assigned an initial probability of 1, *i.e.* a particular basin-state. In each plot, the red line indicates the probability of being in the initial basin-state, green lines indicate those basin-states within the same cluster as the initial state, while dashed black lines are basin-states in other clusters. The simulated duration is 1000 time steps, similar to the length of trajectories in the dataset.

Specificity on *false* data

An interesting question that arose is the objectivity of using the likelihood on MD trajectories when choosing the most suitable model. In particular, we are concerned about the specificity of MDMs on *false* datasets.

We *reverse* the direction of acceleration by creating an "*inverted*" landscape with 11 "*hills*". Langevin dynamics is used to generate 1000 trajectories of 1000 time steps each. The resulting trajectories explore regions of the energy landscape previously seldom visited, see Fig. 4.13. We then score HMM MDMs previously constructed from the "*original*" energy landscape in Fig. 4.5, on *false* data from this "*inverted*" landscape.

The likelihood scores in Fig. 4.14 show an interesting trend. When a model is insufficiently complex ($K = 1$), the *difference* in scores on *true* and *false* data is relatively small. This indicates the model has difficulty distinguishing the different scenarios. However, as model complexity increases, the difference in scores widens significantly. Particularly interesting is the slight increase in score on *false* data beyond $K > 9$. This is when the overlapping of emission probabilities E around basins in the "*original*" energy landscape begins to show slight bias towards the "*false*" data. Although we know the true value of $K = 11$, it suggests a slight preference should be given to models with *almost* as good a score on *true* data.

In practice, although forces can be inverted in MD, it is still too expensive to collect additional *false* data. However, if data of different molecules are available, a suitable penalty term can be estimated, *e.g.* Bayesian information criterion [16]. More importantly, this suggests the potential of using MDMs to classify molecules based on motion dynamics. This is useful because a single DNA mutation may seem minor, but can significantly change a protein's folding and cause diseases such as sickle cell anemia [55].

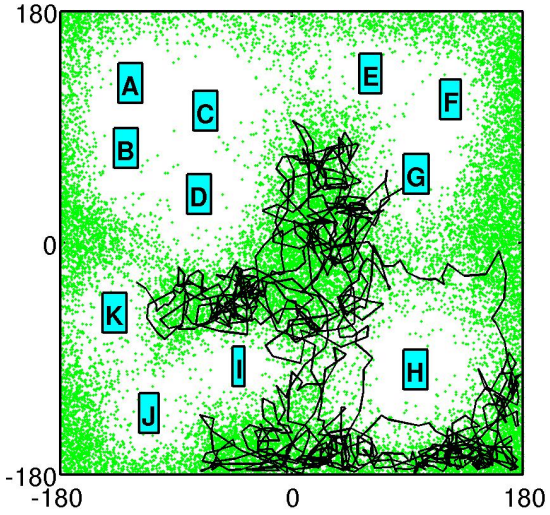


Figure 4.13: *False* dataset from the "*inverted*" landscape with 11 "*hills*". Green dots are the conformations, while the black line traces a sample trajectory.

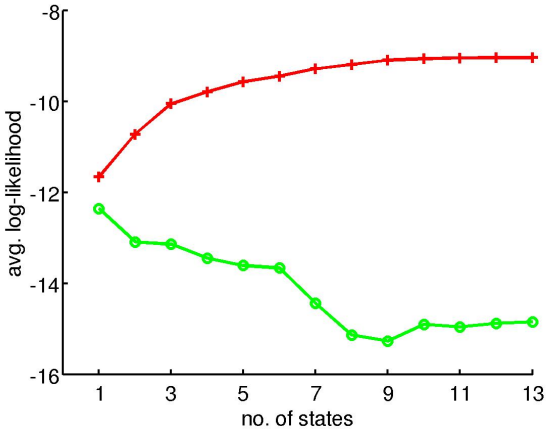


Figure 4.14: Comparison of average log-likelihood scores on the *true* (red) and *false* (green) test datasets. HMM MDMs with $K = 1, 2, \ldots, 13$ states are trained on the *true* dataset from the 11 "*basins*" landscape, and scored on both the *true* and *false* test datasets. As the number of states increases, the MDMs are better able to distinguish between the *true* and *false* datasets, *i.e.* greater separation between the red and green lines.

4.4.2 Villin headpiece

We model the dynamics of villin headpiece (HP-35 NleNle), a protein with 35 amino acids. Due to the large dataset, data preparation is more involved. Similarly, we will first search for the most suitable K-state HMM MDM Θ_K that models villin's dynamics. Then, we will search for the most suitable $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$ and analyze villin's dynamics. In particular, we are interested in understanding villin's motion during the *intermediate* timescale, *before* it has attained the native conformation.

Data preparation

The data for the fast-folding variant of the villin headpiece (HP-35 NleNle) was generated by the Folding@home project [14, 40]. It is one of the largest MD datasets publicly available, consisting of 410 MD trajectories initiated from 9 different unfolded conformations. Each trajectory is 1 µs (10^{-6}s) in duration on the average, with conformations saved every 50 ps (50×10^{-12}s). As such, the dataset contains millions of conformations in high-dimensional space.

For computational efficiency, we cluster the conformations to form *microstates* in the conformation space $\mathcal{C}$. These microstates are ***not*** states in the MDM, and are usually created when the high-dimensional conformation space $\mathcal{C}$ cannot be uniformly sampled. We create the microstates by sampling $10,000$ conformations uniformly along the MD trajectories as the microstate centers. The remaining conformations in the dataset are then clustered to the nearest microstate centers according to the RMSD of all heavy atoms in the protein. Earlier work indicates that we may assume the protein transits directly between microstates that are close according to the RMSD between microstate centers [17, 29].

Furthermore, we construct a distance graph that approximates the dynamics of the protein according to the microstates. The idea is similar to the use of a nonlinear dimension reduction to preserve the relationship between the microstates [84]. Each node of this graph is a microstate and is connected to a small number of other nodes close by in RMSD. An edge of the graph is assigned a weight equal to the RMSD between the end nodes. The distance between two microstates is defined as the length of the shortest path between them in the graph. For large proteins, due to the sparsely sampled high-dimensional conformation space $\mathcal{C}$, this graph-based distance based on microstates better captures the dynamics than direct RMSD between individual conformations.

Search for the most suitable number of states

We applied our model construction algorithm over the microstates, and built HMM MDMs with increasing number of states, all at $\triangle t = 5$ ns timescale. 50% of trajectories is set aside for training $\mathcal{D}_{train}$, and another 25% is used for testing $\mathcal{D}_{test1}$. We save the remaining 25% of trajectories as the *second* test dataset $\mathcal{D}_{test2}$ for the hierarchy.

Fig. 4.15 shows the average log-likelihood score on *first* test dataset $\mathcal{D}_{test1}$. It shows that the score improves significantly when the number of states increases from 1 to 20. Improvement in the score is more gradual between 20 and 40 states. Beyond 40 states, the score remains approximately constant. Therefore, we consider the HMM MDM with about 40 states to be the most suitable HMM Θ_K. (41 states to be specific.)

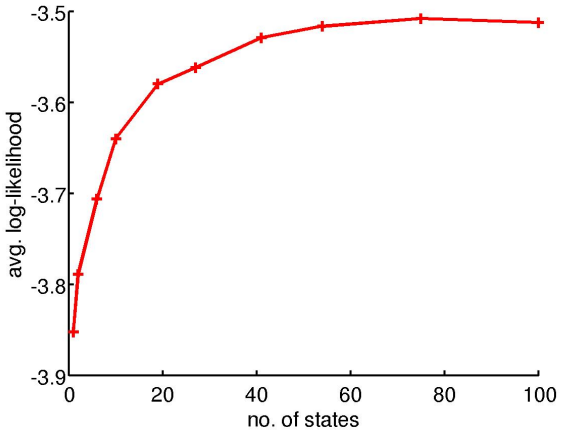


Figure 4.15: Average log-likelihood scores for the villin headpiece HMM MDMs on the first test dataset $\mathcal{D}_{test1}$. We choose $K = 41$ as the most suitable HMM MDM Θ_K.

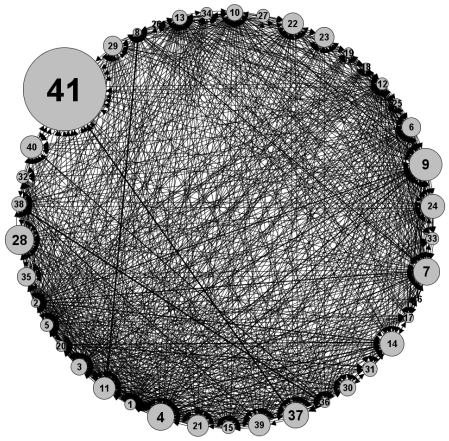


Figure 4.16: 41-state HMM MDM Θ_K of villin headpiece. The largest node, state 41, is the most likely folded state. The size of the nodes corresponds to the stationary distribution probabilities. The weight of an edge corresponds to the transition probability. The complexity of K^2 transitions in a large protein is difficult to visually analyze. A common way to ease analysis is to remove the smaller states and less significant transitions. However, simplifying a model *before* analysis is undesirable because it results in a less accurate model, and deviates from the model *actually* constructed. Hence, constructing an $\boldsymbol{\mathcal{H}}$HMM MDM is crucial for gaining biological understanding.

Fig. 4.16 is the graphical representation of the 41-state HMM MDM Θ_K. Although we can identify state 41 as the most significant state, knowing its 3-D structure does not explain ***how*** it can be achieved, nor the difficulty of achieving it. In addition, due to the complexity of villin's motion, it is difficult to visually examine its dynamics without first simplifying Fig. 4.16. A common way to ease analysis is to remove states with smaller stationary probabilities, and transitions with small probabilities. Another common approach is compute the most probable path between the most likely unfolded state, and the most likely folded state. However, simplifying a model *before* analysis is undesirable because it results in a less accurate model, and deviates from the model originally chosen. Knowing the most probable path, or the best few pathways are also inadequate because it is the *collective* contribution of all possible pathways that will determine a protein's function. Hence, constructing an $\boldsymbol{\mathcal{H}}$HMM MDM for a better spatial and temporal separation of villin's dynamics is crucial for gaining biological understanding.

Search for the most suitable $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$

We search for interesting aspects of villin's dynamics by using the most suitable HMM Θ_K to construct $\boldsymbol{\mathcal{H}}$HMM MDMs with different hierarchies of 41 energy basins. We begin the construction by creating an $\boldsymbol{\mathcal{H}}$HMM with 41 basin-states, and 1 root state. At each iteration, we grow the hierarchy downwards by clustering sibling basin-states into newly created subtrees deeper in hierarchy. We combine the hierarchy from each iteration with Θ_K to construct a candidate $\boldsymbol{\mathcal{H}}$HMM. We then use both $\mathcal{D}_{train}$ and $\mathcal{D}_{test1}$ to optimize parameters of the candidate $\boldsymbol{\mathcal{H}}$HMMs, before using the *second* test dataset $\mathcal{D}_{test2}$ to identify the most suitable $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$.

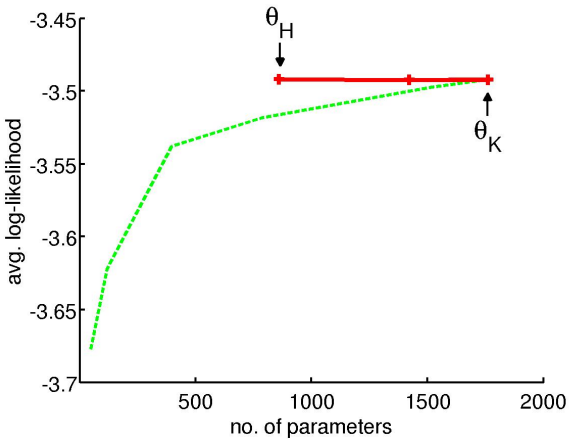


Figure 4.17: Average log-likelihood scores for the villin headpiece $\boldsymbol{\mathcal{H}}$HMM MDMs with different hierarchies on the second test dataset $\mathcal{D}_{test2}$ (red solid line). As we construct the hierarchy with each iteration, we create $\boldsymbol{\mathcal{H}}$HMMs with an *increasing* hierarchy, but a *decreasing* number of parameters. Since all the candidate models have similar score as compared to HMM Θ_K, we choose the model with the least number of parameters as the most suitable $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$. The scores of HMMs with different number of parameters are also shown for reference (green dashed line).

Fig. 4.17 shows the average log-likelihood score on the *second* test dataset $\mathcal{D}_{test2}$. As we construct the hierarchy with each iteration, we create $\boldsymbol{\mathcal{H}}$HMM MDMs with an *increasing* hierarchy, but a *decreasing* number of parameters. The most suitable $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$ has a hierarchy of 41 basin-states corresponding to a nested clustering of 41 energy basins. In addition, due to the collective estimation of transitions across clusters, $\Theta_{\mathcal{H}}$ has reduced the number of parameters from 1762 to 860, while maintaining the accuracy in predicting MD trajectories. $\Theta_{\mathcal{H}}$ also scores better than HMMs with the same number of parameters. This gives us an assurance that the hierarchical clustering of underlying energy basins is suitable for analysis, and is a crucial difference compared to other alternatives that can simplify a model, but without further verification against data.

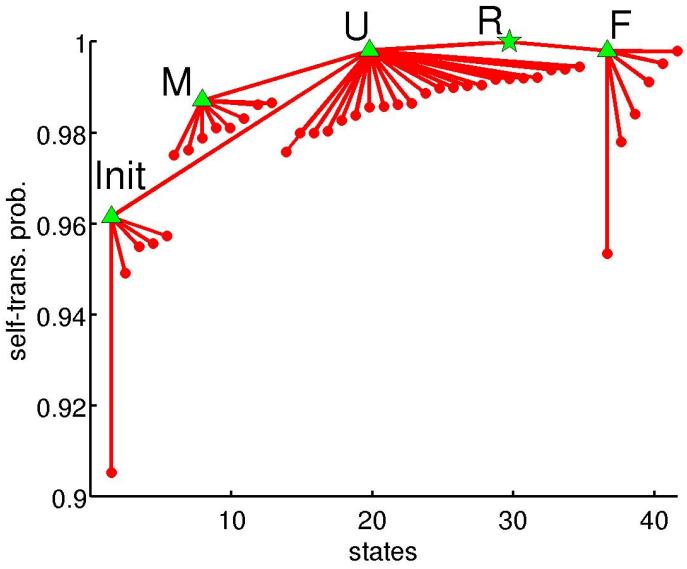
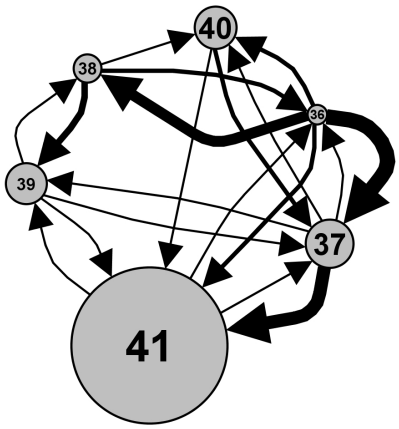
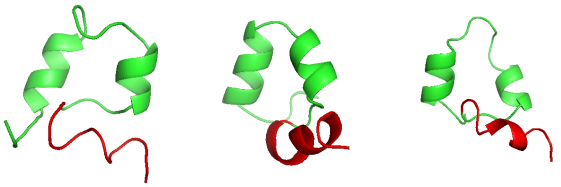


Figure 4.18: Hierarchy of the villin headpiece $\mathcal{H}$HMM MDM $\Theta_{\mathcal{H}}$ with 41 basin-states. The root state R at the top represents the whole MDM (green star). The folded cluster F and the unfolded cluster U are the children of R, the dynamics between them is the slowest to equilibrate, and corresponds to the folding process. Cluster Init is the initial cluster because initial conformations have a probability of 0.674 to be in it. The misfolded cluster M is also within U, and corresponds to the most significant misfold. Green triangles represent macro-states, and the red dots are basin-states corresponding to energy basins. Basin-states are numbered 1 to 41, while macro-states are labeled R, F, U, Init and M. Exit-states are not shown to avoid clutter. Representative conformations can be found in Fig. 4.19 and Fig. 4.20.

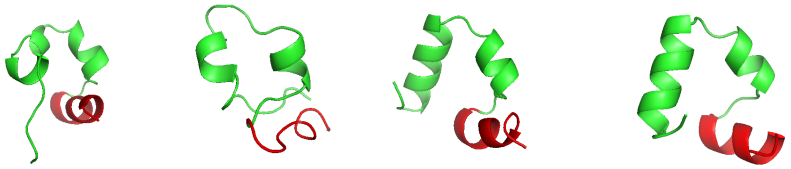
Fig. 4.18 shows that there are two main clusters (U and F) near the top of the hierarchy in $\Theta_{\mathcal{H}}$, where equilibration occurs at the longest timescale. We name these the unfolded cluster U, and the folded cluster F. This is because under the stationary distribution, the native conformation has a probability of 0.628 to be in cluster F (Fig. 4.19). From inference, the initial conformations have a probability of 0.949 to be in cluster U, and basin-states within U represent misfolded conformations (Fig. 4.20). Basin-states are numbered 1 to 41, while macro-states are labeled R, F, U, Init and M.



a) Transitions within the folded cluster F.



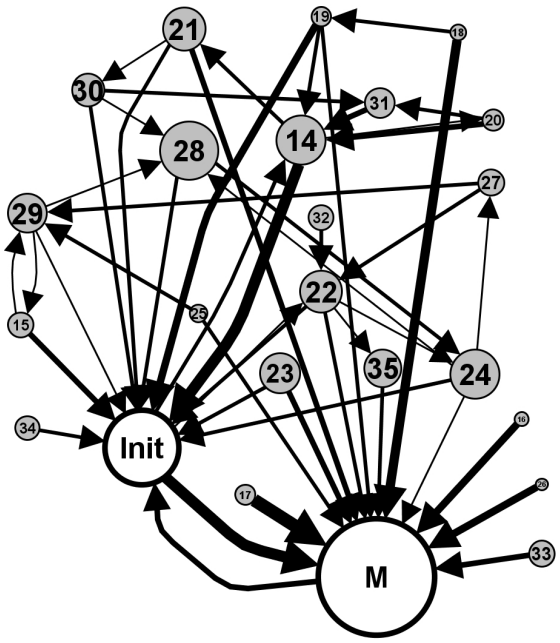
36 37 38



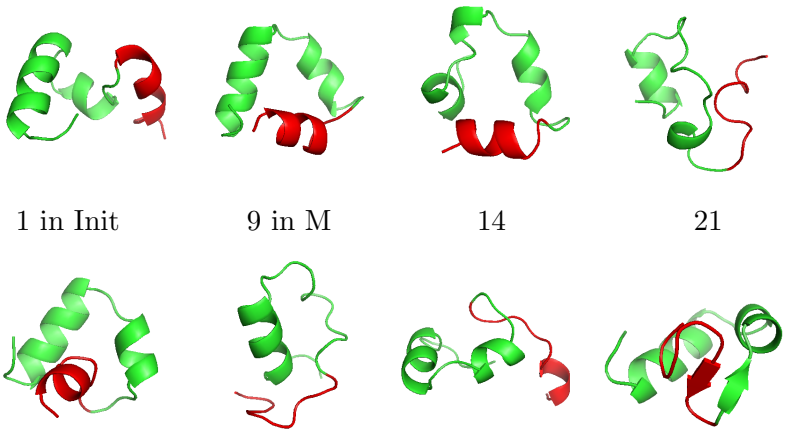
39 40 41 PDB: 2F4K

b) Example conformations.

Figure 4.19: The folded cluster F of the villin headpiece. The example conformation from state 41 has a RMSD of only 3.05 Å to the native (2F4K) conformation. Example conformation from state 37 has a RMSD of 5.31 Å to the native. From inference, the native conformation has the probability of 0.563 to be in state 41. State 41 also has a stationary distribution probability of 0.357.



a) Transitions within the unfolded cluster U.



1 in Init | 9 in M | 14 | 21

22 | 23 | 24 | 28

b) Example conformations.

Figure 4.20: The unfolded cluster U of the villin headpiece. The unfolded cluster U as a whole has a stationary distribution probability of 0.541, even more than the folded cluster F. From inference, cluster Init is the most likely initial cluster, with a probability of 0.674. Cluster M is the most significant misfold, and has a stationary distribution probability of 0.173.

This is our interpretation of villin protein's dynamics:

- **The folded cluster F** consists of 6 basin-states, see Fig. 4.19. From inference, the native conformation has the probability of 0.563 to be in state 41 alone. Structurally, state 41 is the closest to the native conformation, with a RMSD of 3.05 Å (all heavy atoms). State 37 is also very similar to the native conformation, with a RMSD of 5.31 Å. Since transition from state 37 to state 41 is more likely than the reverse, state 41 is the most dominant state under stationary distribution, with probability of 0.357.

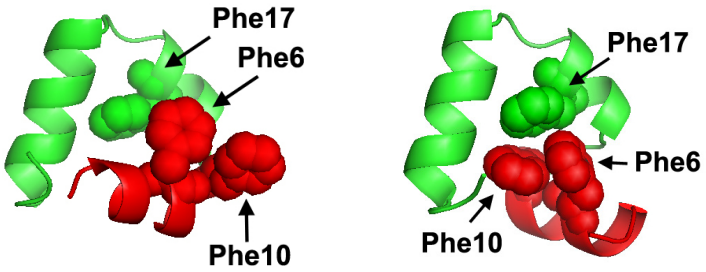
- **The unfolded cluster U** is surprisingly rather stable, and has a stationary distribution probability of 0.541 (Fig. 4.20). However, conformations represented by the states in cluster U either do *not* have *all* 3 helices, or have helix 1 in the *wrong* orientation.

  Additionally, there are two inner clusters. **Cluster Init** is the most likely *intial* cluster, with a probability of 0.674 out of cluster U's probability of 0.949 based on inference on the initial conformations. Cluster Init also has a relatively low self-transition probability (Fig. 4.18), and is easier to escape compared to most basin-states in $\Theta_{\mathcal{H}}$. When a trajectory escapes from cluster Init, it is most likely to transit to **cluster M**, which represents the most significant *misfolded* conformation with helix 1 in the *wrong* orientation (Fig. 4.20). The stability of the misfolded cluster M is an important factor that limits the folding of villin protein. However, cluster M only has a stationary distribution probability of 0.173 out of cluster U's probability of 0.541. Therefore, the other states within cluster U *collectively* contribute significantly to the difficulty of folding.

The clustering of states 14 to 35 within cluster U offers an important clue about the folding of villin protein. Structurally, states 14 to 35 represent *wrongly* folded conformations. Although each of these states has a small stationary distribution probability (0.0136 on average), their collective probability (0.298) is comparable to the most folded state, *i.e.* 41 in cluster F (0.357). Consequently, states 14 to 35 represent many opportunities where a villin protein can get trapped as it tries to fold. Once a trajectory escapes from states 14 to 35, it is likely to reach the misfolded cluster M (Fig. 4.20), while attempting to fold again. Collectively, states 14 to 35 constitute a significant obstacle to the folding of villin protein.

In summary, our model $\Theta_{\mathcal{H}}$ suggests that the folding of villin protein follows a general progression from the initial cluster Init, to the misfolded cluster M, before finally reaching the folded cluster F. However, as the protein explores pathways to reach the energetically favorable native conformation, there are also many wrongly folded conformations where it can be temporarily trapped (states 14 to 35).

Since a "*successful*" transit to the native conformation can occur quickly (Fig. 1.3 on page 22). Therefore, it is the presence of the many "*failed*" attempts in cluster U that indicates the actual difficulty of folding. Consequently, identifying states 14 to 35 in the unfolded cluster U is crucial for understanding the *actual* difficulty of villin's folding process. This is especially so when compared to the folded cluster F. The much fewer states in cluster F suggests that once the protein has made the difficult transition to cluster F, it remains stable in just a few energetically favorable conformations.



Misfold M PDB: 2F4K

Figure 4.21: Phenylalanine residues of the villin headpiece. Phe6 and Phe10 are in helix 1, Phe17 is in helix 2. A fourth residue Phe35 is at amino acid 35, but does not constitute the hydrophobic core. Helix 1 is in red, and is orientated towards the *left* in M, but towards the *right* in 2F4K. Hydrophobic residues tend to clump together in water.

Structurally, $\Theta_{\mathcal{H}}$ suggests that helix 1 of the villin headpiece protein appears to be the reason that has limited its folding, see Fig. 4.21. More specifically, $\Theta_{\mathcal{H}}$ suggests that it is *both* the helical structure and the correct orientation of helix 1 that is difficult to achieve. This is agreeable with earlier work suggesting that the presence of helix 1 is one of the possible reasons that has allowed certain initial conformations to fold faster [40]. Since the phenylalanine residues have also been shown to be vital for wild-type villin to achieve the native conformation [43], the packing of Phe6 and Phe10 in helix 1, in close contact with Phe17 in helix 2 in the native conformation, is worth further investigation.

The hierarchy in $\Theta_{\mathcal{H}}$ has beneficially allowed us to analyze a more complex model with better structural resolution. In our previous work *without* a hierarchy [27], since only a simpler model could be analyzed, the most folded state we could identify has a worse RMSD of 4.12 Å from the native conformation, versus the better 3.05 Å in current analysis with $\Theta_{\mathcal{H}}$.

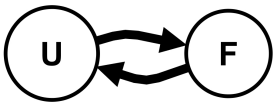


Figure 4.22: Transitions between the unfolded cluster U and the folded cluster F of the villin headpiece. We constructed a 2-state MDM by merging the basin-states within each cluster into one macro-state. The estimated transition probabilities between U and F are $p(F|U) = 0.00189$ and $p(U|F) = 0.00196$. The stationary distribution probability of F is 0.459, and is actually less than that of U.

In addition, an $\mathcal{H}$HMM MDM identifies the suitable transitions and timescale to analyze. For example, transitions between cluster U to cluster F occur at the longest timescale and is explicitly represented in the hierarchy. Without prior knowledge, identifying this *without* a hierarchy involves searching for small probabilities from numerous more probable transitions.

Although it is possible to construct multiple models at different timescales, it is unlikely that all the interesting timescales will be known prior to construction. In particular, constructing villin models at even longer timescales (*e.g.* 1 µs) will require much more data than currently available. Consequently, $\mathcal{H}$HMM MDM beneficially allows the construction of a *single* model, and detailed analysis of dynamics occurring at *multiple* timescales.

More importantly, our analysis is a lot more informative than the RMSD versus time plot in Fig. 1.3 (page 22). Our approach is also more intuitive than earlier analysis by Ensign *et al.* [40], which compares specific atomic distances based on prior knowledge of the protein:

- Separate RMSD of helix 1, helix 2, and helix 3 to native conformation.
- Pair-wise Cartesian distances between Phe6, Phe10, and Phe17.

We do *not* require such *prior* knowledge. Instead, we let data define the important conformations, build an $\mathcal{H}$HMM MDM to model the transitions, and let the hierarchy identify the interesting dynamics.

Simulating dynamics

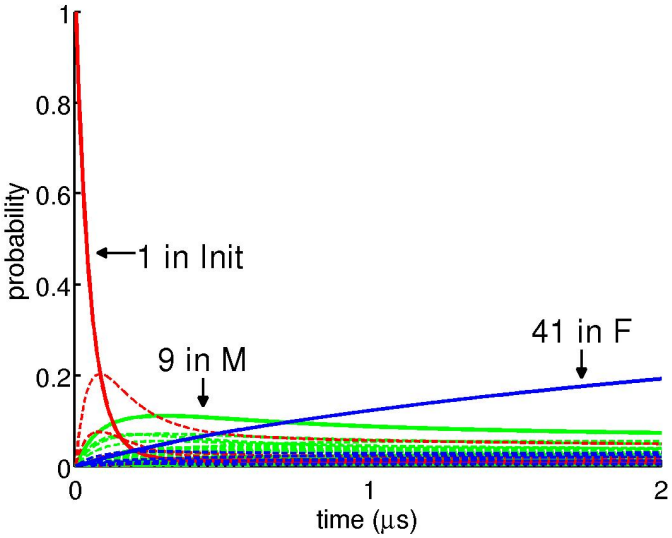
We also simulated the dynamics of villin headpiece using the most suitable $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$, see Fig. 4.23.

Fig. 4.23*a* shows the dynamics of the folding process, *i.e.* starting from the most likely initial state (1 in cluster Init). We can see that the probability of being in state 1 drops very rapidly. At the same time, the probability to be in the rest of the initial cluster Init first increases, but then decreases. This suggests that the initial conformations are highly unstable.

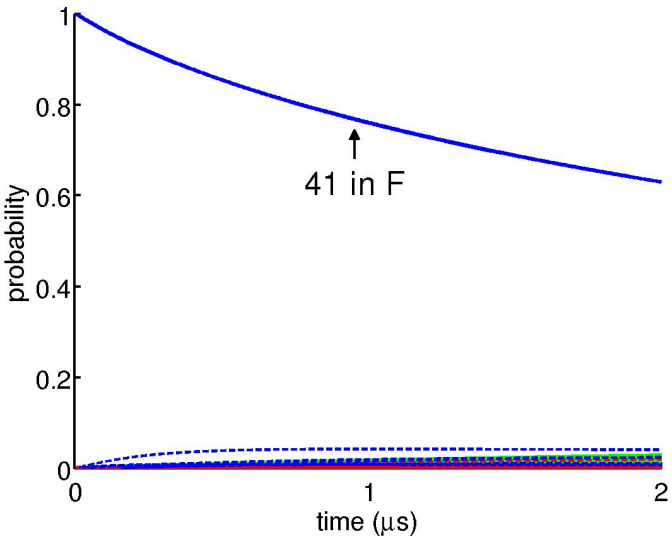
At about 0.3 µs, the most probable state is the most significant misfold (9 in cluster M), which persists relatively well for the rest of the simulation. This suggests that the misfolded conformation is actually rather stable.

From about 1 µs, the most likely folded state (41 in cluster F) begins to dominate, but has yet to stabilize before the end of the 2 µs simulation, indicating that the equilibration to native conformation is likely to be longer. This is agreeable with analysis by Ensign *et al.* [40] that 5 out of the 9 initial conformations generated trajectories that *briefly* visited conformations deemed to be folded. While a 6th initial conformation did not generate any trajectory able to achieve the native conformation at all.

More crucially, similar information can also be derived from the explicit representation in the hierarchy of $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$, *without* simulation. Additionally, without the earlier analysis using $\Theta_{\mathcal{H}}$, we would not know which are the important states to initialize the simulation. Lastly, since the most likely folded state (41 in cluster F) is *significantly* more stable than any other *individual* state (Fig. 4.23*b*), the *collective* contribution of the unfolded cluster U to the actual difficulty of folding is hard to detect and appreciate *without* $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$.



a) Starting from the most likely initial state (1 in cluster Init).



b) Starting from the most likely folded state (41 in cluster F).

Figure 4.23: Dynamics of the villin headpiece simulated using $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$. In each simulation, one branch in the hierarchy is assigned an initial probability of 1. Since we are concerned with the generation of conformations, only the basin-states are plotted. Red lines are states in cluster Init, green lines are states in the rest of cluster U, blue lines are states in cluster F. Solid lines correspond to the more important states, 1 in cluster Init, 9 in cluster M, and 41 in cluster F. The simulated duration is 2 μs, similar to the length of trajectories shown in Fig. 1.3 on page 22.

4.5 Discussions

Our $\mathcal{H}$HMM MDM approach to modeling protein motion dynamics is also applicable to other molecular motions, *e.g.* DNA, RNA. More specifically, $\mathcal{H}$HMM MDM will be especially useful when molecular motion involve complex interactions across different timescales. By searching for the most suitable $\mathcal{H}$HMM MDM $\Theta_{\mathcal{H}}$, the hierarchy will allows us to identify the interesting transitions and the suitable timescales to analyze them. A transition occurring near the top of the hierarchy indicates a difficult conformational change that occurs over long-timescales. While a transition near the bottom of the hierarchy corresponds to quick fluctuations among closely related conformations. This provides an intuitive way to understand *how* a molecule achieves its biological function through its motion.

More importantly, states within each cluster represent the variety of conformations closely associated with each other. By examining the structure of representative conformations within each cluster, it is possible to understand the reasons that might have prevented a molecule from undergoing *further* conformational change. More interesting is the comparison between different clusters with similar timescales. A cluster with a huge number of states indicates frequent transitions among a variety of conformations. Whereas a cluster with a small number of states corresponds to a few energetically favorable conformations. This is useful for biology because understanding the structural characteristics *within* a cluster of energy basins can reveal the reasons behind its stability, or instability, and offer clues as to how a protein can be better designed.

Additionally, $\mathcal{H}$HMM MDMs of different molecules can be used in comparison studies. For example, a protein can be mutated *in silico* and simulated with MD. By constructing different $\mathcal{H}$HMM MDMs on differently

mutated molecules, comparison between their motion dynamics can be made by examining the details of each model. A lowering in the longest timescale will correspond to a favoring of the equilibration process, *e.g.* folding. Whereas an increase in the number of underlying energy basins can indicate greater instability. Therefore, it is possible to better understand the effect of different mutations, *before* wet lab experiments are carried out.

Furthermore, each macro-state in the $\mathcal{H}$HMM MDM can be simulated individually to generate a sequence of conformations based on the underlying cluster of energy basins. Since $\mathcal{H}$HMM MDM generalizes over the velocities of molecular motion, simulation is useful when representative trajectories are needed for study. Simulation is also useful when predictions have to be made. For example, inference is based on the comparison between the simulated dynamics and trajectories. When two molecules M1 and M2 are to be compared to assess their compatibility, besides comparing the hierarchy of their corresponding models, inference can be done with the *model* of M1 on the *trajectories* of M2, and vice-versa. If the cross-comparison results in a good likelihood score, it suggests the two molecules exhibit similar dynamics, and might be able to interact similarly with other molecules if other factors such as chemical compatibility are also satisfied.

A potential question is the approximation of transitions between different parts of the hierarchy. For example, predicting a trajectory's *entry* into a cluster of energy basins relies on the equilibration of dynamics in the rest of the hierarchy. Therefore, regardless of where a trajectory came from, the descent into a new energy basin follows a common set of *downward* transitions. This may seem strange because a trajectory's proximity to the energy basins may seem helpful in determining which basin is more likely to be entered *first*. However, after descent, a trajectory quickly equilibrates

among energy basins in the new cluster, compensating for any potential difference in the *first* basin entered. Moreover, the collective estimation of horizontal transitions between clusters allows a better prediction of the *correct* cluster to transit to, and this is of greater importance.

Another question is the hierarchy explicitly represents dynamics at timescales longer than the $\triangle t$ timescale used to construct the model. In fact, it is possible for the slowest conformational change near the top of the hierarchy to have an expected timescale longer than the longest MD trajectory used in training. A question that can arise is: "How is that possible?" Our answer is it is due to the Markovian property, which allows the model to concatenate short simulation trajectories into much longer ones [29, 30]. Therefore, by constructing the $\boldsymbol{\mathcal{H}}$HMM MDM at the $\triangle t$ timescale, dynamics at longer timescales can be simulated under the Markovian assumption. Consequently, it is crucial to only cluster energy basins with fast equilibrating transitions between them.

Lastly, although a simple protein can fold directly and may not have hierarchical dependencies in its dynamics, the interlocking between different parts of a large protein will likely lead to many misfolded conformations. As we have seen in the villin protein, having a huge number of states can be especially problematic when trying to understand the sequence of events occurring during the folding process. Consequently, although an $\boldsymbol{\mathcal{H}}$HMM MDM may not be necessary for small proteins, $\boldsymbol{\mathcal{H}}$HMMs will be especially crucial when the dynamics is substantially more complex, and gaining an understanding is going to be difficult without an intuitive model.

Chapter 5

Computation of Ensemble Properties

An important purpose of modeling is to summarize salient features of the underlying phenomenon. The choice of modeling approach predetermines the properties that is to be preserved as well as the noise that is to be discarded. The process of training a model with data in effect distills desirable characteristics of the phenomenon down into a compact representation in the form of model parameters.

Assessing the accuracy of MDMs by comparing likelihood scores on new MD trajectories is one way of validation. However, it is an *indirect* validation against reality because MD simulation itself relies on further assumptions to justify its accuracy. Since nature is the golden benchmark against which all models should be validated, the most *direct* validation is the comparison of verifiable quantities against those obtained from wet lab experiments. For this, the advantage of modeling protein dynamics with MDMs is that they can be analyzed systematically to efficiently compute verifiable *ensemble properties* of molecular motion.

5.1 The Importance of Ensemble Properties

The computational modeling of biological phenomenon is only possible due to the culmination of scientific advancements over the centuries. From biology to biophysical theories, then from MD simulation to MDMs, when we are so far from biology, the inevitable question is: "Are we *still* accurate?"

Science is based on the prediction of empirical observations. Due to the molecular nature of proteins, they are difficult to isolate and their individual motions are impossible to measure precisely [78, 79]. Observations made in the wet lab are usually based on the *collective* behavior of an *ensemble* of protein molecules, *e.g.* protein solution in a test-tube. Therefore, ensemble properties that are quantifiable offer the golden benchmark against which computational models can be directly validated against reality.

However, not all comparisons between computed and wet lab ensemble properties are equal. For a particular ensemble property, it is also critical for there to be a direct correspondence between how the property can be obtained computationally, and the scientific explanation of the property. It is only when such correspondence exists, that a validation can serve as a further confirmation of the theory.

Such a correspondence exists between MDMs and molecular motion. Despite variability in the motion pathway of individual molecules, molecules of the same protein eventually fold into the same native conformation, and perform the same biological function. Conceptually, each possible motion trajectory can be translated into a sequence of MDM transitions. In addition, if a MDM is successful in capturing the motion dynamics of the protein, then the MDM should also have captured the characteristics along myriad pathways. Consequently, it should be possible to compute ensemble properties by averaging over *all* possible sequences of MDM transitions.

5.2 Mean First Passage Time (MFPT)

A model of protein dynamics has to be able to predict the *right* conformation at the *right* time. Computational techniques are able to model the atomic details of molecular motion, and various quantities can be computed based on molecular structures at a particular point in time. However, many computable quantities are not easily comparable across different modeling techniques, nor directly comparable against wet lab experiments. For example, the extent of secondary structure formation in a particular conformation can be calculated based on distances between atoms, but this is difficult to measure precisely in the wet lab.

The time for a protein to fold is both measurable and comparable. Measuring a protein's folding time, or rate, is one of the main wet lab experimental techniques used to study proteins [78, 79]. The reason is because a protein's folding time reflects the ease of achieving the native conformation, and is an important way to understand how a protein folds. For example, a mutation can cause a protein to fold more slowly because of the structural hurdles it introduces. Hence, although laborious, by measuring the folding times of similar proteins with slightly different mutations, scientists can compare the structural factors influencing the folding rate, and deduce how the protein folds.

The folding time corresponds to the Mean First Passage Time (MFPT). More specifically, the MFPT of a conformation q to fold is the *expected* time for a protein to reach a folded conformation, *starting* from q. By further requiring conformation q to be unfolded, the MFPT is directly comparable with protein folding times measurable in the wet lab. Consequently, the comparison of the computed MFPT against wet lab folding time provides a crucial experimental validation for MDMs.

Although it is tempting to simulate many folding trajectories starting from q, terminating them when a folded conformation is reached, and then take the average length as an estimate for MFPT. This is impractical because explicit simulation is computationally expensive. In addition, a reliable estimate is not guaranteed because trajectories may not have reached a folded conformation within the allocated simulation time. More importantly, according to the definition, the *expected* time requires the averaging over *infinite* trajectories that started from q, *i.e.* $\mathcal{D}_i = (q, \ldots, q_t, \ldots, q_T)$, where $\{q_t \notin C_{\mathrm{F}} \mid t = 1, \ldots, T-1\}$, $\{q_T \in C_{\mathrm{F}} \mid T = 0, \ldots, \infty\}$, and $C_{\mathrm{F}} \subset C$ is the set of folded conformations.

Here, we describe a practical algorithm to compute MFPT using MDM. Our computation proceeds in two stages. First, we compute the MFPTs for all ***states*** $i \in \mathcal{S}$. The MFPT for a *state* i is the expected time for a protein to reach a folded conformation, *starting* from *state* i. Note that this is *different* from the MFPT of a conformation q that we require. Therefore, in the second stage, we need to incorporate the initial ***conformation*** q into the computation to complete the estimation of MFPT.

Although we will first discuss the computation based on transitions between the K states of an HMM, the case for an $\boldsymbol{\mathcal{H}}$HMM can be similarly derived by resolving the hierarchical dependencies, which we will discuss later.

MFPT of initial *states*

The idea of our approach is to weigh the measurement of time according to the outcome at each step. However, instead of explicitly simulating trajectories, we make use of first-step analysis [103] from Markov chain theory to incorporate the infinite pathways, including cycles, in our computation. This results in a much faster and reliable computation of MFPT.

Let us consider the first-passage time (***FPT***) γ_i of a *single trajectory* simulated by starting in state i at time $t = 0$. This is what can happen in the very *first* time step, where $C_{\text{F}} \subset C$ is the set of folded conformations:

- If the conformation at $t = 0$ is ***folded***, $q_0 \in C_{\text{F}}$, then the trajectory is terminated, and $\gamma_i = 0$ by definition. This event happens with probability $e_i(C_{\text{F}}) = \int_{C_{\text{F}}} e_i(q)\, dq$, where $e_i(q) = \mathcal{N}(q|\mu_i, \sigma_i^2)$ is the emission probability of state i.

- If the conformation at $t = 0$ is ***unfolded***, $q_0 \notin C_{\text{F}}$, then the trajectory continues, and γ_i will depend on the outcome in the *next* time step, *i.e.* $t = 1$. This event happens with probability $1 - e_i(C_{\text{F}})$.

Let us now consider the case for *all possible trajectories*. More specifically, the ***MFPT*** for ***state*** i is $\bar{\gamma}_i = \mathbb{E}(\gamma_i | s_0 = i)$, where the expectation is taken over *all* trajectories that started in state i and end in C_{F}. By conditioning on the events in the first time step, we obtain the following equation for $\bar{\gamma}_i$:

$$\bar{\gamma}_i = 0 \cdot e_i(C_{\text{F}}) + \Big(1 + \sum_{j \in S} p(s_1 = j | s_0 = i) \bar{\gamma}_j\Big) \cdot \big(1 - e_i(C_{\text{F}})\big), \tag{5.1}$$

where $0 \cdot e_i(C_{\text{F}})$ is the case where a *folded* conformation has been reached, and the contribution to $\bar{\gamma}_i$ is 0. However, when an *unfolded* conformation has been reached, we need to consider all possible transitions to the

next time step, $\sum_{j \in S} p(s_1 = j | s_0 = i)$, and the consequent outcome $\bar{\gamma}_j$.

The only unknowns in Eq. 5.1 are the MFPTs $\bar{\gamma}_i$ for $i = 1, 2, \ldots, K$ states corresponding to K energy basins. Since there is one equation for each $\bar{\gamma}_i$, we get a linear system of K equations with K unknowns, which can be solved efficiently using standard numerical methods. More importantly, the algebraic process of solving the linear system *implicitly* enumerates all possible state sequences of the folding trajectories in an efficient way.

At this stage, we can optionally compute the ***MFPT*** of the ***MDM*** $\bar{\gamma}$ by multiplying with the prior probability π_i:

$$\begin{aligned} \bar{\gamma} &= \sum_{i \in S} \mathbb{E}(\gamma_i | s_0 = i) p(s_0 = i) \\ &= \sum_{i \in S} \bar{\gamma}_i \pi_i. \end{aligned} \tag{5.2}$$

Although $\bar{\gamma}$ can be an estimate of folding time, we wish to be more specific about the starting *conformation*. This is because the prior is dependent on initial conformations in the training data, which may not be the same as the initial conformations we wish to compare with.

MFPT of *conformations*

Next, we compute the MFPT for a given *conformation* q_0. Let γ be the ***FPT*** of a *single trajectory* that starts at conformation q_0. Conditioning on the initial state s_0 at $t = 0$, we see that the MFPT of q_0 is given by:

$$\mathbb{E}(\gamma|q_0) = \sum_{i \in S} \mathbb{E}(\gamma|q_0, s_0 = i)p(s_0 = i|q_0). \tag{5.3}$$

We first calculate $p(s_0 = i|q_0)$ using the Bayes rule:

$$p(s_0 = i|q_0) = \frac{p(q_0|s_0 = i)p(s_0 = i)}{\sum_{i \in S} p(q_0|s_0 = i)p(s_0 = i)}, \tag{5.4}$$

where $p(s_0 = i)$ and $p(q_0|s_0 = i)$ can be obtained from the prior probabilities Π and the emission probabilities E of the model, respectively.

Calculating $\mathbb{E}(\gamma|q_0, s_0 = i)$ is more subtle because it is tempting to think that $\mathbb{E}(\gamma|q_0, s_0 = i) = \bar{\gamma}_i$. This is incorrect, because $\bar{\gamma}_i = \mathbb{E}(\gamma|s_0 = i)$ and the additional information provided by q_0 alters the expected value of γ.

To calculate $\mathbb{E}(\gamma|q_0, s_0 = i)$, we need to take an *additional* step and condition once more on the state j at time $t = 1$:

$$\mathbb{E}(\gamma|q_0, s_0 = i) = \sum_{j \in S} \mathbb{E}(\gamma|q_0, s_0 = i, s_1 = j)p(s_1 = j|q_0, s_0 = i) \tag{5.5}$$

$$= \sum_{j \in S} \big(1 + \mathbb{E}(\gamma|s_1 = j)\big)p(s_1 = j|s_0 = i), \tag{5.6}$$

where the last line follows because given the state $s_0 = i$ at time $t = 0$, the state $s_1 = j$ at time $t = 1$ is independent of conformation q_0. The values of $\mathbb{E}(\gamma|s_1 = j)$ can be obtained from the MFPT of the *states* $\bar{\gamma}_i$, for $i = 1, 2, \ldots K$ (Eq. 5.1). Substituting Eq. 5.4 and Eq. 5.6 into Eq. 5.3 gives us the ***MFPT*** of ***conformation*** q_0.

In summary, we compute of MFPT of a conformation q_0 by conditioning on the sequence of events. First, given q_0, we estimate the state s_0 at time $t = 0$. Second, given s_0, we estimate the state s_1 at time $t = 1$. From time $t \geq 1$ onwards, we estimate the probability of reaching a folded conformation $q_t \in C_{\mathrm{F}}$ at every step. This is because those trajectories reaching $q_t \in C_{\mathrm{F}}$ will terminate without further contribution to MFPT.

Other ensemble properties can also be calculated by adjusting the cases of consideration. For example, the probability of folding (p-fold) is a measure of the kinetic distance between a conformation q and the native conformation [38]. Specifically, p-fold of a conformation q is the probability of reaching a folded conformation, *before* an unfolded conformation, starting from q (Section 2.2.2 on page 37). Consider at time t:

- a ***folded*** conformation has been reached, $q_t \in C_{\mathrm{F}}$.
- an ***unfolded*** conformation has been reached, $q_t \in C_{\mathrm{U}}$.
- ***neither*** folded nor unfolded conformation has been reached.

Consequently, for example, we can replace Eq. 5.1 for state i with:

$$\begin{aligned} \mathrm{P}_{\mathrm{fold}i} &= 1 \cdot e_i(C_{\mathrm{F}}) + 0 \cdot e_i(C_{\mathrm{U}}) + \\ &\quad \Big(\sum_{j \in S} p(s_{t+1} = j | s_t = i) \mathrm{P}_{\mathrm{fold}j}\Big) \cdot \big(1 - e_i(C_{\mathrm{F}}) - e_i(C_{\mathrm{U}})\big). \quad (5.7) \end{aligned}$$

Since we are not measuring time, the step count of 1 in the original Eq. 5.1 has been removed.

In practice, when we compare with experimental measures, we are interested in the MFPT for a region C' of conformation space $\mathcal{C}$ rather than a single conformation $q_0 \in \mathcal{C}$. To calculate $\mathbb{E}(\gamma | C')$, we need to modify Eq. 5.3, Eq. 5.4, and Eq. 5.6 slightly by integrating q_0 over C'.

Computational efficiency

Although the calculation of MFPT follows multiple steps, the actual computation is very efficient. In the first stage, we compute the MFPT of the *initial states*. This requires solving a system of K linear equations for K energy basins, at the cost of $O(K^3)$. Although $O(K^3)$ may seem significant, K is usually orders of magnitude *smaller* than the length T of MD trajectories. In the second stage, we compute the MFPT of the *conformations*. This requires a single transition from time $t = 0$, to time $t = 1$, which takes $O(K^2)$ to multiply the transition matrix A once. Since the inference on the unfolded and folded conformations only has to be done once, the overall run time is limited by the $O(K^3)$ cost to solve the system of K linear equations.

More importantly, the alternative, but impractical, approach to estimate MFPT is simulation. Although we can calculate the average lengths of the available MD trajectories that satisfy the MFPT criteria, doing so requires checking every conformation. This incurs a cost of $O(NT)$ time, where N is the number of trajectories. and the trajectory length T is substantially larger than K. Furthermore, this requires a sufficient number of long trajectories traversing *both* the unfolded conformations, and the folded conformations. This is difficult to achieve, except for the smallest proteins.

Another alternative is to simulate the MDM over T time steps, checking the probability of reaching folded conformations at every step, and taking the probability weighted average of length as the MFPT. However, it is impossible to simulate a MDM infinitely, and it is also doubtful that convergence can be achieved. Therefore, not only are the alternative approaches to estimate MFPT significantly more costly, they are only viable for the smallest proteins where the dynamics can be well sampled via MD.

MFPT for $\mathcal{H}$HMM MDMs

The MFPT for an $\mathcal{H}$HMM MDM can be similarly computed. Both HMMs and $\mathcal{H}$HMMs model protein dynamics as a probabilistic distribution of conformations that changes over time. The difference is that an HMM models this change via transitions between each of its K "*hidden*" states, while an $\mathcal{H}$HMM models this as transitions between each of the K branches in its hierarchy. Therefore, to compute MFPT for an $\mathcal{H}$HMM, we need to resolve the hierarchical dependencies so that inference on the unfolded and folded conformations can be carried out.

The key is Eq. 4.1 (page 97), which combines the horizontal and vertical transitions across the hierarchy into transitions between states of the *whole* $\mathcal{H}$HMM (*i.e.* branches in the hierarchy). More specifically, we need to calculate the probability to transit from (the whole) $\mathcal{H}$HMM state s_t at time t, to state s_{t+1} at time $t+1$, *i.e.* $\hat{p}(s_{t+1}|s_t)$. Since each state s_t of the whole $\mathcal{H}$HMM corresponds to an HMM "*hidden*" state, therefore after the conversion, the same MFPT equations can be used for calculations.

The cost of converting the $\mathcal{H}$HMM transitions into equivalent K-state HMM transitions is $O(K^2D)$. This is because for each of the K basin-states in the $\mathcal{H}$HMM, it is necessary to propagate the dynamics up and down the hierarchy of depth D, where at most K states is present at each level. However, since each macro-state can have many children, the depth D of the hierarchy with K leaf basin-states should be relatively small. Consequently, the run time of the MFPT computation is still limited by the $O(K^3)$ it takes to solve a system of K linear equations.

5.3 Results

To demonstrate the accuracy of our algorithms, we computed MFPT for alanine dipeptide and villin headpiece.

5.3.1 Alanine dipeptide

Table 5.1: Estimated MFPTs between α_R and $\beta/C5$ regions of the alanine dipeptide conformation space. See Fig. 3.4 for the conformations (page 72).

	MFPT (ns)	
	$K3$	$M6$
$\alpha_R \rightarrow \beta/C5$	5.75	5.91
$\beta/C5 \rightarrow \alpha_R$	76.34	27.35

To further validate our models, we used models $K3$ and $M6$ from Fig. 3.7 to compute MFPTs between the α_R and $\beta/C5$ regions of the conformation space. We designate conformations with ($\phi = -70 \pm 1$, $\psi = -40 \pm 1$, $\omega = 180 \pm 1$) to be within the α_R region, and conformations with ($\phi = -140 \pm 1$, $\psi = 160 \pm 1$, $\omega = 180 \pm 1$) to be within the $\beta/C5$ region. The transition $\alpha_R \rightarrow \beta/C5$ is in the nanosecond timescale similar to results by Smith using NMR observed coupling constant [7]. Although results for $K3$ and $M6$ differ somewhat in details, both indicate the transition $\alpha_R \rightarrow \beta/C5$ is roughly an order of magnitude faster than the reverse transition. This is agreeable with analysis by Chekmarev *et al.* [24].

To assess the efficiency of our algorithm for MFPT computation, we also computed the MFPTs by *explicitly* generating simulation trajectories from our constructed models. It took our algorithm less than 1 s to compute one MFPT. In comparison, it took 120 s to generate a sufficiently large number of simulation trajectories from the same HMM in order to bring the standard deviation of the MFPT estimate down to 1% of its value.

5.3.2 Villin headpiece

Table 5.2: Estimated MFPTs for nine initial conformations of the villin headpiece (HP-35 NleNle).

MFPT (µs)								
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
6.41	6.41	6.35	6.39	6.29	6.40	6.43	6.26	6.32

We also computed the MFPTs for the nine initial conformations of villin headpiece, I_0 to I_8 (see Fig. 5.1). Table 5.2 shows the result using the most suitable $\boldsymbol{\mathcal{H}}$HMM MDM $\Theta_{\mathcal{H}}$ with 41 basin-states. The results lie in the same microsecond range as the experimental measurements of 4.3 µs from laser temperature jump [63] and 10 µs from NMR line-shape analysis [111]. In addition, the MFPTs for I_4 and I_7 are slightly smaller, which is consistent with the computational analysis of Ensign *et al.* in [40].

For comparison, we also tried to compute the MFPTs by explicitly generating trajectories from the constructed models. However, after 30 minutes of computation, the estimated MFPTs are still two orders of magnitude below the microsecond range. In comparison, the results in Table 5.2 were obtained in less than 1 minute of computation.

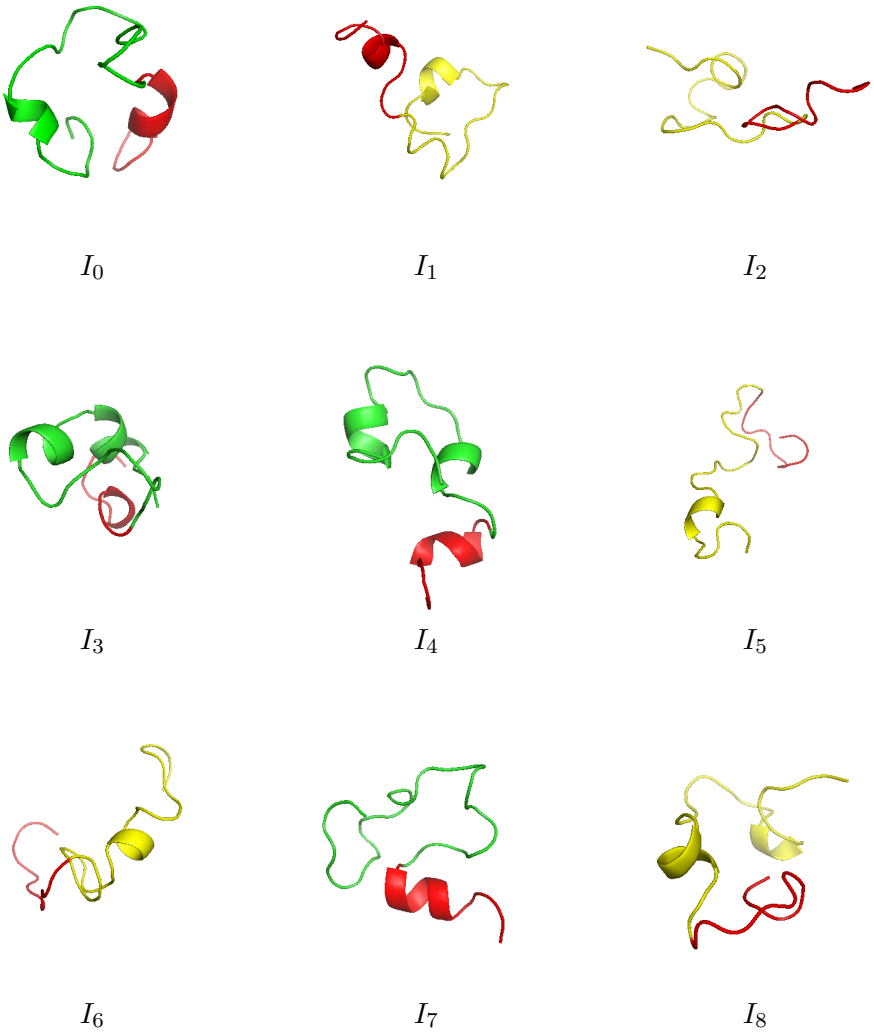


Figure 5.1: Initial conformations of the villin headpiece [40].

Chapter 6

Conclusion

The past decade has witnessed an increasing interest in graphical models of protein motion dynamics at long-timescales. Most recently, the focus has been on cell-based MDMs constructed from MD simulation data. However, existing methods suffer from two main shortcomings. First, defining states by partitioning the protein conformation space into disjoint cells causes violation of the Markovian property. Second, there is no systematic criterion for evaluating the model quality.

Chapter 3 addresses these two shortcomings by defining states as probability distributions of conformations. This reflects the view that a single conformation does not contain enough information to be assigned to a unique state. The resulting HMM-based modeling framework evaluates the model quality by the likelihood of a model given a test dataset of simulation trajectories. In contrast with the cell-based MDMs, our approach enables us to compare models with different number of states and choose the most suitable model according to the likelihood criterion. The results on synthetic energy landscapes and alanine dipeptide illustrate this benefit.

Chapter 4 scales up the modeling approach by searching across timescales. This is crucial because for large proteins with complex motions, biologically interesting dynamics can occur over a *range* of timescales. Without prior knowledge, it is necessary to construct different models at different timescales. Instead, we construct a *single* hierarchical model by searching for clusters of energy basins with fast internal equilibration. The resulting $\mathcal{H}$HMM MDM beneficially identifies the interesting dynamics and the suitable timescales for analysis. This allows us to scale up for larger proteins, yet with a lower cost in the number of parameters. Results on the 11-basins synthetic landscape illustrate scenarios where $\mathcal{H}$HMM MDMs will be useful. Results on the villin headpiece allowed us to appreciate the collective contribution of misfolded conformations to the actual difficulty of folding. This demonstrates the benefits of $\mathcal{H}$HMM MDMs in practical use.

In general, MDMs have several advantages over direct data analysis of MD trajectories. MDMs generalize over the data used to construct them. This allows MDMs to identify states that correspond to biologically significant conformations, and assemble them to provide a global view of the underlying stochastic dynamics. More importantly, MDMs accomplish this *without* relying on prior knowledge of the protein. Instead, data define the important states to capture. This is in contrast with traditional, and often laborious, structural comparisons with reference conformations.

Chapter 5 shows how to exploit MDMs to compute ensemble properties by *implicitly* simulating *infinite* trajectories. The computation of ensemble quantities such as Mean First Passage Time (MFPT) provides the crucial validation of computational models against wet lab experiments. Such tasks are usually difficult or impossible with direct data analysis. Our results on the MFPTs of alanine dipeptide and villin headpiece validate the MDMs.

More importantly, the equations for calculating ensemble quantities can also be easily adapted to estimate other properties. In addition, MDMs are generative, and as we have shown, can too be used for *explicit* simulation.

In the broader context of modeling biological macromolecules. Although MD simulation is still computationally expensive, advances in computer technology are making it more affordable than before. Large simulation data repositories will also become readily available over time. Increasingly, the future challenge will be to gain biological insights from this data by building simple and yet powerful models. The results on the "*inverted*" synthetic landscape with 11 "*hills*" indicate MDMs can also be useful in the cross-comparison of different molecules based on motion dynamics. This can allow scientists to investigate biology at greater details, and gain a better understanding of molecular interactions for the design of novel drugs.

For the more immediate future, it will be interesting to apply our approach to model the dynamics of folded proteins. The conformational flexibility of a folded protein is critical to some of its functions [47]. Here, our approach is likely to scale up well to larger molecules, because transitions between the folded states are often fast and more easily captured by short MD simulations.

In short, this thesis has presented an efficient approach to model a protein's motion dynamics. The resulting Markov Dynamic Model is both accurate in predicting MD trajectories, and intuitive for gaining a biological understanding of the protein. The exploitation to compute ensemble properties crucially allows a MDM to be validated against wet lab experiments. It is hoped that these will serve as the basis of future research, and further our understanding of the wondrous nature.

Bibliography

[1] http://www.wikipedia.org.

[2] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell*. 5th edition, 2007.

[3] F. Allen, G. Almasi, W. Andreoni, D. Beece, B. J. Berne, A. Bright, J. Brunheroto, C. Cascaval, J. Castanos, P. Coteus, P. Crumley, A. Curioni, M. Denneau, W. Donath, M. Eleftheriou, B. Fitch, B. Fleischer, C. J. Georgiou, R. Germain, M. Giampapa, D. Gresh, M. Gupta, R. Haring, H. Ho, P. Hochschild, S. Hummel, T. Jonas, D. Lieber, G. Martyna, K. Maturu, J. Moreira, D. Newns, M. Newton, R. Philhower, T. Picunko, J. Pitera, M. Pitman, R. Rand, A. Royyuru, V. Salapura, A. Sanomiya, R. Shah, Y. Sham, S. Singh, M. Snir, F. Suits, R. Swetz, W. C. Swope, N. Vishnumurthy, T. J. C. Ward, H. Warren, and R. Zhou. Blue gene: A vision for protein science using a petaflop supercomputer. *IBM Systems Journal*, 40(2):310–327, 2001.

[4] Ethem Alpaydin. *Introduction to Machine Learning*. 2nd edition, 2009.

[5] A. Amadei, A.B. Linssen, and H.J. Berendsen. Essential dynamics of proteins. *Proteins: Structure, Function, and Genetics*, 17:412–425, 1993.

[6] Nancy M. Amato, Ken A. Dill, and Guang Song. Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. *J. Comput. Biol.*, 10(3-4):239–255, 2003.

[7] Paul Smith and. The alanine dipeptide free energy surface in solution. *Journal of Chem. Phys.*, 111:5568–5579, 1999.

[8] C. B. Anfinsen. The formation and stabilization of protein structure. *Biochemical Journal*, 128:737–749, 1972.

[9] Mehmet S. Apaydin, Douglas L. Brutlag, Carlos Guestrin, David Hsu, Jean-Claude Latombe, and Chris Varma. Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion. *J. Comput. Biol.*, 10(3-4):257–281, 2003.

[10] A. R. Atilgan, S. R. Durell, R. L. Jernigan, M. C. Demirel, O. Keskin, and I. Bahar. Anisotropy of fluctuation dynamics of proteins with an elastic network model. *Biophys. J.*, 80:505–515, 2001.

[11] I. Bahar, A. Wallqvist, D. G. Covell, and R. L. Jernigan. Correlation between native-state hydrogen exchange and cooperative residue fluctuations from a simple model. *Biochemistry*, 37:1067–1075, 1998.

[12] Ivet Bahar, Ali Rana Atilgan, Melik C. Demirel, and Burak Erman. Vibrational dynamics of folded proteins: Significance of slow and fast motions in relation to function and stability. *Physical Review Letters*, 80:2733–2736, 1998.

[13] Ivet Bahar, Ali Rana Atilgan, and Burak Erman. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Folding and Design*, 2:173–181, 1997.

[14] Adam L. Beberg, Daniel L. Ensign, Guha Jayachandran, Siraj Khaliq, and Vijay S. Pande. Folding@home: Lessons from eight years of volunteer distributed computing. In *IEEE International Symposium on Parallel & Distributed Processing*, 2009.

[15] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000.

[16] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[17] Gregory R. Bowman, Xuhui Huang, and Vijay S. Pande. Using generalized ensemble simulations and Markov state models to identify conformational states. *Methods*, 49:197201, 2009.

[18] Rodney F. Boyer. *Concepts in Biochemistry*. 3rd edition, 2005.

[19] Carl Branden and John Tooze. *Introduction to Protein Structure*. 2nd edition, 1999.

[20] Charles. Brooks and David A. Case. Simulations of peptide conformational dynamics and thermodynamics. *Chemical Review*, 93:2487–2502, 1993.

[21] Hung Bui, Svetha Venkatesh, and Geoff West. Tracking and surveillance in wide-area spatial environments using the Abstract Hidden Markov Model. *International Journal of Pattern Recognition and Artificial Intelligence*, 15:177–196, 2001.

[22] C. Sidney Burrus, Ramesh A. Gopinath, and Haitao Guo. *Introduction to Wavelets and Wavelet Transforms: A Primer*. 1997.

[23] John Cavanagh, Wayne J. Fairbrother, Arthur G. Palmer III, Nicholas J. Skelton, and Mark Rance. *Protein NMR Spectroscopy: Principles and Practice*. 2006.

[24] Dmitriy S. Chekmarev, Tateki Ishida, and Ronald M. Levy. Long-time conformational transitions of alanine dipeptide in aqueous solution: Continuous and discrete-state kinetic models. *J. Phys. Chem. B*, 108:19487–19495, 2004.

[25] Tsung-Han Chiang, Mehmet Serkan Apaydin, Douglas L. Brutlag, David Hsu, and Jean-Claude Latombe. Predicting experimental quantities in protein folding kinetics using stochastic roadmap simulation. In *Proc. ACM Int. Conf. on Research in Computational Molecular Biology (RECOMB)*, 2006.

[26] Tsung-Han Chiang, Mehmet Serkan Apaydin, Douglas L. Brutlag, David Hsu, and Jean-Claude Latombe. Using stochastic roadmap simulation to predict experimental quantities in protein folding kinetics: Folding rates and phi-values. *J. Comput. Biol.*, 14(5):578593, 2007.

[27] Tsung-Han Chiang, David Hsu, and Jean-Claude Latombe. Markov dynamic models for long-timescale protein motion. *Bioinformatics, Special issue on Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, 26:i269i277, 2010.

[28] Fabrizio Chiti and Christopher M. Dobson. Protein misfolding, functional amyloid, and human disease. *Annual Reviews of Biochemistry*, 75:333–366, 2006.

[29] John D. Chodera, Nina Singhal, Vijay S. Pande, Ken A. Dill, and William C. Swope. Automatic discovery of metastable states for the construction of markov models of macromolecular conformational dynamics. *J. Chem. Phys.*, 126:155101, 2007.

[30] John D. Chodera, William C. Swope, Jed W. Pitera, and Ken A. Dill. Long-time protein folding dynamics from short-time molecular dynamics simulations. *Multiscale Modeling & Simulation*, 5(4):1214–1226, 2006.

[31] Qiang Cui and Ivet Bahar. *Normal Mode Analysis: Theory and Applications to Biological and Chemical Systems.* 2005.

[32] Jr. D. E. Koshland. Application of a theory of enzyme specificity to protein synthesis. *Proc. Natl. Acad. Sci. U. S. A.*, 44:98104, 1958.

[33] Payel Das, Mark Moll, Hernan Stamati, Lydia E. Kavraki, and Cecilia Clementi. Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. *Proc. Natl. Acad. Sci. U. S. A.*, 103:9885–9890, 2006.

[34] P. Deuflhard, W. Huisinga, A. Fischer, and Ch. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. *Linear Algebra and its Applications*, 315:3959, 2000.

[35] Peter Deuflhard and Marcus Weber. Robust perron cluster analysis in conformation dynamics. Technical report, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2003.

[36] Ken Dill and Sarina Bromberg. *Molecular Driving Forces: Statistical Thermodynamics in Biology, Chemistry, Physics, and Nanoscience.* 2010.

[37] David L. Donoho and Carrie Grimes. Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data. *Proc. Natl. Acad. Sci. U. S. A.*, 100:55915596, 2003.

[38] Rose Du, Vijay S. Pande, Alexander Yu. Grosberg, Toyoichi Tanaka, and Eugene S. Shakhnovich. On the transition coordinate for protein folding. *J. Chem. Phys.*, 108(1):334–350, 1998.

[39] R. Elber. Long-timescale simulation methods. *Curr. Opin. Struct. Bio.*, 15:151–156, 2005.

[40] Daniel L. Ensign, Peter M. Kasson, and Vijay S. Pande. Heterogeneity even at the speed limit of folding: Large-scale molecular dynamics study of a fast-folding variant of the villin headpiece. *J. Mol. Biol.*, 374:806–816, 2007.

[41] Alan Fersht. *Structure and Mechanism in Protein Science*. 1998.

[42] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, 32:40–62, 1998.

[43] B. S. Frank, D. Vardar, D. A. Buckley, and C. J. McKnight. The role of aromatic residues in the hydrophobic core of the villin headpiece subdomain. *Protein Science*, 11:680–687, 2002.

[44] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. 2nd edition, 2001.

[45] Turkan Haliloglu, Ivet Bahar, and Burak Erman. Gaussian dynamics of folded proteins. *Phys. Rev. Lett.*, 79(16):3090–3093, 1997.

[46] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. 2nd edition, 2005.

[47] Katherine Henzler-Wildman and Dorothee Kern. Dynamic personalities of proteins. *Nature*, 450:964–972, 2007.

[48] Berk Hess, Carsten Kutzner, David van der Spoel, and Erik Lindahl. GROMACS 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. *Journal of Chemical Theory and Computation*, 4:435–447, 2008.

[49] K. Hinsen. Analysis of domain motions by approximate normal mode calculations. *Proteins*, 15:417–429, 1998.

[50] Michael Hirsch and Michael Habeck. Mixture models for protein structure ensembles. *Bioinformatics*, 24(19):2184–2192, 2008.

[51] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4:629–642, 1987.

[52] Reto Horst, Eric B. Bertelsen, Jocelyne Fiaux, Gerhard Wider, Arthur L. Horwich, and Kurt Wüthrich. Direct NMR observation of a substrate protein bound to the chaperonin GroEL. *Proc. Natl. Acad. Sci. U. S. A.*, 102:12748–12753, 2005.

[53] Xuhui Huang, Yuan Yao, Gregory R. Bowman, Jian Sun, Leonidas J. Guibas, Gunnar Carlsson, and Vijay S. Pande. Constructing multi-resolution markov state models (MSMs) to elucidate rna hairpin folding mechanisms. In *Pacific Symposium on Biocomputing*, 2010.

[54] Wilhelm Huisinga, Christof Schütte, and Andrew M. Stuart. Extracting macroscopic stochastic dynamics: Model problems.

Communications on Pure and Applied Mathematics, LVI:0234–0269, 2003.

[55] V. M. Ingram. Gene mutations in human haemoglobin: the chemical difference between normal and sickle cell haemoglobin. *Nature*, 180:326–328, 1957.

[56] Sophie E. Jackson. How do small single-domain proteins fold? *Folding and Design*, 3:R81–R91, 1998.

[57] Guha Jayachandran, Michael R. Shirts, Sanghyun Park, and Vijay S. Pande. Parallelized-over-parts computation of absolute binding free energy with docking and molecular dynamics. *J. Chem. Phys.*, 125:084901, 2006.

[58] I.T. Jolliffe. *Principal Component Analysis*. 2nd edition, 2002.

[59] Lydia E. Kavraki, Peter Svestka, Jean-Claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. on Robotics & Automation*, 12(4):566–580, 1996.

[60] Bettina Keller, Philippe Hunenberger, and Wilfred F. van Gunsteren. An analysis of the validity of markov state models for emulating the dynamics of classical molecular systems and ensembles. *Journal of Chemical Theory and Computation*, 7:1032–1044, 2011.

[61] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. 2009.

[62] A. Kryshtafovych, C. Venclovas, K. Fidelis, and J. Moult. Progress over the first decade of casp experiments. *Proteins*, 61:225–36, 2005.

[63] Jan Kubelka, William A. Eaton, and James Hofrichter. Experimental tests of villin subdomain folding simulations. *J. Mol. Biol.*, 329:625–630, 2003.

[64] Jan Kubelka, James Hofrichter, and William A. Eaton. The protein folding speed limit. *Current Opinion in Structural Biology*, 14:7688, 2004.

[65] Jean-Claude Latombe. *Robot Motion Planning*. 1990.

[66] Andrew R. Leach. *Molecular Modelling: Principles and Applications*. Prentice Hall, 2nd edition, 2001.

[67] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2009.

[68] Cyrus Levinthal. Are there pathways for protein folding? *Extrait du Journal de Chimie Physique*, 65:44–45, 1968.

[69] Michael Levitt. Real-time interactive frequency filtering of molecular dynamics trajectories. *Journal of Molecular Biology*, 220:1–4, 1991.

[70] Michael Levitt, Christian Sander, and Peter S. Stern. Protein normal-mode dynamics: Trypsin inhibitor, crambin, ribonuclease and lysozyme. *J. Mol. Biol.*, 181:423–447, 1985.

[71] Harvey Lodish, Arnold Berk, Chris A. Kaiser, Monty Krieger, Matthew P. Scott, Anthony Bretscher, Hidde Ploegh, and Paul Matsudaira. *Molecular Cell Biology*. 6th edition, 2007.

[72] Stephane Mallat. *A Wavelet Tour of Signal Processing*. 3rd edition, 2008.

[73] Anthony Mittermaier and Lewis E. Kay. New tools provide new insights in NMR studies of protein dynamics. *Science*, 312:224–228, 2006.

[74] Kevin Murphy. Applying the junction tree algorithm to variable-length DBNs. Technical report, UC Berkeley, 2001.

[75] Kevin Murphy. The factored frontier algorithm for approximate inference in DBNs. In *Uncertainty in AI*, 2001.

[76] Kevin Murphy and Mark Paskin. Linear time inference in hierarchical HMMs. In *Neural Info. Proc. Systems*, 2001.

[77] Robert Murray, Victor Rodwell, David Bender, Kathleen M. Botham, P. Anthony Weil, and Peter J. Kennelly. *Harper's Illustrated Biochemistry*. 28th edition, 2009.

[78] Bengt Nölting. *Protein Folding Kinetics: Biophysical Methods*. 2nd edition, 2005.

[79] Bengt Nölting. *Methods in Modern Biophysics*. 3rd edition, 2009.

[80] A.V. Oppenheim and R.W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 3rd edition, 2009.

[81] S. Banu Ozkan, Ken A. Dill, and Ivet Bahar. Fast-folding protein kinetics, hidden intermediates, and the sequential stabilization model. *Protein Science*, 11:1958–1970, 2002.

[82] Stephen C. Phillips, Jonathan W. Essex, and Colin M. Edge. Digitally filtered molecular dynamics: The frequency specific control of molecular dynamics simulations. *Journal of Chemical Physics*, 112:2586–2597, 2000.

[83] Ursula Pieper, Narayanan Eswar, Ben M. Webb, David Eramian, Libusha Kelly, David T. Barkan, Hannah Carter, Parminder Mankoo, Rachel Karchin, Marc A. Marti-Renom, Fred P. Davis, and Andrej Sali. Modbase, a database of annotated comparative protein structure models and associated resources. *Nucleic Acids Research*, 37:D347–D354, 2009.

[84] Erion Plaku and Lydia E. Kavraki. Nonlinear dimensionality reduction using approximate nearest neighbors. In *SIAM Inter. Conf. on Data Mining*, 2007.

[85] Stanley B. Prusiner. Prions. *Proc. Natl. Acad. Sci. U. S. A.*, 95:13363–13383, 1998.

[86] D. C. Rapaport. *The Art of Molecular Dynamics Simulation*. 2004.

[87] Barak Raveh, Angela Enosh, Ora Schueler-Furman, and Dan Halperin. Rapid sampling of molecular motions with prior information constraints. *PLoS Comput. Biol.*, 5:e1000295, 2009.

[88] R. Riek, S. Hornemann, G. Wider, M. Billeter, R. Glockshuber, and K. Wuthrich. NMR structure of the mouse prion protein domain prp (121-131). *Nature*, 382:180–182, 1996.

[89] R. Riek, G. Wider, M. Billeter, S. Hornemann, R. Glockshuber, and K. Wuthrich. Prion protein NMR structure and familial human spongiform encephalopathies. *Proc. Natl. Acad. Sci. U. S. A.*, 95:11667–11672, 1998.

[90] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[91] Gordon S. Rule and T. Kevin Hitchens. *Fundamentals of Protein NMR Spectroscopy.* 2005.

[92] Dennis J. Selkoe. Folding proteins in fatal ways. *Nature*, 426:900–904, 2003.

[93] Richard B. Sessions, Pnina Dauber-Osguthorpe, and David J. Osguthorpe. Filtering molecular dynamics trajectories to reveal low-frequency collective motions: Phospholipase A_2. *J. Mol. Biol.*, 209:617–633, 1988.

[94] David E. Shaw, Paul Maragakis, Kresten Lindorff-Larsen, Stefano Piana, Ron O. Dror, Michael P. Eastwood, Joseph A. Bank, John M. Jumper, John K. Salmon, Yibing Shan, and Willy Wriggers. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330:341–346, 2010.

[95] Amarda Shehu, Lydia E. Kavraki, and Cecilia Clementi. Multiscale characterization of protein conformational ensembles. *Proteins*, 76:837–851, 2009.

[96] A.P. Singh, J.C. Latombe, and D.L. Brutlag. A motion planning approach to flexible ligand binding. In *Proc. Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, pages 252–261, 1999.

[97] Nina Singhal, Christopher D. Snow, and Vijay S. Pande. 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(1):415–425, 2004.

[98] Paul E. Smith, B. Montgomery Pettitt, and Martin Karplus. Stochastic dynamics simulations of the alanine dipeptide using

a solvent-modified potential energy surface. *Journal of Physical Chemistry*, 97:6907–6913, 1993.

[99] Remco Sprangers, Anna Gribun, Peter M. Hwang, Walid A. Houry, and Lewis E. Kay. Quantitative NMR spectroscopy of supramolecular complexes: Dynamic side pores in ClpP are important for product release. *Proc. Natl. Acad. Sci. U. S. A.*, 102:16678–16683, 2005.

[100] William C. Swope, Jed W. Pitera, and Frank Suits. Describing protein folding kinetics by molecular dynamics simulations. 1. theory. *J. Phys. Chem. B*, 108:6571–6581, 2004.

[101] William C. Swope, Jed W. Pitera, Frank Suits, Mike Pitman, Maria Eleftheriou, Blake G. Fitch, Robert S. Germain, Aleksandr Rayshubski, T. J. C. Ward, Yuriy Zhestkov, and Ruhong Zhou. Describing protein folding kinetics by molecular dynamics simulations. 2. example applications to alanine dipeptide and a β-hairpin peptide. *J. Phys. Chem. B*, 108:6582–6594, 2004.

[102] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. 2005.

[103] Howard M. Taylor and Samuel Karlin. *An Introduction to Stochastic Modeling*. Academic Press, 3 edition, 1998.

[104] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[105] Miguel L. Teodoro, George N. Phillips Jr., and Lydia E. Kavraki. A dimensionality reduction approach to modeling protein flexibility.

In *Proc. ACM Int. Conf. on Computational Molecular Biology (RECOMB)*, pages 299–308, 2002.

[106] Miguel L. Teodoro, George N. Phillips Jr., and Lydia E. Kavraki. Understanding protein flexibility through dimensionality reduction. *Journal of Computational Biology*, 10:617–634, 2003.

[107] Georgios Theocharous, Khashayar Rohanimanesh, and Sridhar Mahadevan. Learning hierarchical partially observable markov decision process models for robot navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2001.

[108] Shawna Thomas, Xinyu Tang, Lydia Tapia, and Nancy M. Amato. Simulating protein motions with rigidity analysis. *J. Comput. Biol.*, 14(6):839–855, 2007.

[109] Monique M. Tirion. Large amplitude elastic motions in proteins from a single-parameter, atomic analysis. *Physical Review Letters*, 77:1905–1908, 1996.

[110] Valda J. Vinson. Proteins in motion. *Science*, 324:197, 2009.

[111] Minghui Wang, Yuefeng Tang, Satoshi Sato, Liliya Vugmeyster, C. James McKnight, and Daniel P. Raleigh. Dynamic NMR line-shape analysis demonstrates that the villin headpiece subdomain folds on the microsecond time scale. *J. Am. Chem. Soc.*, 125(20):6032–6033, 2003.

[112] James D. Watson and Francis Crick. Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, 171:737–738, 1953.

[113] Yang Zhang and Jeffrey Skolnick. The protein structure prediction problem could be solved using the current PDB library. *Proc. Natl. Acad. Sci. U. S. A.*, 102:1029–1034, 2005.

[114] Wenjun Zheng. A unification of the elastic network model and the gaussian network model for optimal description of protein conformational motions and fluctuations. *Biophys. J.*, 94:3853–3857, 2008.