STATISTICAL MODELS OF PROTEIN CONFORMATIONAL DYNAMICS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF DEPARTMENT OF CHEMISTRY
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Robert T. McGibbon
March 2016

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Vijay Pande, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Thomas Markland

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Todd Martinez

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

Understanding the conformational dynamics of biological macromolecules at atomic resolution remains a grand challenge at the intersection of biology, chemistry, and physics. Molecular dynamics (MD) — which refers to computational simulations of the atomic-level interactions and equations of motions that give rise to these dynamics — is a powerful approach that now produces immense quantities of time series data on the dynamics of these systems. Here, I describe a variety of new methodologies for analyzing the rare events in these MD data sets in an automatic, statically-sound manner, and constructing the appropriate simplified models of these processes. These techniques are rooted in the theory of reversible Markov chains. They include new classes of Markov state models, hidden Markov models, and reaction coordinate finding algorithms, with applications to protein folding and conformational change. A particular focus herein is on methods for model selection and model comparison, and computationally efficient algorithms.

Acknowledgments

This dissertation represents the culmination of five years of research at Stanford carried out in pursuit of a Ph.D. I feel incredibly fortunate for having had the opportunity to spend this time in such an intellectually vibrant environment, surrounded by brilliant, curious, and motivated peers. My journey to Stanford began with my parents, Rebecca Treiman and Charles McGibbon. Growing up with two professors as parents — not to mention two professors as grandparents as well — long before I considered the notion analytically, I had internalized the idea that scientific discovery was the natural vocation of the intellectually engaged. I am immensely grateful for all their love and support, and for that early lesson. My time here at Stanford has been one of the happiest of my life to date.

No acknowledgments can express the depth of my gratitude towards Vijay Pande, my advisor and mentor. In his group at Stanford, Vijay has created an ideal environment for science, by trusting, inspiring, and motiving his students to formulate pursue their own questions. Not only has he given me important insights and suggestions at the appropriate junctures, but also nurtured the intellectual self-confidence that enabled me to discover those insights myself. I will always remember when Vijay nominated me — against my initial wishes — to give a talk in his place at an important conference in Germany; he trusted me more than I did myself.

I would like to thank all my friends and colleagues in the Pande lab who shared the day to day triumphs and failures of research with me, particularly Christian R. Schwantes, Kyle A. Beauchamp, Thomas J. Lane, Matthew P. Harrigan, Lee-Ping Wang, and Bharath Ramsundar. All of the best ideas described in this dissertation were arrived at collaboratively; all of the worst are mine alone. Outside of the

Pande Lab, Profs. John D. Chodera, Frank Noé, and Xuhui Huang have been critical colleagues.

Much of my work involved software development in collaboration with others at Stanford and around the world. It has been a pleasure to work with and learn the craft of scientific software engineering from Stanford staff including Peter K. Eastman and Yutong Zhao, Stanford students including Kyle A. Beauchamp, Matthew P. Harrigan, and far-flung collaborators including John Chodera (MKSCC, NY), Jason Swails (Rutgers, NJ), Christoph Klein (Vanderbilt, TN), and Martin Scherer (FU Berlin, Germany).

Contents

Abstract **iv**

Acknowledgments **v**

1 Introduction **1**

1.1 Motivation 1

1.2 Theory 7

1.3 Outline 18

2 Learning Kinetic Distance Metrics for Markov State Models of Protein Conformational Dynamics **20**

2.1 Introduction 21

2.2 Methods 22

2.3 Results and discussion 27

2.4 Conclusions 33

3 Understanding Protein Dynamics With L_1-Regularized Reversible Hidden Markov Models **35**

3.1 Introduction 36

3.2 Prior work 38

3.3 Fusion L_1-regularized reversible HMM 39

3.4 Experiments 47

3.5 Discussion and conclusion 53

4 Efficient maximum likelihood parameterization of continuous-time Markov processes **55**
4.1 Introduction . . . 56
4.2 Background . . . 57
4.3 Maximum likelihood estimation . . . 59
4.4 Quantifying uncertainty . . . 66
4.5 Numerical experiments . . . 68
4.6 Conclusions . . . 81

5 Statistical Model Selection for Markov Models **83**
5.1 Introduction . . . 84
5.2 Theory . . . 86
5.3 Methods . . . 91
5.4 Results and discussion . . . 95
5.5 Conclusions . . . 100

6 Variational cross-validation of slow dynamical modes in molecular kinetics **101**
6.1 Introduction . . . 102
6.2 Cross validation . . . 103
6.3 Theory background . . . 106
6.4 Objective function and subspace variational principle . . . 108
6.5 Algorithmic realization . . . 114
6.6 Simulations . . . 116
6.7 Discussion . . . 119
6.8 Conclusions . . . 123
6.A Proofs . . . 124
6.B A tension between the spectral and probabilistic approaches . . . 127
6.C Double well potential integrator and eigenfunctions . . . 129
6.D Landmark UPGMA clustering . . . 131
6.E Simulation Setup . . . 131

7 Identification of simple reaction coordinates from complex dynamics 133
7.1 Introduction 134
7.2 Defining the natural reaction coordinate 135
7.3 A dominant eigenfunction is the natural reaction coordinate 143
7.4 The tICA approximator 150
7.5 A sparse approximator for the dominant eigenfunction 152
7.6 An ADMM solver for the QCQP subproblem 155
7.7 Examples 159
7.8 Conclusions 164
7.A Covariance matrix estimation 165
7.B Projection of point onto an ellipsoid 167
7.C Runtime performance 170

8 MDTraj: a modern, open library for the analysis of molecular dynamics trajectories 171
8.1 Introduction 172
8.2 Capabilities and implementation 173
8.3 Results and discussion 177
8.4 Conclusions 179

Bibliography 180

List of Figures

1.1 Spatial and temporal scales of protein conformational dynamics. . . . 2
1.2 Comparison of predicted and measured protein folding times. 5
1.3 An example two-state Markov model for a hypothetical protein 8

2.1 Voronoi decomposition using the KDML method. 28
2.2 Longest model implied time scales from the KDML method. 29
2.3 Comparison of the KDML kinetic model with an existing model. . . . 30
2.4 Macrostate involvement in the KDML slow dynamical processes. . . . 31
2.5 Representative structure from the native and THR-flip macrostates. . 32
2.6 The flip of ASP15 deliniates a kinetically distinct near native state. . 32

3.1 L_1-HMM analysis of Brownian dynamics simulations. 46
3.2 Ubiquitin dynamics analyzed with an L_1 HMM 48
3.3 Activation of the c-Src Kinase. 52

4.1 A simple eight state Markov process. 69
4.2 Convergence of the estimated rate matrix. 69
4.3 Comparison of the estimated and true off-diagonal rate matrix elements. 70
4.4 Brownian dynamics on the 2-dimensional Müller potential. 72
4.5 Quantile-quantile plot of the approximated asymptotic variances. . . 72
4.6 Violin plots of the relative error between continuous-time and discrete-time Markov models. 74
4.7 The FiP35 WW protein, in its native state. 76
4.8 Comparison of implied exponential relaxation timescales. 77

4.9 The maximum likelihood rate matrix, $\hat{\mathbf{K}}$. 77
4.10 Convergence of selected rate matrix elements. 78
4.11 Performance of our Markov process estimator. 81

5.1 Systems studied in this chapter. 93
5.2 Comparison of four model selection criteria for Müller potential. . . . 96
5.3 Comparison of model selection criteria for Fip35WW domain with tICA. 98
5.4 Comparison of model selection criteria for Fip35WW domain with PCA. 99

6.1 Model selection for MSMs of a double well potential. 117
6.2 Comparison of 8 methods for building MSMs under 5-fold cross validation, evaluated using the rank-6 GMRQ. 120

7.1 Predictions made by the natural reaction coordinate for Smoluchowski diffusion on two-well potential. 139
7.2 An example two-dimensional potential energy surface with two pathways and a natural reaction coordinate. 146
7.3 The log-norm regularizer used in this chapter. 153
7.4 A 2-fluorobiphenyl derivative simulated in this chapter. 159
7.5 tICA and sparse tICA results for simulations of a 2-fluorobiphenyl derivative. 160
7.6 Probability density function of the von Mises distribution. 161
7.7 The ARG 42 ϕ angle over the course of a 1 ms native state simulation of BPTI. 162
7.8 The near-native and ARG42-flipped conformations of BPTI. 163
7.9 Comparison of the runtime of a specialized QCQP solver. 170

8.1 List of MDTraj-supported file formats 173
8.2 Atom selection language. Lines 2 and 3 are equivalent, although the latter text-based syntax may be more natural for users. 175
8.3 MDTraj's interactive WebGL-based protein and trajectory 176
8.4 Demonstration of principal components analysis (PCA) with MDTraj, `scikit-learn` and `matplotlib`. 177

8.5 Demonstration of solvent-accessible surface area calculation done in parallel with MDTraj and IPython. 178

Chapter 1

Introduction

1.1 Motivation

Proteins are the molecular machines of the cell, responsible for virtually all of the structural and functional tasks essential to life. They comprise the cellular components that, among many other tasks, metabolize nutrients, sense the outside world, regulate genetics, and recognize pathogens. This functional versatility may appear remarkable when considering that proteins are linear polymers composed of just twenty independent building blocks; their functionality is due to exquisite spatial and temporal structuring.

Over the past sixty years, breakthroughs in structural biology have yielded atomic-resolution structural models for many thousands of life's essential proteins. This history began with pioneering work of John Kendrew who first glimpsed the three-dimensional structure of the oxygen-binding protein myoglobin using techniques developed by Max Perutz — work for which the pair shared the 1962 Nobel Prize in chemistry. It continues today with the over 9,000 structural models added to the world-wide protein data bank per year.

Although these static structural models are immensely valuable, they are incomplete; proteins at physiological temperatures are, in fact, highly dynamic, and these motions are essential to the biological functions they play. No machine can operate frozen in place.

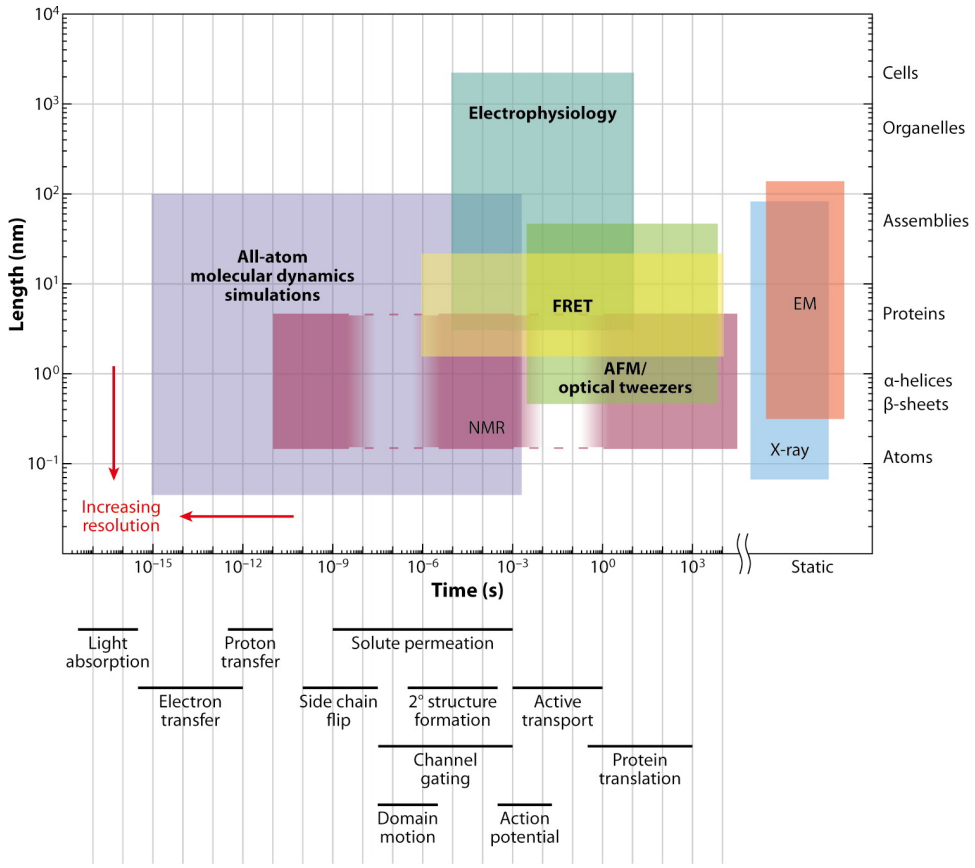


Figure 1.1: Spatial and temporal scales of protein conformational dynamics. The temporal and spatial resolution of the different techniques are indicated with colored boxes, methods capable of yielding information about single molecules, as opposed to ensembles, are indicated in boldface, and the characteristic timescales of certain molecular processes are shown below the time axis. This figure is reproduced with permission from Dror et al.[1]

Protein dynamics span an enormous range of timescales, from the femtosecond atomic-level rearrangements following the photoexcitation of a chromophore to large-scale structural changes that involve complex rearrangements of many interconnected components such as protein folding over seconds or more.[2] A panoply of experimental methods have been developed to study these dynamics, including various nuclear magnetic resonance and electron paramagnetic resonance techniques, fluorescence resonance energy transfer and infrared spectroscopies, and force-based methods based on atomic force microscopy and optical tweezers.[3,4] Generally, however, these techniques experience a tradeoff between spatial and temporal resolutions — the most structurally-detailed methods, such as X-ray crystallography, lack temporal information,[5] while methods with the highest time resolution often resolve only a low-dimensional proxy for structure.[6,7] Detailed experimental characterization of the dynamical processes, intermediate conformations and transition pathways in macromolecular systems remains exceptionally challenging.[8,9]

The fundamental physical laws governing protein dynamics, quantum mechanics and the time-dependent Schrödinger equation, have been known for some ninety years. An attractive alternative to these difficult experiments, then, is to computationally simulate the dynamics of proteins *ab initio*. Such simulations would intrinsically possess both high spatial and temporal resolution. Generally, however, the computational complexity of quantum mechanical calculations for systems as large as biological macromolecules has ruled out their direct application for the study of protein dynamics.* Instead, efforts in the field have focused on a technique — usually referred to as molecular dynamics (MD) — that represents the atomic nuclei explicitly, but the electrons only implicitly, and models the system with an empirically derived and computationally efficient potential energy function (called a "force field"). The nuclear dynamics are then simulated by discretizing Hamilton's equations of motions, usually with a time step on the order of 1 femtosecond.

Since the first protein MD simulation of bovine pancreatic trypsin inhibitor (BPTI) was reported 1977, the field has advanced principally along two fronts.[12] The first is

*Note that because of the recent development of GPU-accelerated quantum chemistry software, quantum mechanical simulations of protein dynamics are beginning to make a significant impact in the field.[10,11]

the development of more accurate protein and solvent force fields, which are parameterized by adjusting coefficients to fit measured experimental and quantum chemical data.[13–16] But even with the most accurate force field, complex collective motions like protein folding and conformational change can take milliseconds of simulation time or longer to occur. When simulated using a protocol whose individual timesteps are on the order of 1 femtosecond, the resulting calculations require simulations of length at least 10^{12} steps in order to observe one such event. And these events are stochastic; statistically robust results require even more simulation time. These calculations are thus extremely computationally demanding. The second front in the maturation of MD, therefore, has been the development of faster software, hardware, and computing platforms for performing these simulations. Indeed, while the first BPTI simulation only covered 9.2 ps of simulation time, modern simulations often have equilibration periods (from which the data is completely discarded) many times longer than that, and reach millisecond-scale production periods.

Although algorithmic innovations have been critical for the speedup in MD simulations — any work would be remiss in failing to acknowledge the contributions of Darden and coworkers in the development of efficient methods for the treatment of long-range electrostatic interactions[17] — innovations in computing hardware have played the primary role in the speedup of MD; Moore's law and the exponential year-over-year improvements in commodity CPUs play a crucial role here.[18] But two additional trends are worth highlighting: the development of specialized hardware and distributed computing projects for MD.

In 2008, D.E. Shaw Research reported the development of Anton, a special-purpose massively parallel machine for MD simulations.[20] Unlike traditional computers that are fully programmable and can be used to perform any task from physics-based calculations to image rendering or web surfing, Anton was built from application-specific integrated circuits (ASICs). These chips featured specialized data paths and control logic specifically designed for computing the energies and forces for MD, connected together with specialized network elements to reduce latencies. While not the first project to attempt specialized hardware for MD,[21–23] Anton was the first to achieve tens (and later with Anton 2, hundreds) of microseconds of simulation time

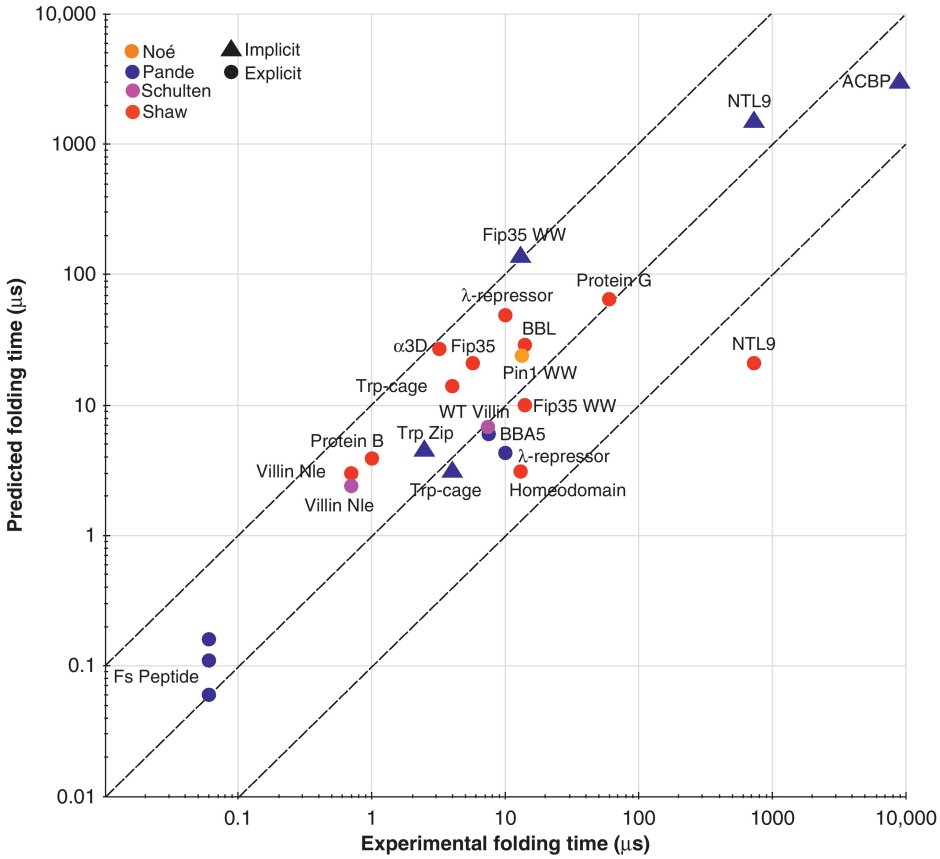


Figure 1.2: Comparison of protein folding timescales as measured experimentally and predicted by all-atom molecular dynamics simulations. The laboratories which performed the simulations are indicated by the point colors. Simulations done using an explicit molecular representation of the solvent are shown with circles, and those using implicit solvation methods are shown with triangles. The progression towards simulations of slower-folding proteins indicates the advance of molecular dynamics methodologies. This figure is reproduced with permission from Lane et al.[19]

per day on biologically-relevant all-atom MD simulations.[24]

Instead of simply running a *single* simulation faster on a specialized machine, a number of efforts have harnessed the collective power of hundrends of thousands of processors volunteered for science by private citizens across the globe. These distributed computers can be used to run many thousands of independent MD trajectories crisscrossing the protein energy landscape that can subsequently be analyzed in unison. This effort was begun by the Folding@Home project,[25] and later replicated by GPUGRID.[26] Both of these distributed computing projects, in addition to traditional high-performance computing and desktop workstation platforms, have benefitted dramatically from the rise in general purpose computing on graphical processing units (GPUs). Once exclusively for visualization, GPUs have become fully programmable, massively parallel co-processors that are often an order or magnitude or more faster than CPUs for scientific calculations.[27] The net result has been a steady increase in the size, complexity, and timescales accessible with MD simulation.

1.1.1 Analysis of MD simulations

As the hardware and software for performing MD simulations has matured, the resulting datasets — collections of trajectories tracking the Cartesian positions (and perhaps momenta) of every atom in a system — have become larger and larger. Increasingly, the quantitative analysis of these dataset has become a limiting factor in the application of MD.[19,28] It is this analysis that is the focus of this work.

The direct visualization of raw MD trajectories is neither scalable nor quantitative. Traditionally, researchers have simplified the analysis problem with physically motivated projections into one- and two-dimensional spaces, where the interpretation of MD simulations is more straightforward. For example, for protein folding simulations one can posit a reaction coordinate such as the fraction of native contacts, Q, and then analyze each frame from the simulation only on the basis of this scalar.[29] This simplification aids in the analysis very much, and while there is good reason to assume that Q may be a suitable collective variable to describe the folding process,[30] it seems clear that the full information content of an MD dataset is not being used

in any analysis restricted to a small number of pre-enumerated, physically motivated scalar projections.

While the complexity can be daunting, we must approach it head on. Although raw MD simulations may have been viewed in the past as a scientific result, I argue now that they are best viewed as an intermediate product. An additional set of models are required to turn these data into something approaching knowledge. These data-driven statistical models for protein conformational dynamics should, in my view, have the following properties:

- The models should be oriented toward quantitatively describing the long-timescale processes in the underlying physical system.
- The models should be suitably complex, capable of smoothly adapting to — rather than assuming — the structure of the data.
- The models should be interpretable and provide answers to specific scientific questions that may be difficult to answer via experiment alone.

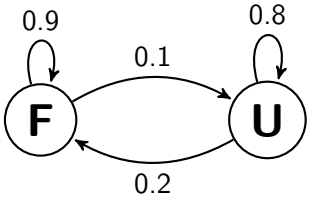
In this thesis, I describe the development of new classes of statistical models for these purposes. Before outlining my contributions, I provide some details on the theoretical framework that underlies my work.

1.2 Theory

Aspects of this section have been adapted with permission from Schwantes, C. R.,[†] McGibbon, R. T.[†] and Pande, V. S., Perspective: Markov models for long-timescale biomolecular dynamics. *J. Chem. Phys.*, 2015, 141, pp 090901.[31] Copyright 2014 American Institute of Physics.

The purpose of this section is to give some background on the mathematical tools and approach used herein. Beyond as discussed above, I take most of the background on molecular dynamics itself as a given; readers who are unfamiliar with the technique and its applications in biological physics are referred to one of the excellent books or

[†]C. R. Schwantes and R. T. McGibbon contributed equally to this work.



$$\mathbf{T} = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$$

Figure 1.3: An example two-state Markov model for a hypothetical protein. The network diagram (left) shows the two states, with arrows indicating the probabilities of the corresponding transitions. The matrix representation (right) collects the transition probabilities, with $\mathbf{T}_{ij} = P(X_{t+1} = j | X_t = i)$.

reviews that have been published on the subject.[32–35] Instead, I begin with Markov chains and Markov processes.

All common molecular dynamics simulations are Markov chains, and placing their analysis in this context gives us a wide set of conceptual and mathematical tools for their analysis.

1.2.1 The Markov Property

Markov chains and Markov processes, named after Russian mathematician Andrei Markov, are stochastic processes: collections of time-indexed random variables representing the stochastic evolution of some system. In a Markov chain, the index variable (here, time) is taken to be discrete, whereas in a Markov process the index variable is taken to be continuous. Both are characterized by the Markov property, which can be thought of as *memorylessness*. A stochastic process satisfies the Markov property if the conditional distribution of the future state of the process is always independent of its past history when conditioned on the present state. Next, we give some specific examples of different processes satisfying this property.

1.2.2 Discrete-state Markov Chain

A discrete-state Markov chain can be used to model the time evolution of a system whose state can be described by a single value in a discrete set. For an example relevant to the body of this work, consider a two-state model for the dynamics of

a protein whose conformation is measured as either "folded" or "unfolded" every millisecond. A simple model of such a discrete-state Markov chain is shown in Fig. 1.3. According to this model, from one millisecond to the next, there is a probability of 0.1 that the protein will change its state to unfolded if it is currently folded, and a probability of 0.2 that the protein will fold if it is unfolded. The self-transition probabilities of 0.9 and 0.8 are implied by conservation of probability — in every step the protein must either jump to the other state or remain in its current state. For such a Markov chain, $X_t = \{X_1, X_2, X_3, \ldots\}$, the Markov property can be written as

$$P(X_{t+1}|X_t = x_t, X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \ldots) = P(X_{t+1}|X_t = x_t). \tag{1.1}$$

That is, the conditional probability distribution of the future state, X_{t+1}, depends on the current state, but is not further informed by any additional information about the past history of the process. Expressed in a more pithy way, the process does not "remember where it came from".

In addition to thinking about the process from the perspective of a single instance of the system with particular stochastic jump probabilities, we will often also take a wider perspective and consider an ensemble of identical systems, distributed among states according some initial probability distribution, $\vec{p}_t$. For this example, $\vec{p}_t$ would be a two-dimensional vector, with entries summing to 1, containing the probability that at any given time, any particular member of the ensemble is in either of the two states,

$$[\vec{p}_t]_i = P(X_t = i). \tag{1.2}$$

From this perspective, the value of writing the transition probabilities in matrix form is very clear; the update of this distribution over time can be computed by a simple matrix multiplication,

$$\vec{p}_{t+1} = \vec{p}_t \, \mathbf{T}. \tag{1.3}$$

Furthermore, leveraging the Markov property this update rule can be applied

iteratively to give a description of the dynamics at longer times,

$$\vec{p}_{t+n} = \vec{p}_t \, \mathbf{T}^n. \tag{1.4}$$

We will see later how the eigendecomposition of $\mathbf{T}$ also gives a mathematically natural and physically interpretable view of the long timescale dynamics.

These types of models, and extensions thereof, are discussed in Chapter 2, Chapter 3, Chapter 5 and Chapter 6, where they are used as a way of simplifying and interpreting the results of atomic-resolution molecular simulations. The principle question is generally how the decomposition of a protein's conformation space into these discrete states should be accomplished. When equipped with such a state decomposition, these discrete-state Markov chain models for protein conformation dynamics are called Markov state models (MSMs).

1.2.3 Discrete-state Markov processes

The model described in Fig. 1.3 can be adapted in a number of ways while retaining the Markov property. Two modifications are obvious: first, moving from discrete time into continuous time, and second, moving from a discrete state space to a continuous state space.

Note that the model introduced above only described the evolution of the system at a discrete time interval, while physical systems are usually more natural described with time as a continuous variable. In additional to this physical rationale, the move to time as a continuous variable brings into play very useful mathematical tools based on differentiability.

The natural continuous-time generalization of the discrete-state Markov chain is a discrete-state Markov process. The process can be specified in terms of the transition *rates* — as opposed to transition jump probabilities — between states, defined by

$$k_{ij} = \lim_{\Delta t \to 0} P(X_{t+\Delta t} = j | X_t = i)/\Delta t, \qquad i \neq j. \tag{1.5}$$

For convenience, we also generally define $k_{ii} = -\sum_{j \neq i} k_{ij}$. Applying this approach

yields the continuous-time generalization of Eq. (1.3).

$$\frac{d\vec{p}(t)}{dt} = \vec{p}(t)\,\mathbf{K}. \tag{1.6}$$

In the ensemble limit, we can interpet $\vec{p}(t)$ as simply the fraction of the ensemble in each state — and this is simply a normalized concentration measure of the concentration of different chemical species. Therefore, this model is identical to the familiar models from chemical kinetics

$$\mathrm{U} \underset{k_b}{\overset{k_f}{\rightleftharpoons}} \mathrm{F} \tag{1.7}$$

$$\frac{[U]_t}{dt} = -k_f[U] + k_b[F] \tag{1.8}$$

$$\frac{[F]_t}{dt} = k_f[U] - k_b[F] \tag{1.9}$$

The formal solution to Eq. (1.6) is

$$p(t) = p(0)\,\exp(\mathbf{K}t). \tag{1.10}$$

From here, we can see that the continuous-time Markov processes can encode a family of disctete-time Markov chains with the special form $\mathbf{T}(\tau) = \exp(\mathbf{K}\tau)$. New statistical methods developed by the author for fitting Markov models of this type are discussed in Chapter 4 of this work.

1.2.4 Markov process in continuous state spaces

Finally, we complete our introductory discussion of Markov processes with a quick look at those that take place in a continuous state space like $\mathbb{R}^N$. A foundational instance of such a process is the Langevin equation. In fact, the models introduced

above are generally used in the context of this work as statistical methods for simplifying, approximating, and interpreting a Markov process in a continuous state space.

As an example Markov process relevant for molecular dynamics, consider the Langevin equation for a particle with mass m and viscous friction coefficient λ in one dimension:

$$m\frac{d^2x(t)}{dt^2} = -\lambda\frac{dx(t)}{dt} + R(t), \tag{1.11}$$

or, rewritten in terms of the velocity as

$$\frac{dv(t)}{dt} = -\frac{\lambda}{m}v(t) + \frac{1}{m}R(t), \tag{1.12}$$

where $R(t)$ is a Gaussian white noise random variable. That is, for all t, the probability distribution $R(t)$ is a standard normal distribution, and its correlation function is $\mathbb{E}[R(t)R(t')] = \delta(t-t')$. One can solve for $v(t)$ by the method of integrating factors, yielding

$$v(t) = v(0)e^{-\gamma t} + \frac{1}{m}\int_0^t e^{-(t-s)\gamma}\, R(s)\, ds, \tag{1.13}$$

which shows that Langevin equation satisfies the Markov property. The random Weiner integral is independent of v. The future state of the particle, $v(t)$, depends on the current state, $v(0)$, but not in any way on the past history of the system.

1.2.5 Reversibility

These Markov models are quite general, and can be used to describe many types of timeseries in different fields of science and engineering. In applying them to describe molecular systems that evolve at thermal equilibrium, one is well-advised to enforce certain know physical constraints. Chief among these is reversibility.

Reversibility, or detailed balance, is a property that comes from the fact that Newtonian dynamics are symmetric to time reversal; the laws are unchanged if time

advances in the positive or negative direction. Detailed balance is expected for molecular processes taking place in thermal equilibrium, and enforcing this property any statistical estimation of the parameters of a Markov chain will generally enhance the model quality.

The basic symmetry with respect to the direction of time for a discrete state Markov chain is that any sample path of length n, $x_0, x_1, \ldots, x_n$, and its time-reversed path, $x_n, x_{n-1}, \ldots, x_0$, are equally probable. Consider the two-step chain, where we have

$$P(X_{t-1} = i, X_t = j) = P(X_{t-1} = j, X_t = i) \tag{1.14}$$

$$P(X_{t-1} = i) \cdot P(X_t = j | X_{t-1} = i) = P(X_{t-1} = j) \cdot P(X_t = i | X_{t-1} = j), \tag{1.15}$$

$$\mu_i \mathbf{T}_{ij} = \mu_j \mathbf{T}_{ji} \tag{1.16}$$

Here, $\mu_i = P(X_t = i)$ is interpreted as the *equilibrium* probability of state i. Therefore we see immediately that the notion of reversibility introduces the physical principle of a unique equilibrium distribution into the models. Furthermore, Eq. (1.16) is very powerful statistically, as it significantly constrains the number of unique independent parameters that will need to be estimated to parameterize a model.

1.2.6 Transfer Operator Approach

Because our goal is often to statistically parameterize discrete-state Markov models in a way that makes them agree as closely as possible with an underlying processes in a continuous state space, it is necessary to build certain analytical and conceptual bridges between the types of models discussed above. How can we, for example, build a Markov chain in discrete state space that best approximates a continous-time Markov process in a continuous state space?

We accomplish this via an operator formalism which describes a discrete-time Markov process in a continuous state space. Indeed, at their root, the methods described herein all work to build some sort of numerical model of this operator. Here

we highlight some important details; we refer the interested reader to the excellent monograph by Schütte, Huisinga, and Deuflhard.[36]

Consider a time-homogeneous, reversible, ergodic Markov processes, $\mathbf{X}_t$, in phase space, Ω, which is reversible with respect to a positive stationary density $\mu(\mathbf{x})$. Each of these constraints is important, and physical. Time-homogeneity refers to the fact that the probabilities of moving between different states do not themselves change as a function of time — physically, the equations of motion and the Hamiltonian are time invariant. Ergodicicity is the property that there do not exist two or more regions of Ω that are dynamically disconnected. Any trajectory that initially starts in one region of phase space will eventually sample every other region. The reversibility constraint, which is a kind of symmetry on the transition probabilities, was discussed above.

At time t, an ensemble of such processes can be described by a probability distribution, $p_t(\mathbf{x})$, which can change as a function of time. The evolution of this probability density over a time interval τ can be described by the action of an operator, $\mathcal{T}(\tau)$, the transfer operator. Intuitively, we expect $p_t(\mathbf{x})$ to be the infinite-dimensional, functional generalization of the probability vector $\vec{p}$ described in the discrete-state case, and $\mathcal{T}(\tau)$ to be the operator generalization of the discrete-state transition probability matrix, $\mathbf{T}$.

For each time t, define the equilibrium-weighted probability density, $u_t(\mathbf{x}) = p_t(\mathbf{x})\mu(\mathbf{x})^{-1}$. Then the transfer operator, $\mathcal{T}(\tau)$ is defined as the operator that evolves $u_t(\mathbf{x})$ forward in time to $u_{t+\tau}(\mathbf{x})$,

$$u_{t+\tau}(\mathbf{x}) = \mathcal{T}(\tau) \circ u_t(\mathbf{x}) = \frac{1}{\mu(\mathbf{x})} \int_{\mathbf{y}\in\Omega} d\mathbf{y}\, P(\mathbf{X}_{t+\tau} = \mathbf{x} \,|\, \mathbf{X}_t \in \mathbf{y} + d\mathbf{y})\, \mu(\mathbf{y}) u_t(\mathbf{y}), \tag{1.17}$$

where the transition probability density, $P(\mathbf{X}_{t+\tau} = \mathbf{x} \,|\, \mathbf{X}_t \in \mathbf{y}+d\mathbf{y})$, is the probability that, if the process is currently in the neighborhood $d\mathbf{y}$ around the coordinate $\mathbf{y}$, it will advance to $\mathbf{x}$ in time τ.

The use of $\mathcal{T}(\tau)$ is somewhat of a mathematical "trick". A more intuitive operator can also be defined from the transition probability density, $P(\mathbf{X}_{t+\tau} = \mathbf{x} \,|\, \mathbf{X}_t \in \mathbf{y}+d\mathbf{y})$,

by considering simply how $p_t(\mathbf{x})$ evolves forward in time to $p_{t+1}(\mathbf{x})$,

$$p_{t+\tau}(\mathbf{x}) = \mathcal{P}(\tau) \circ p_t(\mathbf{x}) = \int_{\mathbf{y}\in\Omega} d\mathbf{y} g(\mathbf{x}), P(\mathbf{X}_{t+\tau} = \mathbf{x} \,|\, \mathbf{X}_t \in \mathbf{y} + d\mathbf{y})\, p_t(\mathbf{y}). \tag{1.18}$$

This operator is called the propagator. Both the transfer operator and propagator are self-adjoint with respect to different inner products. The property of an operator being self-adjoint is the conceptual analog of a matrix being symmetric. The general property is that for any two functions f and g, a self-adjoint operator, $\mathcal{Q}$, is one such that

$$\langle f | \mathcal{Q} \circ g \rangle = \langle Q \circ f | g \rangle \tag{1.19}$$

for some *inner product*, $\langle \cdot | \cdot \rangle$.

Both for the propagator and transfer operator however, we will need norms with a different measure, such as $\langle \cdot | \cdot \rangle_\mu = \int_\Omega d\mathbf{x}\ \mu(\mathbf{x}) f(x) g(\mathbf{x})$. For the transfer operator, expanding both $\langle f | \mathcal{T}(\tau) \circ g \rangle_\mu$ and $\langle \mathcal{T}(\tau) \circ f | g \rangle_\mu$ using the shorthand $p(\mathbf{x}, \mathbf{y}) = P(\mathbf{X}_{t+\tau} = \mathbf{x} \,|\, \mathbf{X}_t \in \mathbf{y} + d\mathbf{y})$ gives

$$\langle \mathcal{T}(\tau) \circ f | g \rangle_\mu = \left\langle \left(\frac{1}{\mu(\mathbf{y})} \int_\Omega d\mathbf{x}\ f(\mathbf{x}) \mu(\mathbf{x}) p(\mathbf{x}, \mathbf{y}) \right) \middle| g \right\rangle_\mu \tag{1.20}$$

$$= \int_{\Omega\times\Omega} d\mathbf{x} d\mathbf{y}\ f(\mathbf{x}) g(\mathbf{y}) \mu(\mathbf{x}) p(\mathbf{x}, \mathbf{y}) \tag{1.21}$$

$$\langle f | \mathcal{T}(\tau) \circ g \rangle_\mu = \left\langle f \middle| \left(\frac{1}{\mu(\mathbf{x})} \int_\Omega d\mathbf{y}\ g(\mathbf{y}) \mu(\mathbf{y}) p(\mathbf{y}, \mathbf{x}) \right) \right\rangle_\mu \tag{1.22}$$

$$= \int_{\Omega\times\Omega} d\mathbf{x} d\mathbf{y}\ f(\mathbf{x}) g(\mathbf{y}) \mu(\mathbf{y}) p(\mathbf{y}, \mathbf{x}). \tag{1.23}$$

We can see that the critical property that makes these two expressions equal is reversibility — the property that $\mu(\mathbf{x}) p(\mathbf{x}, \mathbf{y}) = \mu(\mathbf{y}) p(\mathbf{y}, \mathbf{x})$. For the propagator, $\mathcal{P}(\tau)$, the situation is slightly different because using the inner product, $\langle \cdot | \cdot \rangle_\mu$, we would lack the proper factors of μ to prove self-adjointness using detailed balance. Instead, the reader can verify that $\mathcal{P}(\tau)$ is self-adjoint with respect to the μ^{-1}-weighted inner product, $\langle f | g \rangle_{\mu^{-1}} = \int_\Omega d\mathbf{x}\ \frac{f(\mathbf{x}) g(\mathbf{x})}{\mu(\mathbf{x})}$.

Both operators have a countable set of eigenfunctions and associated eigenvalues.

$$\mathcal{P}(\tau) \circ \phi_i = \lambda_i \phi_i \tag{1.24}$$

$$\mathcal{T}(\tau) \circ \psi_i = \lambda_i \psi_i, \tag{1.25}$$

with identical eigenvalues. The eigenfunctions are related by $\psi_i(\mathbf{x}) = \phi_i(\mathbf{x})/\mu(\mathbf{x})$

Furthermore, these eigenfunctions are orthonormal with respect to one another (using the proper inner products).

$$\langle \phi_i | \phi_j \rangle_{\mu^{-1}} = \langle \phi_i | \psi_j \rangle = \langle \psi_i | \psi_j \rangle_{\mu} = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{otherwise} \end{cases} \tag{1.26}$$

These eigenfunction form a complete basis, meaning that any starting distribution u_t or p_t can be written as a linear combination of the basis functions, such as

$$u_t = \sum_i a_i \psi_i. \tag{1.27}$$

Multiplying both sides by $\mu(\mathbf{x})\psi_j(\mathbf{x})$ and integrating, we get

$$\int_{\Omega} \mu(\mathbf{x}) \psi_j(\mathbf{x}) u_t(\mathbf{x}) = \langle \psi_j | u_t \rangle_{\mu} = \sum_i a_i \langle \psi_j | \psi_i \rangle_{\mu} = a_j, \tag{1.28}$$

where all of the cross-terms drop because of the orthogonality. Thus, the coefficients in this basis-function expansion are simply the projection of u_t onto the basis function.

Because the action of $\mathcal{T}(\tau)$ on its eigenfunctions is simple, and the operator is linear, this expansion lets us write out an expression for $\mathcal{T}(\tau)$ acting on an arbitrary initial distribution,

$$u_{t+\tau}(\mathbf{x}) = [\mathcal{T}(\tau) \circ u_t](\mathbf{x}) = \sum_{i=1}^{\infty} \lambda_i \langle \psi_i, u_t \rangle_{\mu} \psi_i(\mathbf{x}). \tag{1.29}$$

Or, levering the Markov property, we can write out the same expression for longer

times as well,

$$u_{t+n\tau}(\mathbf{x}) = [\mathcal{T}(\tau)]^n \circ u_t(\mathbf{x}) = \sum_{i=1}^{\infty} \lambda_i^n \langle \psi_i, u_t \rangle_\mu \psi_i(\mathbf{x}). \tag{1.30}$$

The eigenvalues, λ_i, are real, and the eigenpairs $(\lambda_i, \psi_i(\mathbf{x}))$ can be taken to be sorted in decreasing order by eigenvalue. According to the Perron-Frobenius theorem, λ_1 is equal to one and corresponds to the stationary eigenfunction, $\psi_1(\mathbf{x}) = 1$. The remaining eigenvalues lie in $(-1, 1)$, and are associated with eigenfunctions which describe directions of collective motion in phase space — dynamical processe — along which the system relaxes towards equilibrium. Eq. (1.30) can therefore be rewritten as

$$u_{t+n\tau}(\mathbf{x}) = 1 + \sum_{i=2}^{\infty} \exp\left(-\frac{n\tau}{t_i}\right) \langle \psi_i, u_t \rangle_\mu \psi_i(\mathbf{x}), \tag{1.31}$$

where t_i are the characteristic relaxation timescales of each dynamical mode, defined in terms of the associated eigenvalues,

$$t_i = -\frac{\tau}{\ln \lambda_i}. \tag{1.32}$$

At long times (large n), the terms corresponding to large eigenvalues (long relaxation timescales) dominate the summation, as small-eigenvalue modes equilibrate and decay quickly. The goal for accurate numerical models of $\mathcal{T}(\tau)$ is thus to resolve these dominant eigenmodes.

A discrete-state Markov model can be derived from this continuous-state model by partitioning the state space, Ω. Let $S = \{s_i\}_{i=1}^N$ be a set of N non-overlapping subsets of Ω that partition the space, that is,

$$s_i \subseteq \Omega \quad \text{for all } i, \tag{1.33}$$

$$\cup_{i=1}^k s_i = \Omega, \tag{1.34}$$

$$s_i \cap s_j = \emptyset \quad \text{for all } i \neq j. \tag{1.35}$$

Then, we can build a discrete-state Markov model with transition matrix $\mathbf{T}$ from the continuous-state description $\mathcal{T}(\tau)$.

$$\begin{aligned}
\mathbf{T}_{ij} &= \mathrm{P}(\mathbf{X}_{t+\tau} \in s_j \,|\, \mathbf{X}_t \in s_i) && (1.36)\\
&= \frac{\mathrm{P}(\mathbf{X}_{t+\tau} \in s_j \text{ and } \mathbf{X}_t \in s_i)}{\mathrm{P}(\mathbf{X}_t \in s_i)} && (1.37)\\
&= \frac{\int_{\mathbf{x}\in\Omega} d\mathbf{x} \int_{\mathbf{y}\in\Omega} d\mathbf{y}\, \mathbf{1}_{s_i}(\mathbf{x})\mu(\mathbf{x})\mathrm{P}(\mathbf{x}\to\mathbf{y};\tau)\mathbf{1}_{s_j}(\mathbf{y})}{\int_{\mathbf{x}\in\Omega} d\mathbf{x}\, \mathbf{1}_{s_i}(\mathbf{x})\mu(\mathbf{x})} && (1.38)\\
&= \frac{\left\langle \mathbf{1}_{s_j}, \mathcal{T}(\tau)\circ \mathbf{1}_{s_i}\right\rangle_\mu}{\left\langle \mathbf{1}_{s_i}, \mathbf{1}_{s_i}\right\rangle_\mu}. && (1.39)
\end{aligned}$$

where

$$\mathbf{1}_{s_i}(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in s_i \\ 0 & \mathbf{x} \notin s_i. \end{cases} \qquad (1.40)$$

1.3 Outline

The fundamental approach I have taken is to use computational methods to build an approximate representation of the transfer operator, $\mathcal{T}(\tau)$, from molecular dynamics simulations in a way which is systematic, automated, and statistically rigorous. The remainder of this work details my efforts of the past five years advance statistical modeling of protein dynamics within this paradigm.

Chapter 2 describes a improved method for constructing Markov state models, which describe the kinetics using a discrete-state Markov chain, focusing on the determination of which conformation should belong to which state. Chapter 3 describes a new class of hidden Markov models which removes the need for this exact spatial partitioning between the states, and Chapter 4 introduces a new method for estimating Markov process models in which the time domain is treated continuously. In chapters Chapter 5 and Chapter 6, I discuss methods for model selection to choose between multiple plausible statistical models for the same MD dataset. Chapter 6

presents what I consider to be my most significant contribution, describing a new variational theorem on the approximation quality of Markov state models and showing how it can be used for model selection and optimization. Chapter 7, which is new, unpublished work, introduces a new type of estimator for simple, interpretable reaction coordinates that builds on these theoretical tools developed for Markov state models. In Chapter 8, I describe some of the open source software we have developed to enable the results described in the previous chapters.

Chapter 2

Learning Kinetic Distance Metrics for Markov State Models of Protein Conformational Dynamics

Statistical modeling of long timescale dynamics with Markov state models (MSMs) has been shown to be an effective strategy for building quantitative and qualitative insight into protein folding processes. Existing methodologies, however, rely on geometric clustering using distance metrics such as root mean square deviation (RMSD), assuming that geometric similarity provides an adequate basis for the kinetic partitioning of phase space. Here, inspired by advances in the machine learning community, we introduce a new approach for learning a distance metric explicitly constructed to model kinetic similarity. This approach enables the construction of models, especially in the regime of high anisotropy in the diffusion constant, with fewer states than was previously possible. Application of this technique to the analysis of two ultralong molecular dynamics simulations of the FiP35 WW domain identifies discrete near-native relaxation dynamics in the millisecond regime that were not resolved in previous analyses.

This chapter is adapted with permission from McGibbon, R. T. and Pande, V. S., Learning Kinetic Distance Metrics for Markov State Models of Protein Conformational Dynamics. *J. Chem. Theory Comput.*, **2013**, *9(7)*, pp 2900–2906.[37]

2.1 Introduction

Molecular dynamics (MD) simulation is a powerful computational tool for probing complex molecular processes in atomistic detail, including vesicle fusion,[38] protein–ligand binding,[39] protein folding[40,41] and mis-folding,[42] conformational change,[34] and crystallization.[43] Advances in computing capabilities including commodity and specialized hardware,[20,44] faster simulation codes,[45–48] optimization for graphical processing units and game consoles,[49,50] and distributed and cloud computing frameworks[25,26] currently enable the collection of milliseconds of aggregate simulation data.

These systems are characterized by complex and high-dimensional conformational state spaces, with the possibility for multiple independent slow degrees of freedom. Without prior knowledge of the relevant conformational states, the objective and unbiased analysis of simulations of these systems is a significant challenge. Recently, Markov state models (MSMs) have emerged as an attractive, scalable approach to the analysis of these massive data sets.[51,52] Using clustering to divide the conformational space, MSMs model the system's dynamics as a memory-less jump process between discrete conformational states. This approach avoids the possibly deceptive projection of the system's dynamics onto a low-dimensional order parameter space,[53,54] and enables the systematic combination of multiple short simulations for the prediction of long-time scale phenomena.[52,55,56] Furthermore, the same set of theoretical and computational tools can be used to build detailed models for quantitative prediction and course-grained models for insight.[57–60] In addition, MSM methodologies facilitate efficient allocation of compute resources during the sampling phase by adaptive exploration of conformational space.[61–64]

Despite the advantages of MSMs, a number of challenges remain. In particular,

methods for discretizing the conformational space into a set of states using purely geometric criteria such as the Cartesian root-mean-square deviation (RMSD) suffer when there exist slow conformational transitions between geometrically proximate states, such as register shift dynamics in β topologies[59] or subtle but slow conformational change dynamics. The choice of distance metric with which to cluster conformations has recently received attention,[65,66] but a general framework for choosing the optimal distance metric for describing molecular kinetics is lacking. Here, we present a new approach. Motivated by recent work in semisupervised learning,[67,68] and specifically an algorithm by Shen et al.[69] we present a novel algorithm for kinetic distance metric learning which constructs metrics explicitly designed to enable kinetic clustering of conformations from MD data sets.

Whereas here we consider MSMs built directly from molecular dynamics simulations, other approaches exist as well for producing network models for the dynamics of biomolecular systems. These include approaches that proceed by direct enumeration of potential energy basins.[70–73] In such methods, an effective clustering of configurations on the potential energy landscape is accomplished by considering the set of local energy minima and transition rates estimated by unimolecular rate theory. While these methods fit naturally into a familiar chemical kinetics framework, they suffer from an exponential scaling in complexity with respect to the system's number of degrees of freedom.

2.2 Methods

MSM construction methodologies based on the clustering of MD trajectories typically involve the use of some geometric criterion such as RMSD, grouping the sampled conformations into a large number of microstates. In addition to the choice of distance metric, considerable attention has been focused on the choice of clustering algorithm.[59,74,75] On this discrete state space, the model is parametrized by estimating transition probabilities or rates between the microstates from the available simulation data. The resulting transition or rate matrix can then be analyzed by various means to find transition pathways between states of interest and kinetically

metastable macrostates.[57,58,76] The model can also be used to predict experimental observables such as those measured by temperature jump florescence and infrared spectroscopy[77,78] triplet-triplet energy transfer[79] and single molecule fluorescence resonance energy transfer (FRET).[80]

Two major sources of error characterize the MSM approach. The first is error introduced by the Markov approximation, the assumption that the future evolution of the system is independent of its history given its present state. Although Hamiltonian dynamics are Markovian in the full $6N$-dimensional space of the atomic positions and their conjugate momenta,[81] the dynamics between discrete conformational states are not, as the projection formally introduces a nontrivial memory kernel into the equations of motion.[82] The neglect of these memory effects in the MSM formalism is a source of systematic error. Although memory effects can be modeled by building higher-order or variable-order Markov models,[83,84] the number of parameters to be estimated increases exponentially with the order of the Markov chain, limiting the utility of this approach to systems with large or high-dimensional state spaces. Although this systematic error due to non-Markovity can be systematically reduced by increasing the number of microstates,[85] this bias reduction is countered by an increase in the second major source of error, statistical uncertainty in the estimation of the pairwise transition probabilities. As the microstates become smaller and more numerous, they become less likely to contain internal free energy barriers and more Markovian. Unfortunately, due to the quadratic scaling of the number of transition probabilities or rates that require estimation with respect to the number of states, these estimations become noisy and liable to overfitting if the number of states is increased without bound.

Other avenues for mitigating the systematic bias due to non-Markovian projected dynamics include methods such as the weighted ensemble (WE).[86–88] Similar to MSMs, the WE also utilizes a discrete partitioning of phase space to define reaction rates. In contrast to MSMs, which are generally applied as a postprocessing technique to analyze an ensemble of short-time trajectories, WE methodologies leverage the state decomposition during the sampling to stop and start trajectories as they explore phase space carrying different probabilistic weights. This need for state

definitions before the WE production simulation leads to a substantially increased computational cost.

To reduce the error due to non-Markovian memory effects within the MSM framework without increasing the size of the state space, we seek a kinetic clustering at the microstate level, such that conformations that can kinetically interconvert quickly are grouped together in contrast to conformations separated by longer time scales. For this purpose, we propose that a gold-standard distance metric for such a kinetic clustering would be the *commute time*, the mean first passage time for the round trip commute between infinitesimal voxels in phase space surrounding two structures. Such a metric would measure the mutual kinetic accessibility of two structures, such that low distances are associated with rapid interconversion, while maintaining the proper symmetry with respect to exchange. Unfortunately, with N frames of MD simulation data, the estimation of N^2 commute times between all pairs of sampled conformations is not achievable without performing a considerable amount of further simulation.

Instead, we introduce a new distance metric for clustering conformations obtained by MD simulation into microstates with the explicit inclusion of kinetic information. Adapting an algorithm from Shen et al. for distance metric learning in the large-margin machine learning framework,[69] we attempt to learn a Mahalanobis distance metric from molecular dynamics trajectories, which, operating on the structural features of a pair of conformations, is able to predict whether these conformations are kinetically close or kinetically distant. Using this approach, we find that it is possible build more Markovian MSMs with fewer states than is possible with existing methods and identify previously hidden slow conformational transitions.

2.2.1 Algorithm

Our goal is to learn a geometric distance metric that maximizes the margin of separation between kinetically close and kinetically distant conformations, which we call "kinetically discriminatory metric learning" (KDML). We take as input a collection of N triplets of structures, (a, b, c), such that conformations a and b are subsequent

frames from a single MD trajectory separated by a short lag time, τ_{close}, whereas structure c is selected from further down the trajectory, at a longer delay time, τ_{far}, from a. The structures must be projected into a suitable vector space such as the space of all heavy-atom heavy-atom pairwise distances or the space of backbone dihedral angles.

We then look for a distance metric that can distinguish the fact that in each triplet, conformations a and b are likely "kinetically close" whereas a and c are "kinetically distant". We choose to restrict our attention to squared Mahalanobis distance metrics, which generalize squared weighted euclidean metrics. The distance metric is an inner product.

$$d^{\mathbf{X}}(\vec{a},\vec{b}) = (\vec{a}-\vec{b})^T\mathbf{X}(\vec{a}-\vec{b}) \tag{2.1}$$

$$= (\vec{a}-\vec{b})^T\mathbf{Q}^T\mathbf{\Lambda}\mathbf{Q}(\vec{a}-\vec{b}) \tag{2.2}$$

$$= [\mathbf{Q}(\vec{a}-\vec{b}]^T\mathbf{\Lambda}[\mathbf{Q}(\vec{a}-\vec{b}] \tag{2.3}$$

The Mahalanobis matrix, $\mathbf{X}$, is required to be symmetrical and positive semidefinite, which guarantees that $d^{\mathbf{X}}$ is symmetric and that $d^{\mathbf{X}}(\vec{a},\vec{b}) \geq 0$. Two special cases of $\mathbf{X}$ are of interest. When $\mathbf{X}$ is diagonal, we have a squared weighted euclidean metric, and when $\mathbf{X} = \mathbf{I}$, the distance metric is simply the squared euclidean distance. In the general case, as demonstrated by Eq. (2.3), the squared Mahalanobis distance can be viewed as a squared weighted euclidean distance after projection onto the eigenbasis of $\mathbf{X}$.

Ideally, the data would permit a metric capable of correctly classifying all of the training examples, yielding a positive margin, ρ_i on each training example i, where $\rho_i = d^{\mathbf{X}}(\vec{a}_i,\vec{c}_i) - d^{\mathbf{X}}(\vec{a}_i,\vec{b}_i)$.

However, in general, it will not be possible to satisfy all of these constraints simultaneously. Instead, we seek a metric that will admit as large margin as possible on as many of the training examples as possible.

Following Shen,[69] we define our objective function f, which we seek to maximize with respect to $\mathbf{X}$ and ρ.

$$f(\mathbf{X}, \rho) = \alpha\rho - \frac{1}{N}\sum_{i=1}^{N} \lambda(d^{\mathbf{X}}(\vec{a}_i, \vec{c}_i) - d^{\mathbf{X}}(\vec{a}_i, \vec{b}_i) - \rho) \tag{2.4}$$

Here, ρ represents the "target" margin, and $\lambda(\cdot)$ is a smooth hinge loss function that penalizes margins on the training data less than the target margin ρ. The parameter α controls the balance between the desire to maximize the margin and to minimize the losses. We find $\mathbf{X}$ relatively insensitive to α for $0 < \alpha < 1$.

The objective function, f, is maximized subject to the constraints that $\mathrm{Tr}(\mathbf{X}) = 1$, $\rho \geq 0$, and $\mathbf{X}$ is positive semidefinite. The constraint on the trace of $\mathbf{X}$ removes the scale ambiguity of the Mahalanobis matrix. For efficiency with gradient based optimization techniques, we employ the Huber loss function $\lambda = \lambda_{\mathrm{Huber}}$, a differential extensions of the hinge loss.

$$\lambda_{\mathrm{Huber}}(\xi) = \begin{cases} 0 & \text{if } \xi \geq h \\ \frac{(h-\xi)^2}{4h^2} & \text{if } -h < \xi < h \\ -\xi & \text{if } \xi \leq -h \end{cases} \tag{2.5}$$

For the special case that $\mathbf{X}$ is diagonal, a change of variables allows the optimization to be performed without constraints. This yields an efficient solution via Broyden-Fletcher-Goldfarb-Shanno (BFGS) method. For the general $\mathbf{X}$, we employ a specialized gradient descent algorithm[69] that outperforms general purpose semidefinite programming solvers by taking advantage of the fact that a trace-one positive semidefinite matrix can be written as a convex linear combination of rank-one trace-one matrices[89] to construct gradient descent steps that naturally preserve the positive semidefiniteness.

The algorithms described above have been implemented as plugins for the MSMBuilder package (version 2.6 and later), which is released under the GNU General Public License and available at `https://github.com/msmbulder/msmbuilder`. This code itself is available at `https://github.com/pandegroup/KDML` and is released

under the same terms.

2.3 Results and discussion

2.3.1 Toy system

We first demonstrate the algorithm for a simple toy system: two-dimensional Brownian motion with an anisotropic diffusion constant. Despite its extreme simplicity, this model captures one essential feature of biomolecular conformational dynamics: the time scales corresponding to orthogonal degrees of freedom can be vastly different. An effective kinetic clustering of such systems requires that the characteristic length scale of the clusters in the directions with the slow diffusion be lower than those in the direction of fast diffusion. Such a clustering, with states that are elongated in the direction of fast motion, is statistically efficient in the sense that it uses no more states, and thus parameters, than are necessary to capture the intrastate dynamics.

For our toy system, we set the ratio of the diffusion constants in the x and y dimensions at 10:1. We apply the diagonal KDML learning algorithm described, training on $N = 4981$ triplets of structures sampled from 100 trajectories of length 500 steps. The time lag between "kinetically close" structures was taken to be $\tau_{\text{close}} = 1$ time step, with $\tau_{\text{far}} = 10$. The resulting Mahalanobis matrix is

$$\mathbf{X}_1 0 = \begin{pmatrix} 0.9915 & 0 \\ 0 & 0.0085 \end{pmatrix}.$$

The Mahalanobis matrix shows that in the kinetic distance metric, motion in the x direction is up-weighted, since motion along this degree of freedom is slower. When we use this kinetic distance metric to run clustering with the hybrid k-medoids algorithm, the clusters produced are geometrically elongated in the y direction. A comparison of the clusters produced using both kinetic metric as well as a standard euclidean distance is shown in Fig. 2.1.

In two dimensions, the effect of the distance metric on the state decomposition are obvious — the ratio of the weights on each degree of freedom are translated roughly

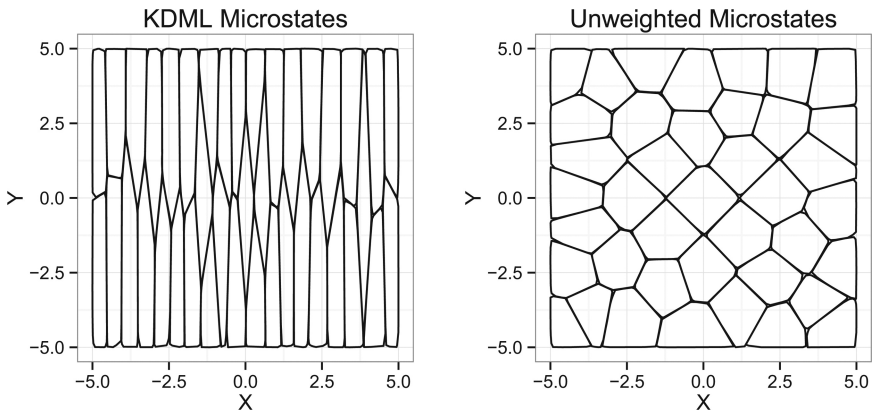


Figure 2.1: Voronoi decomposition of a two-dimensional space into 40 microstates using both the KDML and unweighted euclidean distance metric with the hybrid K-medoids clustering algorithm.(41) Using the kinetic distance metric, the microstates are geometric elongated in the direction of fast motion and are less likely to contain internal free energy barriers, leading to fewer non-Markovian memory effects without increasing statistical error.

into the geometrical aspect ratio of the clusters. Since the mean time to displace by one unit of distance in the x direction is longer than the mean time to displace by the same amount in the y direction, a successful clustering than minimizes the sum of the within-state mean first passage times is one that, similar to the clustering produced by the kinetic distance metric, is not geometrically spherical but instead is elongated in the y direction.

To benchmark the performance of this distance metric, we compute the longest two implied time scales for MSMs built with both an unweighted metric and the KDML metrics, with a variety of numbers of states. The implied time scales, which are computed from the eigenvalues of the transition matrix as $\tau_{\text{lag}}/\ln\lambda_i$ describe the time scales for characteristic dynamical modes of the kinetic network. If system is Markovian, the implied time scales are invariant to changes in the lag time; internal consistency demands that models built at a lag time of τ be equivalent to those built with a lag time of 2τ, provided that for every two steps propagated in the first model we take only one time step in the second model. In practice, non-Markovian behavior is generally manifest as erroneously fast time scales that increase with lag time.[90] As shown in Fig. 2.2, when compared to models built using the standard

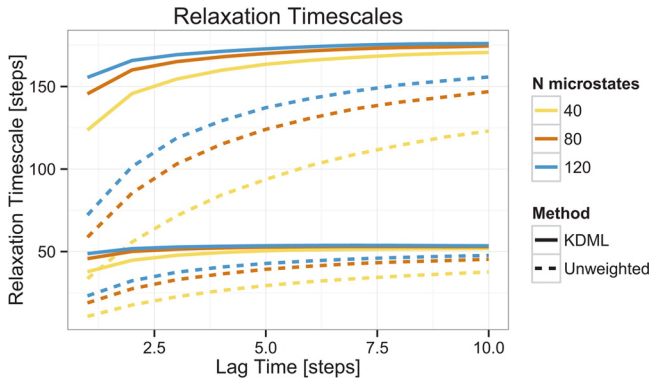


Figure 2.2: Longest two implied time scales for the MSMs produced by different procedures. Dashed curves are for models built using the standard euclidean distance metric, whereas solid lines are for those built using KDML. Models built using KDML are slower and are more Markovian compared to those built using the euclidean metric, even with fewer states.

distance metric, those built using the kinetic distance metric show longer time scales that converge more quickly with respect to lag time, indicating the quicker onset of Markovian behavior. Furthermore, the longest implied time scale for models built using the standard distance metric with 120 states shows behavior similar to that of models built with only 80 states using the kinetic metric. Because the number of fit parameters in the model goes as the square of the number of states, this corresponds to similar performance with less than half the number of adjustable parameters. In situations of comparative data sparsity, this enables a better balance between systematic and statistical error.

2.3.2 FiP35 WW

Next, we apply our approach to a real data set via a reanalysis of two one hundred microsecond simulations of the FiP35 WW domain, performed by D. E. Shaw et al.[91] A previous analysis of this data set using Markov State Model methods revealed the existence of two folding pathways.[77] However, in order to achieve a structural resolution in the microstate clustering of greater than 4.5 Å, it proved necessary in that study to use more than 26,000 microstates. Parameterizing this model required formally estimating more than 6.7×10^8 pairwise transition probabilities. Because of

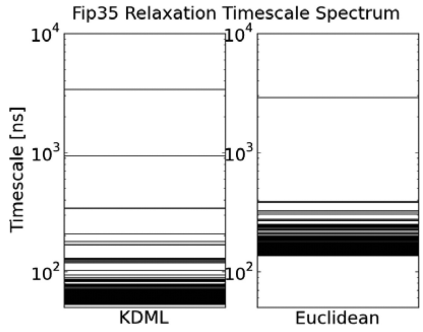


Figure 2.3: Two Markov state models for the FiP35 WW Domain. The model built using the KDML procedure (left) resolves a number of discrete, slow eigenprocesses that were suppressed using the unweighted euclidean distance metric. Both models contain 5000 microstates, a 75 ns lag time, and were built using the same clustering procedure.[75]

the obvious potential for overfitting, we ask if it is possible to use KDML to recover more structural and kinetic detail with fewer states.

To construct our model, we begin by representing each conformation in the data set by a vector encoding the sine and cosine of its backbone Φ and Ψ dihedral angles. This has the effect of vectorizing the conformations, as well as breaking the periodic symmetry of the angle measurement. Using these vectors, we apply the KDML procedure with k = 20,000 triplets sampled with $\tau_{\text{close}} = 2$ ns and $\tau_{\text{far}} = 20$ ns. One hundred rounds of optimization lead to a distance metric that reweights structural features in the protein coordinate space based on their kinetic relevance.

Comparing models built using the KDML procedure or an unweighted euclidean distance metric, we find two striking results show in Fig. 2.3. First, the folding time scale is unchanged by the introduction of the KDML method. In fact, given trends in the estimates of the folding time scale with respect to changes in the number of states (data not shown), we estimate that this analysis significantly underestimates the folding time scale. However, more strikingly, our analysis shows the emergence of new discrete time scales in the relaxation spectrum that were not observed with the euclidean distance metrics. While faster than the folding time scale, the dynamical processes in the hundreds of nanoseconds to microsecond regime are of significant interest (Fig. 2.4).

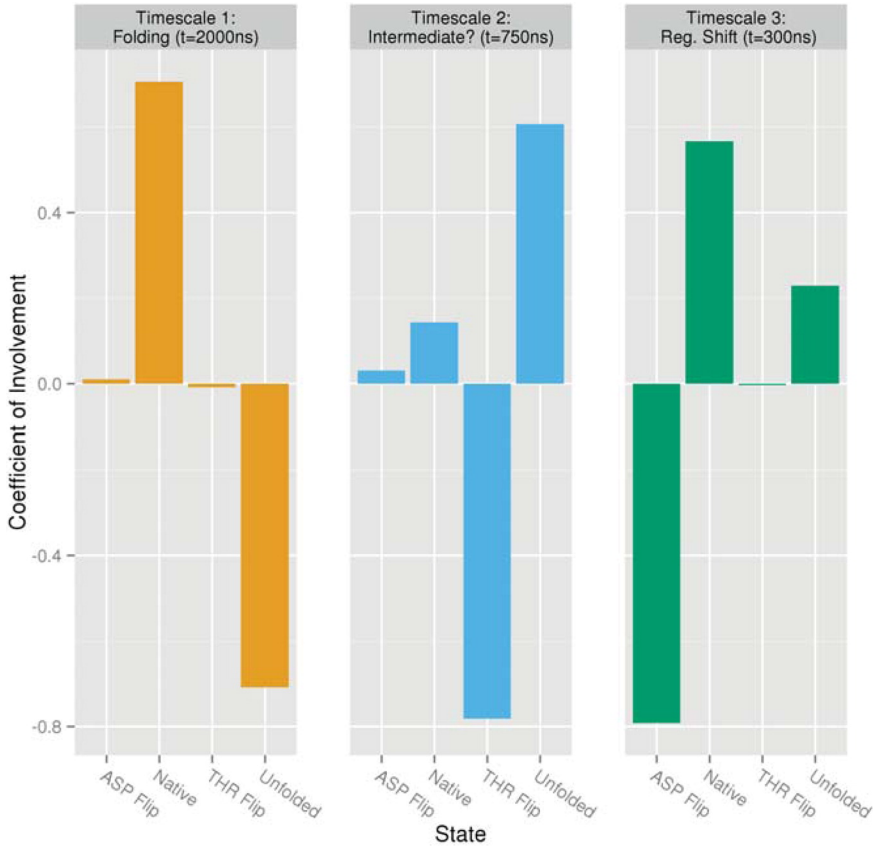


Figure 2.4: Four state macrostate model is able to capture the three slowest dynamic processes in the KDML model of FiP35. The slowest dynamical process represents the direct folding pathway. The second and third dynamical processes are largely associated with two states not previously identified.

Figure 2.5: Representative structure from the simulation's native basin (left) and THR Flip macrostate (right). The subtle structural difference between the two conformational basins, which corresponds to a repacking of the hydrogen bond network in the second hairpin, is identified only via the KDML algorithm.

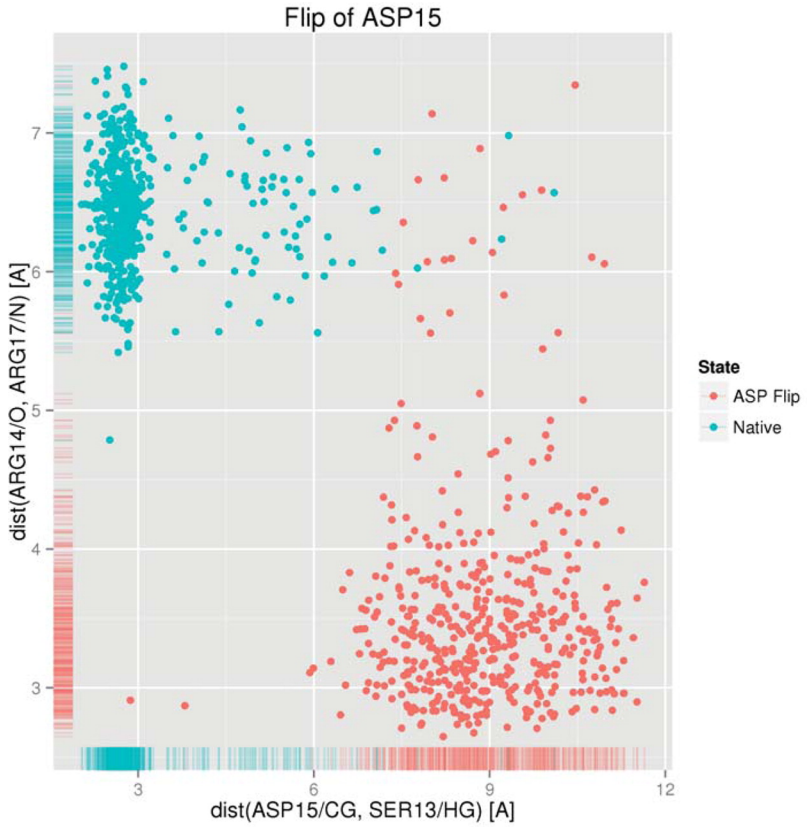


Figure 2.6: Flip of ASP15 and the hydrogen bonding structure in the first loop of Fip35 distinguishes the native state from a kinetically distinct near native state.

To probe the structural dynamics corresponding to these new time scales, we construct a four macrostate model directly from our five thousand microstate model, which seeks to optimally capture the slowest three dynamical processes using the PCCA+ algorithm.[58] Visual analysis of the four states reveals that they correspond to the expected folded and unfolded states in addition to two near-native states characterized by rearrangements in the hydrogen bonding structure in the two loop regions. Specifically, one of the states (Fig. 2.5) shows a reorganization of THR25 forming a nonnative hydrogen bond with HIS23. In the other loop, our final macrostate shows a reorganization of the hydrogen bonding network around ASP15 in which a native hydrogen bond between ASP15 and SER13 is broken and the inflection of the chain at the loop is altered to instead create a set of hydrogen bonds across the loop, including between ARG17O and ARG14N, as shown in Fig. 2.6.

Transitions involving the folding and unfolding of the near native THR flip state occur faster than unfolding of the native state and dominate the second time scale detected in our analysis with a relaxation time of 750-1000 ns. Dynamics between the native and ASP flip state on the other hand occur faster and dominate the third discrete time scale on the order of 300 ns.

We believe that these discrete metastable but near native states were missed by conventional RMSD-based clustering analysis because despite interconverting with the native state slowly, the structural distinctions are subtle. Folding, on the other hand, may be robust to changes in the distance metric because the conformation difference between the folded and unfolded states is so vast.

2.4 Conclusions

By optimizing the distance metric, we have shown that it is possible to build more accurate Markov state models from finite simulation data. An optimal distance metric is one that, while grouping structures geometrically, incorporates a measure of kinetic proximity. In this work, drawing on recent advances in the machine learning community, we approach that goal within a large-margin Mahalanobis distance

metric learning framework. These learned KDML distance metrics allow the construction of Markov state models with fewer states and thus less statistical error than those built using conventional techniques without the bias in the model due to the increase prevalence of non-Markovian effects in this regime. Furthermore, such distance metrics may enable further accuracy improvements within other frameworks for the analysis of biomolecular conformational dynamics, which leverage a discrete-state partitioning, such as weighted ensemble methods.

Reanalyzing two-hunded microseconds of atomistic MD simulation of the FiP35 WW domain by Shaw et al.,[91] we show that the KDML metric permits the identification of key metastable states and discrete relaxation time scales, which were obscured in previous analyses. These metastable near-native states involve subtle structural changes, making them difficult to identify with conventional distance metrics. Such slow dynamics, which nonetheless involve only small changes in gross structural distance metrics such as RMSD, may be a common feature of large proteins, especially those with functionally relevant conformation changes.

Remaining challenges in the analysis of ultralarge molecular dynamics data sets include the enhancement of the connection between the identification of metastable states and the direct search for slow structural degrees of freedom, the direct incorporation of dynamical information into clustering procedures, and further efforts to automate and systematize the Markov state model construction process from MD, for example by optimal selection of parameters within a likelihood maximization framework.

Chapter 3

Understanding Protein Dynamics With L_1-Regularized Reversible Hidden Markov Models

We present a machine learning framework for modeling protein dynamics. Our approach uses L_1-regularized, reversible hidden Markov models to understand large protein datasets generated via molecular dynamics simulations. Our model is motivated by three design principles: (1) the requirement of massive scalability; (2) the need to adhere to relevant physical law; and (3) the necessity of providing accessible interpretations, critical for both cellular biology and rational drug design. We present an EM algorithm for learning and introduce a model selection criteria based on the physical notion of convergence in relaxation timescales. We contrast our model with standard methods in biophysics and demonstrate improved robustness. We implement our algorithm on GPUs and apply the method to two large protein simulation datasets generated respectively on the NCSA Bluewaters supercomputer and the Folding@Home distributed computing network. Our analysis identifies the conformational dynamics of the ubiquitin protein critical to cellular signaling, and elucidates the stepwise activation mechanism of

the c-Src kinase protein.

This chapter is adapted from McGibbon, R. T., Ramsundar, B., Sultan, M. M., Kiss, G., and Pande, V. S., Understanding Protein Dynamics with L_1-Regularized Reversible Hidden Markov Model. *Proceedings of the 31 st International Conference on Machine Learning*, Beijing, China, 2014.[92]

3.1 Introduction

Protein folding and conformational change are grand challenge problems, relevant to a multitude of human diseases, including Alzheimer's disease, Huntington's disease and cancer. These problems entail the characterization of the process and pathways by which proteins fold to their energetically optimal configuration and the dynamics between multiple long-lived, or "metastable," configurations on the potential energy surface. Proteins are biology's molecular machines; a solution to the folding and conformational change problem would deepen our understanding of the mechanism by which microscopic information in the genome is manifested in the macroscopic phenotype of organisms. Furthermore, an understanding of the structure and dynamics of proteins is increasingly important for the rational design of targeted drugs. [93]

Molecular dynamics (MD) simulations provide a computational microscope by which protein dynamics can be studied with atomic resolution. [94] These simulations entail the forward integration of Newton's equations of motion on a classical potential energy surface. The potential energy functions in use, called forcefields, are semi-emprical approximations to the true quantum mechanical Born-Oppenheimer surface, designed to reproduce experimental observables.[95] For moderately sized proteins, this computation can involve the propagation of more than a million physical degrees of freedom. Furthermore, while folding events can take milliseconds (10^{-3} s) or longer, the simulations must be integrated with femtosecond (10^{-15} s) timesteps, requiring the collection of datasets containing trillions of data points.

While the computational burden of performing MD simulations has been a central

challenge in the field, significant progress has been achieved recently with the development of three independent technologies: ANTON, a special-purpose supercomputer using a custom ASICs to accelerate MD;[20] Folding@Home, a distributed computing network harnessing the desktop computers of more than 240,000 volunteers;[25] and Google Exacycle, an initiative utilizing the spare cycles on Google's production infrastructure for science.[96]

The analysis of these massive simulation datasets now represents a major difficulty: how do we turn data into knowledge?[19] In contrast to some other machine learning problems, the central goal here is not merely prediction. Instead, we view analysis — often in the form of probabilistic models generated from MD datasets — as a tool for generating scientific insight about protein dynamics.

Useful probabilistic models must embody the appropriate physics. The guiding physical paradigm by which chemical dynamics are understood is one of *states* and *rates*. States correspond to metastable regions in the configuration space of the protein and can often be visualized as wells on the potential energy surface. Fluctuations within each metastable state are rapid; the dominant, long time-scale dynamics can be understood as a jump process moving with various rates between the states. This paradigm motivates probabilistic models based on a discrete-state Markov chain. *A priori*, the location of the metastable states are unknown. As a result, each metastable state should correspond to a latent variable in the model. Hidden Markov models (HMMs) thus provide the natural framework.

Classical mechanics at thermal equilibrium satisfy a symmetry with respect to time: a microscopic process and its time-reversed version obey the same laws of motion. The stochastic analogue of this property is reversibility (also called detailed balance): the equilibrium flux between any two states X and Y is equal in both directions. Probabilistic models which fail to capture this essential property will assign positive probability to systems that violate the second law of thermodynamics.[97] Hence, we enforce detailed balance in our HMMs.

In addition to the constraints motivated by adherence to physical laws, suitable probabilistic models should, in broad strokes, incorporate knowledge from prior experimental and theoretical studies of proteins. Numerous studies indicate that only

a subset of the degrees of freedom are essential for describing the protein's dominant long time-scale dynamics (see Cho, Levy, and Wolynes and references therein).[29] Furthermore, substantial prior work indicates that protein folding occurs via a sequence of localized shifts.[98] Together, these pieces of evidence motivate the imposition of L_1-fusion regularization.[99] The L_1 term penalizes deviations amongst states along uninformative degrees of freedom, thereby suppressing their effect on the model. Furthermore, the pairwise structure of the fusion penalty minimizes the number of transitions which involve global changes: many pairs of states will only differ along a reduced subset of the dimensions.

The main results of this paper are the formulation of the L_1-regularized reversible HMM and the introduction of a simple and scalable learning algorithm to fit the model. We contrast our approach against standard frameworks for the analysis of MD data and demonstrate improved robustness and physical interpretability.

This paper is organized as follows. Section 2 describes prior work. Section 3 introduces the model and associated learning algorithm. Section 4 applies the model to three systems: a toy double well potential; ubiquitin, a human signaling protein; and c-Src kinase, a critical regulatory protein involved in cancer genesis. Section 5 provides discussion and indicates future directions.

3.2 Prior work

Earlier studies have applied machine learning techniques to investigate protein structure prediction — the problem of discovering a protein's energetically optimal configuration — using CRFs, belief propagation, deep learning, and other general ML methods.[100–103] But proteins are fundamentally dynamic systems, and none of these approaches offer insight into kinetics; rather, they are concerned with extracting static information about protein structure.

The dominant computational tool for studying protein dynamics is MD. Traditional analyses of MD datasets are primarily visual and non-quantitative. Standard approaches include watching movies of a protein's structural dynamics along simulation trajectories, and inspecting the time evolution of a small number of pre-specified

degrees of freedom.[104,105] While these methods have been successfully applied to smaller proteins, they struggle to characterize the dynamics of the large and complex biomolecules critical to biological function. Quantitative methods like PCA can elucidate important (high variance) degrees of freedom, but fail to capture the rich temporal structure in MD datasets.

Markov state models (MSMs) are a simple class of probabilistic models, recently introduced to capture the temporal dynamics of the folding process. In an MSM, protein dynamics are modeled by the evolution of a Markov chain on a discrete state space. The finite set of states is generated by clustering the set of configurations in the MD trajectories.[75] MSMs can be viewed as fully observable HMMs. More recently, HMMs with multinomial emission distributions have been employed on this discrete state space. [106]

Although MSMs have had a number of notable successes,[80,107] they are brittle and complex. Traditional MSMs lack complete data likelihood functions, and learning cannot be easily characterized by a single optimization problem. For these reasons, MSM learning requires significant manual tuning. For example, because clustering is purely a preprocessing step, the likelihood function contains no guidance on the choice of the metastable states. Moreover, the lack of uncertainty in the observation model necessitates the introduction of a very large number of states, typically more than ten thousand, in order to cover the protein's phase space at sufficient resolution. This abundance of states is statistically inefficient, as millions of pairwise transition parameters must be estimated in typically-sized models, and renders interpretation of learned MSMs challenging.

3.3 Fusion L_1-regularized reversible HMM

We introduce the L_1-regularized reversible HMM with Gaussian emissions, a generative probabilistic model over multivariate discrete-time continuous-space time series. As discussed in Section 1, we integrate necessary physical constraints on top of the core hidden Markov model.[108]

Let $\{Y_t\}$ be the observed time series in $\mathbb{R}^D$ of length T (*i.e.*, the input simulation

data), and let $\{X_t\}$ be the corresponding latent time series in $\{1, \ldots, K\}$, where K is a hyperparameter indicating the number of hidden states in the model. Each hidden variable x_t corresponds to a metastable state of the physical system. The emission distribution given $X_t = k$ is a multivariate normal distribution parameterized by mean $\mu_k \in \mathbb{R}^D$ and diagonal covariance matrix $\text{Diag}(\sigma_k^2) \in \mathbb{R}^{D \cdot D}$ (where $\sigma_k^2 \in \mathbb{R}^D$ is the vector of diagonal covariance elements). We use the notation $(\mu_k)_j$ to indicate the jth element of the vector μ_k.

Controlling the means $\{\mu_k\}$ is critical for achieving physically interpretable models. As discussed in Section 1, we wish to minimize the differences between μ_k and $\mu_{k'}$ to the extent possible. Consequently, we place a fusion L_1 penalty on our log likelihood function, which adds the following pairwise cost:[99]

$$\lambda \sum_{k,k'} \sum_j \tau_{k,k'}^{(j)} \left| (\mu_k)_j - (\mu_{k'})_j \right|.$$

Here, λ governs the overall strength of the penalty, while the adaptive fusion weights, $\{\tau_{k,k'}^{(j)}\}$, control the contribution from each pair of states.[109] During learning, the adaptive fusion weights are computed as

$$\tau_{k,k'}^{(j)} = |(\tilde{\mu}_k)_j - (\tilde{\mu}_{k'})_j|^{-1},$$

where the $\{\tilde{\mu}_k\}$ are the learned metastable state means in the absence of the penalty. The intuition motivating the adaptive strength of the penalty is that if degree of freedom j is informative for separating states k and k', the corresponding fusion penalty should be applied lightly.

The reversible time evolution of the model is parameterized by an irreducible, aperiodic, row-normalized K by K stochastic matrix $\mathbf{T}$, which satisfies detailed balance. Mathematically, the detailed balance constraint is

$$\forall k, k', \ \pi_k \mathbf{T}_{k,k'} = \pi_{k'} \mathbf{T}_{k',k},$$

where row vector π is the stationary distribution of $\mathbf{T}$. The stationary distribution π

also parameterizes the initial distribution over the metastable states. By the Perron–Frobenius theorem, π is the dominant left eigenvector of $\mathbf{T}$ with eigenvalue 1 and is not an independent parameter in this model.

The initial distributions and evolution of $\{X_t, Y_t\}$ satisfy the following equations:

$$\begin{aligned} X_0 &\sim \sum_{k=1}^{K} \pi_k \delta_k, \\ X_{t+1} &\sim \sum_{k=1}^{K} \mathbf{T}_{X_t,k}\, \delta_k, \\ Y_t &\sim \mathcal{N}(\mu_{X_t}, \sigma^2_{X_t}). \end{aligned}$$

The complete data likelihood $\{x_t, y_t\}$ is

$$\begin{aligned} &\mathcal{L}(\{x_t\}, \{y_t\} | \mathbf{T}, \mu, \sigma) = \\ &\qquad \pi_{x_0} \prod_{t=1}^{T-1} \mathbf{T}_{x_{t-1}, x_t} \prod_{t=0}^{T-1} \mathcal{N}(y_t; \mu_{x_t}, \sigma^2_{x_t}). \end{aligned}$$

The hyperparameter Δ controls the discretization interval at which a protein's coordinates are sampled to obtain $\{y_t\}$. In the absence of downsampling by Δ, subsequent samples y_t, y_{t+1} would be highly correlated. On the other hand, subsequent samples from an HMM are conditionally independent given the hidden state. Choice of Δ large enough recovers this conditional independence (*vide infra*).

3.3.1 Learning

The model is fit using expectation-maximization. The E-step is standard, while the M-step requires modification to enforce the detailed balance constraint on $\mathbf{T}$ and the adaptive fusion penalty on the $\{\mu_k\}$.

E-step

Inference is identical to that for the standard HMM, using the forward-backward algorithm to compute the following quantities:[108]

$$\gamma_i(t) = \mathbb{P}(X_t = i|\{y_t\}),$$
$$\xi_{ij}(t) = \mathbb{P}(X_t = i, X_{t+1} = j \mid \{y_t\}).$$

M-step

Both the penalty on $\{\mu_k\}$ and the reversibility constraint affect only the M-step. The M-step update to the means in the t-th iteration of EM consists of maximizing the penalized log-likelihood function

$$\mu_k^{(t+1)} = \underset{\mu_k}{\operatorname{argmin}} \sum_i^N \sum_k^K \gamma_k(i) \frac{(x_i - \mu_k)^2}{2(\sigma_k^2)^{(t)}} + \lambda \sum_{k,k'} \sum_j \tau_{k,k'}^{(j)} \left|\mu_{k,j} - \mu_{k',j}\right|.$$

The $\{\mu_k\}$ update is a quadratic program, which can be solved by a variety of methods. We compute $\{\mu_k\}$ by iterated ridge regression. Following Guo et. al (2010) and Fan and Li (2001),[109,110] we use the local quadratic approximation

$$\left| \mu_{k,j}^{(t,s+1)} - \mu_{k',j}^{(t,s+1)} \right| \approx \frac{\left(\mu_{k,j}^{(t,s+1)} - \mu_{k',j}^{(t,s+1)}\right)^2}{2\left|\mu_{k,j}^{(t,s)} - \mu_{k',j}^{(t,s)}\right|} + \frac{1}{2}\left|\mu_{k,j}^{(t,s)} - \mu_{k',j}^{(t,s)}\right|.$$

where s is the iteration index for this procedure within the t-th M-step. This approximation is based on the identity

$$|x - y| = \frac{(x-y)^2}{2|x-y|} + \frac{1}{2}|x - y|.$$

Under the approximation, we obtain a generalized ridge regression problem which can be solved in closed form during each iteration s. Note that this approximation is only valid when $|\mu_{k,j}^{(t,s)} - \mu_{k',j}^{(t,s)}| > 0$. For numerical stability, we threshold $|\mu_{k,j}^{(t,s)} - \mu_{k',j}^{(t,s)}|$ to zero at values less than 10^{-10}.

The variance update is standard:

$$\sigma_k^2 = \frac{\sum_t \gamma_k(t)(y_t - \mu_k)^T(y_t - \mu_k)}{\sum_t \gamma_k(t)}.$$

The transition matrix update is

$$\mathbf{T} = \arg\max_{\mathbf{T}} \sum_{ij} \log(\mathbf{T}_{ij}) \sum_t \xi_{ij}(t).$$

Because the Gaussian emission distributions have infinite support, $\mathbf{T}$ is irreducible and aperiodic by construction. However, we must explicitly constrain $\mathbf{T}$ to satisfy detailed balance.

Lemma 1. *$\mathbf{T}$ satisfies detailed balance if and only if $\mathbf{T}_{ij} = \frac{\mathbf{W}_{ij}}{\sum_k \mathbf{W}_{ik}}$, where $\mathbf{W} = \mathbf{W}^T$.*

Proof. If $\mathbf{T}$ satisfies detailed balance, then let $\mathbf{W}_{ij} = \pi_i \mathbf{T}_{ij} = \pi_j \mathbf{T}_{ji} = \mathbf{W}_{ji}$. Then note

$$\frac{\mathbf{W}_{ij}}{\sum_k \mathbf{W}_{ik}} = \frac{\pi_i \mathbf{T}_{ij}}{\sum_k \pi_i \mathbf{T}_{ik}} = \frac{\mathbf{T}_{ij}}{\sum_k \mathbf{T}_{ik}} = \mathbf{T}_{ij}$$

To prove the converse, assume $\mathbf{T}_{ij} = \frac{\mathbf{W}_{ij}}{\sum_k \mathbf{W}_{ik}}$, with $\mathbf{W} = \mathbf{W}^T$. Let $\pi_i = \sum_k \mathbf{W}_{ik}$. Then $\pi_i \mathbf{T}_{ij} = \mathbf{W}_{ij} = \mathbf{W}_{ji} = \pi_j \mathbf{T}_{ji}$. □

Substituting the results of Lemma 1, we rewrite the transition matrix update as

$$\mathbf{W} = \arg\max_{\mathbf{W}} \Big(\Big[\sum_{ij} \log(\mathbf{W}_{ij}) - \log \pi_i \Big] \sum_t \xi_{ij}(t) \Big).$$

We compute the derivative of the inner term with respect to $\log \mathbf{W}_{ij}$ and optimize with L-BFGS.[111]

3.3.2 Model Selection

There are two free model parameters: K and Δ. The number of metastable states, K, is expected to be small — at most a few dozen. To choose K, we can use the AIC or BIC selection criteria, or alternatively enumerate a few small values.

The choice of Δ is more difficult than the choice of K, as changing the discretization interval alters the support of the likelihood function. Recall that choosing Δ too small results in subsequent samples y_t, y_{t+1} becoming highly correlated, while the model satisfies the conditional independence assumption $Y_t \perp\!\!\!\perp Y_{t+1} \,|\, X_t$. Moreover, small Δ increases data-storage requirements, while Δ too large will needlessly discard data. Thus a balance between these two conflicting directives is necessary.

We use the physical criterion of convergence in the relaxation timescales to evaluate when Δ is large enough. The propagation of the dynamics from an initial distribution over the hidden states, X_t, can be described by

$$P(y_{t+n} \,|\, x_t) = \sum_{k=0}^{K-1} \mathcal{N}(y_{t+n}; \mu_k, \sigma_k^2) \, \left(x_t^T \, \mathbf{T}^n\right)_k .$$

Diagonalize $\mathbf{T}$ in terms of its left eigenvectors ϕ_i, right eigenvectors ψ_i, and eigenvalues λ_i (such a diagonalization is always possible since $\mathbf{T}$ is a stochastic matrix).

$$P(y_{t+n} \,|\, x_t) = \sum_{k=1}^{K} \left[\mathcal{N}(y_{t+n}; \mu_k, \sigma_k^2) \left(\sum_{i=1}^{K} \lambda_i^n \, \langle x_t, \psi_i \rangle \, \phi_i \right)_k \right]$$

Since π is the stationary left eigenvector of $\mathbf{T}$, and the remaining eigenvalues lie in the interval $-1 < \lambda_i < 1$, the collective dynamics can be interpreted as a sum of exponential relaxation processes.

$$P(y_{t+n} \,|\, x_t) = \sum_{k=1}^{K} \left[f_k(y_{t+n}) \left(\pi + \sum_{i=2}^{K} e^{-n/\tau_i} \, \langle x_t, \psi_i \rangle \, \phi_i \right)_k \right]$$

In the equation, we define $f_k(y) = \mathcal{N}(y; \mu_k, \sigma_k^2)$. Each eigenvector of $\mathbf{T}$ (except the

first) describes a dynamical mode with characteristic relaxation timescale

$$\tau_i = -\frac{1}{\ln \lambda_i}.$$

The longest timescales, τ_i, are of central interest from a molecular modeling perspective because they describe dynamical modes visible in time-resolved protein experiments and are robust against perturbations.[78,112] We choose Δ large enough to converge the τ_i: for adequately large Δ, we expect $\tau_i(\Delta)$ to asymptotically converge to the true relaxation timescale τ_i^*. For simple systems, we may evaluate τ_i^* explicitly, while for larger systems, we choose Δ large enough so that $\tau_i(\Delta)$ no longer changes with further increase in the discretization interval.

3.3.3 Implementation

We implement learning for both multithreaded CPU and NVIDIA GPU platforms. In the CPU implementation, we parallelize across trajectories during the E-step using OpenMP. The largest portion of the run time is spent in LOG-SUM-EXP operations, which we manually vectorize with SSE2 intrinsics for SIMD architectures. Parallelism on the GPU is more fine grained. The E-step populates two $T \times K \times K$ arrays with forward and backwards sweeps respectively. To fully utilize the GPU's massive parallelism, each trajectory has a team of threads which cooperate on updating the $K \times K$ matrix at each time step. Specialized CUDA kernels were written for $K = 4, 8, 16$ and 32 along with a generic kernel for $K > 32$.

Even in log space, for long trajectories, the forward-backward algorithm can suffer from an accumulation of floating point errors which lead to catastrophic cancelation during the computation of $\gamma_i(t)$. This risk requires that the forward-backward matrices be accumulated in double precision, whereas the rest of the calculation is safe in single precision.

The speedup using our GPU implementation is 15× compared to our optimized CPU implementation and 75× with respect to a standard numpy implementation using $K = 16$ states on a NVIDIA GTX TITAN GPU / Intel Core i7 4 core Sandy Bridge CPU platform. Further scaling of the implementation could be achieved by

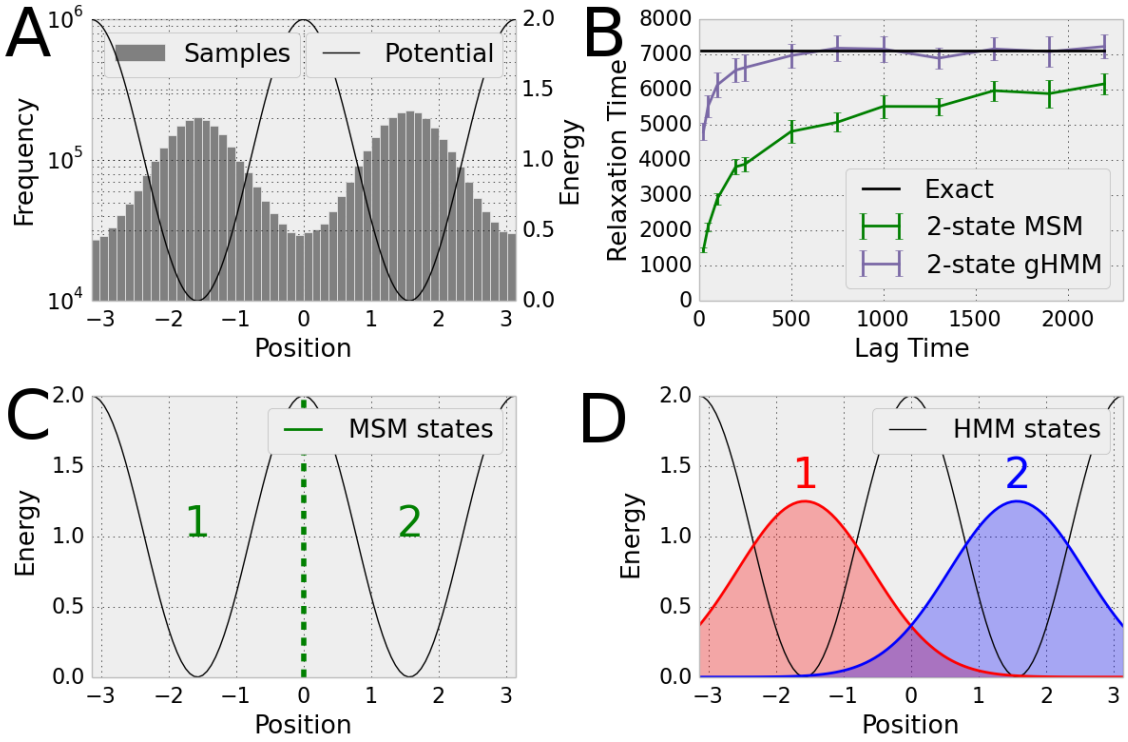


Figure 3.1: Simulations of Brownian dynamics on a double well potential (**A**) illustrate the advantages of the HMM over the MSM. When the dynamics are discretized at a time interval of > 500 steps, the 2-state HMM, unlike the 2-state MSM achieves a quantitatively accurate prediction of the first relaxation timescale (**B**). The MSM (**C**) features hard cutoffs between the states wheres the HMM (**D**) each have infinite support.

splitting the computation over multiple GPUs with MPI.

3.4 Experiments

3.4.1 Double Well Potential

We first consider a one-dimensional diffusion process y_t governed by Brownian dynamics. The process is described by the stochastic differential equation

$$\frac{dy_t}{dt} = -\nabla V(y_t) + \sqrt{2D}R(t)$$

where V is the reduced potential energy, D is the diffusion constant, and $R(t)$ is a zero-mean delta-correlated stationary Gaussian process. For simplicity, we set $D = 1$ and consider the double well potential

$$V(y) = 1 + \cos(2y)$$

with reflecting boundary conditions at $y = -\pi$ and $y = \pi$. Using the Euler-Maruyama method and a time step of $\Delta t = 10^{-3}$, we produced ten simulation trajectories of length 5×10^5 steps each. The histogrammed trajectories are shown in Fig. 1(**A**). The exact value of the first relaxation timescale was computed by a finite element discretization of the corresponding Fokker-Planck equation.[113]

We applied both a two-state MSM and two-state HMM, with fusion L_1 regularization parameter $\lambda = 0$, to the simulation trajectories. The MSM states were fixed, with a dividing surface at $y = 0$, as shown in Fig. 1(**C**). The HMM states were learned, as shown in Fig. 1(**D**). Both the MSM and the HMM display some sensitivity with respect to the discretization interval, with more accurate predictions of the relaxation timescale at longer lag times.

The two-state MSM is unable to accurately learn the longest timescale, τ_1, even with large lag times, while the two-state HMM succeeds in identifying τ_1 with $\Delta \geq 500$ Fig. 1(**B**).

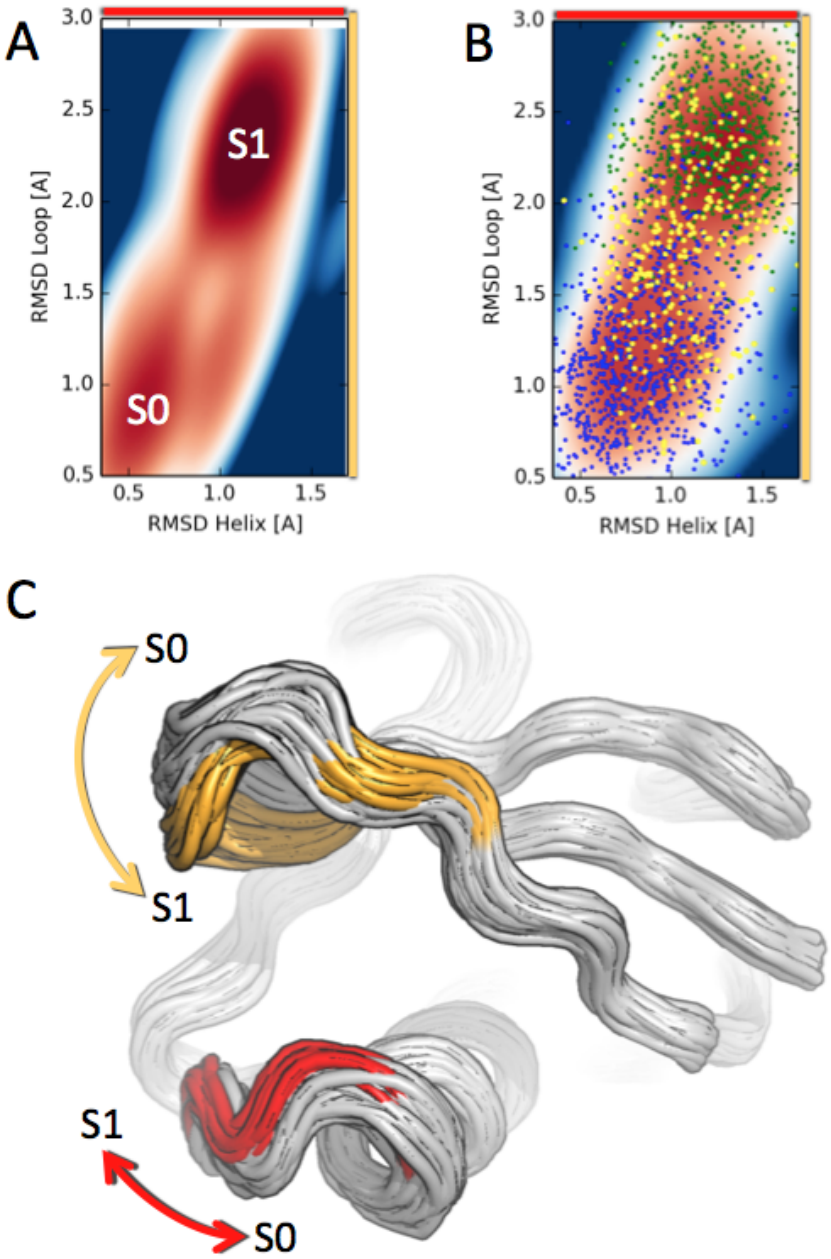


Figure 3.2: Dynamics of Ub. **(A)** The HMM identifies the two metastable states of Ub, varying primarily in the loop and helix regions (axes in yellow and red respectively). **(B)** The MSM fails to cleanly separate the two underlying physical states. Three post-processed macrostates from the MSM are shown (in blue, green, and yellow). **(C)** A structural rendering of the conformational states of the Ub system. S0, shown in grey, binds to the UCH family of proteins, and S1 (with characteristic structural differences to S0 in red and yellow) binds to the USP family.

3.4.2 Ubiquitin

Ubiquitin (Ub) is a regulatory hub protein at the intersection of many signaling pathways in the human body.[114] Among its many tasks are the regulation of inflammation, repair of DNA, and the breakdown and recycling of waste proteins. Ubiquitin interacts with close to 5000 human signaling proteins. Understanding the link between structure and function in ubiquitin would elucidate the underlying framework of the human signaling network.

We obtained a dataset of MD simulations of human Ub consisting of 3.5 million data points. The protein, shown in Fig. 3.2, is composed of 75 amino acids. The simulations were performed on the NCSA Blue Waters supercomputer. The resulting structures were featurized by extracting the distance from each amino acid's central carbon atom to its position in the simulations' starting configurations. HMMs were constructed with 2 to 6 states. We chose Δ by monitoring the convergence of the relaxation timescales as discussed in Sec. 3.3.2, and set the L_1 fusion penalty heuristically to a default value of $\lambda = 0.01$. In agreement with existing biophysical data,[115] the HMMs correctly determined that Ub was best modeled with 2 states (Fig. 3.2**A**). For ease of representation, the learned HMM is shown projected onto two critical degrees of freedom (discussed below).

For comparison, we generated MSM models with 500 microstates (Fig. 3.2**B**) and projected upon the same critical degrees of freedom. We used a standard kinetic lumping post-processing step to identify 3 macrostates (shown in green, blue, and yellow respectively); the lumping algorithm collapsed when asked to identify 2 macrostates. [60] Contrast the simple, clean output of the 2 state HMM in Fig. 3.2(**A**) with the standard MSM of Fig. 3.2(**B**). Note how significant post-processing and manual tuning would be required to piece together the true two-state structural dynamics of Ub from the MSM output.

We display a structural rendering of the Ub system in Fig. 3.2(**C**). The imposed L_1 penalty of the HMM suppresses differences among the uninformative degrees of freedom depicted in grey. The remaining portions of the protein (shown in color) reveal the two critical axes of motion of the Ub system: the hinge dynamics of the loop region displayed in yellow and a kink in the lower helix shown in red. We use

these axes in the simplified representations shown in Figs. 3.2(**A,B**).

The states S0 and S1 identified by the HMM have direct biological interpretations. Comparison to earlier experimental work reveals that configuration S0 binds to the UCH family of proteins, while configuration S1 binds to the USP family instead.[116] The families play differing roles in the vital task of regenerating active Ub for the cell-signaling cycle.

Together, MD and the HMM analysis provide atomic insight into the effect of protein structure on ubiquitin's role in the signaling network. Our analysis approach may have significant value for protein biology and for the further study of cellular signaling networks. Although experimental studies of protein signaling provide the gold standard for hard data, they struggle to provide structural explanations — knowing why a certain protein is more suited for certain signaling functions is challenging at best. In contrast, the MD/HMM approach can provide a direct link between structure and function and give a causal basis for observed protein activity.

3.4.3 c-Src Tyrosine Kinase

Protein kinases are a family of enzymes that are critical for regulating cellular growth whose aberrant activation can lead to uncontrolled cellular proliferation. Because of their central role in cell proliferation, kinases are a critical target for anti-cancer therapeutics. The c-Src tyrosine kinase is a prominent member of this family that has been implicated in numerous human malignancies.[117]

Due to the protein's size and complexity, performing MD simulations of the c-Src kinase is a formidable task. The protein, shown in Fig. 3.3**A**, consists of 262 amino acids; when surrounding water molecules — necessary for accurate simulation — are taken into account, the system has over 40,000 atoms. Furthermore, transition between the active and inactive states takes hundred of microseconds. Adequate sampling of these processes therefore requires hundreds of billions of MD integrator steps. Simulations of the c-Src kinase were performed on the Folding@Home distributed computing network, collecting a dataset of 4.7 million configurations from 550 μs of sampling, for a total of 108 GB of data.[118]

In order to understand the molecular activation mechanism of the c-Src kinase, we analyzed this dataset using the L_1 regularized reversible HMM. We featurized the configurations by extracting the distance from each amino acid's central carbon atom to its position in an experimentally determined inactive configuration. We built HMMs with 2 to 6 states, and singled out the 3 state model for achieving a balance of complexity and interpretability. As with Ub, we chose Δ by monitoring the convergence of the relaxation timescales, same default L_1 fusion penalty of $\lambda = 0.01$.

The L_1-regularized reversible HMM elucidates the c-Src kinase activation pathway, revealing a stepwise mechanism of the dynamics. A projection of the learned HMM states onto two key degrees of freedom is shown in Fig. 3.3**B**. Fig. 3.3**C** shows a structural representation of the means of the three states, highlighting a sequential activation mechanism. The transformation from the inactive to the intermediate state occurs first by the unfolding of the A-loop (the subsection of the protein highlighted in red). Activation is completed by the inward rotation of the C-helix (highlighted in orange) and rupture of a critical side chain interaction between two amino acids on the C-helix and the A-loop respectively.

Although the protein structure is complex, the activation process takes place only in a small portion of the overall protein; the random fluctuations of the remaining degrees of freedom are largely uncoupled from the activation process. As with Ub, the L_1 penalty suppresses the signal from unimportant degrees of freedom shown in grey. In contrast to the simplicity of HMM approach, a recent MSM analysis of this dataset found similar results, but required 2,000 microstastates and significant post-processing of the models to generate physical insight into the activation mechanism.[118]

The identification of the intermediate state along the activation pathway has substantial implications in the field of rational drug design. Chemotherapy drugs often have harmful side effects because they target portions of proteins that are common across entire families, interfering with both the uncontrolled behavior of tumor proteins as well as the critical cellular function of healthy proteins. Intermediate states, such as the one identified by the HMM, are more likely to be unique to each kinase protein; future therapeutics that target these intermediate states could have significantly fewer deleterious side effects.[119]

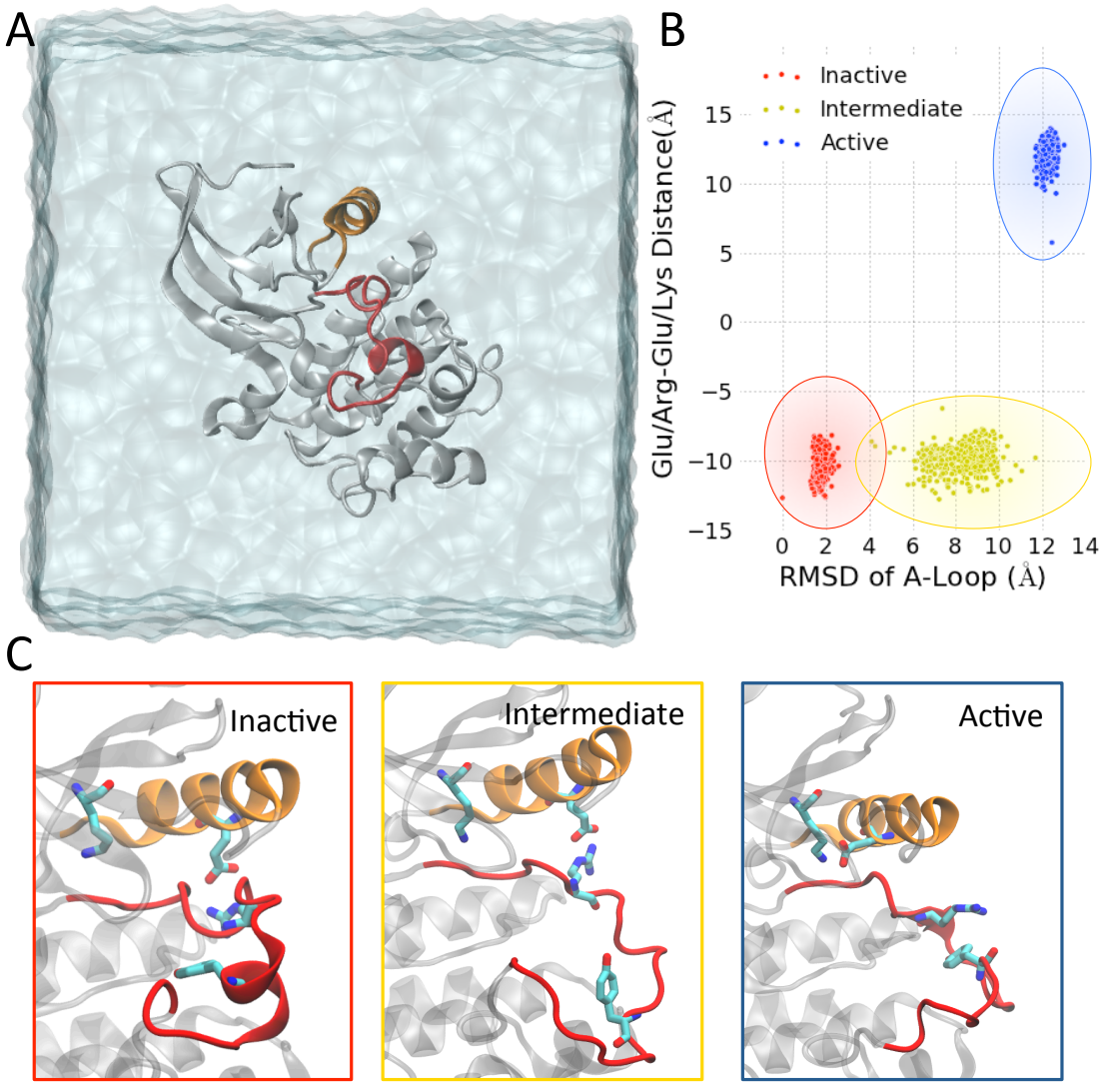


Figure 3.3: Activation of the c-Src Kinase. (**A**) Structure of the protein system. (**B**) The 3 state HMM, projected onto two degrees of freedom representing the positions of the A-loop (shown in red) and C-helix (shown in orange) respectively. (**C**) Structural renderings of the means of the hidden states showing atomistic details of the activation pathway.

3.5 Discussion and conclusion

Currently, MSMs are a dominant framework for analyzing protein dynamics datasets. We propose replacing this methodology with L_1-regularized reversible HMMs. We show that HMMs have significant advantages over MSMs: whereas the MSM state decomposition is a prepreprocessing procedure without guidance from a complete-data likelihood function, the HMM couples the identification of metastable states with the estimation of transition probabilities. As such, accurate models require fewer states, aiding interpretability from a physical perspective.

The switch is not without tradeoffs. MSMs are backed by a significant body of theoretical work: the MSM is a direct discretization of an integral operator which formally controls the long timescale dynamics known as the transfer operator. This connection enables the quantification of approximation error in the MSM framework [97]. No such theoretical guarantees yet exist for the L_1-regularized reversible HMM because the evolution of Y_t is no longer unconditionally Markovian. However, because the HMM can be viewed as a generalized hidden MSM, there is reason to believe that analogues of MSM theoretical guarantees extend to the HMM framework.

While the L_1-regularized reversible hidden Markov model represents an improvement over previous methods for analyzing MD datasets, future work will likely confront a number of remaining challenges. For example, the current model does not learn the featurization and treats Δ as a hyperparameter. Bringing these two aspects of the model into the optimization framework would reduce the required amount of manual tuning. Adapting techniques from Bayesian nonparametrics, unsupervised feature learning and linear dynamical systems may facilitate the achievement of these goals.

Our results show that structured statistical analysis of massive protein datasets is now possible. We reduce complex dynamical systems with thousands of physical degrees of freedom to simple statistical models characterized by a small number of metastable states and transition rates. The HMM framework is a tool for turning raw molecular dynamics data into scientific knowledge about protein structure, dynamics and function. Our experiments on the ubiquitin and c-Src kinase proteins extract

insight that may further the state of the art in cellular biology and rational drug design.

Acknowledgments

We thank Diwakar Shukla for kindly providing the c-Src kinase MD trajectories. B.R. was supported by the Fannie and John Hertz Foundation. G.K and V.S.P acknowledge support from the Simbios NIH Center for Biomedical Computation (NIH U54 Roadmap GM072970). V.S.P. acknowledges NIH R01-GM62868 and NSF MCB-0954714.

Chapter 4

Efficient maximum likelihood parameterization of continuous-time Markov processes

Continuous-time Markov processes over finite state-spaces are widely used to model dynamical processes in many fields of natural and social science. Here, we introduce an maximum likelihood estimator for constructing such models from data observed at a finite time interval. This estimator is dramatically more efficient than prior approaches, enables the calculation of deterministic confidence intervals in all model parameters, and can easily enforce important physical constraints on the models such as detailed balance. We demonstrate and discuss the advantages of these models over existing discrete-time Markov models for the analysis of molecular dynamics simulations.

This chapter is adapted with permission from McGibbon, R. T. and Pande, V. S., Efficient maximum likelihood parameterization of continuous-time Markov processes. *J. Chem. Phys.*, **2015**, *143*, 034109.[120]

4.1 Introduction

Estimating the parameters of a continuous-time Markov jump process model based on discrete-time observations of the state of a dynamical system is a problem which arises in many fields of science, including physics, biology, sociology, meteorology, and finance.[121–124] Diverse applications include the progression of credit risk spreads,[125] social mobility,[126] and the evolution of DNA sequences in a phylogenetic tree.[127] In chemical physics, these models, also called master equations, describe first-order chemical kinetics, and are the principle workhorse for modeling chemical reactions.[128]

For complex physical systems, the derivation of kinetic models from first principles is often intractable. In these circumstances, the parameterization of models from data is often a superior approach. As an example, consider the dynamical behavior of solvated biomolecules, such as proteins and nucleic acids. Despite the microscopic complexity of their equations of motion, relatively simple multi-state kinetics often arise, as exemplified by the ubiquity of two- and few-state Markov process models for protein folding. [59,129–133]

Due in part to the unavailability of computationally efficient and numerically robust estimators for continuous-time Markov models, in the field computational biophysics, discrete-time Markov models have been widely used to fit and interpret the output of molecular dynamics (MD) simulations. Also called Markov state models (MSMs), these methods describe the molecular kinetics observed in an MD simulation as a jump process with a discrete-time interval generally on the order of $\sim$ 10 — 100 ns.[75,134] These models provide convenient estimators for key quantities of interest for molecular systems, such as the free energies of various metastable conformational states, the timescales of their interconversion, and the dominant transition pathways.[70,135–137]

In this work, we introduce an efficient maximum likelihood estimator for continuous-time Markov models on a finite state space from discrete-time data. The source of data used here is identical to that employed in fitting discrete-time Markov chain models — namely, the number of observed transitions between each pair of states within a specified time interval. We demonstrate the properties of these models on

simple systems, and apply them to the analysis of the folding of the FiP35 WW protein domain.

4.2 Background

Consider a time-homogenous continuous-time Markov process $\{X(t) : t \geq 0\}$ over a finite state space, $\mathscr{S} = \{1, \ldots, n\}$. The process is determined completely by an $n \times n$ matrix $\mathbf{K}$, variously called its rate matrix, infinitesimal generator,[138] substitution matrix,[139] or intensity matrix.[140]

For an interval $\tau > 0$, begin with the $n \times n$ matrix, $\mathbf{T}(\tau)$, of probabilities that the process jumps from one state, i, to another state, j,

$$\mathbf{T}(\tau)_{ij} = P(X(t+\tau) = j \mid X(t) = i), \tag{4.1}$$

which, by time-homogeneity is assumed to be independent of t. The process's rate matrix, $\mathbf{K}$, is defined as

$$\mathbf{K} \equiv \lim_{\tau \to 0+} \frac{\mathbf{T}(\tau) - \mathbf{I}_n}{\tau}. \tag{4.2}$$

Given $\mathbf{K}$ and any time interval, τ, the transition probability matrix, $\mathbf{T}(\tau)$, can be expressed a matrix exponential

$$\mathbf{T}(\tau) = \exp(\mathbf{K}\tau) \equiv \sum_{i=0}^{\infty} \frac{\tau^i \mathbf{K}^i}{i!}. \tag{4.3}$$

A particular rate matrix $\mathbf{K}$ corresponds to a valid continuous-time Markov process if and only if its off-diagonal elements are nonnegative and its row sums equal zero. These constraints are necessary to ensure that the probabilities propagated by the dynamics remain positive and sum to one. We denote by $\mathscr{K}$ this set of admissible

rate matrices,

$$\mathscr{K} = \Bigg\{ \mathbf{K} = \{k_{ij}\} \in \mathbb{R}^{n\times n} : k_{ij} \geq 0 \text{ for all } i \neq j, \quad k_{ii} = -\sum_{j\neq i} k_{ij} \Bigg\}. \tag{4.4}$$

Furthermore, we denote by $\mathscr{T}$ the set of all embeddable transition probability matrices, that is, those which could originate as the transition probability matrix, $\mathbf{T}(\tau)$, induced by some continuous-time Markov process,

$$\mathscr{T} = \left\{ \mathbf{T} \in \mathbb{R}^{n\times n} : \exists\, \mathbf{K} \in \mathscr{K}\, s.t. \mathbf{T} = \exp(\mathbf{K}) \right\}. \tag{4.5}$$

It is well-known that set $\mathscr{T}$ is a strict subset of the set of all stochastic matrices; not all stochastic matrices are embeddable.[141,142] A complete description of the topological structure of $\mathscr{T}$ as well as the necessary and sufficient conditions for a stochastic matrix to be embeddable are open problems in the theory of Markov processes.

Although Eq. (4.2) serves as the definition of the rate matrix of a continuous-time Markov process, it is generally not directly suitable as a method for parameterizing Markov models, particularly for applications in chemical kinetics. The attempt to numerically approximate the limit in Eq. (4.2) from empirically measured transition probabilities would be valid if the generating process were exactly Markovian. However, in chemical kinetics, a Markov process model — the chemical master equation — is an approximation valid only for timescales longer than the molecular relaxation time.[143,144] A suitable Markov model which is predictive over long timescales must capture both the instantaneous kinetics as well as, to use the vocabulary of Mori-Zwanzig formalism, the effective contribution of the integrated memory kernel.[82,145]

Our goal is to address this parameterization problem. The primary contribution of this work is an efficient algorithm for estimating $\mathbf{K}$ from observed discrete-time observations. We adopt a direct maximum likelihood approach, with $O(n^3)$ work per iteration. Many constraints on the solution, such as detailed balance or specific sparsity patterns on $\mathbf{K}$ can be introduced in a straightforward manner without additional

cost.

Prior work on this subject is numerous. Crommelin and Vanden-Eijnden proposed a method for estimating $\mathbf{K}$ in which a discrete-time transition probability matrix is first fit to the observed data, followed by the determination of the rate matrix, $\mathbf{K}$ such that $\exp(\mathbf{K}\tau)$ is nearest to the target empirical transition probability matrix.[146,147] The nature of this calculation depends on the norm used to define the concept of "nearest": under a Frobenius norm, this problem has a closed form solution, while the norm of Crommelin and Vanden-Eijnden leads to a quadratic program. A similar approach was advocated by Israel et al.[148]

Kalbfleisch and Lawless proposed a maximum likelihood estimator for $\mathbf{K}$.[149] Without constraints on the rate matrix, their proposed optimization involves the construction and inversion of an $n^2 \times n^2$ Hessian matrix at each iteration of the optimization, rendering it prohibitively costly ($O(n^6)$ scaling per iteration) for moderate to large state spaces.

A series of expectation maximization (EM) algorithms are described by Asmussen, Nerman and Olsson, Holmes and Rubin, Bladt and Sørensen, and Hobolth and Jensen [139,140,150,151] These algorithms treat the state of the system between observation intervals as an unobserved latent variable, which when interpolated via EM leads to more efficient estimators. A review of these algorithms is presented by Metzner et al.[138] At best, each iteration of the proposed methods scales as $O(n^5)$.

4.3 Maximum likelihood estimation

4.3.1 Log-likelihood and gradient

We take our source of data to be one or more observed discrete-time trajectories from a Markov process, $x = \{x_0, x_\tau, \ldots, x_{N\tau}\}$, in a finite state space, observed at a regular time interval.

The likelihood of the data given the model and the initial state is given in terms of the transition probability matrix as the product of the transition probabilities assigned to each of the observed jumps in the trajectory.

$$P(x|\mathbf{K}, x_0) = \prod_{k=0}^{N-1} \mathbf{T}(\tau)_{x_{k\tau}, x_{(k+1)\tau}}. \tag{4.6}$$

When more than one independent trajectory is observed, the data likelihood is a product over trajectory with individual terms given by Eq. (4.6).

Because many transitions are potentially observed multiple times, Eq. (4.6) generally contains many repeated terms. Define the observed transition count matrix $\mathbf{C}(\tau) \in \mathbb{R}^{n \times n}$,

$$\mathbf{C}(\tau)_{ij} = \sum_{k=0}^{N-1} \mathbf{1}(x_{k\tau} = i) \cdot \mathbf{1}(x_{(k+1)\tau} = j). \tag{4.7}$$

Collecting repeated terms, the likelihood can be rewritten more compactly as

$$P(x|\mathbf{K}, x_0) = \prod_{i,j} \mathbf{T}(\tau)_{ij}^{\mathbf{C}(\tau)_{ij}}. \tag{4.8}$$

Suppose that the rate matrix, $\mathbf{K}$ is parameterized by a vector, $\theta \in \mathbb{R}^b$ of independent variables, $\mathbf{K} = \mathbf{K}(\theta)$. In the most general case, every element of the rate matrix may individually be taken as an independent variable, with $b = n^2 - n$. As discussed in Section 4.3.2, other parameterizations may be used to enforce certain properties on $\mathbf{K}$. The logarithm of data likelihood is

$$\begin{aligned} \mathcal{L}(\theta; \tau) &\equiv \ln P(x|\mathbf{K}(\theta), x_0), & (4.9)\\ &= \sum_{i,j} \mathbf{C}_{ij}(\tau) \ln \mathbf{T}(\tau)_{ij}, & (4.10)\\ &= \sum_{i,j} \Big(\mathbf{C}(\tau) \circ \ln \exp\big(\tau\, \mathbf{K}(\theta)\big) \Big)_{ij}, & (4.11) \end{aligned}$$

where $\ln(\mathbf{X})$ is the element-wise natural logarithm, $\exp(\mathbf{X})$ matrix exponential, and $\mathbf{X} \circ \mathbf{Y}$ is the Hadamard (element-wise) matrix product. Note that the element-wise logarithm and matrix exponential are not inverses of one another.

The most straightforward parameter estimator — the maximum likelihood estimator (MLE) — selects parameters which maximize the likelihood of the data,

$$\theta^{MLE} = \arg\max_{\theta} \mathcal{L}(\theta; \tau). \tag{4.12}$$

To maximize Eq. (4.12), we focus our attention on quasi-Newton optimizers that utilize the first derivatives of $\mathcal{L}(\theta;\tau)$ with respect to θ. This requires an efficient algorithm for computing $\nabla_\theta \mathcal{L}(\theta;\tau)$. We achieve this by starting from the eigendecomposition of $\mathbf{K}$,

$$\mathbf{K} = \mathbf{V}\,\mathrm{diag}(\lambda)\mathbf{U}^T, \tag{4.13}$$

where the columns of $\mathbf{U}$ and $\mathbf{V}$ contain the left and right eigenvectors of $\mathbf{K}$ respectively, jointly normalized such that $\mathbf{V}^{-1} = \mathbf{U}^T$, and λ are the corresponding eigenvalues. Assuming that $\mathbf{K}$ has no repeated eigenvalues, the directional derivatives of the induced transition probability matrix, $\partial \mathbf{T}(\tau)_{ij}/\partial\theta_u$ are given by [149,152]

$$\partial \mathbf{T}(\tau)_{ij}/\partial\theta_u = \mathbf{V}\left((\mathbf{U}^T(\partial\mathbf{K}/\partial\theta_u)\mathbf{V}) \circ \mathbf{X}(\lambda, t)\right)\mathbf{U}^T, \tag{4.14}$$

where $\mathbf{X}(\lambda, t)$ is an $n \times n$ matrix with entries

$$[\mathbf{X}(\lambda,t)]_{ij} = \begin{cases} \tau\ \exp(\tau\lambda_i), & i = j, \\ \frac{\exp(\tau\lambda_i) - \exp(\tau\lambda_j)}{\lambda_i - \lambda_j}, & i \neq j. \end{cases} \tag{4.15}$$

The elements of the gradient of the log-likelihood can then be constructed as

$$\frac{\partial \mathcal{L}(\theta;\tau)}{\partial\theta_u} = \sum_{ij} \left(\mathbf{D} \circ \mathbf{V}\Big(\left(\mathbf{U}^T(\partial\mathbf{K}/\partial\theta_u)\mathbf{V}\right) \circ \mathbf{X}(\lambda,t)\Big)\mathbf{U}^T\right)_{ij}, \tag{4.16}$$

where $\mathbf{D}_{ij} = \mathbf{C}(\tau)_{ij}/\mathbf{T}_{ij}$.

A direct implementation of Eq. (4.16) requires at least 4 $n \times n$ matrix multiplies for each element of θ, indexed by u. If the parameter vector, θ, contains $O(n^2)$ parameters, then computing the full gradient will require $O(n^5)$ floating point operations

(FLOPs). However, two properties of the Hadamard product and matrix trace can be exploited to dramatically reduce the computational complexity of constructing the gradient vector to $O(n^3)$ FLOPs.

$$\sum_{ij}(\mathbf{A} \circ \mathbf{B})_{ij} = \mathrm{Tr}\left(\mathbf{A}\mathbf{B}^T\right), \tag{4.17}$$

$$\mathrm{Tr}\left(\mathbf{A}^T(\mathbf{B} \circ \mathbf{C})\right) = \mathrm{Tr}\left(\mathbf{B}^T(\mathbf{A} \circ \mathbf{C})\right). \tag{4.18}$$

Using these identities, the gradient of the log-likelihood can be rewritten as

$$\frac{\partial \mathcal{L}(\theta;\tau)}{\partial \theta_u} = \sum_{ij}\left(\partial \mathbf{K}/d\theta_u \circ \underbrace{\left(\mathbf{U}\Big((\mathbf{V}^T\mathbf{D}\mathbf{U}) \circ \mathbf{X}(\lambda,t)\Big)\mathbf{V}^T\right)}_{\mathbf{Z}}\right)_{ij}. \tag{4.19}$$

Note that because $\mathbf{Z}$ is independent of u, it can be constructed once at the beginning of a gradient calculation at a cost of $O(n^3)$ FLOPs, and reused for each index, u. The remainder of the work involves constructing the derivative matrix $\partial\mathbf{K}/\partial\theta_u$, which is generally quite sparse, and a single inexpensive sum of a Hadamard product. Overall, this rearrangement reduces the complexity of constructing the full gradient vector from $O(n^5)$ to $O(n^3)$ FLOPs.

4.3.2 Reversible parameterization

In the application of these models to domain-specific problems, additional constraints on the Markov process may be known, and enforcing these constraints during parameterization can enhance the interpretability of solutions as well as provide a form of regularization.

For many molecular system, it is known that the underlying dynamics are reversible, and this property can be enforced in Markov models as well. A Markov process is reversible when the rate matrix, $\mathbf{K}$, satisfies the detailed balance condition with respect to a stationary distribution, π, towards which the process relaxes over

time.

$$\pi \mathbf{K} = 0, \tag{4.20}$$

$$\pi_i k_{ij} = \pi_j k_{ji} \qquad \forall\, i \neq j. \tag{4.21}$$

This constraint can be enforced on solutions through the design of the parameterization function, $\mathbf{K}(\theta)$. If $\mathbf{K}$ is reversible, Eq. (4.21) implies that a real symmetric $n \times n$ matrix, $\mathbf{S}$, can be formed, which we refer to as the symmetric rate matrix, such that

$$\mathbf{S} = \mathbf{S}^T = \mathrm{diag}(\sqrt{\pi})\, \mathbf{K}\, \mathrm{diag}(\sqrt{\pi})^{-1}. \tag{4.22}$$

Because of this symmetry and the constraint on the row sums of $\mathbf{K}$, only the upper triangular (exclusive of the main diagonal) elements of $\mathbf{S}$, and the stationary vector, π, need be directly encoded by the parameter vector, θ, to fully specify $\mathbf{K}$. Furthermore, since the elements of π are constrained to be positive, working with the element-wise logarithm of π can enhance numerical stability. For the elements of $\mathbf{S}$, which are only constrained to be nonnegative, the same logarithm transformation is inapplicable, as it is incompatible with sparse solutions that set one or more rate constants equal to zero. For these reasons, we use a parameter vector, of length $b = \binom{n+1}{2}$, with $\theta = (\theta^{(S)}, \theta^{(\pi)})$. The first $\binom{n}{2}$ elements, notated $\theta^{(S)}$, encode the off-diagonal elements of $\mathbf{S}$. The remaining n elements are notated $\theta^{(\pi)}$, and are used to construct the stationary distribution, π. From $\mathbf{S}$ and π, the off-diagonal and diagonal elements of $\mathbf{K}$ are then constructed from Eq. (4.22). In explicit notation, the construction is

$$\mathrm{vech}(\mathbf{S})_i = \theta_i^{(S)} \quad i \in \{1, \ldots, n(n-1)/2\}, \tag{4.23}$$

$$\pi_i = \frac{\exp(\theta_i^{(\pi)})}{\sum_{j=1}^{n} \exp(\theta_j^{(\pi)})} \quad i \in \{1, \ldots, n\}, \tag{4.24}$$

$$\mathbf{K}_{ij} = \begin{cases} [D(\sqrt{\pi})^{-1} \mathbf{S} D(\sqrt{\pi})]_{ij}, & i \neq j \\ -\sum_{j \neq i} \mathbf{K}_{ij}, & i = j, \end{cases} \tag{4.25}$$

where vech($\mathbf{A}$) is the row-major vectorization of the elements of a symmetric $n \times n$ matrix above the main diagonal,

$$\text{vech}(\mathbf{A}) = [a_{1,2}, \ldots, a_{1,n}, a_{2,3}, \ldots, a_{2,n}, \ldots, a_{n-1,n}]^T. \tag{4.26}$$

The necessary gradients of Eq. (4.25), $\partial \mathbf{K}_{ij}/d\theta_u$ are sparse. For fixed $1 \leq u \leq \binom{n}{2}$, the $n \times n$ matrix $\partial \mathbf{K}_{ij}/d\theta_u$ over all i, j contains only four nonzero entries, whereas for $\binom{n}{2} < u \leq \binom{n+1}{2}$, the same matrix contains $3n - 2$ nonzero entries. The sum of its Hadamard product with $\mathbf{Z}$ in Eq. (4.19) can thus be computed in $O(1)$ or $O(n)$ time. For the remainder of this work, we focus exclusively on this reversible parameterization for $\mathbf{K}(\theta)$.

4.3.3 Optimization

Equipped with the log-likelihood and an efficient algorithm for the gradient, we now consider the construction of maximum likelihood estimates, Eq. (4.12). Among the first-order quasi-Newton methods tested, we find Limited-memory Broyden-Fletcher-Goldfarb-Shanno optimizer with bound constraints (L-BFGS-B) to be the most successful and robust.[153,154]

To begin the optimization, we choose the initial guess for θ according to the following procedure. First, we fit the maximum likelihood reversible transition probability matrix computed using Algorithm 1 of Prinz et al.[97] Next, we compute its principle matrix logarithm, $\widetilde{\mathbf{K}}$, using an inverse scaling and squaring algorithm, and scaling by τ.[155] Generally, the MLE reversible transition matrix is not embeddable, and thus the principle logarithm is complex or has negative off-diagonal entries, and does not correspond to any valid continuous-time Markov process. We take the initial guess from $\theta^{(\pi)}$ directly from the stationary eigenvector of the MLE transition matrix, and $\theta^{(S)}$ from the nearest (by Frobenius norm) valid rate matrix to $\widetilde{\mathbf{K}}$, given by $\max(\text{Re}(\widetilde{\mathbf{K}}), 0)$.[142]

The optimization problem is non-convex in the general case and may have multiple local minima. Varying the optimizer's initialization procedure can thus mitigate

the risk of convergence to a low quality local minimum. One alternative initialization $\mathbf{K}$ is the pseudo-generator, $\mathbf{K}_p = (\mathbf{T}(\tau) - I_n)/\tau$, which arises from a first-order Taylor approximation to the matrix exponential. After the optimization has terminated, a useful check is to compare the maximum likelihood transition matrix $\mathbf{T}(\tau)$ estimated during initialization with the exponential of the recovered rate matrix, $\exp(\tau \mathbf{K}^{MLE})$. Large differences between the two matrices, or their eigenspectra / relaxation timescales, may be symptomatic of non-embedability of the data or a convergence failure of the optimizer. If the data are available at a lag time shorter than τ, convergence failures can often also be circumvented by using a converged rate matrix obtained from a model at a shorter lag time as an initial guess for a model at a longer lag time.

4.3.4 Implementation notes

Because $\mathbf{S}$ is symmetric, it can be diagonalized efficiently at cost of $O(4n^3/3)$ FLOPs. The eigenvectors can then be rotated by $D(\sqrt{\pi})$ to give the eigenvectors of $\mathbf{K}$. Compared to diagonalizing the non-symmetric matrix $\mathbf{K}$ directly, this can yield a speedup of 2-10× in the critical diagonalization step required to compute the gradient vector.

For each pair of states with an observed transition count, (i, j) such that $\mathbf{C}(\tau)_{ij} > 0$, the gradient expressions Eq. (4.16), and Eq. (4.19) are only defined when $\mathbf{T}_{ij} > 0$. A sufficient condition to ensure this property is that $\mathbf{K}$ be irreducible,[156] but this cannot be straightforwardly ensured throughout every iteration of the L-BFGS-B optimization without heavy-handed measures such as complete positivity of $\mathbf{K}$. In practice, we find that replacing any zeros values in $\mathbf{T}$ with a small constant, such as 1×10^{-20}, when computing the matrix $\mathbf{D}$ in Eq. (4.19) is sufficient to avoid this instability.

Furthermore, note that calculation of $\mathbf{X}(\lambda, t)$ by direct implementation of Eq. (4.15) can suffer from a substantial loss of accuracy for close-lying eigenvalues. The matrix can instead be computed in a more precise manner using the `exprel`$(x) \equiv (e^x - 1)/x$ or `exmp1` $\equiv e^x - 1$ routines, which are designed to be accurate for small x and are available in numerical libraries such as SLATEC, GSL, and the upcoming release of

SciPy.[157–159]

4.4 Quantifying uncertainty

Since all data sets are finite, statistical uncertainty in any estimate of a probabilistic model is unavoidable. Therefore, key quantities of interest beyond the maximum likelihood rate matrix itself, $\mathbf{K}^{MLE} = \mathbf{K}(\theta^{MLE})$, are estimates of the sampling uncertainty in $\mathbf{K}^{MLE}$, and estimates of the sampling uncertainty in quantities derived from $\mathbf{K}^{MLE}$, such as its stationary eigenvector, π, its eigenvalues, λ_i, and relaxation timescales.

In the large sample size limit, the central limit theorem guarantees that the distribution of θ^{MLE} converges to a multivariate normal distribution with a covariance matrix which can be estimated by the inverse of the Hessian of the log-likelihood function evaluated at θ^{MLE}, assuming that the MLE does not lie on a constraint boundary.[160] This can be thought of as a second order Taylor expansion for the log-likelihood surface at the MLE; the log-likelihood is approximated as a paraboloid with negative curvature whose peak is at the MLE and whose width is determined by the Hessian matrix at the peak. The exponential of the log-likelihood, the likelihood surface, is then Gaussian, and the multivariate delta theorem can be used to derive expressions for the asymptotic variance in scalar functions of θ^{MLE}.[160] Computationally, the critical component is the computation of the Hessian matrix,

$$\mathbf{H}_{uv}(\theta;\tau) = \frac{\partial^2 \mathcal{L}(\theta;\tau)}{\partial\theta_u \partial\theta_v}, \tag{4.27}$$

$$= \sum_i^n \sum_j^n \mathbf{C}_{ij} \left(\frac{\partial^2 \mathbf{T}_{ij}/\partial\theta_u\partial\theta_v}{\mathbf{T}_{ij}} - \frac{(\partial \mathbf{T}_{ij}/\partial\theta_u)(\partial \mathbf{T}_{ij}/\partial\theta_v)}{\mathbf{T}_{ij}^2} \right). \tag{4.28}$$

and its inverse.

4.4.1 Approximate Analytic Hessian

Direct calculation of the Hessian requires both the evaluation of the first derivatives of $\mathbf{T}$ as well as the more costly second derivatives. A more efficient alternative, as pointed out by Kalbfleisch and Lawless, is to approximate the second derivatives by estimates of their expectations.[149]

Let $C_i = \sum_j \mathbf{C}_{ij}$. Taking the expected value of $\mathbf{C}_{ij}$ conditional on C_i, we approximate $\mathbf{C}_{ij} \approx \mathbf{T}_{ij} C_i$. This makes it possible to factor $\mathbf{C}_{ij}$ out of the summation over j in Eq. (4.28), and exploit the property that $\sum_j^n \partial^2 \mathbf{T}_{ij} / \partial\theta_u \partial\theta_v = 0$, simplifying Eq. (4.28) to

$$\mathbf{H}_{uv}(\theta;\tau) \approx -\sum_{ij} \frac{C_i}{\mathbf{T}_{ij}} \frac{\partial \mathbf{T}_{ij}}{\partial \theta_u} \frac{\partial \mathbf{T}_{ij}}{\partial \theta_v}. \tag{4.29}$$

Equipped with the approximator Eq. (4.29), the asymptotic variance-covariance matrix of θ is calculated as the matrix inverse of the Hessian, $\mathbf{\Sigma} = \mathbf{H}^{-1}$, and the asymptotic variance in each derived quantity $g(\theta)$ is estimated using the multivariate delta method.[160]

$$\mathrm{Var}(g(\theta)) \approx \nabla g(\theta^{MLE})^T \, \mathbf{\Sigma} \, \nabla g(\theta^{MLE}) \tag{4.30}$$

For example, the asymptotic variance in the stationary distribution can be calculated as

$$\mathrm{Var}(\pi_k) \approx \sum_{i,j}^{n} \frac{\partial \pi_k}{\partial \theta_i^{(\pi)}} \mathbf{\Sigma}_{ij}^{(\pi)} \frac{\partial \pi_k}{\partial \theta_j^{(\pi)}}, \tag{4.31}$$

where $\mathbf{\Sigma}^{(\pi)}$ represent the lower $n \times n$ block of the asymptotic variance covariance matrix and

$$\frac{\partial \pi_i}{\partial \theta_j^{(\pi)}} = \begin{cases} \pi_i - \pi_i^2, & i = j, \\ -\pi_i \pi_j, & i \neq j, \end{cases} \tag{4.32}$$

Other key quantities of interest for biophysical applications include the exponential

relaxation timescales of the Markov model

$$\tau_i = -(\lambda_i)^{-1} \quad i \in \{2, \ldots, n\}. \tag{4.33}$$

The asymptotic variance in the relaxation timescales, τ_i, is

$$\mathrm{Var}(\tau_i) \approx \sum_{uv} \frac{\partial \tau_i}{\partial \theta_u} \Sigma_{uv} \frac{\partial \tau_i}{\partial \theta_v}, \tag{4.34}$$

where $\frac{\partial \tau_i}{\partial \theta_u}$ follows from standard expressions for derivatives of eigensystems,[161]

$$\frac{\partial \tau_i}{\partial \theta_u} = \frac{1}{\lambda_i^2} \left[\mathbf{U}^T \frac{\partial \mathbf{K}(\theta)}{\partial \theta_u} \mathbf{V} \right]_{ii}. \tag{4.35}$$

The sampling uncertainty in other derived properties which depend continuously on θ can be calculated similarly.

When the MLE solution lies at the boundary of the feasible region, with one or more elements of $\theta^{(S)}$ equal to zero, we adopt an active set approach to approximate $\mathbf{\Sigma}$. We refer to the elements of $\theta^{(S)}$ which do not lie on a constraint boundary as free parameters. The Hessian block for the free parameters is constructed and inverted, and the variance and covariance of the constrained elements as well as their covariance with the free parameters is taken to be zero.

4.5 Numerical experiments

We performed numerical experiments on three datasets, which demonstrate different aspects of our estimator for continuous-time Markov processes. Where appropriate, we compare these models to reversible discrete-time Markov models which directly estimate $\mathbf{T}(\tau)$, parameterized via Algorithm 1 of Prinz et al.[97]

4.5.1 Recovering a Known Rate Matrix

First, we constructed a simple synthetic eight state Markov process with known rates. The network is shown in Fig. 4.1. The largest non-zero eigenvalue of $\mathbf{K}$ is $\lambda_2 \approx -9.40\times$

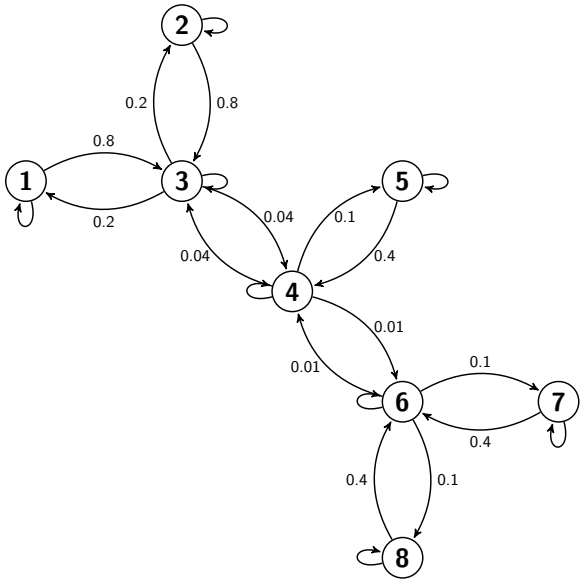


Figure 4.1: A simple eight state Markov process. Connected states are labeled with the pairwise rate constants, $\mathbf{K}_{ij}$. Self transition rates (not shown), $\mathbf{K}_{ii}$, are equal to the negative sum of each states outgoing transition rates, in accordance with Eq. (4.4).

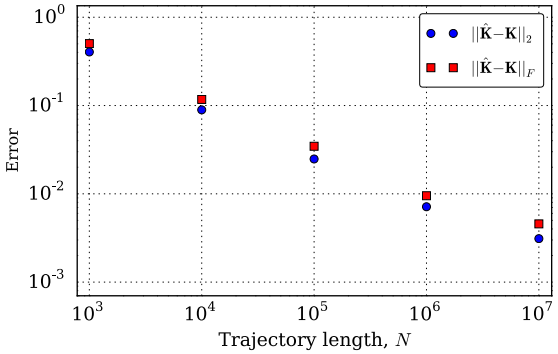


Figure 4.2: Convergence of the estimated rate matrix, $\hat{\mathbf{K}}$, to the true generating rate matrix in Fig. 4.1 for discrete-time trajectories of increasing length simulated from the process in Fig. 4.1 with a time step of 1. Using either a 2 norm (blue) or Frobenius norm (red), we see roughly power law convergence over the range of trajectory lengths studied.

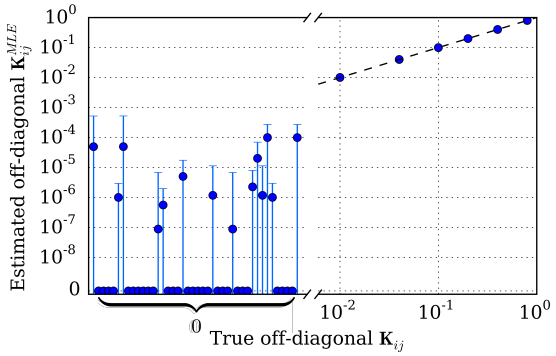


Figure 4.3: Comparison of the estimated and true off-diagonal rate matrix elements for a trajectory of length $N = 10^7$ simulated from the process in Fig. 4.1 with a time step of 1. The true non-zero elements of $\mathbf{K}$ are well-estimated, as shown in the right portion of the plot; here, error bars are small enough to be fully obscured by point markers. On the other hand, the estimator spuriously estimates non-zero rates between many of the states which are not connected in the underlying process. However, the 95% confidence intervals for these spurious rates each overlap with zero.

10^{-3}, which corresponds to a slowest exponential relaxation timescale, $\tau_2 \approx 106.4$ (arbitrary time units).

From this model, we simulated discrete-time data with a collection interval of 1 time unit by calculating the matrix exponential of $\mathbf{K}$ and propagating the discrete-time Markov chain. In Fig. 4.2, we show the convergence of the models estimated from this simulation data to the true model, as the length of the simulated trajectories grows. As expected, the fit parameters get more accurate as the size of the data set grows. We observe approximately power law convergence as measured by the 2-norm and Frobenius norm over the range of trajectory lengths studied.

The true rate matrix for this continuous time Markov process is sparse — only 7 of the 28 possible pairs of distinct states are directly connected in Fig. 4.2. Can this graph structure be recovered by our estimator? This task is challenging because of the nature of the discrete-time data. The observation that the system transitioned from state i (at time t) to state j (at time $t+1$) does *not* imply that $\mathbf{K}_{ij}$ is non-zero. Instead, the observed $i \to j$ transition may have been mediated by one or more other states — the process may have jumped from i to k, and then again from k to j, all

within the observation interval.

When the rate matrix, $\mathbf{K}$ is irreducible, the corresponding transition probability matrix $\mathbf{T}(\tau)$ is strictly positive for every positive lag time, τ.[156] This implies that in the limit that the trajectory length, N, approaches infinity, at least 1 transition count will almost surely be observed between any pair of states, regardless of the sparsity of $\mathbf{K}$.

In Fig. 4.3, we attempt to resolve the underlying graph structure using the model estimated with a trajectory of length $N = 10^7$. The plot compares the estimated rate matrix elements with the true values. We find that all of the true connections are well-estimated, and that many of the zero rates are also correctly identified. However, the maximum likelihood estimator also identifies very low, but non-zero rates between many of the states which are in fact disconnected.

We computed 95% (1.96σ) confidence intervals for each of the estimated rate matrix elements, $\mathbf{K}_{ij}^{MLE}$. For each of the spuriously non-zero elements, these confidence intervals overlapped with zero. None of the confidence intervals for the properly non-zero rates overlapped with zero. These uncertainty estimates can therefore be used, in combination with the MLE, to identify the underlying graph structure.

This example demonstrates that some degree of sparsity-inducing regularization or variable selection may be required to robustly identify the underlying graph structure in Markov process.

4.5.2 Accuracy of Uncertainty Estimates

How accurate are the approximate asymptotic uncertainty expressions derived in Section 4.4? To answer this question, we performed a numerical experiment with twenty independent and identically distributed collections of trajectories of Brownian dynamics on a two-dimensional potential. One of those trajectories is shown superimposed on the potential in Fig. 4.4, along with the 8×8 grid used to discretize the process. The Brownian dynamics simulations were performed following the same procedure described in McGibbon, Schwantes and Pande.[162]

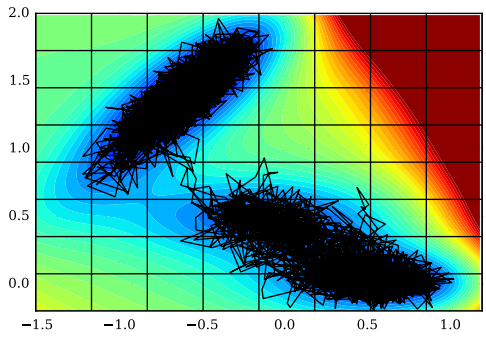


Figure 4.4: Brownian dynamics on the 2-dimensional Müller potential was discretized by projecting the simulated trajectories onto an 8×8 grid. A typical trajectory is shown in black. The resulting discrete-state process can be approximated as a continuous-time Markov process.

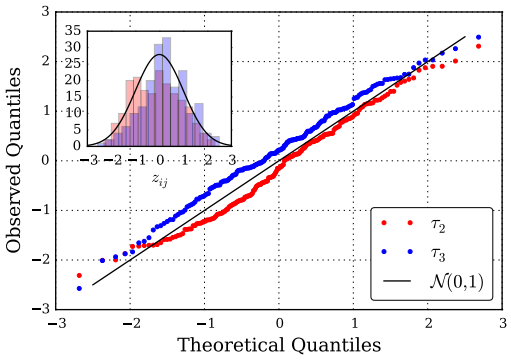


Figure 4.5: Quantile-quantile plot of the standardized differences, Eq. (4.36), between estimated relaxation timescales, τ_2 and τ_3, on twenty i.i.d. datasets. If the estimated timescales are normally distributed with the calculated asymptotic variances, the quantiles of their standardized differences would match exactly with the theoretical quantiles of the standard normal distribution.

To assess the accuracy of the asymptotic approximations, we compare the empirical distribution of the estimated parameters over the separate data sets with the theoretical distribution which would be expected based on the Gaussian approximation. Consider a scalar model parameter g, such as one of the relaxation timescales or equilibrium populations. Fitting a model separately on each of the twenty data sets yields estimates, $\{(\hat{g}_1, \sigma^2_{\hat{g}_1}), \ldots (\hat{g}_{20}, \sigma^2_{\hat{g}_{20}})\}$. If these estimates are accurate, then $\hat{g}$ is normally distributed, $\hat{g} \sim \mathcal{N}(g, \sigma^2_{\hat{g}})$. Our goal is to examine the accuracy of the estimated variances, $\sigma^2_{\hat{g}}$. Note that the true value of g is unknown, but subtracts out when examining standardized differences between the estimates, which, assuming normality, should follow a standard normal distribution,

$$z_{ij} = \frac{\hat{g}_i - \hat{g}_j}{\sqrt{\sigma^2_{\hat{g}_i} + \sigma^2_{\hat{g}_j}}} \overset{?}{\sim} \mathcal{N}(0, 1). \tag{4.36}$$

In Fig. 4.5, we compare the empirical and theoretical distributions of z_{ij}, $(i, j) : 1 \leq i \leq 20, i < j \leq 20$, for estimates of the first two relaxation timescales using a quantile-quantile (Q-Q) plot, a powerful method of comparing distributions. The observation that Q-Q plot runs close to the $y = x$ line is encouraging, and shows that the observed deviates are close to normally distributed, and that the approximator's variance estimates are of the appropriate magnitude. This suggests that the asymptotic error expressions can be of practical utility for practitioners.

4.5.3 Comparison with discrete-time MSMs

In a data-limited regime, are continuous-time Markov models more capable than discrete-time MSMs? We extended the analysis in Section 4.5.1 to a larger class of generating processes in order to address this question. We began by sampling random 100-state Markov process rate matrices from scale free random graphs.[163]. Details of the random rate matrix generation are described in Section 4.6.

From each random rate matrix, $\mathbf{K}$, we sampled three discrete-time trajectories of different lengths. Each trajectory was used individually to fit both a continuous-time and discrete-time Markov model, and the parameterized models were then compared

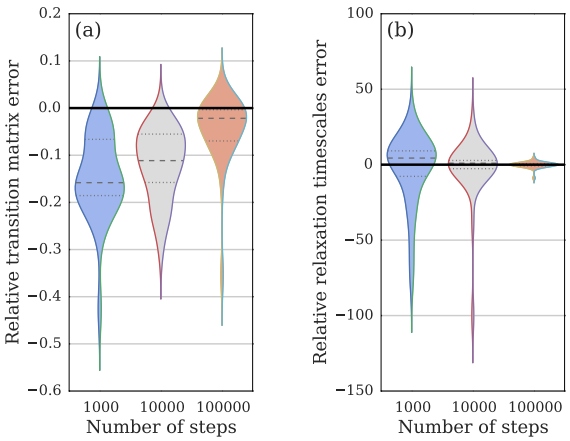


Figure 4.6: Violin plots of the relative error between continuous-time and discrete-time Markov models for kinetics on random graphs. Values below zero indicate lower error for the continuous-time model, whereas values above zero indicate the reverse. The shape displays the data density, computed with a Gaussian kernel density estimator. Panel (a): as measured by the Frobenius-norm error in the estimated transition matrices, $||\hat{\mathbf{T}} - \mathbf{T}||_F$, the continuous-time model achieves lower errors, with a larger advantage for shorter trajectories. Panel (b): as measured by the max-norm error in the estimated relaxation timescales, $\max_i |\hat{\tau}_i - \tau_i|$, the two models are not distinguishable.

to the underlying system from which the trajectories were simulated to assess the convergence properties of the approaches.

In Fig. 4.6, we consider two notions of error. The first norm measures error in the elements of the estimated transition matrix, $||\hat{\mathbf{T}} - \mathbf{T}||_F$. Unlike the experiment in Fig. 4.2, we used the $\hat{\mathbf{T}}$ as the basis of the measure so that the continuous-time and discrete-time models could be compared on an equal footing. The second error norm we consider is the max-norm error in the estimated relaxation timescales, $\max_i |\hat{\tau}_i - \tau_i|$, which measures a critical spectral property of the models. In both panels of Fig. 4.6, the distribution of the difference in error between the continuous-time and discrete-time models is plotted; values below zero indicate that the continuous-time model performed better for a particular class of trajectories, whereas values above zero indicate the reverse. For each condition, we performed $N = 30$ replicates.

Our results show that as measured by the transition matrix error, the continuous-time Markov process model is more accurate in the regimes considered. A binomial sign test rejects the hypothesis that the two estimators give the same error for all three conditions (two-sided p values of $[2 \times 10^{-9}, 2 \times 10^{-9}, 1 \times 10^{-3}]$ for trajectories of length $[10^3, 10^4, 10^5]$ steps, respectively). The relative advantage of the continuous-time Markov model decreases as the trajectory length increases — its advantage is in the sparse data regime when no transition counts have been observed between a significant number of pairs of states.

In contrast, as measured by the relaxation timescale estimation error, we observe no significant difference between the continuous-time and discrete-time estimators. A binomial sign test does not definitively reject the hypothesis that the two estimators give the same error for any of the three conditions (two-sided p values of $[0.02, 0.36, 0.85]$ for trajectories of length $[10^3, 10^4, 10^5]$ steps, respectively). Neither estimator is consistently more accurate in recovery of the dominant spectral properties of the dynamics.

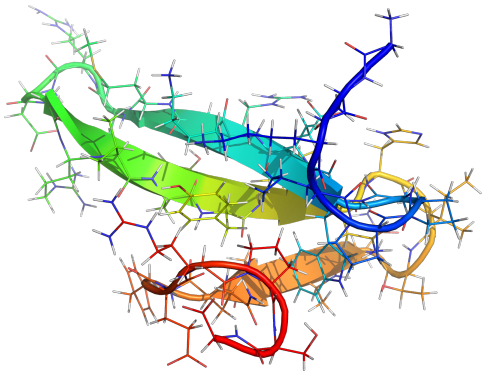


Figure 4.7: The FiP35 WW protein, in its native state. We analyzed two $100\mu s$ MD trajectories of its folding performed by D.E. Shaw Research to estimate a Markov process model for its conformational dynamics.[91]

4.5.4 Application to Protein Folding and Lag Time Selection

How can these models be applied to the analysis of molecular dynamics (MD) simulations of protein folding? We obtained two independent ultra-long $100\mu s$ MD simulations of the FiP35 WW protein,[164] a small 35 residue β-sheet protein (Fig. 4.7), performed by D.E. Shaw Research on the ANTON supercomputer.[91]

In order to focus on the construction of discrete-state Markov models, we initially projected every snapshot of the MD trajectories, which were available at a 200 ps time interval, into a discrete state space with 100 states in a way consistent with prior work.[162] Briefly, this involved the extraction of the distance between the closest non-hydrogen atoms in each pair of amino acids in each simulation snapshot,[165] followed by the application of time-structure independent components analysis (tICA) to extract the four most slowly decorrelating degrees of freedom,[166,167] which were then clustered into 100 states using the k-means algorithm.[168,169]

Although the equations of motion for a protein's dynamics in an MD simulation are Markovian, the generating process of the data analyzed by our model is not. The pre-processing procedure which projects the original dynamics from a high-dimensional continuous state space (the position and momenta of the constituent atoms) into a lower dimensional continuous space or discrete state space is not information preserving, and destroys the Markov property.[82,145] For chemical dynamics,

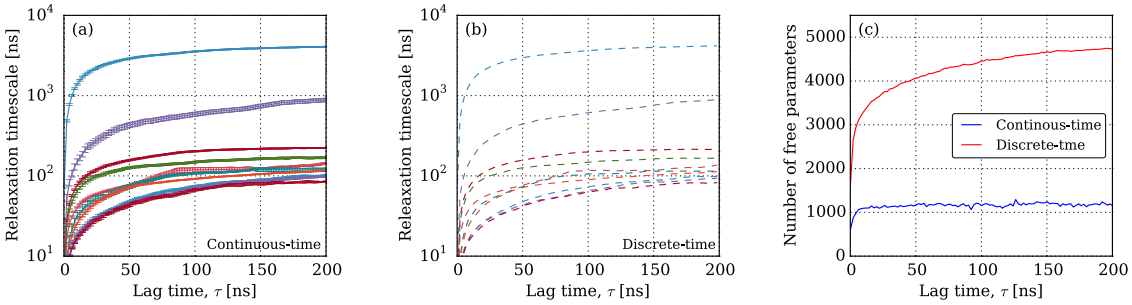


Figure 4.8: Implied exponential relaxation timescales of parameterized (a) continuous-time Markov process model and (b) discrete-time Markov model as a function of lag time. The relaxation timescales computed by the two algorithms coincide almost exactly ($r^2 = 0.999978$). (c) The number of free (non-zero) parameters estimated by the discrete-time and continuous-time models respectively; the continuous-time Markov model achieves a more parsimonious representation of the data.

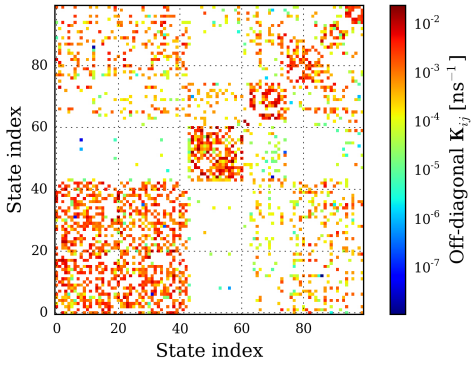


Figure 4.9: The maximum likelihood rate matrix, $\hat{\mathbf{K}}$, computed at a lag time, τ, of 100 ns. The state indices were sorted in seven macrostates using the PCCA algorithm.[57]

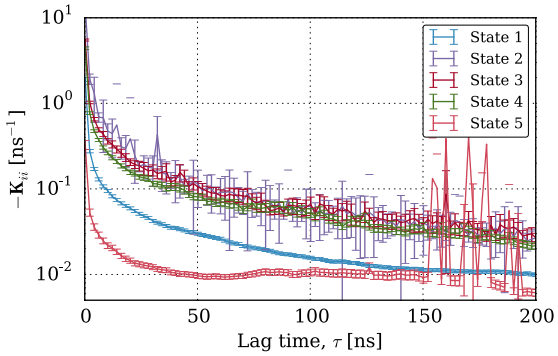


Figure 4.10: Convergence of selected rate matrix elements as a function of lag time. A plausible method for lag time selection would be to choose τ such that some or all of these entries are determined to have plateaued.

qualitative features of the non-Markovianity are well-understood. Consider, for example, a metastable system with two states, A and B, the system in state A may stochastically oscillate across the boundary surface many times without committing to state B. Whereas for a Markov process, the probability distribution of the waiting time that the system spends in any states before exiting is exponential, chemical dynamics are expected to show a higher propensity for short waiting times, corresponding to so-called recrossing events.[144] This effect is more pronounced when viewing the process at short lag times — the bias induced by approximating the process as Markov decreases with lag time.[85]

For the FiP35 WW domain, we observe that the change in the relaxation timescales of the continuous-time and discrete-time Markov models with respect to lag time are essentially identical, as shown in Fig. 4.8. For both model classes, the estimated relaxation timescales increase and converge with respect to lag time. This is consistent with our results in Fig. 4.6 (b), which suggest that the estimated timescales are the same for both models, especially as the length of the trajectories grow. While fitting the models in Fig. 4.8, we observed a small number (2-4) of convergence failures at long lag times, which were notable due to a dramatic discontinuity in the relaxation timescales curve. This problem was solved by reinitializing the optimization at these lag times from the converged solutions at adjacent lag times.

Because of the essentially unchanged nature of the relaxation timescale spectrum, we suggest that when choosing a particular lag time, the same approach be used for discrete-time and continuous-time Markov models. Ideally, this entails the selection of a lag time large enough that the relaxation timescales are independent of lag time.[170,171] For the continuous-time Markov model, other techniques may be appropriate as well. For example, in Fig. 4.10 we show the convergence of selected diagonal entries of the rate matrix as a function of lag time. As described in the context of transition state theory, these rate constants should plateau with increasing τ, which provides another related basis on which select the parameter.[172,173]

The most significant difference between the continuous-time and discrete-time estimators in this case is the sparsity of the parameterized models. In Fig. 4.8(c), we compare the number of non-zero independent parameters for both models as a function of τ. Of the $\binom{n+1}{2} = 5050$ independent parameters for both the continuous-time and discrete-time models, only $\approx$ 1200 are nonzero for the continuous-time model, regardless of lag time. In contrast, the number of nonzero parameters for the discrete-time model model continues to increase with lag time.

We anticipate that the sparsity of $\mathbf{K}$ may aid in the analysis and interpretation of Markov models. In Fig. 4.9, we show the MLE rate matrix computed at $\tau = 100\ ns$. The state indices were sorted such that states grouped together via Peron cluster cluster analysis (PCCA) were given adjacent indices.[57] The evident block structure of the matrix visually indicates that the protein's conformational space consists of a small number of regions with generally high within-region rate constants, but weak between-region coupling. Although a detailed analysis of the biophysics of these conformations is beyond the scope of this work, visual analysis of these structures indicate that the model resolves folded and unfolded, as well as partially folded intermediate states.

In interpreting the 1.96σ error bars on the relaxation timescales in Fig. 4.8(a), cautionary note is warranted. Our error analysis considers the number of observed transitions between states but does not take into account any notion of uncertainty in the proper definitions of the states themselves, or the error inherent in approximating a non-Markovian process with a Markov process. The observation in Fig. 4.8(a) that

the magnitude of the systematic shift in the timescales with respect to lag time is much larger than the error bars suggests that the Markov approximation (a model misspecification) is a larger source of error, for this dataset, than the statistical uncertainty in model parameters. For these reasons, we caution that these error bars should be interpreted as lower bounds rather than upper bounds.

4.5.5 Performance

In order to assess the performance of our maximum likelihood estimator, we compared it with an algorithm by Holmes and Rubin, which solves the same Markov process parameterization problem using an expectation-maximization approach.[139] Because the original code was unavailable, we reimplemented the algorithm following the description by Metzner et al., where it is denoted "Algorithm 4: Enhanced MLE-method for the reversible case".[138] The algorithm scales as $O(n^5)$, where n is the number of states. Its rate limiting step involves an $O(n^5)$ FLOP contraction of five $n \times n$ matrices into a rank-4 tensor of dimension n on each axis. * For benchmarking, we constructed a variant of the the FiP35 WW protein dataset from Section 4.5.4, in which we varied the number of states between 10 and 100 during clustering. All models were fit on an Intel Xeon E5-2650 using a single CPU core.

As shown in Fig. 4.11, and expected on the basis of the $O(n^5)$ vs. $O(n^3)$ scaling, the performance difference between the algorithms is substantial. For $n = 100$, our algorithm is roughly four orders of magnitude faster per iteration; our algorithm takes on the order of 1 ms per iteration, while the Holmes-Rubin estimator's iteration takes over 10 seconds. Using the L-BFGS-B optimizer's default convergence criteria, roughly three quarters (68/91) of the runs of our algorithm converge in fewer than 100 iterations; a solution is often achieved long before the EM estimator has performed a single iteration.

*Both our algorithm and Holmes-Rubin estimator were implemented the Cython language and compiled to C++. Our implementation of the Holmes-Rubin estimator is available at `https://github.com/rmcgibbo/holmes_rubin`

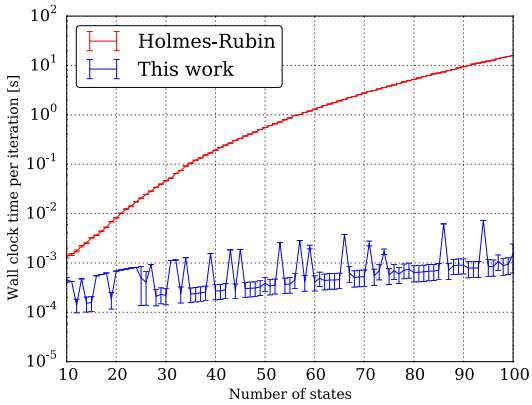


Figure 4.11: Performance of our Markov process estimator, as compared to the Holmes-Rubin EM estimator.[139] Each iteration of our $O(n^3)$ estimator takes on the order of 1 ms, while the $O(n^5)$ Holmes-Rubin estimator takes over 10 seconds per iteration for a 100 state model. Using default convergence criteria, our estimator often achieves a solution long before the EM estimator finishes a single iteration.

4.6 Conclusions

In this work, we have introduced a maximum likelihood estimator for continuous-time Markov processes on discrete state spaces. This model can be used to estimate transition rates between various substates in a dynamical system based on observations of the system at a discrete time interval. Various constraints on the solution, such as detailed balance, can be easily incorporated into the model, and asymptotic error analysis can give confidence intervals in model parameters and derived quantities.

With the efficient parameterization problem solved, these continuous-time Markov models offer several advantages over existing MSM methodologies. As compared to discrete-time MSMs, these models are more interpretable for chemists and biologists because they do not arbitrarily discretize time. Although a lag time is used internally during parameterization, the final estimated quantities are familiar rate constants from chemical kinetics, as opposed to the somewhat unintuitive transition probabilities in a discrete-time MSM. Furthermore, these models are more parsimonious, and unlike the discrete-time MSM are able to detect that many pairs of states are not immediately kinetically adjacent to one another. This makes it possible to more

clearly recover the underlying graph structure of the kinetics. For applications such as the determination of transition pathways in protein dynamics, we anticipate that this property will be valuable.

Many extensions of this model are possible in future work. The simple nature of the constraints on θ make Bayesian approaches, especially Hamiltonian Monte Carlo, particularly attractive.[174] In particular, because of the separation of $\theta^{(\pi)}$ and $\theta^{(S)}$ in the parameterization, strong informative priors on π may be added to extend the work of Trendelkamp-Schroer and Noé.[175] The appropriate sparsity inducing priors on $\theta^{(S)}$ may be a topic of future work.

An implementation of this estimator is available in the MSMBuilder software package at `http://msmbuilder.org/` under the GNU Lesser General Public License.

Acknowledgements

This work was supported by the National Science Foundation and National Institutes of Health under Grant Nos. NIH R01-GM62868, NIH S10 SIG 1S10RR02664701, and NSF MCB-0954714. We thank Mohammad M. Sultan, Matthew P. Harrigan, John D. Chodera, Kyle A. Beauchamp, Ariana Peck, and the reviewers for helpful feedback during the preparation of this work.

Random rate matrices

Scale-free random graphs with 100 states were generated using the Barabási–Albert preferential attachment model with $m = 3$.[163] From the graph's adjacency matrix, we generated a symmetric rate matrix S by sampling a log-normally distributed random variable ($\mu = -3$, $\sigma = 2$) for each connected edge. The stationary distribution, π, was sampled from Dirichlet($\alpha = 1$). The matrix S was then scaled by $50 \cdot (\sum_{ij} S_{ij})^{-1}$, which tuned the relaxation timescales in the range between 10^2 and 10^3 time steps, and used with π in Eq. (4.25) to construct $\mathbf{K}$.

Chapter 5

Statistical Model Selection for Markov Models

Markov state models provide a powerful framework for the analysis of biomolecular conformation dynamics in terms of their metastable states and transition rates. These models provide both a quantitative and comprehensible description of the long-timescale dynamics of large molecular dynamics with a Master equation, and have been successfully used to study protein folding, protein conformational change, and protein-ligand binding. However, to achieve satisfactory performance, existing methodologies often require expert intervention when defining the model's discrete state space. While standard model selection methodologies focus on the minimization of systematic bias and disregard statistical error, we show that by consideration of the states' conditional distribution over conformations, both sources of error can be balanced evenhandedly. Application of techniques that consider both systematic bias and statistical error on two $100\mu s$ molecular dynamics trajectories of the Fip35 WW domain shows agreement with existing techniques based on self-consistency of the model's relaxation timescales, with more suitable results in regimes in which those timescale-based techniques encourage over-fitting. By removing the

> need for expert tuning, these methods should reduce modeling bias and lower the barriers to entry in Markov state model construction.

This chapter is adapted the permission from McGibbon, R. T.,† Schwantes, C. R.† and Pande, V. S., Statistical Model Selection for Markov Models of Biomolecular Dynamics. *J. Phys. Chem. B* **2014**. *118(24)*, pp 6475-6481.[162] Copyright 2013 American Chemical Society. †R.T.M. and C.R.S. contributed equally to this work.

5.1 Introduction

Proteins are highly complex molecular machines, and their dynamics are an essential aspect of biomolecular function. These dynamics span a wide range of length scales, timescales and complexity, including folding and aggregation, conformational change between functional native sub-states, ligand binding, and allostery.[176–179] Whereas classical experimental probes have often been interpreted in two-state frameworks, ensemble measurements with increasingly high temporal resolution as well as sensitive single molecule probes have uncovered a vast array of complex multi-state kinetics.[180,181] But the atomic-resolution characterization of these dynamics is often an enormous challenge — as molecular probes like Förster resonance energy transfer, small-angle x-ray scattering, and nuclear magnetic resonance techniques measure complex projections of the intrinsic structure, generally reporting simultaneously on many degrees of freedom.[182–184]

Computer simulations can complement experiments by providing atomic insight into conformational dynamics. With advances at the algorithmic, hardware, and software levels, modern molecular simulation paradigms, incorporating specialized or accelerated hardware, often in combination with highly parallel distributed computing frameworks, are capable of generating extensive simulation ensembles with relative ease.[25,26,185–188] A central challenge in the field of molecular simulations is now often the kinetic analysis, which typically involves constructing a lower resolution model that captures the system's essential features in an interpretable framework.[19,28] For example, by projecting the data down onto one or two degrees of freedom we create

a simpler model for the system, such as one characterized by diffusion along a single reaction coordinate.[189]

Markov state models (MSMs) are one approach for analyzing MD data sets that are able to smoothly move between high and low-resolution models.[59,97,190,191] High-resolution models maintain quantitative agreement with the underlying simulation data, while low-resolution models capture the salient features of the potential energy landscape, sacrificing some degree of model complexity. In an MSM, the dynamics are modeled as a memory-less jump process between a discrete set of conformational states which partition the phase space. Two key quantities that define the MSM are thus the state definitions, an indicator function basis over phase space, and the pairwise transition probabilities, which parameterize the kinetics. The matrix of transition probabilities can be used to locate the systems transition paths, and its dominant eigenvectors identify the metastable states and long-timescale dynamical modes.[57,192]

Two competing sources of error govern the accuracy of an MSM. The first source of error is systematic and is common to all dimensionality reduction methods, in that simplified representations of complex systems discard information. The second source of error is statistical in nature, in that the parameters of an MSM must be estimated from a finite number of stochastic MD trajectories. As we discuss below, these two sources of error compete in a bias-variance tradeoff.[193] Given a finite size data set, as the number of states increases, the systematic error (the bias) decreases, while the statistical error (the variance) increases.

Here, we seek to build models that are *suitably* complex, given the data, yielding complex descriptions of the system only to the extent that their additional parameters are implied by the observed dynamics. To that end, we introduce a procedure for scoring the likelihood of an MSM, which, together with standard statistical model selection techniques, enables a balanced selection of the model's parameters.

5.2 Theory

5.2.1 Markov State Models

Let $\mathbf{x}_t$ be a ergodic time-homogeneous Markov process the phase space Ω with stationary density $\mu(\mathbf{x})$. Consider an ensemble of such processes at time t, described by a distribution $p_t(\mathbf{x})$. There exists an operator, termed the propagator, that evolves the a distribution forward in time at discrete intervals, τ.

$$p_{t+\tau}(\mathbf{x}) = \mathcal{Q}^{(\tau)} \circ p_t(\mathbf{x}) \tag{5.1}$$

The eigenfunctions of the propagator form a basis for $p(\mathbf{x})$, and the time evolution of the ensemble after n applications of the propagator can be decomposed into a sum of terms along each eigenfunction.

$$p_{t+n\tau}(\mathbf{x}) = \sum_{i=1}^{\infty} \exp\left(-\frac{n \cdot \tau}{t_i}\right) \langle \phi_i, p_t \rangle_{\mu^{-1}} \phi_i(\mathbf{x}), \tag{5.2}$$

where $t_i = -\dfrac{\tau}{\ln \lambda_i}$ are the propagator's relaxation timescales and ϕ_i and λ_i are the eigenfunctions and eigenvalues of the propagator, and $\langle \cdot, \cdot \rangle_{\mu^{-1}}$ denotes the inner product defined as

$$\langle \phi_i, p_t \rangle_{\mu^{-1}} = \int_{\Omega} d\mathbf{x}\ \phi_i(\mathbf{x}) p_t(\mathbf{x}) \frac{1}{\mu(\mathbf{x})}. \tag{5.3}$$

For a more detailed discussion of the propagator and its properties, we refer the reader to Prinz et al.[97]

For classical MD, $\mathcal{Q}^{(\tau)}$ is determined by the Hamiltonian and equations of motion.[36] However, it is not practical to explicitly construct the propagator from the Hamiltonian for complex systems. A Markov model provides an estimate for the propagator. A k-state Markov model is defined on a discrete set of states, $S = \{s_i\}_{i=1}^k$, where $s_i \subseteq \Omega$ and $\bigcup_{i=1}^k s_i = \Omega$. Furthermore, $s_i \cap s_j = \emptyset$ for all $i \neq j$. In words, every point in Ω is assigned to one (and only one) state in the MSM. Let $\sigma(\mathbf{x})$ be the function that maps a point $\mathbf{x} \in \Omega$ to the index i of the state in S such that $x \in s_i$.

The MSM models the dynamics of the propagator with a Markov jump process

on $\{1, \ldots, k\}$. Let $\mathbf{p} \in (\mathbb{R} \cap [0, 1])^k$ be a column vector whose entries sum to one. The elements in $\mathbf{p}$ are defined by a coarse-graining of $p(\mathbf{x})$ over S, as

$$p_i = \int_{\mathbf{x} \in s_i} d\mathbf{x}\, p(\mathbf{x}). \tag{5.4}$$

Consider a probability vector at time t denoted $\mathbf{p}(t)$. The time-evolution of $\mathbf{p}(t)$ is described by

$$\mathbf{p}(t + \tau)^T = \mathbf{p}(t)^T \mathbf{T}, \tag{5.5}$$

where T_{ij} is the probability of transiting to state j in time τ given that the system started in state i. With this construction, the eigenvalues and eigenvectors of $\mathbf{T}$ provide approximations for the eigenvalues and eigenfunctions of $\mathcal{Q}$. In a direct analogy with Eq. (5.2), the eigenvalues λ_i of T correspond to relaxation timescales of the Markov model

$$t_i = -\frac{\tau}{\ln \lambda_i}. \tag{5.6}$$

Let D be a set of molecular dynamics trajectories in Ω. To construct an MSM, both S and $\mathbf{T}$ must be determined from the data. The state-space, S, can be constructed by clustering the points in D or via a grid-based discretization of Ω. Given S, maximum-likelihood estimators for the $O(k^2)$ elements of a reversible transition probability matrix $\mathbf{T}$ exist.[75,97]

Using this procedure, it has been shown that MSM estimates of the eigenvalues λ_i are systematically underestimated in the limit of infinite data, and that the magnitude of this bias goes to zero as $k \to \infty$. This fact has inspired the development of variational methods for model selection, which optimize the MSM parameters in order to maximize the eigenvalues. [194]

Selecting the size of S is an important step in model construction, however there is not currently a widely-accepted method in use. Heuristic criteria, such as manually selecting $k = n_{\text{conformation}}/14$ or clustering D such that each conformation is within a pre-specified distance to its cluster center (e.g. 4.5Å or 1.2Å RMSD) have been previously employed.[59,77,195] Selection of τ based convergence of the relaxation timescales (Eq. (5.6)), as suggested by Swope et al. (2004)[170] and others is more well

established. The rate of this convergence with respect to τ at fixed S has also been used as a metric for evaluating a model's state decomposition.[196]

5.2.2 The Bias-Variance Dilemma

For MSM methods to be valuable to practitioners, they must operate in limited data regimes. Thus, we seek a method for model selection that is sensitive to both the systematic error discussed above, as well as the statistical error associated with estimating the model's parameters. Whereas model selection based on the variational principle described above would tend to increase the number of states without bound, as the magnitude of the eigenvalue bias decreases with the number of states, practical application of this criteria would cause a catastrophic increase in statistical uncertainty in the model. This tradeoff between low-variance (small k) but biased estimators, and high-variance (large k) but unbiased estimators is a general feature of problems in statistical learning.[193] For example, it can be shown that the uncertainty in the posterior distribution of $\mathbf{T}$ given S and D approaches infinity as k approaches infinity.

Introduction to MSMs via the propagator theory motivates the view of the parametrization problem as one of *function estimation*. This perspective has a critical disadvantage: the natural error metric — the difference between the estimated MSM and true propagator is intrinsically uncomputable. The only access to the true propagator is via samples from the Hamiltonian with MD. Therefore, a compelling alternative is to view the problem as one of *prediction*. Since the MSM is a generative model for trajectories in Ω, it is possible to sample "pseudo-trajectories" directly from the MSM. If the MSM were perfect, these "pseudo-trajectories" would be indistinguishable from trajectories generated by the true propagator. Thus, the probability assigned by the MSM, to samples from the true propagator is a measure of the MSM's accuracy. This argument is formalized by Bayes' rule, which establishes a proportionality between the probability of a model given data to the probability of the data given the model

$$P\Big(\{S, \mathbf{T}\} \;\Big|\; D\Big) = P\Big(D \;\Big|\; \{S, \mathbf{T}\}\Big) \frac{P(\{S, \mathbf{T}\})}{P(D)}, \tag{5.7}$$

where $P\Big(\{S, \mathbf{T}\} \;\Big|\; D\Big)$ is the posterior probability of an MSM, $P\Big(D \;\Big|\; \{S, \mathbf{T}\}\Big)$ is the *likelihood* of the data given a model, $P(\{S, \mathbf{T}\})$ is the prior probability of an MSM before observing any data, and $P(D)$ is the probability of the data, a constant.

5.2.3 Likelihood of a Markov State Model

Bayes' rule provides a foundation for model selection, by establishing the proportionality of the probability of a model given the data to the probability of the data given the model. Without a strong prior, model selection is reduced to the search for the model that maximizes the likelihood of the data. With $\{\mathbf{x}_t\}_{t=1}^N$ an observed trajectory of length N, the likelihood can be written as:

$$P\big(\{\mathbf{x}_t\}_{t=1}^N \;\big|\; \{S, T\}\big) d\mathbf{x}^N = \pi_{\sigma(\mathbf{x}_1)} \prod_{t=1}^{N-1} T_{\sigma(\mathbf{x}_t),\sigma(\mathbf{x}_{t+1})} \cdot \prod_{t=1}^{N} P(\mathbf{x}_t|\sigma(\mathbf{x}_t)) \cdot d\mathbf{x}^N, \qquad (5.8)$$

where π_i is the stationary probability of state i according to $\mathbf{T}$ and $P(\mathbf{x}_t|\sigma(\mathbf{x}_t))$ is the probability of sampling a conformation $\mathbf{x}_t$ given that the process is in the state $\sigma(\mathbf{x}_t)$, referred to as the emission distribution of state $\sigma(\mathbf{x}_t)$. There is some flexibility in the choice of emission distribution, though Eq. (5.8) is valid only when the emission distributions satisfy

$$P(\mathbf{x}|i) = 0 \quad \forall\, \mathbf{x} \text{ s.t. } \sigma(\mathbf{x}) \neq i. \qquad (5.9)$$

An alternative approach involving fuzzy clustering, where conformations are not uniquely assigned to a single state, would not impose the constraint in Eq. (5.9), and would also involve a likelihood function that marginalizes over all possible paths.[197] In this work, we consider only models based on crisp partitioning. As such, given the coarse-graining of $p(\mathbf{x})$ in Eq. (5.4), the most natural choice emission distributions are normalized indicator functions

$$P(\mathbf{x} \mid i) = \frac{1}{V_i} \mathbf{1}_{s_i}(\mathbf{x}). \qquad (5.10)$$

An alternative emission distribution was proposed by Kellog, Lange and Baker (2012).[198] Instead of having support on Ω, this distribution is supported on the training set

$\{\mathbf{x}_t\}_{t=1}^N$. It is given by

$$P(\mathbf{x} \mid i) = \left|\{\mathbf{x}_k : \sigma(\mathbf{x}_k) = \sigma(\mathbf{x})\}\right|^{-1} \sum_{t=1}^{N} \delta(\mathbf{x} - \mathbf{x}_t), \tag{5.11}$$

where $|\cdot|$ denotes the cardinality of a set and $\delta(\cdot)$ is the Dirac delta function. As this is a discriminative model, it is unable to generalize and assign probability to new data. In certain circumstances, this is an undesirable property. For example, protocols that involve fitting and validating models on separate data sets (e.g. cross-validation) are impossible when the statistical model lacks the capacity to describe new data.

5.2.4 Statistical Model Selection

Likelihood maximization on the training data (i.e. the data used to fit the model) is insufficient for model selection when the number of parameters varies between proposed models, as more complex models generally exhibit higher empirical likelihoods, often at the cost of larger generalization errors due to over-fitting.[199,200] Statistical learning theory provides a number of methods to overcome this problem. Conceptually, the most straightforward is a full Bayesian treatment in which all unknown model parameters are represented by probability distributions. The evidence for a k-state model is computed by formally integrating Eq. (5.7) over the model parameters and the evidence ratio, or Bayes factor,[201] then provides a rigorous basis of model selection that appropriately punishes overly complex models as they become poorly constrained in parameter space. Unfortunately such approaches are impractical for problems of this size because of the need to integrate over all possible Markov models of a given size.

Instead, we explore three alternative procedures for choosing the number of states in an MSM: cross validation, Schwarz's Bayesian information criterion (BIC)[202] and the Akaike information criterion (AIC).[203] Cross-validation attempts to directly measure the Markov model's generalization error. First, the model is parameterized by building both the state space and transition matrix on a subset of the available molecular dynamics trajectories, then the likelihood is evaluated on the left-out portion.

This scheme can be repeated many times with different partitions of the data set.

$$\mathrm{BIC} \equiv -2 \cdot \ln L + (\ln N) \cdot \kappa \tag{5.12}$$

$$\mathrm{AIC} \equiv -2 \cdot \ln L + (2) \cdot \kappa \tag{5.13}$$

where L is the maximum likelihood, κ is the number of free parameters, and N is the number of data points. Model selection is performed by a minimization of the criterion. The BIC is derived based on the dominant terms in the Laplace approximation to the logarithm of the Bayes factor with a vague prior, while the functional form of the AIC comes from an asymptotic approximation to the Kullback-Leibler divergence between the model and the true distribution.[204,205] The appearance of number of data points N in Eq. (5.12) assumes that the data is independent and identically distributed given the model, which is often poorly justified for time series. Nonetheless, the value in the AIC and BIC comes from their simple form, that they do not require leaving out portions of the available data during fitting, and acceptable performance in practice.[206]

5.3 Methods

5.3.1 Volume Estimation

The uniform distribution emission model presents a computational challenge: its use requires the calculation of the (hyper)volume of the MSM's states, which, when defined by clustering, are high-dimensional Voronoi cells. While trivial in two or three dimensions, this computational geometry task becomes challenging in high-dimensional settings. The computation of such volumes has occupied significant attention in recent years in the computational geometry literature, especially via randomized algorithms.[207–209] We opt to approximate the volumes using Monte Carlo integration, which we find tractable for large systems only when the molecular dynamics data set is first projected into a suitable small vector space of up to perhaps ten dimensions.

A further challenge is the description of the states that are at the edge of the MSM — whose Voronoi cells extend to infinity in at least one direction. In these cases, the Voronoi cells are of unbounded volume. Instead we wish to truncate these states by bounding them by the convex hull of the data set. Because the convex hull of our simulation data sets are computationally inaccessible, we defined the accessible volume of Ω denoted A, to be the set of all points within a distance R within a set of test points Y.

Algorithm 1 Monte Carlo Estimation of the State Volumes

procedure MC VOLUME ESTIMATION(σ, A, M)
  $\mathbf{c} \leftarrow [0, 0, \ldots, 0]$ ▷ $\mathbf{c}$ is a vector of length k
  while $\sum_i c_i \leq M$ **do**
    $\mathbf{x} \leftarrow$ sample from axis-aligned hyper cube containing A
    if $\mathbf{x} \in A$ **then**
      $c_{\sigma(\mathbf{x})} \leftarrow c_{\sigma(\mathbf{x})} + 1$
    end if
  end while
  $\mathbf{v} \leftarrow \mathbf{c} \cdot (\sum_{i=1}^{k} c_i)^{-1}$ ▷ $\mathbf{v}$ contains the relative volumes of each state
  return v
end procedure

The result of this scheme produces the volumes of each state relative to the volume of A. It's important, therefore, to use the same set A when comparing multiple models.

5.3.2 Simulation Protocol and MSM Construction

Two systems were investigated using this likelihood scheme (Fig. 5.1). The first was a simple two dimensional surface with three energy minima, called the Müller potential. The dynamics are governed by

$$\frac{d\mathbf{x}}{dt} = -\nabla V(\mathbf{x})\zeta + \sqrt{2kT\zeta}R(t),$$

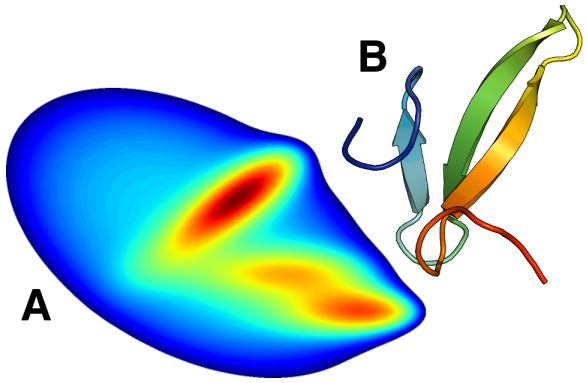


Figure 5.1: Systems studied in this chapter. (A) Brownian dynamics on the two-dimensional Müller potential.[210] (b) 200 μs of dynamics of the Fip35 WW domain[164], courtesy of D.E. Shaw research.[91]

where $\zeta = 10^{-3}$, $kT = 15$, $R(t)$ is a delta-correlated Gaussian process with zero mean, and $V(\mathbf{x})$ was defined as

$$V(\mathbf{x}) = \sum_{j=1}^{4} A_j \cdot \exp\Big(a_j(x_1 - X_j)^2 + b_j(x_1 - X_j)(x_2 - Y_j) + c_j(x_2 - Y_j)^2\Big),$$

where $a = (-1, -1, -6.5, 0.7)$; $b = (0, 0, 11, 0.6)$; $c = (-10, -10, -6.5, 0.7)$; $A = (-200, -100, -170, 15)$; $X = (1, 0, -0.5, -1)$; $Y = (0, 0.5, 1.5, 1)$ as suggested by Müller and Brown (1979).[210] Using the Euler-Maruyama method and a time step of 0.1, we produced two trajectories of length 10^6 time steps. The initial positions were sampled via a uniform distribution over the box: $[-1.5, 2.0] \times [-0.2, 2.0]$.

The first trajectory was clustered using the k-centers clustering algorithm with the Euclidean distance. State volumes were computed for the uniform emission model using $M = 10^5$ rounds of Monte Carlo integration defining the set A using the cluster centers from the 50-state model as the test points Y with a cutoff of $R = 0.28$. The second trajectory was used as a test set. All MSMs were built using a lag time of 30 steps.

Next, we reanalyzed two ultra-long 100 μs molecular dynamics trajectories of the Fip35 WW domain,[164] provided courtesy of D.E. Shaw Research [91] (Amber ff99SB-ILDN force field,[211] TIP3P water model.[212]) We projected the trajectories into a

four dimensional space using time-structure based independent components analysis (tICA) [166,167] on the first trajectory. To calculate the tICs, each conformation was represented as a vector of distances between all pairs of residues separated by at least three amino acids. The distance $d(A, B)$ between residues A and B was determined as follows. Let $\{a_i\}_{i=1}^{n_a}$ and $\{b_i\}_{i=1}^{n_b}$ be the Cartesian coordinates of the heavy (non-hydrogen) atoms in A and B. Then we define

$$d(A, B) \equiv \min_{i,j} ||a_i - b_j||_2,$$

where $||\cdot||_2$ denotes the ℓ_2 norm. The tICs were computed using a correlation lag time of 200 ns. We also performed Principal Components Analysis (PCA) on the same residue-residue distance representation and built MSMs using the top three PCs. In each projection, k-centers was used with the Euclidean distance metric.

For the tICA MSMs, the cluster centers from the 100 state model were used as the test points Y to define A with a cutoff of $R = 0.81$. In the PCA MSMs, the cluster centers from the 500 state model were used as the test points with a cutoff of $R = 2.0$. In each case, $M = 10^6$ rounds of Monte Carlo integration were performed to compute the state volumes. All WW MSMs were built using a lag time of 50 ns, which is the same lag time used in previous analyses.[59]

For both the Müller potential and WW models, a maximum likelihood estimator for reversible transition matrices, with a pseudocount of $1/k$ was used to compute $\mathbf{T}$. The pseudocount ensures that all transitions are assigned nonzero probability, which is especially important for evaluating data that was not used to train the model, as often new transitions will be observed. All analysis and model construction was performed using MSMBuilder. [75]

The BIC and AIC were calculated according to Eq. (5.12) and Eq. (5.13), with N equal to the total number of transitions observed in the data. However, as transitions were counted using the "sliding window" approach, these transitions are not statistically independent.[97] Therefore, the BIC may be over-penalizing the number of parameters due to this over-estimate of N.

5.4 Results and discussion

5.4.1 Müller Potential

How well can likelihood-based methods select parameters for an MSM in the data-rich regime? To answer this question, we simulated a simple two-dimensional potential and built a series of MSMs (see *Methods*). We compared four different model selection criteria: (1) test log-likelihood (2) BIC (3) AIC and (4) implied timescales convergence. The analysis (Fig. 5.2) indicates that the log-likelihood on the training data set continually increases with the addition of further states. The increase is most dramatic at low k, and levels off at high k, indicating that once k is large enough, the marginal increase in the likelihood with respect to further states is small. For this data set, the log-likelihoods computed on a separate test set exhibit a notably similar trend. Above $k = 600$, the test log-likelihood stops increasing. These two trends are to be expected: the additional parameters in the model at higher k never decreases the model's ability to fit the training data. On the other hand, this can lead to the model fitting the statistical noise in the training data set, rather than the system's true dynamics. As result, these "over fit" models at high k are less predictive, as measured by the log-likelihood they assign to new test data.

Leaving out a portion of a molecular dynamics data set from the training to serve as a test set is costly proposition for the applications of MSMs to the study of complex system. For large biomolecular simulations characterized by long intrinsic timescales, sampling the potential exhaustively is a significant challenge,[19] making it difficult to afford discarding half of the data set during the fitting of an MSM. For this reason, we also computed two penalized likelihood model selection criteria, the AIC and BIC, which augment the training set log-likelihood with an explicit penalty on model complexity to avoid over fitting. Applied to our simulations of the Müller potential, both the AIC and BIC penalize model complexity more strongly than the test set log-likelihood.

The likelihood-based methods are consistent with model selection based on maximization of the implied timescales. For this simple data set, the timescales are maximized by models with 200 states, which agrees with the results obtained with

BIC. Model selection based on direct maximization of the implied timescales considers only the systematic error in the MSM, and neglects the statistical error. The consistency between the two approaches on this data set is an indication that the statistical uncertainty is low for these models, which is to be expected given the ease of sampling this two dimensional toy potential.

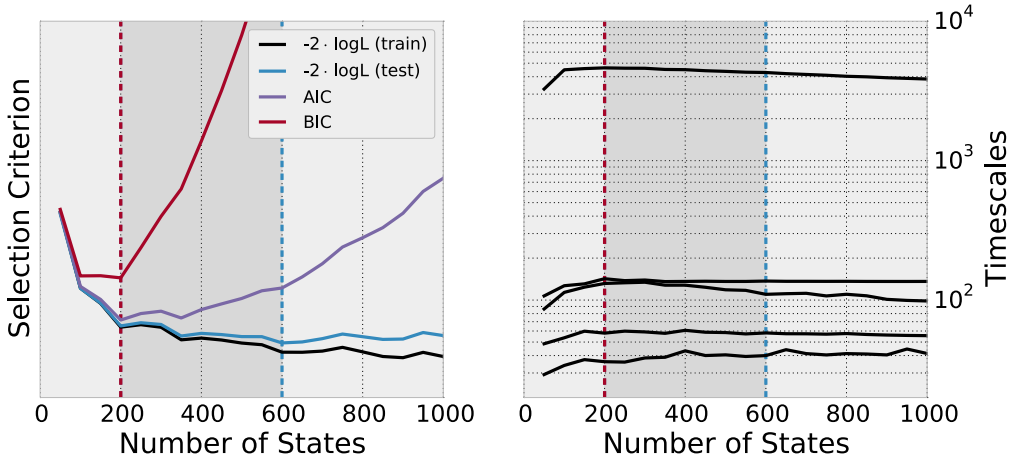


Figure 5.2: Four model selection criteria were used (left panel) to select the state discretization for MSMs built on simulations of the Müller potential. The log-likelihood was evaluated on the training set (black) and a separate test set (blue). These values are plotted as $-2 \cdot \log L$ for ease of comparison with the BIC (red) and AIC (purple). Lower scores indicate better models. Dashed lines are used to indicate criterion-minimizing models. These criteria are consistent with the convergence of the implied timescales (right panel).

As shown in Fig. 5.2, models built with too few states achieve a drastically reduced likelihood, but above a threshold region the likelihood increases relatively slowly. The penalty on the number of parameters in Eq. (5.12) and Eq. (5.13) begins to dominate. The optimal models, according to the BIC, AIC, and test log-likelihood are between 200 and 600 states for this system, which is consistent with the convergence of the relaxation timescales of the models.

The AIC and BIC penalize the larger state models much more than the test set

log-likelihood. It's not clear why the test log-likelihood is more lenient, however, it suggests that when possible, one should use a test log-likelihood approach as opposed to an approximate method like the AIC or BIC. This cross-validation requires fitting multiple models on subsets of the data, and so is not feasible for larger systems in the data-poor regime.

5.4.2 Fip35 WW Domain

How appropriate are likelihood-based model selection criteria for more realistic MD data sets? To answer this questions, we constructed Markov models based on simulations of the Fip35 WW domain[164] performed on the Anton supercomputer.[91] As a preprocessing step, we used time-structure based independent components analysis (tICA) to project the data into a four dimensional space. This procedure was necessary in order to compute the volumes of the MSM states, as described in Algorithm 1. MSMs were constructed in this lower dimensional space with differing number of states using the k-centers clustering algorithm (see *Methods*).

As was the case with the Müller MSMs, the log-likelihood of the training data set increases with the addition of more states, while the test log-likelihood, AIC and BIC lack a monotonic trend (Section 5.4.2). The test log likelihood has a maximum at 650 states, and decreases with the addition of further states. This trend is matched by the AIC, which is computed only on the training data, while the BIC on the other hand seems to over-penalize model complexity. The implied timescales of these MSMs converge rapidly, and selection based on their convergence is consistent with the statistical methods.

We also built a series of models on the same data set by replacing the tICA preprocessing with Principal Components Analysis (PCA). We projected the data set onto the top three principal components, which are the highest-variance uncorrelated linear combinations of the input degrees of freedom (see *Methods*). Because PCA does not explicitly take into account the temporal structure of the data set, we expected that MSMs based on PCA would be less predictive than those based on tICA. Our results, shown in Fig. 5.4, confirm this hypothesis. We built MSMs with as many as

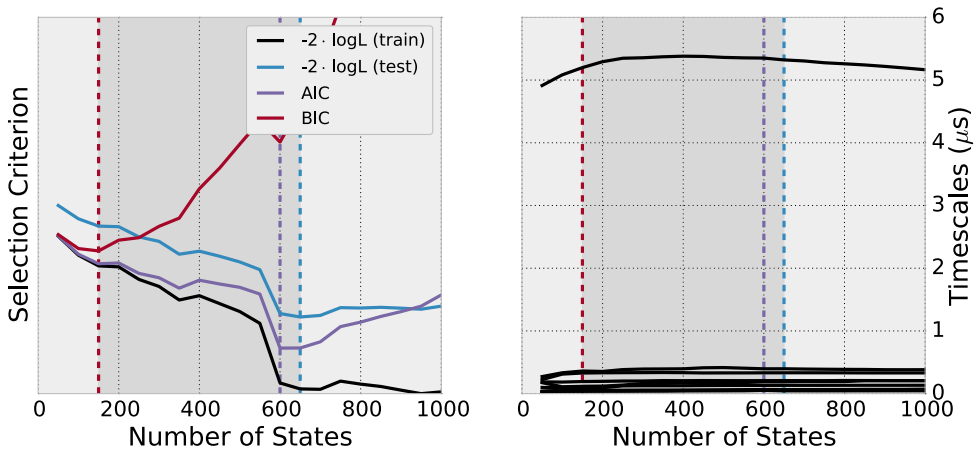


Figure 5.3: MSMs for the Fip35WW domain built with tICA were evaluated based on four model selection criteria. The log-likelihood was evaluated on the training set (black) and a separate test set (blue), and compared with the AIC (purple) and BIC (red) penalized log-likelihoods, computed on the training set. Dashed lines are used to indicate criterion-minimizing models. The models' ten slowest relaxation timescales (shown in the right panel) converge rapidly.

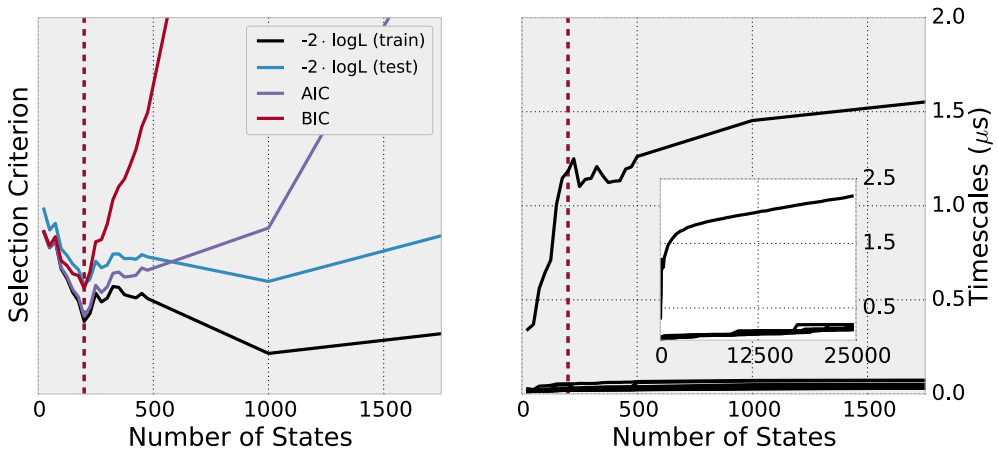


Figure 5.4: The four selection criteria were used to compare MSMs for the Fip35WW domain built with PCA. The longest relaxation timescale (right panel) exhibits a linear increase up to 25,000 states. The AIC (blue), BIC (red) and test set log-likelihood each recommend a model with 200 states.

25,000 states, and saw a linear trend in the longest relaxation timescale with respect to the number of states.

The statistical model selection methods indicate that we are only justified in using models with k on the order of 100. The AIC, BIC and test set log-likelihood criteria each recommend a model built with 200 states, which is also the point just after the most dramatic increase in the longest relaxation timescale occurs with respect to k. On the other hand, the relaxation timescales continue to increase further, even as the test set log likelihood begins to decrease after 1000 states.

The likelihood functions described herein permit the comparison of Markov state models with varying number of states, however, they require that the compared models have the same support. As such, a direct comparison of the likelihoods between models built with tICA and models built with PCA is not justified.

5.5 Conclusions

Markov State Models are powerful and popular frameworks for the analysis of molecular simulations, with a growing literature and two open source software tools.[75,134] There are, however a number of steps in the construction process that require hand-tuning, which limits the use of MSMs to experts and introduces significant biases into the model building process. Additionally, the ability to automatically construct MSMs on the fly while simulations are in progress, an important point for so-called adaptive sampling procedures,[62] is hampered when manual model selection is required.

In the future, we plan to extend this work to the consideration of models without discrete states, where the requirement that states strictly partition phase space into a set of discrete indicator functions is relaxed and the models are parameterized by a direct optimization of Eq. (5.8). This strategy would complement approaches that generalize MSMs beyond discrete states.[194,213]

Chapter 6

Variational cross-validation of slow dynamical modes in molecular kinetics

Markov state models (MSMs) are a widely used method for approximating the eigenspectrum of the molecular dynamics propagator, yielding insight into the long-timescale statistical kinetics and slow dynamical modes of biomolecular systems. However, the lack of a unified theoretical framework for choosing between alternative models has hampered progress, especially for non-experts applying these methods to novel biological systems. Here, we consider cross-validation with a new objective function for estimators of these slow dynamical modes, a generalized matrix Rayleigh quotient (GMRQ), which measures the ability of a rank-m projection operator to capture the slow subspace of the system. It is shown that a variational theorem bounds the GMRQ from above by the sum of the first m eigenvalues of the system's propagator, but that this bound can be violated when the requisite matrix elements are estimated subject to statistical uncertainty. This overfitting can be detected and avoided through cross-validation. These result make it possible to construct Markov state models for protein dynamics in a

way that appropriately captures the tradeoff between systematic and statistical errors.

This chapter is adapted with permission from McGibbon, R. T. and Pande, V. S., Variational cross-validation of slow dynamical modes in molecular kinetics. *J. Chem. Phys.*, **2015**, *142*, 124105.[214]

6.1 Introduction

Conformational dynamics are central to the biological function of macromolecular systems such as signaling proteins, enzymes, and channels. The molecular description of processes as diverse as protein folding, kinase activation, voltage-gating of ion channels, and ubiquitin signaling involve not just the structure of a unique single conformation, but the conformational dynamics between a multitude of states accessible on the potential energy surface.[215–218] These dynamics occur on a range of timescales and have varying degrees of structural complexity: localized vibrations may occur on the 0.1 ps timescale, while large-scale structural changes like protein folding can take seconds or longer.[2] Although many experimental techniques — most notably X-ray crystallography and nuclear magnetic resonance spectroscopy — can yield detailed structural information on functional conformations, the experimental characterization of the dynamical processes, intermediate conformations and transition pathways in macromolecular systems remains exceptionally challenging.[8,9]

Atomistic molecular dynamics (MD) simulations can complement experiment and provide a powerful tool for probing conformational dynamics, allowing researchers to directly visualize and analyze the time evolution of macromolecular systems in atomic detail. Three major challenges for MD simulation of complex systems are the accuracy of the potential energy functions, adequate sampling of conformational space, and quantitative analysis of simulation results. The state-of-the-art on all three fronts has advanced rapidly in recent years. A new generation of increasingly accurate forcefields have recently emerged, such as those which include explicit polarizability and have been parameterized more systematically.[219–223] On the sampling problem,

the introduction of graphical processing units (GPUs) has dramatically expanded the timescales accessible with MD simulation at modest cost, and specialized MD-specific hardware and distributed computing networks have yielded further gains.[20,25–27,96] In this work, we focus on the remaining challenge, the quantitative analysis of MD simulations.

Despite, or perhaps because of their detail, MD simulations require further analysis in order to yield insight into macromolecular dynamics or quantitative predictions capable of being tested experimentally. The direct result of a simulation, an MD trajectory, is a time series of Cartesian positions (and perhaps momenta) of dimension $3N$ ($6N$ if momenta are retained), where N is the number of atoms in the system. Because routine MD simulations may contain tens or hundreds of thousands of atoms, these time series are extremely high-dimensional. A multitude of methods have been proposed for reducing the dimensionality or complexity of MD trajectories and enabling the analysis of the system's key long-lived conformational states, dynamical modes, transition pathways, and essential degrees of freedom.[53,58,190,192,224–226]

In this work, we combine two central ideas from machine learning and chemical physics — hyperparameter selection via cross-validation and variational approaches for linear operator eigenproblems — to create a new method for discriminating between alternative simplified models for molecular kinetics constructed from MD simulations. Towards this end, we prove a new variational theorem concerning the simultaneous approximation of the first m eigenfunctions of the propagator of a high-dimensional reversible stochastic dynamical system, which mathematically formalize the slow collective dynamical motions we wish to identify in molecular systems.

6.2 Cross validation

In seeking to estimate the slowest collective dynamical modes of a molecular system from a finite set of stochastic trajectories, statistical noise is unavoidable. These dynamical modes, which we formally identify as the first m eigenfunctions, ϕ_i, of the propagator, an integral operator associated with the dynamics of a molecular system (vide infra), are functions on $\mathbb{R}^{3N}$ to $\mathbb{R}$. Like the ground state wave function

in quantum chemistry, these eigenfunctions can only be approximately represented in any finite basis set. Reducing this approximation error, a statistical bias, motivates the use of larger and more flexible basis sets. Unfortunately, in an effect known as the *bias-variance tradeoff*,[162,227] larger basis sets tend to exacerbated a competing source of error, the model variance: with a fixed simulation data set but additional adjustable parameters due to a larger basis set, the model estimates of these eigenfunctions become more unstable and uncertain.

As has been known since at least the early 1930s, training a statistical algorithm and evaluating its performance on the same data set generally yields overly optimistic results.[228] For this reason, standard practice in supervised machine learning is to divide a data set into separate training and test sets. The model parameters are estimated using the training data set, but performance is evaluated separately by scoring the now-trained model on the separate test set, consisting of data points that were left out during the training phase. To avoid overfitting, the choice between alternative models is made using test set performance, not training set performance.

However, because researchers may expend significant effort to collect data sets, the exclusion of a large fraction of the of the data set from the training phase can be a costly proposition. k-fold cross-validation is one alternative that can be more data-economical, where the data is split into k disjoint subsets, each of which is cycled as both the training and test set.

Let X be a collection of molecular dynamics trajectories (the data set), which we assume for simplicity to consist of a multiple of k independent MD trajectories of equal length. In k-fold cross validation, the trajectories are split into k equally-sized disjoint subsets, called folds, denoted $X^{(i)}$, for $i \in \{1, 2, \ldots, k\}$. These will serve as the test sets. Let $X^{(-i)} = X \setminus X^{(i)}$ denote the set of all trajectories excluded from fold i; these will serve as the training sets.

Consider an algorithm to estimate the m slowest dynamical modes of the system, g. Examples of such estimators include Markov state models (MSMs)[97] and time-structured independent components analysis (tICA).[166,167] The result of this calculation, the estimated eigenfunctions, $\hat{\phi}_{1:m}$, are taken to be a function of both an input dataset, X, as well as a set of hyperparameters, θ, which many include settings

such as the number of states or clustering algorithm in an MSM or the basis set used in tICA.

$$\hat{\phi}_{1:m} = (\hat{\phi}_1, \hat{\phi}_2, \ldots, \hat{\phi}_m) = g(X, \theta) \tag{6.1}$$

Furthermore, consider an objective function, $O(\hat{\phi}_{1:m}, X')$, which evaluates a set of proposed eigenfunctions, $\hat{\phi}_{1:m}$, and a (possibly new) dataset X', returning a single scalar measuring the performance or accuracy of these eigenfunctions. The mean cross validation performance of a set of hyperparameters is defined by the following expression, which builds k models on each of the training sets and scores them on the corresponding test sets.

$$MCV(\theta) = \frac{1}{k} \sum_{i=1}^{k} O(g(X^{(-i)}, \theta),\, X^{(i)}) \tag{6.2}$$

Model selection can be performed by finding the hyperparameters, $\theta^* = \arg\max_\theta MVC(\theta)$, which maximize the cross validation performance. Many variants of this protocol with different procedures for splitting the training and test sets, such as repeated random subsampling cross-validation and leave-one-out cross validation, are also possible.[229]

The remainder of this work seeks to develop a suitable objective function, O, for estimates of the slow dynamical modes in molecular kinetics that can be used as shown above in a cross-validation protocol. This task is complicated by the fact that no ground-truth values of true eigenfunctions, ϕ_i, are available, either in the training or test sets. Nevertheless, our goal is to construct a *consistent* objective function, such that as the size of a molecular dynamics data set, X grows larger, the maximizer of $O(\cdot, X)$ converges in probability to the true propagator eigenfunctions, $\phi_{1:m}$.

$$\arg\max_{\hat{\phi}_{1:m}} \; O(\hat{\phi}_{1:m}, X) \xrightarrow{p} \phi_{1:m} \tag{6.3}$$

6.3 Theory background

We begin by introducing the notion of the propagator and its eigenfunctions from a mathematical perspective, introducing the key variables and notation that will be essential for the remainder of this work. We largely follow the order of presentation by Prinz et al. (2011) which contains a longer and more thorough discussion.[97]

Consider a time-homogeneous, ergodic, continuous-time Markov process $\mathbf{x}(t) \in \Omega$ which is reversible with respect to a stationary distribution stationary distribution $\mu(\mathbf{x}) : \Omega \to \mathbb{R}^+$. Where necessary for concreteness, we take the phase space, Ω, to be $\mathbb{R}^{3N}$, where N is the number of atoms of a molecular system. The system's stochastic evolution over an interval $\tau > 0$ is described by a transition probability density

$$p(\mathbf{x}, \mathbf{y}; \tau)d\mathbf{y} = \mathbb{P}[\mathbf{x}(t+\tau) \in B_\epsilon(\mathbf{y}) \mid \mathbf{x}(t) = \mathbf{x}], \tag{6.4}$$

where $B_\epsilon(\mathbf{y})$ is the open ϵ-ball centered at $\mathbf{y}$ with infinitesimal measure $d\mathbf{y}$.

Consider an ensemble of such systems at time t, distributed according to some probability distribution $p_t(\mathbf{x})$. After waiting for a duration τ, the distribution evolves to a new distribution,

$$p_{t+\tau}(\mathbf{y}) = \int_\Omega d\mathbf{x}\; p(\mathbf{x}, \mathbf{y}; \tau)\, p_t(\mathbf{x}) = \mathcal{P}(\tau) \circ p_t(\mathbf{y}), \tag{6.5}$$

which defines a continuous integral operator, $\mathcal{P}(\tau)$, called the propagator with lag time τ. The propagator, $\mathcal{P}(\tau)$, admits a natural decomposition in terms of its eigenfunctions and eigenvalues

$$\mathcal{P}(\tau) \circ \phi_i = \lambda_i \phi_i. \tag{6.6}$$

Furthermore, due to the reversibility of the underlying dynamics, $\mathcal{P}(\tau)$ is compact and self-adjoint with respect the a μ^{-1} weighted scalar product,[230]

$$\langle f, g \rangle_{\mu^{-1}} = \int_\Omega d\mathbf{x}\; f(\mathbf{x}) g(\mathbf{x}) \mu^{-1}(\mathbf{x}), \tag{6.7}$$

where f and g are arbitrary scalar functions on Ω. The propagator has a unique largest eigenvalue $\lambda_1 = 1$ with corresponding eigenfunction $\phi_1(\mathbf{x}) = \mu(\mathbf{x})$. The remaining

eigenvalues are real and positive, can be sorted in descending order, and can be normalized to be μ^{-1}-orthonormal. Using the spectral decomposition of $\mathcal{P}(\tau)$, the conformational distribution of an ensemble at arbitrary multiples of τ can be written as a sum of exponentially decaying relaxation processes

$$p_{t+k\tau}(\mathbf{x}) = \sum_{i=1}^{\infty} \lambda_i^k \langle p_t, \phi_i \rangle_{\mu^{-1}} \phi_i, \tag{6.8}$$

$$= \mu(\mathbf{x}) + \sum_{i=2}^{\infty} \exp\left(-\frac{k\tau}{t_i}\right) \langle p_t, \phi_i \rangle_{\mu^{-1}} \phi_i, \tag{6.9}$$

where $t_i = -\dfrac{\tau}{\ln \lambda_i}$. The eigenfunctions $\phi_i(\mathbf{x})$ for $i = 2, ...$ can thus be interpreted as dynamical modes, along each of which the system relaxes towards equilibrium with a characteristic timescale, τ_i. Many molecular systems are characterized by m individual *slow* timescales with eigenvalues close to one, separated from the remaining eigenvalues by a *spectral gap*. These slowest collective degrees of freedom, such as protein folding coordinates and pathways associated with enzyme activation/deactivation, are often identified with key functions in biological systems. The remaining small eigenvalues correspond to faster dynamical processes that rapidly decay to equilibrium. Under these conditions, the long-time dynamics induced by the propagator can be well described by consideration of only these slow eigenfunctions — that is, a rank-m low-rank approximation to the propagator.

Furthermore, not only do these slow eigenfunctions form a convenient basis, in fact they lead to an *optimal* reduced-rank description of the dynamics. That is, each of the partial sums formed by truncating the expansion in Eq. (6.8) at its first m terms is is the closest possible rank-m approximation to $\mathcal{P}(\tau)$ in spectral norm. This statement is made precise by the following theorem.

Theorem 2. *Let $\mathcal{P}$ be compact linear operator which is self-adjoint with respect to an inner product $\langle \cdot, \cdot \rangle_{\mu^{-1}}$. Assume that the eigenvalues λ_i and associated eigenfunctions ϕ_i are sorted in descending order by eigenvalue. Define the rank-m operator $\mathcal{P}_m$ such*

that $\mathcal{P}_m \circ f = \sum_{i=1}^{m} \lambda_i \langle f, \phi_i \rangle_{\mu^{-1}} \phi_i$*, and let* $\mathcal{A}_m$ *be an arbitrary rank* m *operator. Then,*

$$\mathcal{P}_m = \underset{\text{rank}(\mathcal{A}_m) \leq m}{\arg\min} \; ||\mathcal{A}_m - \mathcal{P}||_{\mu^{-1}}. \tag{6.10}$$

Proof. This is the extension of the familiar Eckart-Young theorem to self-adjoint linear operators. The original result is by Schmidt.[231] See Courant and Hilbert (pp 161),[232] and Micchelli and Pinkus (1971)[233] for further details. □

While Theorem 2 is a statement about operator approximation, it can also be viewed as a statement about optimal dimensionality reduction for description of slow dynamics. Over all m-dimensional dimensionality reductions, the one which projects the dynamics onto its first m propagator eigenfunctions preserves to the largest degree information about the long-timescale evolution of the original system.

Note however that rank-constrained propagator, $\mathcal{P}_m$, while optimal by spectral norm is not generally positivity-preserving, as proved in Section 6.B, which is an important property of the propagator necessary for its probabilistic interpretation in Eq. (6.5).

6.4 Objective function and subspace variational principle

In this section we introduce the objective function discussed abstractly in Section 6.2. We show that both the existing time-structure independent components analysis (tICA)[166,167] and Markov state model (MSM)[51,70,97,137,196] methods can be interpreted as procedures which directly optimize this criteria using different restricted families of basis functions. Furthermore, we show that in the infinite-data limit, when the requisite matrix elements can be computed without error, a variational bound governs this objective function: *ansatz* eigenfunctions, $\hat{\phi}_{1:m}$, which differ from the propagator's true eigenfunctions, $\phi_{1:m}$, are always assigned a score which is less than the score of the true eigenfunctions.

Unfortunately, in the more typical finite-data regime, this variational bound can be

violated in a pernicious manner: as the size of the basis set increases past some threshold, models can give continually-increasing training set performance (even breaking the variational bound), even as they get *less* accurate when measured on independent test data sets. This observation underscores the practical value of cross-validation in estimators for the slow dynamical processes in molecular kinetics.

Our results build on the important contributions of Noeé and Nüske (2013) and Nüske et al.,[194,234] who introduced a closely related variational approach for characterizing the slow dynamics in molecular systems. Our novel contribution stems from an approach to the problem through the lens of cross-validation, with its need for a single scalar objective function. While previous work focuses on estimators of each of the propagator eigenfunctions, ϕ_i, one at at time, we focus instead on measures related to the simultaneous estimation of all of the first m eigenfunctions collectively.

Theorem 3. *Let $\mathcal{P}$ be compact linear operator whose eigenvalues $\lambda_1 > \lambda_2 \geq \lambda_3, \ldots$ are bounded from above and which is self-adjoint with respect to an inner product $\langle \cdot, \cdot \rangle_{\mu^{-1}}$. Furthermore, let f be an arbitrary set of m linearly independent functions on $\Omega \to \mathbb{R}$, $f = \{f_i(\cdot)\}_{i=1}^m$. Let $\mathbb{S}^m$ and $\mathbb{S}^m_{++}$ be the space of $m \times m$ real symmetric matrices and positive definite matrices respectively. Define a matrix $P \in \mathbb{S}^m$ with $P_{ij} = \langle f_i, \mathcal{P} \circ f_j \rangle_{\mu^{-1}}$, and a matrix $Q \in \mathbb{S}^m_{++}$ with $Q_{ij} = \langle f_i, f_j \rangle_{\mu^{-1}}$. Define $\mathcal{R}_{\mathcal{P}}[f]$ as*

$$\mathcal{R}_{\mathcal{P}}[f] = \mathrm{Tr}\left(PQ^{-1}\right). \tag{6.11}$$

Then,

$$\mathcal{R}_{\mathcal{P}}[f] \leq \sum_{i=1}^{m} \lambda_i. \tag{6.12}$$

Lemma 4. *The equality in Eq.* (6.12) *holds for $f = \{\phi_1, \phi_2, \ldots, \phi_m\}$, and any set of m functions, f, such that* $\mathrm{span}(f) = \mathrm{span}(\{\phi_1, \phi_2, \ldots, \phi_m\})$.

The proof of Theorem 3 follows from the Ky Fan theorem.[235,236] Its proof, as well as the proof of Lemma 4 can be found in Section 6.A.

This result implies that the slow eigenspace of the molecular propagator can be numerically determined by simultaneously varying a set of *ansatz* functions f to maximize $\mathcal{R}_{\mathcal{P}}[f]$. If the maxima is found, then f are the desired eigenfunctions, up

to a rotation. The matrix P has the form of a time-lagged covariance matrix between the *ansatz* functions, describing the tendency of the system to move from regions of phase space strongly associated one *ansatz* function to another in time τ. The matrix Q acts like a normalization, giving the equilibrium overlap between *ansatz* functions. Note that when the trial functions, f, are μ^{-1}-orthonormal, Q, is simply the identity. Under these conditions, $\mathcal{R}_{\mathcal{P}}[f]$ then assumes a simple form as the sum of the individual Ritz values of the trial functions.

Physically, $\mathcal{R}_{\mathcal{P}}[f]$ can be interpreted as the "slowness" of the lower-dimensional dynamical process formed by projecting a high-dimensional process through the m *ansatz* functions. The maximization of $\mathcal{R}_{\mathcal{P}}[f]$ corresponds to a search for the coordinates along which the system decorrelates as slowly as possible.

Because it is bounded by the sum of the first m true eigenfunctions of the propagator, $\mathcal{R}_{\mathcal{P}}[f]$, is the foundation of the sought objective function for cross-validation of estimators for the slow dynamical modes in molecular kinetics. Unfortunately, it cannot be calculated exactly from a molecular dynamics simulation. Next we show how the requisite matrix elements, P_{ij} and Q_{ij} can be approximated from MD. The noise in these approximations will be a function of both the amount of available simulation data and the size and flexibility of a basis set, leading to the bias variance tradeoff discussed earlier. By the continuous mapping theorem and Theorem 3, consistency of the objective function (in the sense of Eq. (6.3)) is established if these estimators for P and Q are consistent.

6.4.1 Basis Function Expansion

Equipped with this variational theorem, we now consider the construction of an approximation to the dominant eigenspace of $\mathcal{P}(\tau)$ using linear combinations of functions from a finite basis set. This reduces the problem of searching over the space of sets of m functions to a problem of finding a weight matrix that linearly mixes the basis functions.

Let $\{\varphi_a\}_{a=1}^{n}$ be a set of n functions on $\Omega \to \mathbb{R}$, which will be used as basis functions

in which to expand the slowest m propagator eigenfunctions. Physically motived basis functions for protein dynamics might include measurements of protein backbone dihedral angles, the distances between particular atoms, or similar structural coordinates. The basis can also be indicator functions for specific regions of phase space — the "Markov states" in a MSM.

Following Nüske et al.,[234] we expand the m *ansatz* eigenfunctions as μ-weighted linear combinations of the basis functions, $f_i(\mathbf{x}) = \sum_a A_{ia}\mu(\mathbf{x})\varphi_a(\mathbf{x})$, where $A \in \mathbb{R}^{n\times m}$ is a weight matrix of arbitrary expansion coefficients. From the basis functions, we define the time-lagged covariance and overlap matrices $C \in \mathbb{S}^n$ and $S \in \mathbb{S}^n_{++}$ respectively such that $C_{ij} = \langle \mu\varphi_i,\ \mathcal{P}\circ\mu\varphi_j\rangle_{\mu^{-1}}$ and $S_{ij} = \langle \mu\varphi_i,\ \mu\varphi_j\rangle_{\mu^{-1}}$.

Then, by exploiting the linearity of the basis function expansion, the matrices P and Q, can be written as matrix products involving the expansion coefficients, correlation and overlap matrices.

$$P = A^T C A \tag{6.13}$$

$$Q = A^T S A \tag{6.14}$$

These equations can be interpreted in a simple way: the time-lagged correlation and overlap of the *ansatz* functions with respect to on another can be formed from two similar matrices involving only the basis functions, C and S, and the expansion coefficients, A. When the *ansatz* functions, f, are constructed this way, $\mathcal{R}_{\mathcal{P}}[f]$ reduces to the generalized matrix Rayleigh quotient (GMRQ), $\mathcal{R}_{\mathcal{P}}[f] = \mathcal{R}(A; C, S) = \mathcal{R}(A)$

$$\mathcal{R}(A) \equiv \mathrm{Tr}\left(A^T C A \left(A^T S A\right)^{-1}\right) \tag{6.15}$$

Following Lemma 4, we note that $\mathcal{R}(A)$ is a function only of column span of A, and is not affected by rescaling, or the application of any invertible transformation of the columns. Therefore, the optimization of $\mathcal{R}(A)$ can be seen as a single optimization problem over the set of all m-dimensional linear subspaces of Ω. This space is referred

to as a *Grassmann manifold*.[237] Note that when $m = 1$, P and Q are scalars, and $\mathcal{R}(A)$ reduces to a standard generalized Rayleigh quotient.

Furthermore, with fixed basis functions, the training problem, $A^* = \arg\max_A \mathcal{R}(A; C, S)$, is solved directly by a matrix A^* with columns that are the m generalized eigenvectors of C and S with the largest eigenvalues, and this eigenproblem is identical to the one introduced for the tICA method,[166,167] and Ritz method.[194]

6.4.2 Estimation of matrix elements from MD

Equipped with a collection of basis functions, $\{\varphi\}$, how can C and S be estimated from an MD dataset? As previously shown by Nüske et al.,[234] the matrix elements C and S can be estimated from an equilibrium molecular dynamics simulations, $X = \{\mathbf{x}_t\}_{t=1}^T$, by exploiting the ergodic theorem and measuring the correlation between the basis functions, with or without a time lag.

$$
\begin{aligned}
C_{ij} &= \langle \mu\varphi_i, \mathcal{P}(\tau) \circ \mu\varphi_j \rangle_{\mu^{-1}} && (6.16)\\
&= \int_{x\in\Omega} \int_{y\in\Omega} d\mathbf{x}\, d\mathbf{y}\; \mu(\mathbf{y})\; \varphi_i(\mathbf{y})\; p(\mathbf{x}, \mathbf{y}; \tau)\; \varphi_j(\mathbf{x}) && (6.17)\\
&\approx \frac{1}{T-\tau} \sum_{t=1}^{T-\tau} \varphi_i(\mathbf{x}_t)\varphi_j(\mathbf{x}_{t+\tau}) && (6.18)\\
S_{ij} &= \langle \mu\varphi_i, \mu\varphi_j \rangle_{\mu^{-1}} && (6.19)\\
&= \int_{x\in\Omega} \varphi_i(\mathbf{x})\varphi_j(\mathbf{x})\mu(\mathbf{x}) && (6.20)\\
&\approx \frac{1}{T} \sum_{t=1}^{T} \varphi_i(\mathbf{x}_t)\varphi_j(\mathbf{x}_t) && (6.21)
\end{aligned}
$$

Note that for Theorem 3 to be applicable, C is required to be symmetric, a property which is likely to be violated by the estimator Eq. (6.18). For this reason, in practice we use an estimator that averages the matrix computed in Eq. (6.18) with its transpose. We call this method transpose symmetrization, and it amounts to including each trajectory twice in the dataset, once in the forward and once in the

reversed direction, as discussed in Schwantes and Pande.[166]

Markov state models (MSMs)[51,70,97,137,196] are particular case of the proposed method that have been widely applied to the analysis of biomolecular simulations[56,59,78,107,118,238–241], where the basis functions are chosen to indicator functions on a collection of non-overlapping subsets of the conformation space. Given a set of discrete non-overlapping states which partition Ω, $\mathcal{S} = \{s_i\}_{i=1}^n$, such that $s_i \subseteq \Omega$, $\bigcup_{i=1}^n s_i = \Omega$, and $s_i \cap s_j = \emptyset$, and define

$$\varphi_i^{\mathrm{MSM}}(\mathbf{x}_t) = \begin{cases} 1, & \text{if } \mathbf{x}_t \in s_i. \\ 0, & \text{otherwise.} \end{cases} \tag{6.22}$$

Using this basis set, as previously shown by Nüske et al.,[234] estimates of the correlation matrix elements C_{ij} can be obtained following Eq. (6.18) by counting the number of observed transitions between sets s_i and s_j. The overlap matrix S is diagonal with entries, S_{ii}, that estimate the stationary probabilities of the sets, $S_{ii} \approx \int_{\mathbf{x} \in s_i} d\mathbf{x}\, \mu(\mathbf{x})$.

For the particular case of MSM basis sets, in contrast to the somewhat crude transpose symmetrization method, a more elegant enforcement of symmetry of C can be accomplished via a maximum likelihood estimator following Algorithm 1 of Prinz *et al.*[97]

Equipped with these estimators for C and S from MD data, Eq. (6.15) now has a form which is suitable for use as a cross-validation objective function, $O(\hat{\phi}_{1:m}, X')$. The proposed eigenfunctions, which may have been trained on a *different* dataset, are numerically represented by expansion coefficients, $\hat{A}$, and C and S act as sufficient statistics from the test dataset X'; the GMRQ objective function is $\mathcal{R}(\hat{A}; C(X'), S(X'))$.

6.5 Algorithmic realization

The central practical purpose of cross-validation with generalized matrix Rayleigh quotient (GMRQ) is, given an MD dataset, to select a set of appropriate basis functions with which to construct Markov state models (MSMs) for system's kinetics. Note that Eq. (6.22) leaves substantial flexibility in the definition of the basis set, since the partitioning of phase space into states is left unspecified.

Methodologies for constructing these states include clustering the conformations in the dataset using a variety of distance metrics, clustering algorithms, and dimensionality reduction techniques. Let θ be a variable containing the settings for these procedures, which parameterizes a function, $g_\theta^{\text{MSM}}(X)$, that, given a collection of MD trajectories, returns a set of n states, $\mathcal{S}$.

Procedurally, GMRQ-based cross-validation for MSMs is a protocol for assigning a scalar score, $MCV(\theta)$, to the MSM hyperparameters, θ, with the following steps.

1. Separate the full data set into k disjoint folds, as described in Section 6.2.

2. For each fold, i, use the training data set, $X^{(-i)}$, to construct a set of states, $\mathcal{S}^{(-i)} = g_\theta(X^{(-i)})$.

3. Use the states $\mathcal{S}^{(-i)}$ and the training data set $X^{(-i)}$ to build a Markov state model. This entails clustering the dataset to obtain the basis functions (states), $\{\varphi\}$, estimating the training set correlation and overlap matrices $C^{(-i)}$ and $S^{(-i)}$ from the trajectories, and computing their first m generalized eigenvectors, $A = \arg\max_A \mathcal{R}(A; C^{-(i)}, S^{(-i)})$, with a standard generalized symmetric eigensolver (e.g. LAPACK's DSYGV subroutine).[242]

4. These eigenvectors maximize the GMRQ on the training set, but how do they perform when tested on new data? Using the test set data, $X^{(i)}$, and the states, $\mathcal{S}^{(-i)}$, *as determined from the training set*, compute the test set correlation and overlap matrices, $C^{(i)}$ and $S^{(i)}$. These trained eigenvectors, A, are scored on

the test set by $\mathcal{R}(A; C^{(i)}, S^{(i)})$. The key metric for model selection, the cross-validation mean test set generalized matrix Rayleigh quotient is

$$MVC(\theta) = k^{-1} \sum_{i=1}^{k} \mathcal{R}(A;\ C^{(i)}, S^{(i)}). \tag{6.23}$$

As an overfitting diagnostic, we also calculate the cross-validation mean training set GMRQ,

$$MVC'(\theta) = k^{-1} \sum_{i=1}^{k} \mathcal{R}(A; C^{(-i)}, S^{(-i)}). \tag{6.24}$$

5. Finally, the entire procedure is repeated for many choices of θ, and the hyperparameter set that maximize the mean cross validation score is chosen as the best model, $\theta^\star = \arg\max_\theta MVC(\theta)$.

For this approach, one symptom of overfitting — the construction of models that describe the statistical noise in X rather than the underlying slow dynamical processes — is an overestimation of the eigenvalues of the propagator and overestimation of the GMRQ. Related statistical methods, such as kernel principal components analysis which also involve spectral analysis of integral operators under non-negligible statistical error suffer from the same effect, which has been termed variance inflation.[243–245]

Left unspecified in this protocol are three important parameters: the degree of cross validation, k, the number of desired eigenfunctions, m, and the correlation lag time, τ. In our experiments, following common practice in the machine learning community, we use $k = 5$. Especially in the data-limited regime, the tradeoffs involving the choice of k are not entirely clear, as the objective lacks the form of an empirical risk minimization problem.[227,246] For MSMs, substantial attention in the literature has been paid to the selection of the lag time, τ.[51,84,97] With fixed basis function, it has been shown that the eigenfunction approximation error is a decreasing function of the τ, which motivates the use of larger values.[85] On the other hand, larger values of τ limit the temporal resolution of the model. For MSMs of protein folding, the authors' experience suggest that appropriate values for τ are often in the range between

1 and 100 nanoseconds. Finally, we suggest that m, the rank of GMRQ, be selected based on the number of slow dynamical processes in the system, as determined by an apparent gap in the eigenvalue spectrum of $\mathcal{P}(\tau)$, or heuristically to a value between 2 and $\sim$10.

6.6 Simulations

6.6.1 Double well potential

In order to gain intuition about the method, we begin by considering one of simplest possible systems: Brownian dynamics on a double well potential. We consider a one dimensional Markov process in which a single particle evolves according to the stochastic differential equation

$$\frac{dx_t}{dt} = -\nabla V(x_t) + \sqrt{2D}R(t) \tag{6.25}$$

where V is the reduced potential energy, D is the diffusion constant, and $R(t)$ is a zero-mean delta-correlated stationary Gaussian process. For simplicity, we consider the potential

$$V(x) = 1 + \cos(2x) \tag{6.26}$$

with reflecting boundary conditions at $x = -\pi$ and $x = \pi$. Using an Euler integrator, a time step of $\Delta t = 10^{-3}$, and diffusion constant $D = 10^3$, we simulated 10 trajectories starting from $x = 0$ of length 10^5 steps, and saved the position every 100 steps. The potential and histogram of the resulting data points is shown in Fig. 6.1 (b). We computed the true eigenvalues of the system's propagator to machine precision by discretizing the propagator matrix elements on a dense grid (see Section 6.C). The timescale of the slowest relaxation process in this system is $t_2 \approx 7115.3$ steps, and the dataset contains approximately 94 transitions events.

We now consider the construction of Markov state models for this system, and in particular the selection of the number of states, n, using states, $\mathcal{S} = \{s_i\}_{i=1}^n$, which evenly partition the region between $x = -\pi$ and $x = \pi$ into n equally spaced intervals.

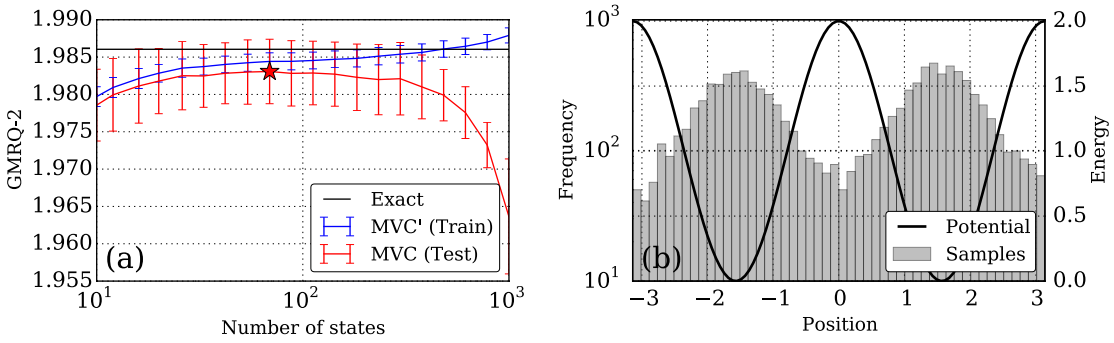


Figure 6.1: Model selection for MSMs of a double well potential. Error bars indicate standard deviations over the 5 folds of cross validation. See text for details.

$$s_i = \Big[-\pi + \frac{2\pi}{n}(i-1),\ -\pi + \frac{2\pi}{n} i \Big) \tag{6.27}$$

When n is too low, we expect that the discretization error in the MSM will dominate, and our basis will not be flexible enough to capture the first eigenfunction of the propagator. On the other hand, because the number of parameters estimated by the MSM is proportional to n^2, we expect that for n too large, our models will be overfit. We therefore use 5-fold cross validation with the GMRQ to select the appropriate number of states, balancing these competing effects. The cross-validation GMRQ for the first two eigenvectors ($m = 2$, $\tau = 100$ steps) of the MSMs is shown in Fig. 6.1 (a), along with the exact value of the GMRQ. The blue training curve gives the average GMRQ over the folds when scoring the models on the *same* trajectories that they were fit with, and is simply equal to the mean sum of the first two eigenvalues of the MSMs, whereas the red curve shows the mean GMRQ evaluated on the left-out test trajectories.

The training GMRQ increases monotonically, and we note with particular emphasis that it increases *past* the exact value when using a large number of states. This indicates that the models built with more than 200 states predict *slower* dynamics than the true propagator. This effect is impossible in the limit of infinite data as

demonstrated by Eq. (6.12) — it is a direct manifestation of overfitting, and indicates why straightforward variational optimization without testing on held-out data or consideration of statistical error fails in a data-limited regime. The training set eigenvectors, the maximizers of the training set GMRQ, are actually exploiting noise in the dataset more so than modeling the propagator eigenfunctions. On the other hand, the test GMRQ displays an inverted U-shaped behavior and achieves a maximum at $k = 61$. These models thus achieve the best *predictive* accuracy in capturing the systems slow dynamics, given the finite data available.

6.6.2 Comparison of Clustering Procedures: Octaalanine

What methods of MSM construction are most robustly able to capture the long-timescale dynamics of protein systems? To address this question, we performed a series of analyses of 27 molecular dynamics trajectories of terminally-blocked octaalanine, a small helix forming peptide. We used 8 different methods to construct the state discretization using clustering with three distance metrics and three clustering algorithms.

For clustering, we considered three distance metrics. The first was the backbone ϕ and ψ dihedral angles. Each conformation was represented by the sine and cosine of these torsions for a total of 32 features per frame, and distances for clustering were computed using a Euclidean metric. Second, we considered the distribution of reciprocal interatomic distances (DRID) distance metric introduced by Zhou and Caflisch,[66] using the $C\alpha$, $C\beta$, C, N, and O atoms in each residue. Finally, we considered the Cartesian minimal root mean square deviation (RMSD) using the same set of atoms per residue.[247] We also considered three clustering algorithms, k-centers,[75] a landmark version of UPGMA hierarchical clustering (see Section 6.D), and k-means.[168]

For each pair of distance metric and clustering algorithm (excluding k-means & RMSD which are incompatible),[134] we performed 5-fold cross validation using between 10 and 500 states for the clustering. For this experiment, we heuristically chose a lag time of $\tau = 10$ ps, and $m = 6$, to capture the first five dynamical processes in addition

to the stationary distribution. The results are shown in Fig. 6.2, with blue curves indicating the mean GMRQ on the training set, and red curves indicating the mean performance on the held-out sets. We find that in all cases, the performance on the training set is *optimistic*, in the sense that the *ansatz* eigenvectors fit during training score more poorly when reëvaluated on held out data. Furthermore, although the training curves all continue to increase with respect to the number of states within the parameter range studied — which might be interpreted from a variational perspective as the quality of the models continually increasing — the performance on the test sets tends to peak at a moderate number of states and then decrease. We interpret this as a sign of overfitting: when the number of states is too large, with models fitting the statistical noise in the dataset rather than the underlying slow dynamical processes. Of the parameters studied, the best performance appears to be using the combination of k-means clustering with the dihedral distance metric, using between 50 and 200 states. We also note that k-centers appears to yield particularly poor models for all distance metrics, which may be rationalized on the basis that, by design, the algorithm selects outlier conformations to serve as cluster centers.[75]

6.7 Discussion

Some amount of summarization, coarse-graining or dimensionality reduction of molecular dynamics data sets is a necessary part of their use to answer questions in biological physics. In this work, we argue that the goal of this effort should essentially be to find the dominant eigenfunctions of the system's propagator, an unknown integral operator controlling the system's dynamics. We show that this goal can be formulated as the variational optimization of a single scalar functional, which can be approximated using trajectories obtained from simulation and a parametric basis set. Although overfitting is a concern with finite simulation data, this risk can be mitigated by the use of cross-validation.

When the basis sets are restricted to mutually-orthogonal indicator functions or linear functions of the input coordinates, this method corresponds to the existing MSM and tICA methods. Unlike previous formulations, it provides a method by

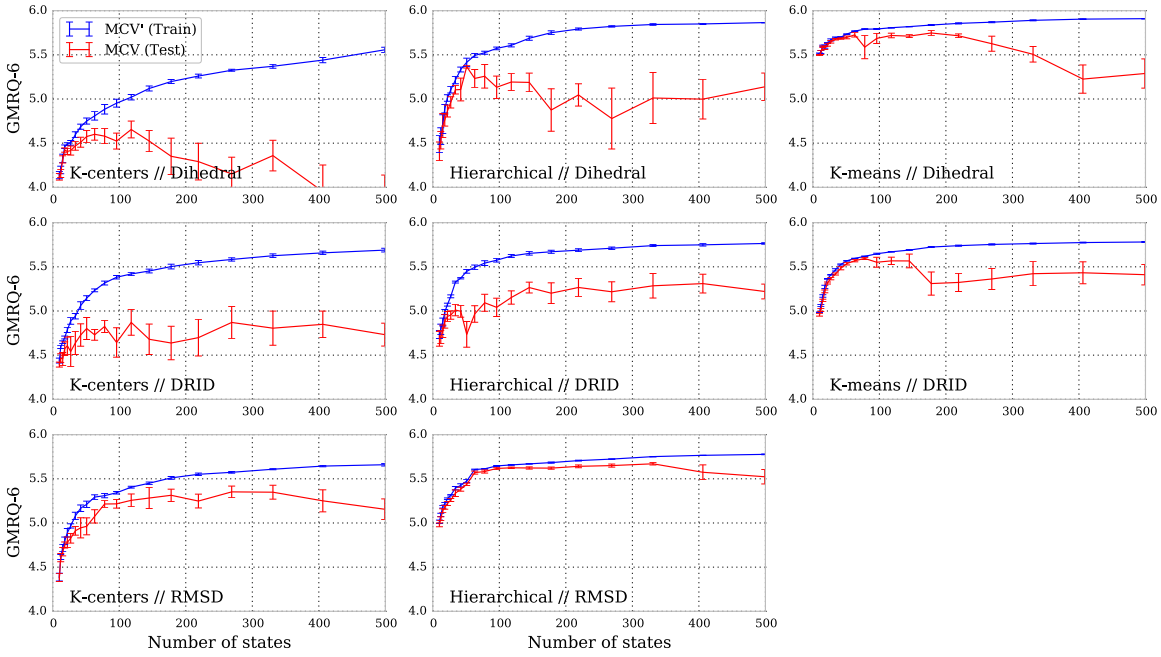


Figure 6.2: Comparison of 8 methods for building MSMs under 5-fold cross validation, evaluated using the rank-6 GMRQ. We used the k-centers, k-means, and landmark-based ($n_{\text{landmarks}} = 5000$) UPGMA hierarchical clustering algorithms, with the DRID[66] and backbone dihedral angle featurizations. Error bars indicate the standard error in the mean over the cross validation folds.

which MSM and tICA solutions can be "scored" on new data sets that were not used during parameterization, making it possible to measure the generalization performance of these methods and choose the various hyperparameters required for each method, such as the number of MSM states or clustering method. Furthermore, the extension to other families of basis functions (e.g Gaussians) is straightforward, and GMRQ provides a natural quantitative basis on which to conclude whether these new methods are superior to existing basis sets.

6.7.1 Connections to quantum mechanics and machine learning

The variational theorem for eigenspaces in this work has strong connections to work in two other related fields: excited state electronic structure theory in quantum mechanics and multi-class Fisher discriminant analysis in machine learning. In quantum mechanics, Theorem 3 is analogous to what has been called the ensemble or trace variational principle in that field,[248–251] which bounds the sum of the energy of the first m eigenstates of the Hamiltonian by the trace of a matrix of Ritz values. While the goal of finding just the ground-state eigenfunction ($m = 1$) is more common in computational quantum chemistry, the simultaneous optimization of many eigenstates is critical for many applications including band-structure calculations for materials in solid state physics.

Furthermore, in machine learning, this work has an analog in the theory multi-class Fisher discriminant analysis.[252] Here, the goal is to find a low-rank projection of a labeled multi-class dataset which maximizes the between-class variance of the dataset while controlling the within-class variances. The optimal discriminant vectors are shown to be the first k generalized eigenvectors of an eigenproblem involving these two variance matrices — the problem shares the same structure as Eq. (6.15) in this work.[253] We anticipate that this parallel will aid the development of improved algorithms for the identification of slow molecular eigenfunctions, especially with respect to regularization and sparse formulations.[254,255]

6.7.2 Comparison to likelihood maximization

While we focus on the identification of the dominant eigenfunctions of the system's propagator, a different viewpoint is that analysis of MD should essential entail the construction of probabilistic, generative models over trajectories, fit for example by maximum likelihood or Bayesian methods.

As we show in Section 6.4.2, and Nüske et al.,[234] have shown earlier, MSMs arise naturally from a maximization of Eq. (6.12) when the *ansatz* eigenfunctions are constrained to be linear combinations of a set of mutually orthogonal indicator functions. However, MSMs can also be viewed directly as probabilistic models, constructed by maximizing a likelihood function of the trajectories with respect to the model parameters. This probabilistic view has, in fact, been central to the field, driving the development of improved methods for example in model selection,[162,198] parameterization,[256] and coarse-graining.[60,257] To what extent does this imply that the variational and probabilistic views are equivalent?

In Section 6.B we show that while these two views may coincide for the particular choice of basis set with MSMs, they need not be equivalent in general. In fact, the GMRQ-optimal model formed by the first m eigenfunctions of the propagator need not be positivity preserving, which is essential to form a probabilistic likelihood function in the sense of Kellogg et al. or McGibbon, Schwantes and Pande.[162,198] On the other hand, the two views *are* tightly coupled; their connection is given by the error bounds proved by Sarich, Noé, and Schütte.[85] When the model gives a good approximation to the slow propagator eigenspace (low eigenfunction approximation error, high GMRQ), a good approximation to the long-timescale transition probabilities is obtained.

Cross validation with the log likelihood requires either a generative model for the high dimensional data, such as a hidden Markov model (HMM),[92] or dimensionality reduction before model comparison. This is a major disadvantage, because accurate and tractable generative models for time series with tens or hundreds of thousands dimensions are not generally available. However, treating dimensionality reduction as a preprocessing and applying probabilistic models afterwards, as done by McGibbon, Schwantes and Pande,[162] does not enable quantitative comparison between alternative competing dimensionality reduction protocols. With the GMRQ, on the other

hand, the need for a reference state decomposition or high-dimensional generative model is eliminated,[257] and different dimensionality reduction procedures can easily be compared in a quantitative manner, as shown in Fig. 6.2.

6.8 Conclusions

The proliferation of new and improved methods for constructing low-dimensional models of molecular kinetics given a set of high-resolution MD trajectories has been a boon to the field, but the lack of a unified theoretical framework for choosing between alternative models has hampered progress, especially for non-experts applying these methods to novel biological systems. In this work we have presented a new variational theorem governing the estimation of the space formed by the span of multiple eigenfunctions of the molecular dynamics propagator. With this method, a single scalar-valued functional scores a proposed model on a supplied data set, and the use of separate testing and training data sets makes it possible to quantify and avoid statistical overfitting. During the training step, time-structure independent components analysis (tICA) and Markov state models (MSMs) are specific instance of this method with different types basis functions. This method extends these tools, making it possible to score trained models on new datasets and perform hyperparameter selection.

We have applied this approach to compare eight different protocols for Markov state model construction on a set of MD simulations of the octaalanine peptide. We find that of the methods tested, k-means clustering with the dihedral angles using between 50 and 200 states appears to outperform the other methods, and that the k-centers cluster method can be particularly prone to poor generalization performance. To our knowledge, this work is the first to enable such quantitative and theoretically well-founded comparisons of alternative parameterization strategies for MSMs.

We anticipate that this work will open the door to more complete automation and optimization of MSM construction. Free and open source software fully implementing these methods is available in the MDTraj and MSMBuilder3 packages from `http://mdtraj.org` and `http://msmbuilder.org`, along with example and

tutorials.[165] While the lag time, τ and rank, m, of the desired model must be manually specified, other key hyperparameters that control difficult-to-judge statistical tradeoffs, such as the number of states in an MSM, can be chosen be optimizing the cross-validation performance. Furthermore, given recent advances in automated hyperparameter optimization in machine learning, we anticipate that this search itself can be fully automated.[258]

Acknowledgements

This work was supported by the National Science Foundation and National Institutes of Health under Nos. NIH R01-GM62868, NIH S10 SIG 1S10RR02664701, and NSF MCB-0954714. We thank the anonymous reviewers for their many suggestions for improving this work, Christian R. Schwantes, Mohammad M. Sultan, Sergio Bacallado, and Krikamol Muandet for helpful discussions, and Jeffrey K. Weber for access to the octaalanine simulations.

6.A Proofs

6.A.1 Proof of Theorem 3

The eigenfunctions, ϕ_i, of $\mathcal{P}(\tau)$ form a complete basis. Expand each $f_i = \sum_a w_{ai}\phi_a$ with coefficients $W \in \mathbb{R}^{\infty\times m}$ with $W_{ni} = \langle f_i, \phi_n\rangle_{\mu^{-1}}$.

$$P_{ij} = \langle f_i,\ \mathcal{P} \circ f_j \rangle_{\mu^{-1}} \tag{6.28}$$
$$= \Big\langle \sum_a W_{ai}\phi_a,\ \mathcal{P} \circ \sum_b W_{bj}\phi_b \Big\rangle_{\mu^{-1}} \tag{6.29}$$
$$= \sum_a W_{ai} W_{aj} \lambda_a \tag{6.30}$$
$$Q_{ij} = \langle f_i, f_j \rangle_{\mu^{-1}} \tag{6.31}$$
$$= \Big\langle \sum_a W_{ai}\phi_a,\ \sum_b W_{bj}\phi_b \Big\rangle_{\mu^{-1}} \tag{6.32}$$
$$= \sum_a W_{ai} W_{aj} \tag{6.33}$$

We define the diagonal matrix $D(\lambda)$ with $D_{ii} = \lambda_i$. Then, Eq. (6.30) and Eq. (6.33) can be rewritten in matrix form:

$$P = W^T D(\lambda) W, \tag{6.34}$$
$$Q = W^T W. \tag{6.35}$$

Let $F = Q^{1/2} \in \mathbb{S}^m_{++}$ be the (unique) positive definite square root of Q, which is guaranteed to exist because Q is positive definite, and $B = WF^{-1}$. Then, rearrange the objective function using the cyclic property of the trace:

$$\mathcal{R}_{\mathcal{P}}[f] = \mathrm{Tr}\left(\underbrace{W^T D(\lambda) W}_{P} \underbrace{(FF)^{-1}}_{Q^{-1}}\right), \tag{6.36}$$
$$= \mathrm{Tr}\left(F^{-1} W^T D(\lambda) W F^{-1}\right), \tag{6.37}$$
$$= \mathrm{Tr}\left(B^T D(\lambda) B\right). \tag{6.38}$$

Note that $B^T B = F^{-1} W^T W F^{-1} = I_m$. Therefore, by application of the Ky Fan theorem,[235,236]

$$\mathcal{R}_{\mathcal{P}}[f] \leq \sum_i^m \lambda_i, \tag{6.39}$$

and the equality holds when $f = \{\phi_1, \phi_2, \ldots, \phi_m\}$.

6.A.2 Proof of Lemma 4

Following Absil et al.,[237] let $f = \{f_1, f_2, \ldots, f_m\}$, and $M \in \mathbb{R}^{m\times m}$ be an arbitrary invertible matrix. Define a new collection of functions $g = \{g_1, g_2, \ldots, g_m\}$, such that $g_j = \sum_{i=1}^{m} M_{ij} f_i$, and a matrix $W' \in R^{\infty\times m}$ such that $W'_{ni} = \langle g_i, \phi_n\rangle_{\mu^{-1}}$. Expanding the matrix elements of W', we note that

$$W' = WM. \tag{6.40}$$

Then, using Eq. (6.34) and Eq. (6.35), $\mathcal{R}_{\mathcal{P}}[g]$ can be written as a matrix expression involving W', and subsequently rewritten involving W and M. Expansion of the matrix products and application of the cyclic property of the trace confirms that $\mathcal{R}_{\mathcal{P}}[g] = \mathcal{R}_{\mathcal{P}}[f]$:

$$\begin{aligned}
\mathcal{R}_{\mathcal{P}}[g] &= \mathrm{Tr}\left(W'^T D(\lambda) W' (W'^T W')^{-1}\right), && (6.41)\\
&= \mathrm{Tr}\left((WM)^T D(\lambda)(WM)((WM)^T(WM))^{-1}\right), && (6.42)\\
&= \mathrm{Tr}\left(M^T W^T D(\lambda) W M^{-1}(W^T W)^{-1} M^{-T}\right), && (6.43)\\
&= \mathrm{Tr}\left(W^T D(\lambda) W (W^T W)^{-1}\right), && (6.44)\\
&= \mathcal{R}_{\mathcal{P}}[f]. && (6.45)
\end{aligned}$$

The significance of this result is that it demonstrates $\mathcal{R}_{\mathcal{P}}$ to be invariant to linear transformations of f which preserve the space spanned by the functions. Much like the Ritz value of an trial vector is invariant to rescaling, or the angle between two planes is invariant to linear transformations of the basis vectors defining the planes, $\mathcal{R}_{\mathcal{P}}[f]$ is only a functional of the space spanned by f. This space — the set of all m-dimensional linear subspaces of a vector or Hilbert space — is referred to as a *Grassmann manifold*.[237]

6.B A tension between the spectral and probabilistic approaches

Here we show, by way of a simple analytical example, the extent to which the variational and probabilistic approaches to the analysis of molecular dynamics data are indeed distinct. By explicitly constructing the propagator eigenfunctions for a Brownian harmonic oscillator, we show that the rank-m truncated propagator, $\mathcal{P}_m(\tau)$, built from the first m eigenpairs of $\mathcal{P}(\tau)$ is not in general a nonnegativity-preserving operator. That is, for some valid initial distributions, $p_t(\mathbf{x})$, the propagated distribution, $\tilde{p}^{(m)}_{t+\tau}(\mathbf{x}) = \mathcal{P}_m(\tau) \circ p_t(\mathbf{x})$, fails to be non-negative throughout Ω and thus does not represent a valid probability distribution.

$$p^{(m)}_{t+\tau}(\mathbf{x}) \ngeq 0 \; \forall \; \mathbf{x} \in \Omega \tag{6.46}$$

This indicates that variational and probabilistic approaches have the potential to be almost contradictory in what they judge to be "good" models of molecular kinetics.

Consider the diffusion of a Brownian particle in the potential $U(x) = x^2$. For simplicity, we take the temperature and diffusion constant to be unity. This is an Ornstein-Uhlenbeck process, and the dynamics are described by the Smoluchowski equation,

$$\frac{\partial}{\partial t} p_t(x) = \mathcal{L} \circ p_t(x), \tag{6.47}$$

with infinitesimal generator $\mathcal{L}$ given by

$$\mathcal{L} = \frac{\partial^2}{\partial x^2} + 2\frac{\partial}{\partial x} x, \tag{6.48}$$

and stationary distribution $\mu(x) = \pi^{-1/2} e^{-x^2}$.

We can expand the generator in terms of its eigenfunctions, $\phi_n(x)$, and eigenvalues,

ξ_n, defined by,

$$\mathcal{L} \circ \phi_n(x) = \xi_n \phi_n(x), \tag{6.49}$$

which can be recognized as the Hermite equation whose solutions are related to the Hermite polynomials, H_n. For $n = \{0, 1, \ldots\}$ the solutions are

$$\phi_n(x) = c_n e^{-x^2} H_n(x), \tag{6.50}$$

$$\xi_n = -2n, \tag{6.51}$$

$$c_n^2 = (2^n n! \pi)^{-1}, \tag{6.52}$$

where the normalizing constants, c_n, are chosen such that $\langle \phi_n, \phi_m \rangle_{\mu^{-1}} = \delta_{nm}$.

The propagator $\mathcal{P}(\tau)$ can be formed by integrating Eq. (6.46) with respect to t, giving

$$\mathcal{P}(\tau) = e^{\tau \mathcal{L}}. \tag{6.53}$$

$\mathcal{P}(\tau)$ shares the same eigenfunctions as $\mathcal{L}$. Its eigenvalues, λ_n, are related to the eigenvalues of $\mathcal{L}$ by

$$\lambda_n = e^{-\tau \xi_n}. \tag{6.54}$$

We now define the rank-m truncated propagator, $\mathcal{P}_m(\tau)$, such that

$$\mathcal{P}_m(\tau) \circ p_t = \sum_{n=0}^{m-1} \lambda_n \langle p_t, \phi_n \rangle_{\mu^{-1}} \phi_n \tag{6.55}$$

$$= \sum_{n=0}^{m-1} e^{-2n\tau} c_n e^{-x^2} H_n(x) \left[\int_{-\infty}^{\infty} dx' \; c_n \sqrt{\pi} \, p_t(x') H_n(x') \right] \tag{6.56}$$

Consider an initial distribution, $p_t(x) = \delta(x - x_0)$, propagated forward in time by $\mathcal{P}_m$. Let $\tilde{p}_\tau^{(m)} = \mathcal{P}_m(\tau) \circ \delta(x - x_0)$. Then, Eq. (6.56) simplifies to

$$\tilde{p}_\tau^{(m)}(x) = \sum_{n=0}^{m-1} \frac{1}{2^n n! \sqrt{\pi}} e^{-2n\tau} \, e^{-x^2} H_n(x) H_n(x_0). \tag{6.57}$$

Consider now the specific case of $m = 2$. Using the explicit expansion $H_0(x) = 1$, and $H_1(x) = 2x$, we have

$$\tilde{p}_\tau^{(2)}(x) = \frac{1}{\sqrt{\pi}} e^{-x^2} \left(1 + 2x x_0 e^{-2\tau}\right). \tag{6.58}$$

Note that Eq. (6.58) has a zero when $x = -e^{2\tau}/2x_0$, and that

$$\tilde{p}_\tau^{(2)}(x) < 0 \text{ when } \begin{cases} x < -e^{2\tau}/2x_0 & \text{if } x_0 > 0 \\ x > -e^{2\tau}/2x_0 & \text{if } x_0 < 0. \end{cases} \tag{6.59}$$

Because of this non-positivity, $\tilde{p}_\tau^{(2)}(x)$ is not a valid probability distribution.

This example demonstrates that the rank-m truncated propagator need not, in general, preserve the positivity of distributions it acts on. Therefore, if such a model of the dynamics are fit or assessed via maximum-likelihood methods on datasets consisting of observed transitions, despite being *optimal* by spectral norm, the true rank-m truncated propagator may appear to give a log likelihood of $-\infty$. The variational and probabilistic approaches to modeling molecular kinetics can indeed be very different.

6.C Double well potential integrator and eigenfunctions

To discretize the Brownian dynamics stochastic differential equation in Eq. (6.25) with reflecting boundary conditions at $-\pi$ and π, we used the Euler integrator,

$$x_{t+1} = bc\left(x_t + \left(\nabla V(x_t) + \sqrt{2D}R(t)\right)\Delta t\right), \tag{6.60}$$

where steps that went outside the boundaries by a given distance were reflected back into the interval with a matching distance to the boundary:

$$bc(x) = \begin{cases} 2\pi - x & \text{if } x > \pi, \\ -2\pi - x & \text{if } x < -\pi, \\ x & \text{otherwise.} \end{cases} \tag{6.61}$$

We computed the propagator eigenvalues by discretizing the interval into n MSM states $\{s_i\}_{i=1}^n$, following Eq. (6.27), and computing the matrix elements without stochastic sampling. This calculation is more straightforward by working with the the transition matrix $T \in \mathbb{R}^{n \times n}$:

$$T_{ij} = \mathbb{P}\left[x_{t+\tau} \in s_j | x_t \in s_i\right], \tag{6.62}$$

Instead of the correlation and overlap matrices, C and S, directly. Note that as shown by Nüske et al.[234] and Prinz et al.,[97], $T = S^{-1}C$. Thus the eigenvalues of T are identical to the generalized eigenvalues of (C, S).

To calculate the matrix elements of T, we consider each each state, s_i, represented by its left endpoint, x_i. For each pair of states (i, j), we calculate the probability of the random force required to transition between them in one step, from Eq. (6.60), taking into account the fact that because of the reflecting boundary conditions, the transition could have taken place via a transition outside the interval followed by a reflection.

Let δx be the width of the states, $\delta x = n^{-1}2\pi$. For each $i \in \{1, \ldots n\}$ and $k \in \{-n, \ldots, n\}$, we calculate the action for a step from x_i to $x_i + k\delta x$.

$$a_{ik} = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{((x_i + k\delta x) - x_i + \nabla V(x_i)\Delta t)}{\sqrt{2D}}\right). \tag{6.63}$$

Let $j'_{ik} \in \{1, \ldots, n\}$ be the index of state which contains $bc(x_i + k\delta x)$. We calculate T_{ij} by summing the appropriate action terms which, because of the boundary

conditions, get reflected into the same ending state:

$$T_{ij} = \sum_{k=-n}^{n} a_{ik}\, \delta_{j'_{ik},j}. \tag{6.64}$$

where $\delta_{i,j}$ is the Kronecker delta. This calculation is implemented in the file `brownian1d.py` in the MSMBuilder3 package. The eigenvalues of T converge rapidly as n increases. Our results in Fig. 6.1 use $n = 500$.

6.D Landmark UPGMA clustering

Landmark-based UPGMA (Unweighted Pair Group Method with Arithmetic Mean) agglomerative clustering is a simple scalable hierarchical clustering which does not require computing the full matrix of pairwise distances between all data points. The procedure first subsamples l "landmark" data points at regular intervals from the input data. These data points are then clustered using the standard algorithm, resulting in n clusters. [259] Let S_n be the set of landmark data points assigned by the algorithm to the cluster n, and $d(x, x')$ be the distance metric employed. Then, each remaining data point in the training set as well as new data points from the test set, x^*, are assigned to cluster, $s(x^*) \in \{1, \ldots, n\}$, whose landmarks they are on average closest to:

$$s(x^*) = \arg\min_n \frac{1}{|S_n|} \sum_{x \in S_n} d(x^*, x). \tag{6.65}$$

6.E Simulation Setup

We performed all-atom molecular dynamics simulations of terminally-blocked octaalanine (Ace-$(Ala)_8$-NHMe) in explicit solvent using the GROMACS 4 simulation package,[45] the AMBER ff99SB-ILDN-NMR forcefield,[260] and the TIP3P water model.[212] The system was energy minimized, followed by 1 ns of equilibration using the velocity rescaling thermostat (reference temperature of 298K, time constant of 0.1 ps),[261] Parrinello-Rahman barostat (reference pressure of 1 bar, time constant of

1 ps, isotropic compressibility of 5×10^{-5} bar),[262] and Verlet integrator (time step of 2 fs). Production simulations were performed in the canonical ensemble using the same integrator and thermostat. Nonbonded interactions in all cases were treated with the particle mesh Ewald method, using a real space cutoff distance for Ewald summation as well as for van der Waals interactions of 10.0 Å.[17] Twenty six such simulations were performed, with production lengths between 20 and 150 ns each. The total aggregate sampling was 1.74 μs.

Chapter 7

Identification of simple reaction coordinates from complex dynamics

Reaction coordinates are widely used throughout chemical physics to model and understand complex chemical transformations. We introduce a definition of the natural reaction coordinate, suitable for condensed phase and biomolecular systems, as a maximally predictive one-dimensional projection. We then show this criterion is uniquely satisfied by a dominant eigenfunction of an integral operator associated with the ensemble dynamics. We present a new sparse estimator for these eigenfunctions which can search through a large candidate pool of structural order parameters and build simple, interpretable approximations that employ only a small number of these order parameters. Example applications with a small molecule's rotational dynamics and simulations of bovine pancreatic trypsin inhibitor show that this approach can filter through statistical noise to identify simple reaction coordinates from complex dynamics.

This chapter is previously unpublished work written by the the author and V. S. Pande, being prepared for submission to *J. Chem. Phys.*

7.1 Introduction

The reaction coordinate — a single collective variable that quantifies progress in a chemical reaction — is a ubiquitous concept in chemical kinetics.[263,264] Reaction coordinates are, for example, required for computing reaction rates using transition state theory,[263–265] computing kinetically meaningful free energy barriers,[266] and accelerating conformational sampling in many biomolecular simulation protocols.[267–271] Their most important use, however, is often in facilitating insight into chemical reaction mechanisms.[272–274]

Implicit in the concept is the notion that the measurement of reaction coordinate is dynamically informative, and provides a proxy for the rate-limiting dynamical processes of the system. Reactions in soft matter and condensed phase systems, such as the folding of a protein or an enzyme-catalyzed chemical transformation take place in a high-dimensional phase space that may include many uninvolved solute and solvent coordinates. In this regime, identification of reaction coordinates is difficult.[274] Physical intuition may suffice to determine these critical degrees of freedom for low-dimensional systems, such as simple bimolecular gas-phase reactions. But in more complex processes involving tens of thousands or more atoms, rough energy landscapes, and/or solvent dynamics, methods to identify the reaction coordinate that rely merely on physical intuition or trial and error can be *ad hoc* and unsystematic.[29,275–277]

We recognize that the identification of a system's reaction coordinate(s) is invaluable for physical interpretation of complex molecular systems, that researchers now have access to extremely large data sets of unbiased molecular dynamics (MD) simulations of biologically relevant macromolecules, and that the interpretation of these data is often a major bottleneck.[19] We therefore aim to develop a method to use these MD data sets to *infer* accurate and interpretable reaction coordinates. Our approach

builds on time-structure independent components analysis (tICA).[166,167] But these tICA-derived reaction coordinates can be a black box; they are difficult to interpret physically because of their abstract construction as linear combinations of a large number of structural features. In contrast, our new estimator explicitly adds a sparsity consideration into the formulation to filter through statistical noise and identify simple physical reaction coordinates from complex dynamics.

The structure of this paper is as follows: First, we define the *natural reaction coordinate(s)* based on a set of intuitive mathematical properties that these collective variables should satisfy. After introducing these properties, we discuss their relationship to other commonly used definitions of the reaction coordinate. Next, we prove that this definition is satisfied by the leading eigenfunctions of an integral operator governing the ensemble dynamics.* Finally, we introduce and demonstrate a practical new estimator which can approximate these reaction coordinates as extremely sparse, interpretable, regularized linear combinations of structural order parameters.

7.2 Defining the natural reaction coordinate

Although (or perhaps because) the idea of the reaction coordinate is so widely used in chemical kinetics, the community has not always agreed on its precise meaning. A number of different definitions thereof have been proposed, including the minimum energy path or intrinsic reaction coordinate (MEP),[279–282] the minimum action path (MAP),[283–287] and the committor function.[288,289]

In order to proceed in the face of this definitional ambiguity, we begin from first principles and propose a set of properties that a natural reaction coordinate should satisfy for any time-homogeneous, reversible, ergodic Markov process. This approach is geared towards conformational dynamics of soft matter systems, and we make none of the assumptions common in chemical kinetics about the existence of two metastable

*For systems that evolve under Langevin dynamics, the operator is a backward Fokker-Planck operator.[278] For a discrete-time reversible Markov chain like thermostated Hamiltonian dynamics integrated with a finite-timestep integrator, the associated operator is a backward transfer operator.[36]

states, about the relative importance of entropic or enthalpic barriers, about low temperature, or about the number of pathways that are possible. This level of generality does come with a trade off; it makes it impossible to leverage quasi-equilibrium approximations, and our algorithms will require equilibrium sampling. The mathematical properties which we specify require that the natural reaction coordinate (a) be a dimensionality reduction that (b) is defined only by the system's dynamics, and that (c) is the maximally predictive projection about the future evolution of the system. Below, we describe and define each of these criteria in detail.

7.2.1 A dimensionality reduction from Ω to $\mathbb{R}$

The natural reaction coordinate should be a function which maps any point in the system's full phase space to a single real number. Notating the reaction coordinate as q, and phase space as Ω,[†] we may specify this as

$$q : \Omega \to \mathbb{R}.$$

The reason for this form is that it should be well-defined to calculate how "far along" the reaction coordinate any conformation is, or to speak about the mean value of the reaction coordinate for some equilibrium or non-equilibrium ensemble of conformations. Reaction coordinates taking this form include geometric or physical observables which could, in principle, be as simple as the distance between two specific atoms. Later on, we will show how the formulation also extends naturally to multiple orthogonal reaction coordinates.

On the other hand, path-based definitions of the reaction coordinate such as the MEP or MAP do not take this form. Instead of functions from Ω to $\mathbb{R}$, a path through phase space is a function from $\mathbb{R}$ to Ω. These paths map an arc length to a phase space coordinate, and the value of the reaction coordinate is undefined for all

[†]We use the phrase 'phase space' to refer to either a position, momenta phase space, or a position-only configuration space, depending on the underlying dynamics. For thermostatted Hamiltonian or Langevin dynamics, $\Omega = \mathbb{R}^{6N}$, where N is the number of atoms. For overdamped Langevin, also called Brownian or Smoluchowski, dynamics, $\Omega = \mathbb{R}^{3N}$. In periodic boundary conditions, the position space is some $3N$-dimensional torus, but the exact definition of Ω is not critical for our purposes.

conformations in Ω that are not on this path. For the minimum energy path, this issue was discussed by Natanson *et al.*, [290] who showed that while a reaction coordinate of the form $\Omega \to \mathbb{R}$ could be defined by introducing a projection operator onto the MEP, there was considerable ambiguity in the choice of projection function. This ambiguity was present even for reactive systems containing only 3 atoms without roughness, and are exacerbated in high-dimensional and condensed phase systems. This is one factor which makes the $\Omega \to \mathbb{R}$ formulation more attractive than the $\mathbb{R} \to \Omega$ formulation.

7.2.2 Uniquely determined by the dynamics

The natural reaction coordinate should be uniquely defined by the equations of motion that govern the underlying dynamics in Ω, which include the system's Hamiltonian, boundary conditions, and integration scheme. We wish to define the natural reaction coordinate in a way that does not depend on particular "reaction" or "product" conformations or subsets of phase space.

Although it may appear intuitive to define the reaction coordinate in terms of two end points or two states, this definition has a number of formal and practical drawbacks. Subdividing phase space into non-overlapping reactant and product states, $A \subset \Omega$, $B \subset \Omega$, $A \cap B = \emptyset$, is a useful device, but this is a construct imposed by the modeller, not the underlying Hamiltonian. All experimentally measurable observables, such as ensemble averages, single-molecule time series, or time-correlation functions of a spectroscopic quantity are independent of whether the modeller labels certain regions of phase space as A or B.

For systems containing a small number of atoms, it is often relatively obvious how these states should be determined: e.g. for a bond-forming reaction, one can simply measure whether the distance between the atoms is greater than a certain cutoff. And when the states are metastable, many quantities which might formally depend on the exact specification of the states' boundaries in fact have a very weak dependence thereon, as long as the perturbed state boundaries are still metastable.[291] But in high-dimensional systems where entropy plays a dominant role, and when confronted with significant roughness in the energy landscape on energy scales less than $k_B T$,

it can be very difficult in practice to identify these metastable states. Furthermore, many systems have more than two metastable states.

Consider protein folding dynamics, where A and B would generally be taken to be the protein's folded and unfolded states. A number of practical definitions of the folded or unfolded state, based on metrics including root-mean-square deviations to a crystal structure, numbers of native contacts, or radii of gyration are defensible. None, however, are obviously mandated. If the definition of the natural reaction coordinate depends on the exact line-drawing between folded and unfolded, each definition of the state boundaries would lead to a slightly different natural reaction coordinate, with no criteria to judge which is optimal.

In our view, a formal defintion of the natural reaction coordinate should be unique and independent of any partitioning of phase space into regions, and only a function of the system's underlying dynamics. As a dimensionality reduction, the natural reaction coordinate should teach us about the system's metastable states, not the other way around.

7.2.3 Maximally predictive projection

Finally, the key property that we use to define the natural reaction coordinate relates to its ability to optimally predict the dynamics. Of all possible one-dimensional measurements of the state of some high-dimensional dynamical system, the natural reaction coordinate should be the most informative about the future evolution of the system. This relates to the expectation, common in chemical kinetics, that the dynamics along the reaction coordinate are rate-limiting, and that all other degrees of freedom in the system equilibrate more rapidly. The maximally predictive single coordinate will measure progress with respect to the rate-limiting bottlenecks, as the orthogonal coordinates can more reliably be assumed to be at, or near, equilibrium.

We now formalize this notion mathematically. To begin, we define the following quantities:

- The system has a unique equilibrium distribution over phase space, $\mu(x) : \Omega \to \mathbb{R}$. Note that $\forall x,\ \mu(x) > 0$ and $\int_\Omega dx\ \mu(x) = 1$.

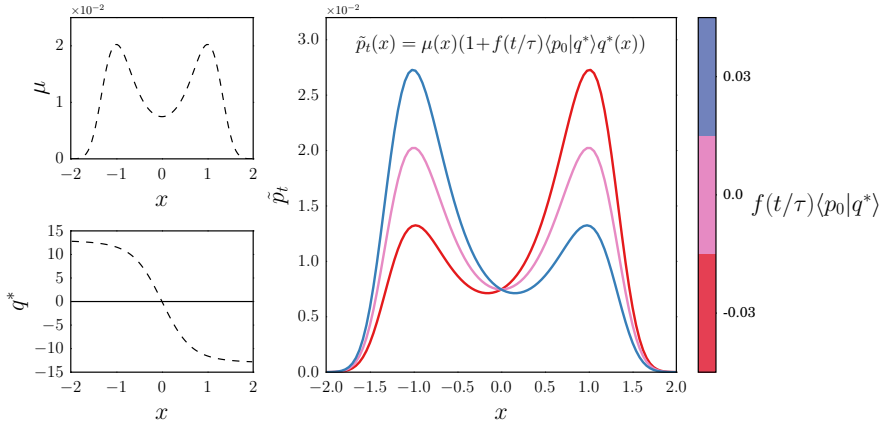


Figure 7.1: Predictions, $\tilde{p}_t$, made by the natural reaction coordinate, q^*, for Smoluchowski diffusion on two-well potential, $U(x) = (x-1)^2(x+1)^2$ with a uniform diffusion constant, $D = 1$. The upper left panel shows the stationary distribution, $\mu(x)$, and the lower left panel shows the natural reaction coordinate, q^*, which changes sign between the two metastable states. The main panel shows the family of possible predictions, $\tilde{p}_t$ that can be made by Eq. (7.1) using this choice of q, indicating the variable partitioning of density between the two basins. For an arbitrary initial distribution, p_0, this coordinate minimizes the worst-case predictive error about the future ensemble p_t given only knowledge of the current ensemble's projection onto q. As discussed in Section 7.3, q^* was calculated from the second eigenfunction of the Smoluchowski operator, which was determined in this case using the FiPy PDE solver.[292]

- Initially, the state of an ensemble is described by a (generally non-equilibrium) probability distribution over phase space, $p_0(x) : \Omega \to \mathbb{R}$.
- We consider an *ansatz* reaction coordinate, $q(x) : \Omega \to \mathbb{R}$, and an associated scalar, τ, which will be interpreted as a timescale of the dynamics along the *ansatz* reaction coordinate.
- The scalar projection of the initial distribution, p_0, along the reaction coordinate is measured as $\langle q|p_0\rangle = \int_\Omega dx\ q(x)p_0(x)$.
- At some later time, $t > 0$, the system will have evolved from p_0 to a new distribution over phase space, $p_t(x) : \Omega \to \mathbb{R}$, according to the underlying equations of motion for the dynamics. Note that while p_t is a probability distribution, it is not a random variable; it is produced deterministically from p_0 and the system's equations of motion.

Now, consider the task of constructing an approximation to p_t. This approximation, $\tilde{p}_t$, is constrained to depend only on $\mu(x)$, $q(x)$, τ, t, and $\langle q|p_0\rangle$. That is, given knowledge of the equilibrium distribution, the *ansatz* reaction coordinate, its timescale, and *no other* information about the current ensemble, p_0, beyond its projection onto the *ansatz* reaction coordinate, our goal is to construct a prediction of the future ensemble at some later time t.

A basic dimensional analysis argument and the constraint that $\int_\Omega \tilde{p}_t = 1$ is sufficient to establish that, assuming that q is measured in a system of units such that it has mean zero and unit variance in the equilibrium ensemble, the functional form of $\tilde{p}_t$ given q must be

$$\tilde{p}_t(x) = \mu(x) + f(t/\tau)\langle q|p_0\rangle q(x)\mu(x), \tag{7.1}$$

for some non-random function, f, that is independent of x. Later on, we will show that f is necessarily an exponential, $f(t/\tau) = e^{-t/\tau}$. For diffusion on a double well potential, a diagrammatic example of the family of predictions, $\tilde{p}_t$, that can be made given a particular choice of q is shown in Fig. 7.1.

Even with full knowledge of the Hamiltonian and equations of motion, this prediction will not be exact because the one-dimensional measurement, $\langle q|p_0\rangle$, gives incomplete information about $p_0(x)$. We define the error in the prediction, $E_{p_0}[q]$, as the μ^{-1}-weighted mean squared error,

$$\begin{aligned} E_{p_0}[q] &= ||p_t(x) - \tilde{p}_t(x)||^2_{\mu^{-1}} \\ &= \int_\Omega dx\ \mu^{-1}(x)(p_t(x) - \tilde{p}_t(x))^2. \end{aligned} \tag{7.2}$$

Note that this error depends on the arbitrary initial distribution. To remove this dependency, we consider the worst-case error by maximizing over all possible initial distributions,

$$E[q] = \max_{p_0} E_{p_0}[q], \tag{7.3}$$

$$q^* = \arg\min_q E[q]. \tag{7.4}$$

The functional $E[q]$ thus measures how well the measurement of an arbitrary collective variable can be used to predict the future state of the system. We define the *natural reaction coordinate*, q^*, as the minimizer of $E[q]$. It is, in this sense, the collective variable which is maximally informative about the system's dynamics.

7.2.4 Alternative Definitions

The approach we have taken is not the only one possible. Note first the choice of error functional, Eq. (7.2). While it may not be initially intuitive, the μ^{-1}-weighting on the norm is the logical choice for a mean squared error. It is the μ^{-1} measure, combined with detailed balance, that ensures, for example, that minimizer, q^*, is strictly independent of t (see Section 7.3.1). A different choice, like the Kullback-Leibler divergence of Wasserstein distance would be possible,[293] but lead to substantially different results. Additionally, observe that in contrast to many other formulations,[294–296] our approach is not based on the explicit construction of a one-dimensional Smoluchowski-like diffusion along the reaction coordinate.

Next, we turn our discussion to an alternative reaction coordinate definition, the committor function. This quantity was first introduced by Onsager as the splitting probability for ion-pair recombination.[297] The committor is defined based on the prior identification of two non-overlapping states, $A \subset \Omega$, $B \subset \Omega$, $A \cap B = \emptyset$, which do not fully partition phase space, $A \cup B \subset \Omega$. Then, the committor, $p_A(x)$, is defined as the probability that a trajectory initiated from x would enter the set A before entering B.[288,289] In the context of protein folding, where A is taken to be the protein's folded state, the committor is often referred to as p-fold.[298,299] The committor, $p_A(x)$, takes a value of 1 for conformations inside A, and 0 for conformations inside B. The condition $\{x : p_A(x) = 1/2\}$ defines a transition state ensemble or separatrix — the set of conformations equally likely to commit to either state A or state B.

Using the concept of the ensemble of *transition paths*, which are defined as trajectory segments following the moment at which the system has exited the set A and up until the systems enters the set B, without re-entering A, Hummer proved an important result.[291] He showed that, for diffusive dynamics, the probability of being on a transition path given that the system is at x, $\mathbb{P}(\mathrm{TP}_{AB}|x)$, is determined by the committor alone, $\mathbb{P}(\mathrm{TP}_{AB}|x) = 2p_A(x)(1 - p_A(x))$. This implies also that the separatrix can be identified as the set of conformations which are most likely to be on reaction paths.

A number of computational methods build approximations to the transition path ensemble, committor or isocommittor surfaces. These include transition path sampling (TPS),[288,289] transition interface sampling,[300] and the finite temperature string method.[301,302]

Most of the existing algorithms that identify physical reaction coordinates from molecular simulations are based on committor analysis or TPS.[303–310] In the simplest version, one initializes a large collection of trajectories from isosurfaces of an *ansatz* reaction coordinate and measures which of the two basins, A or B, they commit to. If this coordinate is a good approximation to the committor, the measured splitting fraction will be narrowly peaked around the characteristic value.[311] Criteria based on this observation can then be used to screen an *ansatz* reaction coordinate, or optimize the parameters of a model for the reaction coordinate.[304] More efficient maximum

likelihood method which fit a parametric model for the reaction coordinate from TPS data further refine this approach.[306,308]

By design, these algorithms rest on the pre-identification of the A and B states, and are not naturally suited to systems with more than two metastable states, although multiple-state extensions are available.[312] For the reasons discussed in Section 7.2.2, we thus dispute the claim that the committor should be taken as the *perfect* or *exact* reaction coordinate.[304,305,313,314] The authors' experiences with large-scale simulations of protein folding and activation on Folding@Home have shown that it can be difficult to locate and precisely define these metastable states. This suggests that, for an important class of problems, the metastable states should be constructed from the output of some model, as opposed to being treated as a modelling input.[25,56,118] These considerations motivate our formulation of the natural reaction coordinate in a manner independent of the choice to label certain regions of phase space as A or B.

7.3 A dominant eigenfunction is the natural reaction coordinate

In this section, we demonstrate that the natural reaction coordinate, as defined by the minimizer of Eq. (7.4), is the second leading eigenfunction of an integral operator associated with a system's Markovian dynamics in Ω. For simplicity, we work here with a discrete-time Markov chain, $\{X_0, X_1, X_2, \ldots\}$, such as a typical all-atom MD simulation with a finite time step integrator, assuming only that the prediction interval t is greater than 1 step (typically on the order of 2 fs). Afterwards, we note why the same results apply if the underlying dynamics are a continuous-time Markov process, and discuss the natural generalization to multiple reaction coordinates.

We note that others have also used the term reaction coordinate for these eigenfunctions.[224,315]

7.3.1 Preliminaries

The one-step dynamics of a system's Markovian evolution forward in time can be completely described in terms of a stochastic transition density kernel,

$$p(x, y)dy = \mathbb{P}(X_{t+1} \in B_\epsilon(y)|X_t = x), \tag{7.5}$$

where $B_\epsilon(y)$ is the open ϵ-ball centered at y with infinitesimal measure dy. Essentially, this kernel measures the conditional probability of jumping from x to y in one step.[‡] Integrating over the initial ensemble, p_t, gives a Chapman-Kolmogorov equation for the evolution of the ensemble to p_{t+1},

$$p_{t+1}(y) = \int_\Omega dx\ p_t(x)p(x, y). \tag{7.6}$$

By assumption, we consider only ergodic and reversible Markov processes. Ergodicicity is the property that there do not exist two or more regions of Ω that are dynamically disconnected. That is, the integrated transition density is strictly positive, $\int_{y\in A} p(x, y) > 0$ for all x and all non-empty subsets of Ω, A. The reversibility condition is that the Markov chain obeys a detailed balance equation with respect to its stationary measure, $\mu(x)$,

$$\mu(x) \cdot p(x, y) = \mu(y) \cdot p(y, x). \tag{7.7}$$

For molecular dynamics, $\mu(x)$ is the equilibrium distribution associated with the thermodynamic ensemble that the system is sampling, such as the Boltzmann distribution at constant temperature, and reversibility can be interpreted as a type of generalized symmetry on the function $p(x, y)$.

The form of our maximally predictive projection formulation suggests that the reaction coordinate acts like a perturbation to the equilibrium distribution. This

[‡]For example, if the underlying dynamics are overdamped Langevin on a potential energy function $U(x)$ in units of kT with unit diffusion constant, simulated using an Euler-Maruyama integrator with a unit time step, the stochastic transition density kernel, $p(x, y)dy$, would be the probability density function of a Gaussian distribution with mean $\bar{y} = x - \nabla U(x)$ and variance $\sigma^2 = 2$.

suggests that we consider the equations for the time evolution of a new function, $u_t(x) \equiv p_t(x)/\mu(x)$, which measures the same information as $p_t(x)$, but encoded the excess or depletion of probability in an ensemble with respect to the stationary distribution. Applying the Chapman-Kolmogorov equation to the time evolution of u_t, we have

$$u_{t+1}(y) = \frac{1}{\mu(y)} \int_\Omega dx\ u_t(x)\mu(x)p(x,y). \tag{7.8}$$

This equation is taken to define the action of the one step backward transfer operator, $\mathcal{T}(1)$, which is uniquely defined by the transition density kernel,

$$u_{t+1}(y) = [\mathcal{T}(1) \circ u_t](y). \tag{7.9}$$

The transfer operator has many properties — we refer the interested reader to the monograph of Schütte, Huisinga, and Deuflhard for mathematical details. [36] For our purposes, the most relevant properties are that $\mathcal{T}(1)$ is compact and self-adjoint, and thus has a complete, countable set of real eigenfunctions and eigenvalues,

$$\mathcal{T}(1) \circ \psi_i = \lambda_i \psi_i, \tag{7.10}$$

which we number in decreasing order by eigenvalue magnitude. Each ψ_i can be assumed to be normalized such that they are orthonormal with respect to the μ-weighted inner product,

$$\langle \psi_i | \psi_j \rangle_\mu = \int_x dx\ \mu(x)\psi_i(x)\psi_j(x) = \delta_{ij}. \tag{7.11}$$

Furthermore, the largest eigenvalue is $\lambda_1 = 1$, with associate eigenfunction $\psi_1(x) = 1$, and the absolute values of the remaining eigenvalues lie within the unit interval, $|\lambda_i| < 1$.[36]

These properties imply that the action of $\mathcal{T}(1)$ on u_t can be written as a spectral

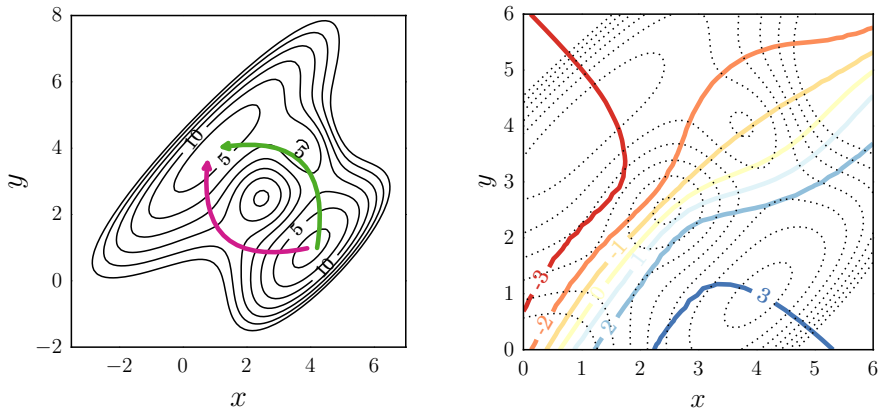


Figure 7.2: An example two-dimensional potential energy surface (left panel) with two of the possible pathways shown in magenta and green. The right panel shows a contour plot of the natural reaction coordinate, $\psi_2(x, y)$, for Smoluchowski dynamics at $kT = 5$ with a homogeneous diffusion constant, $D = 1$, overlaid on the potential energy surface, which is shown with dotted contours. We emphasize that while the natural reaction coordinate, $\psi_2 : \Omega \to \mathbb{R}$, provides a measure of progress with respect to any path between the two minima, it cannot be viewed as a single pathway itself.

decomposition,

$$[\mathcal{T}(1) \circ u_t](x) = \sum_{i=1}^{\infty} \lambda_i \langle u_t | \psi_i \rangle_\mu \psi_i(x). \tag{7.12}$$

By repeatedly applying the single-step $\mathcal{T}(1)$ operator, we can also build the multi-step $\mathcal{T}(t)$ operator. Because of the linearity of the operator and orthonormality of the eigenfunctions, each repeated application only pulls out another factor of the eigenvalue in the sum. The spectral decomposition of $\mathcal{T}(t)$ is thus

$$[\mathcal{T}(t) \circ u_t](x) = \sum_{i=1}^{\infty} \lambda_i^t \langle u_t | \psi_i \rangle_\mu \psi_i(x). \tag{7.13}$$

7.3.2 Analysis of the error functional

We now apply this spectral decomposition of the transfer operator to the analysis of the error functional from Section 7.2.3 and prove why the natural reaction coordinate

is equal to the second transfer operator eigenfunction, $q^* = \psi_2$.

First, observe that the form of the prediction about the future state of the system made using the reaction coordinate, Eq. (7.1), can also be written as some operator that maps $p_0 \to \tilde{p}_t$, or equivalently in u-notation as an approximate transfer operator, $\tilde{\mathcal{T}}(t)$, that maps $u_0 \to \tilde{u}_t$, where $\tilde{u}_t(x) \equiv \tilde{p}_t(x)/\mu(x)$.

$$\tilde{u}_t = \langle u_0|1\rangle_\mu + f(t/\tau)\langle u_0|q\rangle_\mu q(x) \tag{7.14}$$

$$= \tilde{\mathcal{T}}(t) \circ u_0. \tag{7.15}$$

The approximate transfer operator, $\tilde{\mathcal{T}}(t)$, is rank 2; it has two non-zero eigenvalues, 1 and $f(t/\tau)$, with associated eigenfunctions 1 and $q(x)$ respectively.

Next, we rewrite the error the error functional, Eq. (7.2) in u-notation as well,

$$E_{u_0}[q] = \int_\Omega dx\ \mu(x)(u_t(x) - \tilde{u}_t(x))^2 \tag{7.16}$$

$$= ||(\mathcal{T}(t) - \tilde{\mathcal{T}}(t)) \circ u_0||_\mu^2 \tag{7.17}$$

$$E[q] = \max_{u_0} ||(\mathcal{T}(t) - \tilde{\mathcal{T}}(t)) \circ u_0||_\mu^2, \tag{7.18}$$

where the maximum is understood to be taken over properly normalized u_0, $||u_0||_\mu = 1$.

Now, we prove by why $\psi_2 = \min_q E[q]$. Observe that for any q, there exists a function $v(x)$ in the span of the first three eigenfunctions of $\mathcal{T}$, $v = a_1\psi_1 + a_2\psi_2 + a_3\psi_3$, which is normalized, $\langle v|v\rangle_\mu = 1$, and which is in the null space of $\tilde{\mathcal{T}}$, $\tilde{\mathcal{T}} \circ v = 0$.[§] Since $E[q]$ is the maximum of $E_{\mu_0}[q]$ over all possible μ_0, it also must be greater than the

[§]To be concrete, set $a_1 = 0$, and choose a_2 and a_3 to satisfy $a_2\langle\psi_2|q\rangle_\mu = -a_3\langle\psi_3|q\rangle_\mu$ and $a_2^2 + a_3^2 = 1$.

error incurred for this particular starting distribution, $\mu_0 = v$. Thus,

$$E[q] \geq ||(\mathcal{T}(t) - \tilde{\mathcal{T}}(t)) \circ v||_\mu^2 \tag{7.19}$$

$$= ||\mathcal{T}(t) \circ v||_\mu^2 \tag{7.20}$$

$$= \sum_{i=1}^{3} \lambda_i^{2t} a_i^2 \tag{7.21}$$

$$\geq \lambda_3^{2t}, \tag{7.22}$$

where the third line only includes a sum up to $i = 3$ because, by construction, v is in the span of the first three eigenfunctions. The final line follows because of the ordering of the eigenvalues and the normalization of v, implying $\sum_{i=1}^{3} a_i^2 = 1$.

Interpreting this inequality, we see that the worst-case prediction error for any *ansatz* reaction coordinate, q, is always greater than or equal to λ_3^{2t}. Furthermore, for the particular choice $q = \psi_2$ and $f(t/\tau) = \lambda_2^t$, the equality is achieved, $E[\psi_2] = \lambda_3^{2t}$.[¶] If we define $\tau \equiv -1/\ln \lambda_2$, we see also that $f(t/\tau)$ can be written as $f(t/\tau) = e^{-t/\tau}$. Therefore, ψ_2 is the natural reaction coordinate, the minimizer of $E[q]$.

The reader may recall that this argument is equivalent to the Eckart-Young Theorem on the optimal low-rank approximation of a matrix.[316] For self-adjoint linear operators, the original result is by Schmidt.[231] See Courant and Hilbert (pp. 161),[232] and and Micchelli and Pinkus[233] for further details.

7.3.3 Continuous-time Markov processes

When the generating process is a continuous-time Markov process, $\mathcal{T}(1)$ has an infinitesimal generator, $\mathcal{L}$,

[¶]To demonstrate that $E[\psi_2] = \lambda_3^{2t}$, note that for this choice of q and $f(t/\tau)$, $\tilde{\mathcal{T}}(t)$ is equal to the sum of the first two terms in the spectral decomposition of $\mathcal{T}(t)$. The squared spectral norm of the difference between the two operators is the square of the largest eigenvalue of the difference operator. The first two eigenpairs having been subtracted out, the square of the largest remaining eigenvalue is λ_3^{2t}.

$$\mathcal{L} = \lim_{t \to 0} \frac{\mathcal{T}(t) - \mathcal{I}}{t}. \tag{7.23}$$

The set of eigenfunctions of $\mathcal{L}$ and $\mathcal{T}(t)$ are equivalent, so for these processes, ψ_2 can be defined in either manner.

7.3.4 Multiple reaction coordinates

One attractive property of this definition of the reaction coordinate is that it generalizes naturally to multiple orthogonal reaction coordinates ordered by timescale.

Recall that the maximally predictive projection criterion from Section 7.2.3 assumed that the approximation, $\tilde{p}_t$, was to be formed only from knowledge of the equilibrium distribution and the *ansatz* reaction coordinate. The multiple coordinate generalization follows from modifying this criteria to assume knowledge of μ and the first $k-1$ eigenfunctions, μ and $\psi_2, \ldots, \psi_{k-1}$. Additionally, assume that the projection of the initial distribution onto each coordinate is available. Then, another application of the Eckart-Young Theorem shows that the maximally predictive remaining *ansatz* coordinate is ψ_k. Multiple orthogonal natural reaction coordinates can thus be defined in a stepwise manner, and shown to be equal to the leading eigenfunctions, $\psi_2, \ldots, \psi_k$.

Systems contains k metastable states will have $k-1$ eigenfunctions whose associated eigenvalues are close to one, separated from the remaining eigenvalues by a so-called spectral gap.[97]

7.3.5 Two-dimensional example

In the left panel of Fig. 7.2, we show an example potential with two possible pathways between the dominant basins. The potential is given by the following expression,[295]

$$\begin{aligned} U(x,y) =& [1 - 0.5\tanh(y-x)](x+y-5)^2 \\ &+ 0.2[((y-x)^2-9)^2 + 3(y-x)] \\ &- 15e^{-(x-2.5)^2-(y-2.5)^2} - 20e^{-(x-4)^2-(y-4)^2}. \end{aligned} \tag{7.24}$$

For Smoluchowski dynamics at $kT = 5$ with a homogeneous diffusion constant, $D = 1$, the natural reaction coordinate, $\psi_2(x, y)$, is shown with solid contour lines in the right panel of Fig. 7.2. Although ψ_2 can be calculated without explicitly notating any two regions A and B as the reactant or product state, it provides a natural measure of progress of any conformation or ensemble between the two dominant metastable states in the upper left and lower right regions of the potential.

7.4 The tICA approximator

Markov state models (MSMs) and time-structure independent components analysis (tICA) are two widely used approximators for ψ_2 that can be parameterized directly from molecular dynamics trajectories.[97,136,166,167] Other popular estimators include diffusion maps and kernel tICA.[224,278,315,317,318]

In the tICA method, the goal is to find the optimal variational approximation to ψ_2 using a linear combination of basis functions. These basis functions are generally structural order parameters that can be evaluated easily for each snapshot in a simulation, such as the distance between certain pairs of atoms or some nonlinear transformation thereof, torsion angles between quartets of atoms, or root-mean-squared deviations to certain landmark conformations.

Assume that there are m linearly-independent basis functions, where typical values of m are in the hundreds to thousands. Without loss of generality, we assume that the basis functions have been mean-subtracted, so that they have zero mean in the equilibrium ensemble. We label the collection of basis functions as $\{\chi_j\}_{j=1}^m$.

Because $\mathcal{T}$ is self-adjoint, it can be shown that the true eigenfunction, ψ_2, satisfies a variational theorem,[194,234]

$$\begin{aligned} \psi_2 = \arg\max_q \ & \langle q|\mathcal{T}(t) \circ q\rangle_\mu \\ & \langle \mu|q\rangle = 0 \\ & \langle q|q\rangle_\mu = 1. \end{aligned} \tag{7.25}$$

Because inner products of the form $\langle q|\mathcal{T}(t) \circ q\rangle_\mu$ can be interpreted as the value of the autocorrelation function of a mean-zero, unit variance observable at time t,[166,194,234] we see as well that ψ_2, in addition to being the most predictive collective variable, as discussed above, is the most slowly decorrelating collective variable under the system's dynamics.

As in variational quantum chemistry methods, this quantity serves as a figure of merit for the optimization of a trial function. Expanding the *ansatz* as $q = \sum_i a_i \chi_i$, the maximization is equivalent to the quadratic optimization problem

$$\begin{aligned} \mathbf{a}^* = \arg\max_{\mathbf{a}} \ & \mathbf{a}^T \mathbf{C}(t) \mathbf{a} \\ & \mathbf{a}^T \mathbf{\Sigma} \mathbf{a} = 1. \end{aligned} \tag{7.26}$$

The solution, $\mathbf{a}^*$, yielding the best approximation to ψ_2 in the span of the basis set is the generalized eigenvector associated with the largest generalized eigenvalue of the matrices $\mathbf{C}(t)$ and $\mathbf{\Sigma}$.[319]

The symmetric matrix, $\mathbf{C}(t)$, and positive-definite matrix, $\mathbf{\Sigma}$, have elements given by,

$$C_{ij}(t) = \langle \chi_i|\mathcal{T}(t) \circ \chi_j\rangle_\mu = \mathbb{E}\left[\chi_i(X_t) \cdot \chi_i(X_0)\right] \tag{7.27}$$

$$\Sigma_{ij} = \langle \chi_i|\chi_j\rangle_\mu = \mathbb{E}\left[\chi_i(X_0) \cdot \chi_j(X_0)\right], \tag{7.28}$$

where the expectations are understood to be taken over the stochastic process. As discussed in detail by Schwantes and Pande[166] and Pérez-Hernández *et al.*,[167] the matrix elements can be estimated by empirical averages over the snapshots in molecular

dynamics trajectories. The matrix $\mathbf{C}(t)$ is a collection of time-lagged correlations between the basis functions, and $\mathbf{\Sigma}$ is a covariance matrix of the basis functions. In Section 7.A, we discuss the use of shrinkage estimators in approximating $\mathbf{\Sigma}$ from timeseries data.

7.5 A sparse approximator for the dominant eigenfunction

The tICA method has one obvious drawback: the solution, our approximate natural reaction coordinate, is a linear combination of all m basis functions, and the loadings are typically non-zero. This makes the solutions difficult to interpret in a mechanistic manner, because hundreds or thousands of different interatomic distances and/or torsion angles, for example, can be combined together into a single collective variable. Because an important property of reaction coordinates is their role in facilitating physical interpretation of the underlying molecular system, we consider it desirable to reduce the number of explicitly used variables.

These same interpretability issues arise with numerous methods in machine learning and statistics. For example, in multivariate linear regression, a response variable is modeled as the linear combination of input variables. Interpretable models, with only a small number of non-zero coefficients, can be obtained using variable selection methods such as the lasso.[320]

In this section, we introduce a new *sparse* approximator for ψ_2. The solution will share the same form as the tICA approximation, $q = \sum_i a_i \chi_i$, except that the vast majority of the expansion coefficients, a_i, will be zero.

One general approach for building sparsity-inducing estimators is to augment the objective function — in our case, Eq. (7.26) — with a regularization term that penalizes model complexity and steers the optimization towards solutions that fit the data well, but also remain simple. By scaling the strength of this term, the modeller can trade off between the two goals.

Arguably the most natural sparsity-inducing regularizer would be the ℓ_0 norm, a

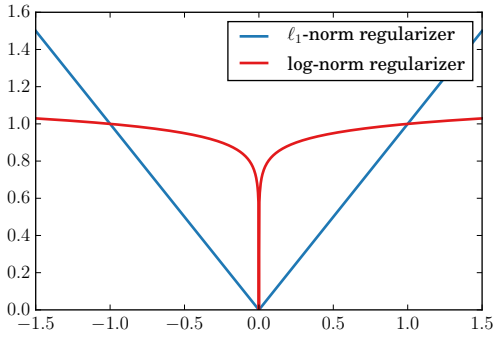


Figure 7.3: The log-norm regularizer used in this work, $\frac{\log(1+|x|/\epsilon)}{\log(1+1/\epsilon)}$, with $\epsilon = 10^{-6}$, as compared to the ℓ_1 norm. The log-norm is a closer approximation to the ℓ_0 norm, and is attractive computationally for this problem because it leads to a more efficient optimization algorithm than the ℓ_1.

penalty proportional to the number of non-zero elements in the solution vector. Unfortunately, ℓ_0-penalized problems generally require an NP-hard combinatorial search. For many problems, such as linear regression, the most common numerically-tractable regularizers which lead to sparse solutions are based on the ℓ_1 norm, which is sometimes interpreted as a relaxation of ℓ_0.[321,322]

However, both the ℓ_0 and ℓ_1 versions of Eq. (7.26) are unsuitable. As discussed by Sriperumbudur, Torres, and Lanckriet,[323] the addition of either an ℓ_0 or ℓ_1 penalty to the Eq. (7.26) objective leads to the intractable problem of maximizing a non-concave objective function. They considered an alternative relaxation of the ℓ_0 penalty,

$$||\mathbf{x}||_0 = \sum_{i=1}^{p} 1_{\{|x_i| \neq 0\}} = \lim_{\epsilon \to 0} \sum_{i=1}^{p} \frac{\log(1 + |x_i|/\epsilon)}{\log(1 + 1/\epsilon)}. \tag{7.29}$$

Choosing a fixed $\epsilon > 0$ yields a regularizer that is concave (see Fig. 7.3), which is a property that will allow the sparse tICA method with this choice regularizer to be optimized efficiently as a difference of convex programs.[324] Therefore, to define this

sparse tICA algorithm, we adopt the following formulation:[‖]

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \mathbf{x}^T\mathbf{C}\mathbf{x} - \rho\sum_{i=1}^{n}\frac{\log(1+|x_i|/\epsilon)}{\log(1+\epsilon)} \\ \text{subject to} \quad & \mathbf{x}^T\mathbf{\Sigma}\mathbf{x} \leq 1, \end{aligned} \tag{7.30}$$

where $\rho \geq 0$ is the regularization strength. At $\rho = 0$, the problem reduces to standard tICA. Larger values of ρ will induce sparsity in the solution vectors.

Investigating sparse generalized eigenvalue problems, Sriperumbudur, Torres, and Lanckriet[323] showed that Algorithm 2 is a globally convergent method for solving Eq. (7.30). The algorithm is iterative, and refines an initial guess. Each iteration requires solving Eq. (7.31), a quadratically-constrained quadratic program (QCQP).

Algorithm 2 Sriperumbudur, Torres, and Lanckriet[323]

Require: $\mathbf{C} \in \mathbb{S}^n, \mathbf{\Sigma} \in \mathbb{S}^n_{++}, \rho > 0, \epsilon > 0$

Choose $\tau > \max(0, -\lambda_{min}(\mathbf{C}))$, $\mathbf{x}^{(0)} \in \{\mathbf{x} : \mathbf{x}^T\mathbf{\Sigma}\mathbf{x} \leq 1\}$

$\rho_\epsilon = \rho/\log(1+\epsilon^{-1})$

while not converged **do**

$w_i^{(l)} \leftarrow \rho_e\tau^{-1}(|x_i^{(l)}| + \epsilon)^{-1}$

$\mathbf{b}^{(l)} \leftarrow (\tau^{-1}\mathbf{C} + \mathbf{I}_n)\mathbf{x}^{(l)}$

$$\begin{aligned} \mathbf{x}^{(l+1)} \leftarrow \underset{\mathbf{x}}{\text{argmin}} \quad & ||\mathbf{x} - \mathbf{b}^{(l)}||_2^2 + ||\mathbf{D}(\mathbf{w}^{(l)})\mathbf{x}||_1 \\ \text{subject to} \quad & \mathbf{x}^T\mathbf{\Sigma}\mathbf{x} \leq 1 \end{aligned} \tag{7.31}$$

end while

These QCQPs are convex. When the number of basis functions, m, is small (less than a few hundred), we have found that they can be solved quickly and with high accuracy by off-the-shelf convex optimization libraries. However, for sparse tICA, our interest is in searching for sparse linear combinations from libraries of many thousands of possible structural order parameters. In this regime, more efficient algorithms are

[‖]At this point, we switch the notation slightly for clarity of presentation. $\mathbf{x} \in \mathbb{R}^m$ will be the vector of sparse tICA expansion coefficients being optimized, and we take the t-dependence of $\mathbf{C}(t)$ to be implicit, so we simply use the notation $\mathbf{C}$.

necessary.

7.6 An ADMM solver for the QCQP subproblem

We now derive a new, efficient solver for Eq. (7.31) using the alternating direction method of multipliers (ADMM). ADMM is a general method for constructing optimization algorithms for problems of the form

$$\begin{aligned} \underset{\mathbf{x},\mathbf{z}}{\text{minimize}} \quad & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{z} = \mathbf{c} \end{aligned} \tag{7.32}$$

where $f(\mathbf{x})$ and $g(\mathbf{z})$ are convex, but not necessarily smooth, functions. See Boyd *et al.*[325] for a comprehensive review. We take $f(\mathbf{x})$ to be the original objective function from Eq. (7.31),

$$f(\mathbf{x}) = \frac{1}{2}||\mathbf{x} - \mathbf{b}||_2^2 + ||\mathbf{D}(\mathbf{w})\mathbf{x}||_1, \tag{7.33}$$

where $\mathbf{D}(\mathbf{w})$ is matrix with the vector $\mathbf{w}$ along the diagonal, and $g(\mathbf{z})$ to encode the constraint,

$$g(\mathbf{z}) = \begin{cases} 0 & \text{if } \mathbf{z}^T\mathbf{\Sigma}\mathbf{z} \leq 1 \\ \infty & \text{otherwise,} \end{cases} \tag{7.34}$$

where $\mathbf{A} = \mathbf{B} = \mathbf{I}_n$, and $\mathbf{c} = 0$. The ADMM algorithm, in so-called scaled form, consists of the following iterations.

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} \left(f(\mathbf{x}) + \frac{\varrho}{2}||\mathbf{x} - \mathbf{z}^{(k)} + \mathbf{u}^{(k)}||_2^2 \right) \tag{7.35}$$

$$\mathbf{z}^{(k+1)} = \arg\min_{\mathbf{z}} \left(g(\mathbf{z}) + \frac{\varrho}{2}||\mathbf{x}^{(k+1)} - \mathbf{z}^{(k)} + \mathbf{u}^{(k)}||_2^2 \right) \tag{7.36}$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{x}^{(k+1)} - \mathbf{z}^{(k+1)},$$

where ϱ is a scalar that acts like a step size parameter, and can be adjusted over the course of the optimization to maintain stability.

By splitting the objective function into two parts, f and g, the algorithm can alternate taking steps that minimize over the variables $\mathbf{x}$ and $\mathbf{z}$ separately, with the $\mathbf{u}$ variable serving to pull these variables towards each other and enforce the constraint that $\mathbf{x} = \mathbf{z}$ at convergence.

The advantage of this formulation is that, as we now show, both the $\mathbf{x}$ and the $\mathbf{z}$ optimization steps can be performed very efficiently.

7.6.1 ADMM x update

The $\mathbf{x}$ optimization, Eq. (7.35), can be rewritten as

$$\arg\min_{\mathbf{x}} \frac{1}{2}||\mathbf{x} - \mathbf{b}||_2^2 + ||\mathbf{D}(\mathbf{w})\mathbf{x}||_1 + \frac{\varrho}{2}||\mathbf{x} - \mathbf{v}||_2^2, \tag{7.37}$$

where $\mathbf{v} = \mathbf{z}^{(k)} - \mathbf{u}^{(k)}$. This function is component-wise separable over the elements of $\mathbf{x}$, $f(\mathbf{x}) = \sum_i f_i(x_i)$. The minimization, Eq. (7.37), can thus be carried out as n separate scalar minimizations,

$$\arg\min_{x_i} \frac{1}{2}(x_i - b_i)^2 + w_i|x_i| + \frac{\varrho}{2}(x_i - v_i). \tag{7.38}$$

Although this objective function is not differentiable, it is a simple application of subdifferential calculus to compute a simple closed-form expression for the minimizer. The explicit solution is

$$x_i = \frac{1}{\varrho + 1} S_{w_i}(b_i + \varrho v_i), \tag{7.39}$$

where S, the soft-thresholding function, is defined as

$$S_\kappa(a) = \begin{cases} a - \kappa \text{ if } a > \kappa \\ 0 \text{ if } |a| \leq \kappa \\ a + \kappa \text{ if } a < -\kappa. \end{cases} \tag{7.40}$$

This simple form and component-wise separability means that the ADMM $\mathbf{x}$ update can be computed extremely rapidly.

7.6.2 ADMM z update

Because $g(\mathbf{z})$ is a hard boundary function, the $\mathbf{z}$ update, Eq. (7.36), can be interpreted as the projection of a point $\mathbf{a} = \mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)}$ onto the constraint set, $\{\mathbf{z} : \mathbf{z}^T \boldsymbol{\Sigma} \mathbf{z} \leq 1\}$, a hyper-ellipsoid. The problem can be rewritten as

$$\mathbf{z}^* = \begin{array}{ll} \underset{\mathbf{z}}{\arg\min} & ||\mathbf{z} - \mathbf{a}||^2 \\ \text{subject to} & \mathbf{z}^T \boldsymbol{\Sigma} \mathbf{z} \leq 1. \end{array} \tag{7.41}$$

For the nontrivial case in which the point $\mathbf{a}$ lies outside the ellipsoid, $\mathbf{a}^T \boldsymbol{\Sigma} \mathbf{a} > 1$, the solution, $\mathbf{z}^*$, is on the border of the ellipsoid, $\mathbf{z}^{*T} \boldsymbol{\Sigma} \mathbf{z}^* = 1$. By precomputing the eigendecomposition of $\boldsymbol{\Sigma}$, this can be solved efficiently using Kiseliov's method which is detailed in Section 7.B.[326]

An open source implementation of the estimator is available in the MSMBuilder software package at `http://msmbuilder.org`.

7.6.3 Further orthogonal reaction coordinates

Like tICA, our algorithm is not restricted to finding a single reaction coordinate, but can also identify sparse approximations to the other long-timescale eigenfunctions, $\psi_3, \ldots, \psi_k$. Unlike in the tICA method, in which the full set of solutions can be computed simultaneously with a single call to a standard generalized eigensolver, each sparse reaction coordinate must be estimated with a separate calculation.

As with most iterative sparse principal components analysis methods, we obtain the remaining generalized eigenvectors by subtracting the influence of the solution from the matrix $\mathbf{C}$, and then restarting optimization using the deflated matrix. The tradeoffs between methods for this deflation step have been discussed by Mackey.[327] Based on the recommendations therein, we have adopted Mackey's Schur complement deflation strategy.

7.6.4 Hyperparameter selection and implementation notes

In order to use sparse tICA in practice, a value of the regularization strength, ρ, must be chosen. When $\rho = 0$, sparse tICA reduces to the standard tICA algorithm, and larger values of ρ will increase the sparsity. We recommend two possible methods of choosing ρ. First, with cross-validation, the modeller may split the data set into two or more portions, optimize the reaction coordinate at different values of ρ using one fraction of the data set, and check the value of the objective function on the left-out data set. For tICA and Markov state models, this approach was discussed McGibbon and Pande.[214] It is equally applicable to sparse tICA.

Alternatively, when the primary goal is to generate physically interpretable reaction coordinates, the modeller may choose the value of ρ to bring the number of non-zero loadings down to a pre-specified number that is amenable to interpretation. When employing this strategy, we recommend that modellers watch the value of the pseudoeiegenvalue (Rayleigh quotient), $\hat{\lambda} = \mathbf{x}^T\mathbf{C}\mathbf{x}/\mathbf{x}^T\mathbf{\Sigma}\mathbf{x}$. It should decrease slightly with increasing ρ, but dramatic drops in $\hat{\lambda}$ may indicate over-regularization.

The procedure also depends on $\epsilon > 0$, which controls the shape of the regularizer. Lower values of ϵ lead to a tighter approximation of the ℓ_0 norm, but can also lead to numerical instabilities as the derivative of the regularizer near zero goes to infinity, as can be seen in Fig. 7.3. Empirically, we have found that $\epsilon = 10^{-6}$ provides a suitable balance. Finally, note that the scalar ϱ is required during the optimization as well. This parameter affects only the convergence rate of the solver, as opposed to the final solution, and can be dynamically adjusted over the course of the optimization using standard methods described by Boyd *et al.* [325]

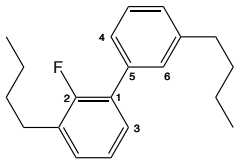


Figure 7.4: A 2-fluorobiphenyl derivative simulated in this work. An overcomplete set of 510 internal coordinates were measured from each frame, which included four dihedral angles (described by carbons 2-1-5-4, 2-1-5-6, 3-1-5-4, and 3-1-5-6) that described the inter-ring torsion angle.

7.7 Examples

7.7.1 Torsional reaction coordinate

We demonstrate our approach on molecular dynamics simulations of a simple 2-fluorobiphenyl derivative, shown in Fig. 7.4. This system is interesting as a toy example because chemical intuition suggests that the rotation of the rings with respect to one another will be hindered. We anticipate the dynamics of the aliphatic tails to be faster and uncoupled to the reaction coordinate. Can our algorithm recover this sparse reaction coordinate?

After parameterization with the generalized Amber forcefield,[328] we simulated the system in the gas phase for 250 ns at 290 K using a Langevin integrator with a friction coefficient of 1 ps^{-1} and timestep of 2 fs using OpenMM 6.3.[186] Snapshots from the simulation were saved every 20 ps. From each simulation snapshot, we recorded the values of an overcomplete set of 510 internal coordinates, which included the distances between all unique pairs of carbon atoms, measured in nanometers, the angles between pairs of bonded atoms, in radians, and the sine and cosine of the dihedral angles between all quartets of bonded atoms. After mean subtraction, these coordinates form our basis functions, χ_i, for tICA and our sparse variant. Despite our chemical intuition, from an algorithmic perspective, finding the reaction coordinate for this system is something like finding a needle in a haystack.

In Fig. 7.5, we show the resulting dominant eigenvector as estimated by tICA and our new approach using increasing values of the regularizer, ρ. The pseudoeigenvalue, $\hat{\lambda}$, is the Rayleigh quotient of the collective variable, related to its timescale by $\hat{\tau} =$

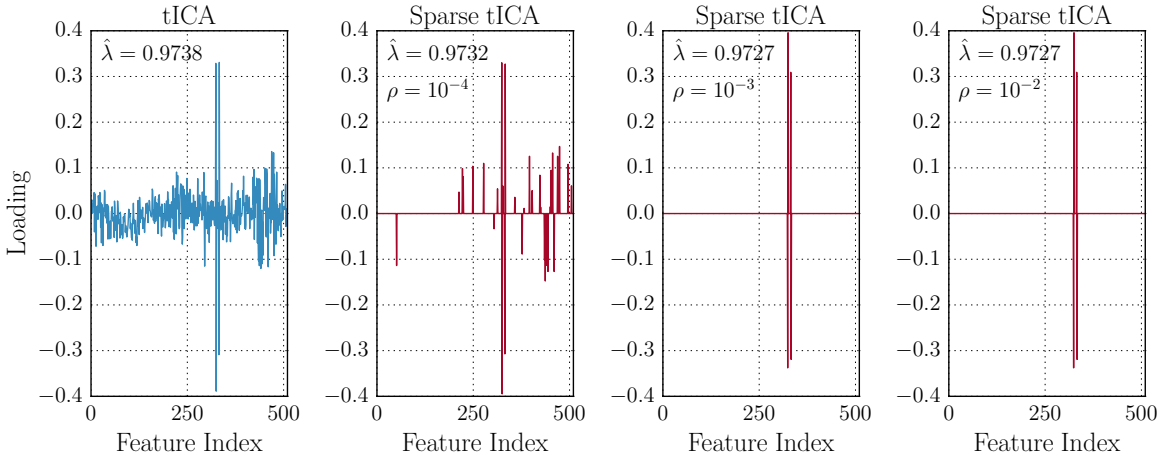


Figure 7.5: tICA and sparse tICA results for simulations of the 2-fluorobiphenyl derivative shown in Fig. 7.4 with increasing values of the regularization strength, ρ. The unregularized tICA results report a reaction coordinate which is a dense linear combination of all 510 input features. In contrast, with increasing values of the regularization strength, ρ, the sparse tICA algorithm filters out this noise to identify only the sines of the four dihedral angles that collectively characterize the inter-ring torsional reaction coordinate, with only a minor decrease in the psuedoeigenvalue, $\hat{\lambda}$.

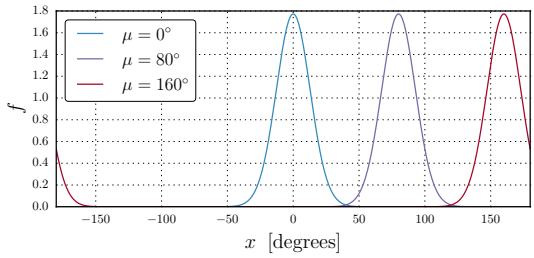


Figure 7.6: Probability density function of the von Mises distribution with $\kappa = 20$ and different values of the location parameter, μ. For an angle x, the function is given by $f(x;\kappa) = \frac{e^{\kappa \cos(x-\mu)}}{2\pi I_0(\kappa)}$, where $I_0(\kappa)$ is the modified Bessel function of order 0. The function has a full-width at half maximum of approximately 30°.

$-1/\ln\lambda_i$. In standard tICA, this value is maximized exclusively, whereas in sparse tICA, this objective is balanced against a penalty that favors zero coefficients. We see in Fig. 7.5 that the tICA solution, as expected, returns a collective variable that is a linear combination of all 510 input coordinates, with a nonzero component on each of the coordinates and significant noise.

In constrast, our sparse tICA algorithm suppresses this noise and identifies sparse collective variables that are formed from linear combinations of only a small number of the input degrees of freedom. This sparsity increases with larger values of the regularization strength, ρ, and only leads to a modest decrease in the approximated timescale associated with the coordinate. For $\rho = 10^{-3}$ and $\rho = 10^{-2}$, only four input coordinates survive. Inspection of these coordinates shows that they are the sines of the four dihedral angles that cross between the rings (atoms 2-1-5-4, 2-1-5-6, 3-1-5-4, and 3-1-5-6 in Fig. 7.4). We interpret these results to show that sparse tICA has, without any prior chemical knowledge, filtered through a collection of structural order parameters, many of which are irrelevant in describing the slowest dynamical process of this molecule, and located the subset which can approximate the natural reaction coordinate.

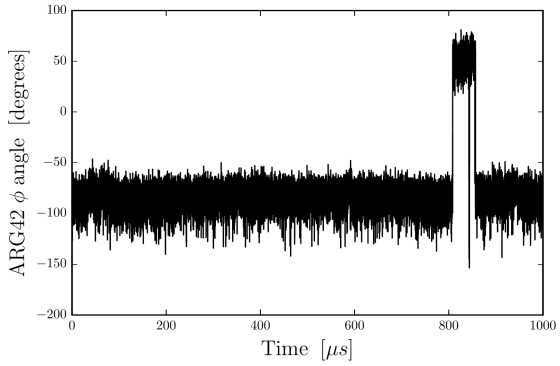


Figure 7.7: The ARG 42 ϕ angle over the course of the 1 ms simulation of native state dynamics of BPTI performed by D.E. Shaw Research.[91]. Our sparse tICA identifies this as the reaction coordinate for a process that involves the opening and hydration of the protein's core.

7.7.2 Bovine pancreatic trypsin inhibitor (BPTI)

In this section, we apply the sparse tICA method to analyze the native state dynamics of the bovine pancreatic trypsin inhibitor (BPTI), a small 58-residue globular protein that has been extensively investigated by experimental and computational methods. We reanalyzed the one millisecond all-atom molecular dynamics simulation performed by D.E. Shaw Research at 300K with explicit solvent.[91] With its rigid disulfide bonds, the system remains folded over the course of the simulation, but samples a number of near-native states.

For each frame in the trajectory data set, sampled every 25 ns, we computed the value of an extensive set of 2880 structural order parameters from the backbone and side chain dihedral angles. For each of the 57 protein backbone ϕ and ψ torsion angles, as well as the 46 χ_1 torsion angles, we computed 18 order parameters by evaluating the probability density function of the von Mises distribution at different values of its location parameter, evenly spaced around the unit circle at 20° increments. A subset of these functions is shown in Fig. 7.6. These functions act like softened indicator functions that wrap appropriately on $(-180°, 180°)$. We hypothesized that this would be a suitable basis in which to expand the reaction coordinates for BPTI, because it is well suited for expressing a function representing flux between two regions on

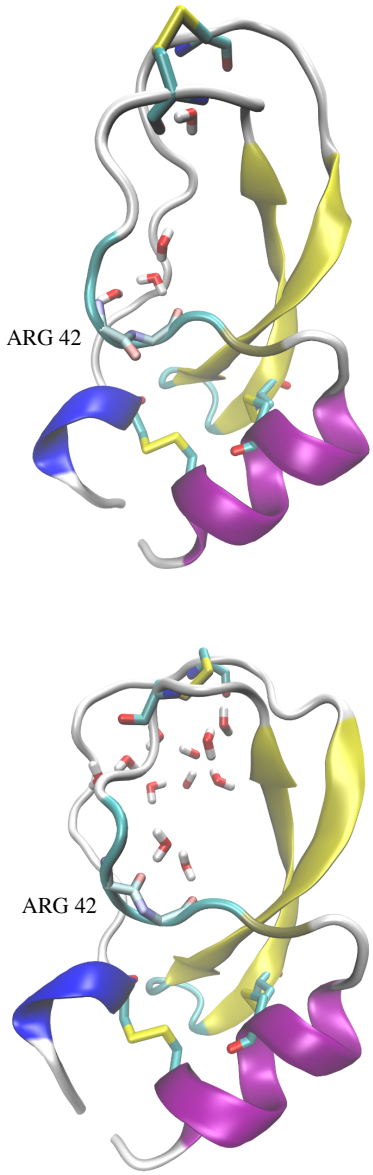


Figure 7.8: The near-native (above) and ARG42-flipped (below) conformations of BPTI from the simulation trajectory. The first panel shows the near-native conformation sampled by the majority of the simulation with a ARG42 ϕ angle between $-50°$ and $-150°$, with the expected four crystallographic waters. Nearly 800 μs into the simulation, the trajectory samples an alternate state in which the protein's core opens and hydrates and the crystallographic waters can exchange with the bulk. In this state, the ARG42 ϕ angle has flipped, putting its oxygen pointing into the now hydrated core.

a Ramachandran plot. Each structural order parameters in our input basis set can thus be interpreted as roughly indicating whether a particular torsion angle is within one of 18 different $\sim 30^\circ$ windows.

Using these input features, we fit a sparse tICA model with $\rho = 0.005$ and observed a surprising result. The first solution depends only on the ϕ dihedral angle of ARG 42. The timeseries of this angle over the course of the simulation is shown in Fig. 7.7, and we see that this degree of freedom makes a single dramatic flip over the course of the simulation. When we inspected conformations from this flipped state, we observed that the protein's core had opened and hydrated. While this large-scale structural change is obvious from visual inspection of the trajectory, fact that the ϕ angle of ARG 42 acts as a switch between these two states was unexpected. While many other degrees of freedom also change between these two states, such as the orientation of the upper disulfide linkage (visible in Fig. 7.8) these degrees of freedom also fluctuate within the near-native state. It is the rare inward flip of ARG 42 which we observe to draw in solvent to hydrate the protein's small core.

7.8 Conclusions

In this work, we have introduced a defintion of the natural reaction coordinate as a function that satisfies a set of simple mathematical properties: that it (a) is a dimensionality reduction that (b) is defined only by the system's dynamics, and that (c) is the maximally predictive projection about the future evolution of the system. The definition is particularly apt for soft-matter systems in which there may be more than two metastable states, or for systems in which identifying and structurally defining the metastable states is challenging. For any time-homogeneous, reversible, ergodic Markov chain such as thermostatted molecular dynamics, these properties are uniquely satisfied by a dominant eigenfunction of the transfer operator associated with the dynamics, ψ_2. This eigenfunction is also the most slowly decorrelating collective variable in the system.

We developed a practical new estimator that builds upon the tICA method for estimating these eigenfunctions. Like tICA, this estimator is used to post-process molecular dynamics trajectories. Unlike the variational tICA method which constructs an approximation to these eigenfunctions using a linear combination of structural order parameters in which all of the coefficients are generally non-zero, our estimator finds sparse solutions. It is thus able both to filter through inevitable statistical noise and identify simple, interpretable strutural order parameters that approximate these natural reaction coordinates, without any prior knowledge of the system.

Application of this method to molecular dynamics simulations of a 2-fluorobiphenyl derivative and BPTI show that the approach can identify reaction coordinates for the slow dynamical processes in these data sets that are readily interpretable. In BPTI, we see that opening and hydration of the protein core is controlled by a flip of a single backbone ϕ angle at ARG 42.

We anticipate that this method will be useful for the analysis of today's large molecular dynamics data sets. An implementation of this estimator is available in the MSMBuilder software package at `http://msmbuilder.org/` under the GNU Lesser General Public License.

Acknowledgments

The authors thank Brooke Husic and Thomas J. Lane for helpful discussions made during the preparation of this chapter, Ariana Peck and Carlos X. Hernández for invaluable copy editing, and the National Institutes of Health under Nos NIH R01-GM62868 for funding. We graciously acknowledge D.E. Shaw Research for providing access to the BPTI trajectory data set.

7.A Covariance matrix estimation

In this section, we discuss some issues related to the estimation of the covariance matrix, $\mathbf{\Sigma}$, from timeseries data such as molecular dynamics simulations. If we consider a single trajectory of length N and collect the results of the evaluation of each of the

zero-meaned m basis functions on each of the T snapshots into a matrix, $\chi \in \mathbb{R}^{m \times N}$, the standard estimator for $\mathbf{\Sigma}$ would be the sample covariance matrix,

$$\mathbf{S} = \frac{1}{N-1}\chi\chi^T. \tag{7.42}$$

Covariance matrix estimation is a ubiquitous problem common to many fields of science and engineering, and a number of issues with this estimator are known. In particular, results from random matrix theory suggest that the eigenspectrum of the estimated covariance matrix, $\hat{S}$, is over-dispersed with respect to the true value. That is, its large eigenvalues are too large, and its small eigenvalues are too small. For a fixed number of basis functions, m, the sample eigenvalues can be shown to converge to the true eigenvalues as N goes to infinity,[329] but when m is allowed to grow with N, keeping m/N fixed, results such as the Marčkenko-Pastur law suggest that the sample eigenvalues are not effective estimators, and do not converge to the true eigenvalues.[330]

In the context of a weight matrix in a generalized eigenvalue problem, misestimation of the small eigenvalues of S is particularly problematic. The generalized eigenvalue problem requires that S be positive-definite — in the extreme case when $\hat{S}$ is rank-deficient, the maximum value of Eq. (7.26) is not defined and we get the matrix equivalent of a division by zero.

The most popular class of stabilized covariance matrix estimators are called shrinkage estimators, and take the form

$$\hat{\mathbf{\Sigma}} = (1-\gamma)\mathbf{S} + \gamma(\mathrm{Tr}(\mathbf{S})/m)\mathbf{I}, \tag{7.43}$$

for some positive constant γ. The interpretation of this expression is that the shrunk covariance matrix is a convex combination of two estimators, the (low bias, but high variance) sample covariance matrix, and the (high bias, but low variance) estimator that assumes all basis functions have identical variances and zero covariance. An estimator of this form was first popularized by Ledoit and Wolf in the context of Markowitz portfolio selection.[331–333] Other shrinkage targets are possible beyond the scaled identity; we refer the reader to the excellent review by Schäfer and Strimmer.[334]

The key insight of Ledoit and Wolf is that, under a Frobenius norm objective on the difference between the shrunk covariance matrix and the true covariance matrix, the asymptotically optimal value of the shrinkage constant, γ, can be estimated directly from $\mathbf{S}$, without knowing the true covariance matrix. Thus, no extra tunable parameters need to be added to the algorithm, which is important for usability.

Further improvements to the Ledoit-Wolf (LW) estimator were made by Chen, Wiesel, and Hero III.[335] First, using the Rao-Blackwell theorem,[336] they produced a more accurate Rao-Blackwellized Ledoit-Wolf (RBLW) estimator for the optimal shrinkage constant that dominates the LW estimator. In addition, unlike the LW estimator, the RBLW estimator can be computed even more efficiently and essentially requires no significant computational work beyond the calculation of the sample covariance matrix, $\mathbf{S}$. The expression for the RBLW-optimal shrinkage constant, γ, is

$$\gamma = \min(\alpha, \beta/U), \tag{7.44}$$

where α, β, and U are given by

$$\alpha = \frac{N-2}{N(N+2)} \tag{7.45}$$

$$\beta = \frac{(m+1)N-2}{N(N+2)} \tag{7.46}$$

$$U = \frac{m\,\mathrm{Tr}(\mathbf{S}^2)}{\mathrm{Tr}^2(\mathbf{S})} - 1. \tag{7.47}$$

We recommend this RBLW estimator for $\mathbf{\Sigma}$ for use with both tICA and sparse tICA.

7.B Projection of point onto an ellipsoid

Here we discuss our method for projecting a point in $\mathbb{R}^N$ onto an ellipsoid, following Kiseliov.[326] Given a point $\mathbf{a}$ outside the ellipsoid and a positive definite matrix $\mathbf{\Sigma}$, the

problem can be written as:

$$\mathbf{z}^* = \begin{array}{ll} \underset{\mathbf{z}}{\arg\min} & ||\mathbf{z} - \mathbf{a}||^2 \\ \text{subject to} & \mathbf{z}^T \mathbf{\Sigma} \mathbf{z} \leq 1. \end{array} \tag{7.48}$$

Because, for our purposes, it will be necessary to solve the problem many times for different values of $\mathbf{a}$ with the same value of $\mathbf{\Sigma}$, it will be advantageous to consider any possible pre-processing of $\mathbf{\Sigma}$ that will speed up the calculation for each $\mathbf{a}$.

For the nontrivial case in which the point $\mathbf{a}$ lies outside the ellipsoid, the solution is on the border of the ellipsoid, $\mathbf{z}^{*T} \mathbf{\Sigma} \mathbf{z}^* = 1$, so we address only the equality. First, consider the Lagrangian, L,

$$L = ||\mathbf{z} - \mathbf{a}||^2 + \mu(\mathbf{z}^T \mathbf{\Sigma} \mathbf{z} - 1). \tag{7.49}$$

The solution to Eq. (7.48) satisfies the condition $\nabla L = 0$, yielding

$$\mathbf{z}^* = (\mathbf{I}_n + \mu^* \mathbf{\Sigma})^{-1} \mathbf{a}. \tag{7.50}$$

The value of the Lagrange multiplier at the solution, μ^*, must be determined to ensure that the constraint is satisfied. This requires solving the scalar equation $G(\mu) = 0$, where $G(\mu)$ is defined as

$$G(\mu) = \mathbf{z}^*(\mu)^T \mathbf{\Sigma} \mathbf{z}^*(\mu) - 1, \tag{7.51}$$

$$\mathbf{z}^*(\mu) = (\mathbf{I}_n + \mu \mathbf{\Sigma})^{-1} \mathbf{a}. \tag{7.52}$$

We solve for the root of G using Newton's method, which requires computing G and $G' = dG/d\mu$. Assuming that the eigendecomposition of $\mathbf{\Sigma}$ has been precomputed, $\mathbf{\Sigma} = \mathbf{V}\mathbf{D}(\mathbf{w})\mathbf{V}^T$, applying the Woodbury matrix identity shows that G and G' can be computed in linear time, without explicitly inverting any matrices or solving any

linear systems, as Eq. (7.52) suggests might be necessary,

$$\mathbf{z}^*(\mu) = (\mathbf{I}_n + \mu\boldsymbol{\Sigma})^{-1}a \tag{7.53}$$
$$= (\mathbf{I}_n + \mu\mathbf{V}\mathbf{D}(\mathbf{w})\mathbf{V}^T)^{-1}\mathbf{a} \tag{7.54}$$
$$= (\mathbf{V}(\mathbf{I}_n + \mu\mathbf{D}(\mathbf{w}))\mathbf{V}^T)^{-1}\mathbf{a} \tag{7.55}$$
$$= \mathbf{V}\mathbf{D}(\mathbf{e})\mathbf{V}^T a, \tag{7.56}$$

where $e_i = (\mu w_i + 1)^{-1}$. Then, expanding $G(\mu)$, we have

$$G(\mu) = \mathbf{z}^*(\mu)^T\boldsymbol{\Sigma}\mathbf{z}^*(\mu) - 1 \tag{7.57}$$
$$= (\mathbf{V}\mathbf{D}(\mathbf{e})\mathbf{V}^T\mathbf{a})^T\mathbf{V}\mathbf{D}(\mathbf{w})\mathbf{V}^T\mathbf{V}\mathbf{D}(\mathbf{e})\mathbf{V}^T\mathbf{a} - 1 \tag{7.58}$$
$$= \mathbf{a}^T\mathbf{V}\mathbf{D}(\mathbf{f})\mathbf{V}^T\mathbf{a} - 1, \tag{7.59}$$

where $f_i = w_i e_i^2 = w_i/(\mu w_i + 1)^2$. The derivative required for Newton's method, $dG/d\mu$, is then very simple to calculate.

This algorithm is summarized in Algorithm 3. The quadratic convergence of Newton's method and low per-step work makes this preferable to alternatives such as the Lin-Han method.[337]

Algorithm 3 Projection of a point onto an ellipsoid

Require: $\mathbf{a} \in \mathbb{R}^n, \boldsymbol{\Sigma} \in \mathbb{S}^n_{++}$
  $\mathbf{w}, \mathbf{V} \leftarrow \text{eigs}(\boldsymbol{\Sigma})$ ▷ Compute eigenvalues and eigenvectors
  $\mathbf{c} \leftarrow \mathbf{V}^T\mathbf{a}$
  if $\mathbf{c}^T\mathbf{D}(\mathbf{w})\mathbf{c} \leq 1$ **then**
    return a ▷ Trivial if **a** is inside the set
  else
    $\mu^{(0)} \leftarrow 1$
    while not converged **do** ▷ Newton's method
      $G^{(k)} \leftarrow -1 + \sum_{i=1}^n c_i^2 w_i/(\mu^{(k)} w_i + 1)^2$
      $G'^{(k)} \leftarrow -2\sum_{i=1}^n c_i^2 w_i^2/(\mu^{(k)} w_i + 1)^3$
      $\mu^{(k+1)} \leftarrow \mu^{(k)} - G^{(k)}/G'^{(k)}$
    end while
    $e_i \leftarrow (\mu^{(k)} w_i + 1)^{-1}$
    return $\mathbf{V}\mathbf{D}(\mathbf{e})\mathbf{c}$
  end if

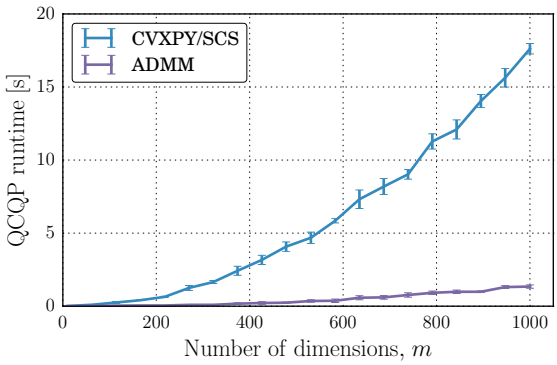


Figure 7.9: Comparison of the runtime of our specialized QCQP solver and a generic solver using CVXPY and SCS.[338,339] We observe a speedup of approximately one order of magnitude. Efficient warm-starting of the QCQP in Algorithm 2 yields further improvements in runtime. Error bars indicate standard deviations over 5 replicates.

7.C Runtime performance

In addition to our ADMM-based solver, we implemented the sparse tICA algorithm using CVXPY and the off-the-shelf SCS solver to solve the QCQP.[338,339] In Fig. 7.9, we compare the runtime of these two approaches. For this comparison, we randomly generated the matrix $\mathbf{\Sigma}$ from a Wishart distribution with m degrees of freedom and an identity scale matrix, and initialized the ADMM solver from a vector, $\mathbf{x}$, with elements drawn from the standard normal distribution. The error bars indicate standard deviations over 5 replicates. The timings were performed on a Mid 2014 Apple Macbook Pro laptop.

We see generally that our solver is roughly an order of magnitude faster on the QCQP than CVXPY with SCS. Our sparse tICA implementation, however, is also able to efficiently warm-start, because the vectors $\mathbf{w}$ and $\mathbf{b}$ also converge during the outer iteration of Algorithm 2. Because of this, we find that when we substitute in the off-the-shelf solver to Algorithm 2, the speedup achieved by our ADMM approach is even more substantial. For example, while converging the first sparse tICA solution with $m = 500$ using our ADMM implementation takes on the order of 0.1 seconds, the same optimization takes approximately 7 minutes using the off-the-shelf solver.

Chapter 8

MDTraj: a modern, open library for the analysis of molecular dynamics trajectories

As molecular dynamics simulations continue to evolve as powerful computational tools for studying complex biomolecular systems, the necessity of flexible and easy-to-use software tools for the analysis of these simulations is growing. We have developed MDTraj, a modern, lightweight, and fast software package for analyzing molecular dynamics simulations. MDTraj reads and writes trajectory data in a wide variety of commonly used formats. It provides a large number of trajectory analysis capabilities including minimal root-mean-square-deviation calculations, secondary structure assignment, and the extraction of common order parameters. The package has a strong focus on interoperability with the wider scientific Python ecosystem, bridging the gap between molecular dynamics data and the rapidly-growing collection of industry-standard statistical analysis and visualization tools in Python. MDTraj is a powerful and user-friendly software package that simplifies the analysis of molecular dynamics data and connects these datasets with the modern interactive data science software ecosystem in Python.

This chapter is adapted with permission from McGibbon, R. T., Beauchamp, K. A., Harrigan, M. P., Klein, C., Swails, J. M., Hernández, C. X., Schwantes, C. R., Wang, L.-P., Lane, T. J., and Pande, V. S., MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophys. J.*, **2015**, *109(8)*, pp 1528–1532.[165]

8.1 Introduction

Molecular dynamics (MD) simulations yield a great deal of information about the structure, dynamics, and function of biological macromolecules by modeling the physical interactions between their atomic constituents. Modern MD simulations, often using distributed computing, graphics processing unit acceleration, or specialized hardware can generate large datasets containing hundreds of gigabytes or more of trajectory data tracking the positions of a system's atoms over time.[40] In order to use these vast and information-rich datasets to understand biomolecular systems and generate scientific insight, further computation, analysis, and visualization are required.[19]

Within the last decade, the Python language has become a major hub for scientific computing. It features a wealth of high-quality open source packages, including those for interactive computing,[340] machine learning,[341] and visualization. [342] This environment is ideal for both rapid development and high performance, as computational kernels can be implemented in C, C++, and FORTRAN but made available within a more user-friendly interactive environment.

In the MD community, the benefits of integration with such industry standard tools have not yet been fully realized because of a tradition of custom file formats and command-line analysis, although significant progress has been made.[45,343–346] In order to address this need, we have developed MDTraj, a modern, open, and lightweight Python library for analysis and manipulation of MD trajectories. The project has the following goals:

1. To serve as a *bridge* between MD data and the modern statistical analysis and scientific visualization software ecosystem in Python.

Package	File format
Many packages	`pdb`, `xyz`, `dcd`
Amber	`prmtop`, `crd`, `netcdf`, `binpos`, `restrt`
Gromacs	`gro`, `xtc`, `trr`
Desmond	`dtr`, `stk`
CHARMM	`psf`
LAMMPS	`lammpstrj`
TINKER	`arc`
HOOMD-Blue	`xml`
OpenMM	`xml`
TRIPOS	`mol2`
MDTraj	`hdf5`

Figure 8.1: List of MDTraj-supported file formats

2. To support a wide range of MD data formats and computations.

3. To run rapidly on modern hardware with efficient memory utilization, enabling the interactive analysis of large datasets.

8.2 Capabilities and implementation

MDTraj is widely interoperable and extremely easy to use. First and foremost, MDTraj can load trajectory and/or topology data from the formats used by a broad range of MD packages, including Amber,[48] Gromacs,[45] Desmond,[47] CHARMM,[347] NAMD,[46] TINKER,[348] LAMMPS,[349] OpenMM,[186] and HOOMD-Blue;[350] see Section 8.2 for a full list of supported file formats. This wide support enables consistent interfaces and reproducible analyses regardless of users' preferred MD simulation packages.

From its inception, MDTraj has been designed to work in concert with other packages for analysis and visualization. No single toolkit can provide all possible ways to analyze molecular simulations, especially given the rapid pace of development in statistics and data science. Rather than attempting to provide all conceivable functionality in one toolkit, MDTraj leverages Python and NumPy to empower users to connect their MD data with the large and rapidly growing ecosystem of data science

tools available more broadly in the community.

MDTraj originated from the trajectory handling portions of MSMBuilder [75], where it now provides a stable base for handling trajectories, computing order parameters and projections, and providing distance metrics — such as minimal root-mean-squared deviation (RMSD) — that are necessary for clustering. Additionally, it is now used inside tools that analyze data from the Folding@home distributed computing architecture [351], a structure based virtual screening pipeline at Google Research, the PyEMMA Markov modeling package [134], the Ensembler and mBuild [352,353] modeling tools, and countless individual analysis scripts. MDTraj is part of the Omnia consortium (`http://omnia.md`) suite of tools, which will be described in a later paper.

Most data analyses for MD involve either (a) extracting a vector of order parameters of each simulation snapshot or (b) defining a distance metric between snapshots. MDTraj makes it very easy to rapidly extract these representations. It includes an extremely fast RMSD engine capable of operating near the machine floating point limit described by Haque, Beauchamp and Pande.[354] Functions for secondary-structure assignment,[355] solvent accessible surface area determination,[356] hydrogen bond identification, [357] residue-residue contact mapping, NMR scalar coupling constants, [358] nematic order parameters,[359] and the extraction of various internal degrees of freedom are similarly available. Where appropriate, these compute kernels are written in C or C++ and heavily optimized with vectorized instructions (SSE3 intrinsics) and multithreading (OpenMP). To enable interoperability, these data are returned to the user as multidimensional NumPy arrays, the standard numeric data storage format for the scientific Python ecosystem.

MDTraj also provides an atom selection language. Often, analysis functions are applied to a subset of atoms in the system. To generate arrays of these indices, the `topology` attribute and full Python grammar can be a powerful combination (Box 8.2, line 2). For users less familiar with Python or making the transition from other packages, a natural text-based selection syntax can be used as well (Box 8.2, line 3). These selection strings can be translated into standard Python syntax for pedagogical purposes, or directly executed.

Ease-of-use is a central and deliberate goal at each level of the the design and

```
In [1]: top = my_traj.topology
In [2]: [atom.index for atom in top.atoms if atom.residue.is_water and atom.name == 'O']
In [3]: top.select("water and name O")
```

Figure 8.2: Atom selection language. Lines 2 and 3 are equivalent, although the latter text-based syntax may be more natural for users.

implementation of MDTraj. This starts with installation. Using the `conda` package manager, users can get started in seconds using the shell command `conda install -c omnia mdtraj`, which downloads and installs pre-compiled binaries of MDTraj (and all of its dependencies) on either Microsoft Windows, Linux, or Mac OS-X.

The package has an extremely simple object model, which makes it very easy for new users to get started. Only a single class, `Trajectory`, needs to be mastered; it contains all relevant information about the MD trajectory, such as the atomic coordinates, unit cell dimensions, and simulation time. Loading files and performing analysis are generally done with functions (e.g. `mdtraj.load`, `mdtraj.compute_<name>`) as opposed to classes to provide a simple and intuitive user experience that minimizes the need to remember complex object workflows.

MDTraj is extensively documented in a consistent format. The package itself contains over 9000 lines of Python "docstrings" that describe each function and class. The website, `http://mdtraj.org`, has the full API documentation, but more importantly contains 14 complete, executable example notebooks demonstrating topics including hydrogen bond identification, Ramachandran plotting, and strategies for memory-limited computation on large datasets. These examples provide new users the patterns to get up and running with their own analyses immediately.

Furthermore, MDTraj includes a unique interactive WebGL-based 3D structure viewer for the IPython notebook adapted from *iview*,[360] shown in Fig. 8.3. Because it combines the analysis input code with results and plots into a single worksheet, the IPython notebook provides one of the most convenient user interfaces for interactive analysis. This convenience is further enhanced by MDTraj's `TrajectoryView` widget, which runs inside the IPython notebook and provides a high-quality and fully interactive 3D rendering of a trajectory. The viewer can save high-quality `png` images

```
import mdtraj as md
from mdtraj.html import TrajectorySliderView

traj = md.load("trajectory.pdb")
TrajectorySliderView(traj, secondaryStructure="ribbon")
```

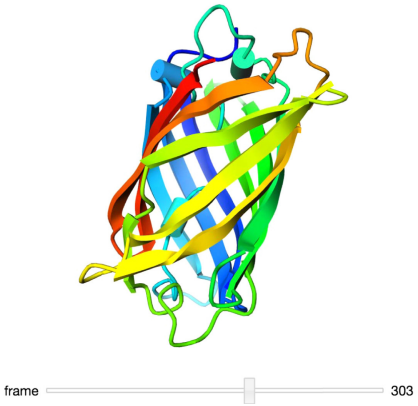


Figure 8.3: MDTraj's interactive WebGL-based protein and trajectory

or `STL` 3D models. MDTraj thus not only provides first-class scriptability but also high quality 3D visualization.

The development, engineering, and testing of MDTraj incorporate modern best practices for scientific computing.[361] The package contains over 1100 unit tests for individual components. These tests are continually run on each incremental contribution on both Windows and Linux, using multiple versions of Python and the required libraries. The project is hosted on GitHub, and development takes place fully openly and collaboratively. Users of MDTraj are often researchers who are interested in analyzing simulations in new ways, a task which involves not only MDTraj library functions but also writing new code. The simple coding style, open source licensing, GitHub pull request based development pattern, [362] and active culture of collaborative code review enable these researchers to rapidly prototype new methods and extend MDTraj. This has been borne out by the MDTraj community, which comprises members from numerous academic and industrial research groups across the world who have contributed to the project over the past two years.

```
import mdtraj as md
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

t = md.load("trajectory.pdb")
pairs = t.top.select_pairs("all", "all")
X = md.compute_distances(t, pairs)

pca = PCA(n_components=2)
Y = pca.fit_transform(X)

plt.hexbin(Y[:, 0], Y[:, 1], bins="log")
```

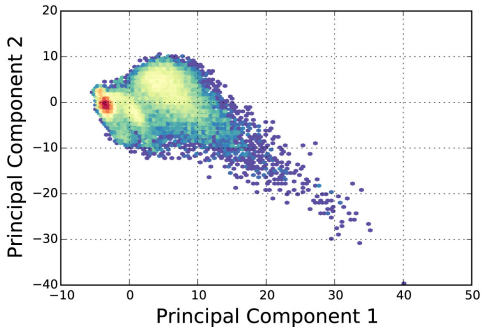


Figure 8.4: Demonstration of principal components analysis (PCA) with MDTraj, `scikit-learn` and `matplotlib`.

8.3 Results and discussion

The capabilities of MDTraj serve as a *bridge*, connecting MD data with statistics and graphics libraries developed for general data science audiences. A key advantage of this design, for users and developers, is access to a much wider range of state-of-the-art analysis capabilities characterized by large feature sets, extensive documentation and active user communities.

A demonstration of this integrative workflow is shown in Fig. 8.4, which combines MDTraj with the `scikit-learn` package for PCA and `matplotlib` for visualization, to determine high-variance collective motions in a protein system. While PCA is a widely used method that is included in a variety of MD analysis packages, the advantage of integrating with the wider data science community is immediately evident when moving on to more complex statistical analysis. For example, a variety of sparse and kernelized PCA-like methods have been recently introduced in the machine learning community,[363] and may be quite powerful for analyzing more complex

```
from IPython.parallel import Client
import numpy as np
import mdtraj as md
import matplotlib.pyplot as plt

c = Client()

traj = md.iterload("trajectory.pdb", chunk=10)
results = c[:].map(md.shrake_rupley, traj)
sasa = results.get()

plt.plot(np.sum(np.vstack(sasa), 1))
```

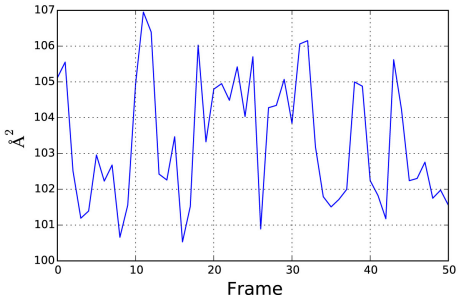


Figure 8.5: Demonstration of solvent-accessible surface area calculation done in parallel with MDTraj and IPython.

protein systems. Because of its open and interoperable design, these cutting-edge statistical tools are readily available to MD researchers with MDTraj, without duplication of developer efforts and independent of the particular MD software used to perform the simulations.

We generally find that file I/O and main memory are more limiting than raw CPU performance for MD analysis. For this reason, simple multi-node parallelization, even over relatively slow interconnects, can often be extremely useful for accelerating calculations. As an example, Fig. 8.5 shows a demonstration of the use of MDTraj with the `IPython.parallel` toolkit to parallelize the calculation of the solvent accessible surface area of a trajectory over the individual snapshots of the trajectory. The code requires separately initializing an array of IPython engine processes on which the calculation is executed. These can be distributed over many nodes on a cluster or in the cloud and linked together by MPI or SSH. Because many simulation datasets contain many separate MD trajectories saved in separate files, a similar pattern can also be used to process individual files in parallel.

8.4 Conclusions

Within the field of trajectory analysis tools, MDTraj stands out due to its ease of use, flexibility and Python-centric design, largely thanks to its organization around the intuitive `Trajectory` object in which data are stored as NumPy arrays. This design significantly enhances extensibility and gives users a great deal of latitude for freely accessing and manipulating the data according to the needs of their research. MDTraj speeds up analysis tasks by implementing computationally intensive operations (such as RMSD) using optimized low-level kernels written in C / C++. Furthermore, MDTraj can read and write a very wide range of trajectory file formats, ensuring interoperability across most molecular dynamics software packages.

Acknowledgments

The authors acknowledge funding from the National Institutes of Health (R01-GM62868, P30-CA008748) and National Science Foundation (MCB-0954714). We are also grateful to the full team of MDTraj contributors: Patrick Riley, Teng Lin, Tim Moore, Ravira Manathan, Joshua Adelman, Chaya Stern, Gert Kiss, Muneeb Sultan, Yutong Zhao, Andrea Zonca, Ondrej Marsalek, Thomas Peulen, Anton Goloborodko, Alexander Götz; as well as participants on the MDTraj discussion forum and issue tracker.

Bibliography

[1] R. O. Dror, R. M. Dirks, J. Grossman, H. Xu, and D. E. Shaw. Biomolecular simulation: a computational microscope for molecular biology. *Annu. Rev. Biophys.*, 41:429–452, 2012.

[2] G. Careri, P. Fasella, E. Gratton, and W. Jencks. Statistical time events in enzymes: a physical assessment. *Crit. Rev. Biochem. Mol. Biol.*, 3(2):141–164, 1975.

[3] C. M. Dobson. Experimental investigation of protein folding and misfolding. *Methods*, 34(1):4–14, 2004.

[4] R. Ishima and D. A. Torchia. Protein dynamics from NMR. *Nat. Struct. Biol.*, 7(9):740–743, 2000.

[5] J. Drenth. *Principles of protein X-ray crystallography.* Springer–Verlag, 2007.

[6] B. Schuler and W. A. Eaton. Protein folding studied by single-molecule FRET. *Curr. Opin. Struct. Biol.*, 18(1):16–26, 2008.

[7] H. S. Chung, K. McHale, J. M. Louis, and W. A. Eaton. Single-molecule fluorescence experiments determine protein folding transition path times. *Science*, 335(6071):981–984, 2012.

[8] H. B. Buergi and J. D. Dunitz. From crystal statics to chemical dynamics. *Acc. Chem. Res.*, 16(5):153–161, 1983.

[9] A. Mittermaier and L. E. Kay. New tools provide new insights in NMR studies of protein dynamics. *Science*, 312(5771):224–228, 2006.

[10] I. S. Ufimtsev, N. Luehr, and T. J. Martinez. Charge transfer and polarization in solvated proteins from ab initio molecular dynamics. *J. Phys. Chem. Lett.*, 2(14):1789–1793, 2011.

[11] H. J. Kulik, N. Luehr, I. S. Ufimtsev, and T. J. Martinez. Ab initio quantum chemistry for protein structures. *J. Phys. Chem. B*, 116(41):12501–12509, 2012.

[12] J. A. McCammon, B. R. Gelin, and M. Karplus. Dynamics of folded proteins. *Nature*, 267(5612):585–590, 1977.

[13] X. Zhu, P. E. M. Lopes, and A. D. MacKerell. Recent developments and applications of the CHARMM force fields. *Wiley Interdiscip. Rev. Mol. Sci.*, 2(1):167–185, 2012.

[14] L.-P. Wang, T. J. Martinez, and V. S. Pande. Building force fields: An automatic, systematic, and reproducible approach. *J. Phys. Chem. Lett.*, 5(11):1885–1891, 2014.

[15] J. A. Maier, C. Martinez, K. Kasavajhala, L. Wickstrom, K. E. Hauser, and C. Simmerling. ff14SB: Improving the accuracy of protein side chain and backbone parameters from ff99SB. *J. Chem. Theory Comput.*, 11(8):3696–3713, 2015.

[16] M. Hülsmann and D. Reith. SpaGrOW — a derivative-free optimization scheme for intermolecular force field parameters based on sparse grid methods. *Entropy*, 15(9):3640, 2013.

[17] T. Darden, D. York, and L. Pedersen. Particle mesh Ewald: An N log(N) method for Ewald sums in large systems. *J. Chem. Phys.*, 98(12):10089–10092, 1993.

[18] M. Vendruscolo and C. M. Dobson. Protein dynamics: Moore's law in molecular biology. *Curr. Biol.*, 21(2):R68–R70, 2011.

[19] T. J. Lane, D. Shukla, K. A. Beauchamp, and V. S. Pande. To milliseconds and beyond: Challenges in the simulation of protein folding. *Curr. Opin. Struct. Biol.*, 23(1):58–65, 2013.

[20] D. E. Shaw, J. C. Chao, M. P. Eastwood, J. Gagliardo, J. P. Grossman, C. R. Ho, D. J. Lerardi, I. Kolossváry, J. L. Klepeis, T. Layman, C. McLeavey, M. M. Deneroff, M. A. Moraes, R. Mueller, E. C. Priest, Y. Shan, J. Spengler, M. Theobald, B. Towles, S. C. Wang, R. O. Dror, J. S. Kuskin, R. H. Larson, J. K. Salmon, C. Young, B. Batson, and K. J. Bowers. Anton, a special-purpose machine for molecular dynamics simulation. *Commun. ACM*, 51(7):91, 2008.

[21] R. Fine, G. Dimmler, and C. Levinthal. Fastrun: A special purpose, hardwired computer for molecular simulation. *Proteins Struct. Funct. Bioinf.*, 11(4):242–253, 1991.

[22] M. Taiji, T. Narumi, Y. Ohno, N. Futatsugi, A. Suenaga, N. Takada, and A. Konagaya. Protein explorer: A petaflops special-purpose computer system for molecular dynamics simulations. In *Proceedings of the 2003 ACM/IEEE conference on Supercomputing - SC '03*. ACM, 2003.

[23] S. Toyoda, H. Miyagawa, K. Kitamura, T. Amisaki, E. Hashimoto, H. Ikeda, A. Kusumi, and N. Miyakawa. Development of MD engine: High-speed accelerator with parallel processor design for molecular dynamics simulations. *J. Comput. Chem.*, 20(2):185–199, 1999.

[24] D. E. Shaw, J. Grossman, J. A. Bank, B. Batson, J. A. Butts, J. C. Chao, M. M. Deneroff, R. O. Dror, A. Even, C. H. Fenton, A. Forte, J. Gagliardo, G. Gill, B. Greskamp, C. R. Ho, D. J. Ierardi, L. Iserovich, J. S. Kuskin, R. H. Larson, T. Layman, L.-S. Lee, A. K. Lerer, C. Li, D. Killebrew, K. M. Mackenzie, S. Y.-H. Mok, M. A. Moraes, R. Mueller, L. J. Nociolo, J. L. Peticolas, T. Quan, D. Ramot, J. K. Salmon, D. P. Scarpazza, U. B. Schafer, N. Siddique, C. W. Snyder, J. Spengler, P. T. P. Tang, M. Theobald, H. Toma, B. Towles, B. Vitale, S. C. Wang, and C. Young. Anton 2: Raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer. In

SC14: International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2014.

[25] M. Shirts and V. S. Pande. Screen savers of the world: Unite! *Science*, 290(5498):1903–1904, 2000.

[26] I. Buch, M. J. Harvey, T. Giorgino, D. Anderson, and G. De Fabritiis. High-throughput all-atom molecular dynamics simulations using distributed computing. *J. Chem. Inf. Model.*, 50(3):397–403, 2010.

[27] J. E. Stone, D. J. Hardy, I. S. Ufimtsev, and K. Schulten. GPU-accelerated molecular modeling coming of age. *J. Mol. Graphics Modell.*, 29(2):116–125, 2010.

[28] P. L. Freddolino, C. B. Harrison, Y. Liu, and K. Schulten. Challenges in protein folding simulations: Timescale, representation, and analysis. *Nat. Phys.*, 6(10):751–758, 2010.

[29] S. S. Cho, Y. Levy, and P. G. Wolynes. P versus Q: Structural reaction coordinates capture protein folding on smooth landscapes. *Proc. Natl. Acad. Sci. U.S.A.*, 103(3):586–591, 2006.

[30] J. N. Onuchic and P. G. Wolynes. Theory of protein folding. *Curr. Opin. Struct. Biol.*, 14(1):70–75, 2004.

[31] C. R. Schwantes, R. T. McGibbon, and V. S. Pande. Perspective: Markov models for long-timescale biomolecular dynamics. *J. Chem. Phys.*, 141(9):090901, 2014.

[32] D. Frenkel and B. Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Computational science series. Elsevier Science, 2001.

[33] M. Tuckerman. *Statistical Mechanics: Theory and Molecular Simulation*. OUP Oxford, 2010.

[34] M. Karplus and J. A. McCammon. Molecular dynamics simulations of biomolecules. *Nat. Struct. Biol.*, 9(9):646–652, 2002.

[35] S. A. Adcock and J. A. McCammon. Molecular dynamics: Survey of methods for simulating the activity of proteins. *Chem. Rev.*, 106(5):1589–1615, 2006.

[36] C. Schütte, W. Huisinga, and P. Deuflhard. *Transfer Operator Approach to Conformational Dynamics in Biomolecular Systems*. Springer, 2001.

[37] R. T. McGibbon and V. S. Pande. Learning kinetic distance metrics for Markov state models of protein conformational dynamics. *J. Chem. Theory Comput.*, 9(7):2900–2906, 2013.

[38] P. M. Kasson, E. Lindahl, and V. S. Pande. Atomic-resolution simulations predict a transition state for vesicle fusion defined by contact of a few lipid tails. *PLoS Comput. Biol.*, 6(6):e1000829, 2010.

[39] T. Hansson, C. Oostenbrink, and W. van Gunsteren. Molecular dynamics simulations. *Curr. Opin. Struct. Biol.*, 12(2):190–196, 2002.

[40] J. L. Klepeis, K. Lindorff-Larsen, R. O. Dror, and D. E. Shaw. Long-timescale molecular dynamics simulations of protein structure and function. *Curr. Opin. Struct. Biol.*, 19(2):120–127, 2009.

[41] G. R. Bowman, V. A. Voelz, and V. S. Pande. Taming the complexity of protein folding. *Curr. Opin. Struct. Biol.*, 21(1):4–11, 2011.

[42] Y.-S. Lin, G. R. Bowman, K. A. Beauchamp, and V. S. Pande. Investigating how peptide length and a pathogenic mutation modify the structural ensemble of amyloid beta monomer. *Biophys. J.*, 102(2):315–324, 2012.

[43] M. Matsumoto, S. Saito, and I. Ohmine. Molecular dynamics simulation of the ice nucleation and growth process leading to water freezing. *Nature*, 416(6879):409–413, 2002.

[44] 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. Flitch, 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 Syst. J.*, 40(2):310–327, 2001.

[45] B. Hess, C. Kutzner, D. van der Spoel, and E. Lindahl. GROMACS 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. *J. Chem. Theory Comput.*, 4(3):435–447, 2008.

[46] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kalé, and K. Schulten. Scalable molecular dynamics with NAMD. *J. Comput. Chem.*, 26(16):1781–1802, 2005.

[47] K. Bowers, E. Chow, H. Xu, R. Dror, M. Eastwood, B. Gregersen, J. Klepeis, I. Kolossvary, M. Moraes, F. Sacerdoti, J. Salmon, Y. Shan, and D. Shaw. Scalable algorithms for molecular dynamics simulations on commodity clusters. In *ACM/IEEE SC 2006 Conference – SC '06*. IEEE, 2006.

[48] D. A. Pearlman, D. A. Case, J. W. Caldwell, W. S. Ross, T. E. Cheatham, S. DeBolt, D. Ferguson, G. Seibel, and P. Kollman. AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. *Comput. Phys. Commun.*, 91(1-3):1–41, 1995.

[49] E. Luttmann, D. L. Ensign, V. Vaidyanathan, M. Houston, N. Rimon, J. Øland, G. Jayachandran, M. Friedrichs, and V. S. Pande. Accelerating molecular dynamic simulation on the cell processor and Playstation 3. *J. Comput. Chem.*, 30(2):268–274, 2009.

[50] M. S. Friedrichs, P. Eastman, V. Vaidyanathan, M. Houston, S. Legrand, A. L. Beberg, D. L. Ensign, C. M. Bruns, and V. S. Pande. Accelerating molecular dynamic simulation on graphics processing units. *J. Comput. Chem.*, 30(6):864–872, 2009.

[51] V. S. Pande, K. Beauchamp, and G. R. Bowman. Everything you wanted to know about Markov state models but were afraid to ask. *Methods*, 52(1):99–105, 2010.

[52] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T. R. Weikl. Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations. *Proc. Natl. Acad. Sci. U.S.A.*, 106(45):19011–19016, 2009.

[53] S. V. Krivov and M. Karplus. Hidden complexity of free energy surfaces for peptide (protein) folding. *Proc. Natl. Acad. Sci. U.S.A.*, 101(41):14766–14770, 2004.

[54] S. Muff and A. Caflisch. Kinetic analysis of molecular dynamics simulations reveals changes in the denatured state and switch of folding pathways upon single-point mutation of a β-sheet miniprotein. *Proteins*, 70(4):1185–1195, 2007.

[55] J. D. Chodera, W. C. Swope, J. W. Pitera, and K. A. Dill. Long-time protein folding dynamics from short-time molecular dynamics simulations. *Multiscale Model. Simul.*, 5(4):1214–1226, 2006.

[56] V. A. Voelz, G. R. Bowman, K. Beauchamp, and V. S. Pande. Molecular simulation of ab initio protein folding for a millisecond folder NTL9(1-39). *J. Am. Chem. Soc.*, 132(5):1526–1528, 2010.

[57] P. Deuflhard, W. Huisinga, A. Fischer, and C. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. *Linear Algebra Appl.*, 315(1):39–59, 2000.

[58] P. Deuflhard and M. Weber. Robust perron cluster analysis in conformation dynamics. *Linear Algebra Appl.*, 398:161–184, 2005.

[59] K. A. Beauchamp, R. T. McGibbon, Y.-S. Lin, and V. S. Pande. Simple few-state models reveal hidden complexity in protein folding. *Proc. Natl. Acad. Sci. U.S.A.*, 109(44):17807–17813, 2012.

[60] G. R. Bowman. Improved coarse-graining of Markov state models via explicit consideration of statistical uncertainty. *J. Chem. Phys.*, 137(13):134111, 2012.

[61] N. S. Hinrichs and V. S. Pande. Calculation of the distribution of eigenvalues and eigenvectors in Markovian state models for molecular dynamics. *J. Chem. Phys.*, 126(24):244101, 2007.

[62] G. R. Bowman, D. L. Ensign, and V. S. Pande. Enhanced modeling via network theory: Adaptive sampling of Markov state models. *J. Chem. Theory Comput.*, 6(3):787–794, 2010.

[63] S. Pronk, E. Lindahl, P. Larsson, I. Pouya, G. R. Bowman, I. S. Haque, K. Beauchamp, B. Hess, V. S. Pande, and P. M. Kasson. Copernicus. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis on - SC11*. ACM, 2011.

[64] J. K. Weber and V. S. Pande. Characterization and rapid sampling of protein folding Markov state model topologies. *J. Chem. Theory Comput.*, 7(10):3405–3411, 2011.

[65] P. Cossio, A. Laio, and F. Pietrucci. Which similarity measure is better for analyzing protein structures in a molecular dynamics trajectory? *Phys. Chem. Chem. Phys.*, 13(22):10421, 2011.

[66] T. Zhou and A. Caflisch. Distribution of reciprocal of interatomic distances: a fast structural metric. *J. Chem. Theory Comput.*, 8(8):2930–2937, 2012.

[67] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Adv. Neural Inf. Process. Syst. 15*, pages 505–512, 2003.

[68] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 577–584, San Francisco, CA, USA, 2001.

[69] C. Shen, J. Kim, and L. Wang. Scalable large-margin Mahalanobis distance metric learning. *IEEE Trans. Neural Networks*, 21(9):1524–1530, 2010.

[70] F. Noé and S. Fischer. Transition networks for modeling the kinetics of conformational change in macromolecules. *Curr. Opin. Struct. Biol.*, 18(2):154–162, 2008.

[71] J. M. Carr and D. J. Wales. Global optimization and folding pathways of selected α-helical proteins. *J. Chem. Phys.*, 123(23):234901, 2005.

[72] J. M. Carr and D. J. Wales. Folding pathways and rates for the three-stranded β-sheet peptide beta3s using discrete path sampling. *J. Phys. Chem. B*, 112(29):8760–8769, 2008.

[73] D. Prada-Gracia, J. Gómez-Gardeñes, P. Echenique, and F. Falo. Exploring the free energy landscape: From dynamics to networks and back. *PLoS Comput. Biol.*, 5(6):e1000415, 2009.

[74] B. Keller, X. Daura, and W. F. van Gunsteren. Comparing geometric and kinetic cluster algorithms for molecular simulation data. *J. Chem. Phys.*, 132(7):074110, 2010.

[75] K. A. Beauchamp, G. R. Bowman, T. J. Lane, L. Maibaum, I. S. Haque, and V. S. Pande. MSMBuilder2: Modeling conformational dynamics on the picosecond to millisecond scale. *J. Chem. Theory Comput.*, 7(10):3412–3419, 2011.

[76] X. Huang, Y. Yao, G. R. Bowman, J. Sub, L. J. Guibas, G. Carlsson, and V. S. Pande. Constructing multi-resolution Markov state models (MSMs) to elucidate RNA hairpin folding mechanisms. In *Biocomputing 2010*, pages 228–239. World Scientific Pub Co Pte Lt, 2009.

[77] T. J. Lane, G. R. Bowman, K. Beauchamp, V. A. Voelz, and V. S. Pande. Markov state model reveals folding and functional dynamics in ultra-long MD trajectories. *J. Am. Chem. Soc.*, 133(45):18413–18419, 2011.

[78] W. Zhuang, R. Z. Cui, D.-A. Silva, and X. Huang. Simulating the T-jump-triggered unfolding dynamics of trpzip2 peptide and its time-resolved IR and two-dimensional IR signals using the Markov state model approach. *J. Phys. Chem. B*, 115(18):5415–5424, 2011.

[79] K. A. Beauchamp, D. L. Ensign, R. Das, and V. S. Pande. Quantitative comparison of villin headpiece subdomain simulations and triplet-triplet energy transfer experiments. *Proc. Natl. Acad. Sci. U.S.A.*, 108(31):12734–12739, 2011.

[80] V. A. Voelz, M. Jäger, S. Yao, Y. Chen, L. Zhu, S. A. Waldauer, G. R. Bowman, M. Friedrichs, O. Bakajin, L. J. Lapidus, S. Weiss, and V. S. Pande. Slow unfolded-state structuring in acyl-CoA binding protein folding revealed by simulation and experiment. *J. Am. Chem. Soc.*, 134(30):12565–12577, 2012.

[81] N. G. Van Kampen. *Stochastic processes in physics and chemistry*. Elsevier Science, 1992.

[82] R. Zwanzig. Memory effects in irreversible thermodynamics. *Phys. Rev.*, 124(4):983–992, 1961.

[83] S. Bacallado. Bayesian analysis of variable-order, reversible Markov chains. *Ann. Stat.*, 39(2):838–864, 2011.

[84] S. Park and V. S. Pande. Validation of Markov state models using shannon's entropy. *J. Chem. Phys.*, 124(5):054118, 2006.

[85] M. Sarich, F. Noé, and C. Schütte. On the approximation quality of Markov state models. *Multiscale Model. Simul.*, 8(4):1154–1177, 2010.

[86] H. Feng, R. Costaouec, E. Darve, and J. A. Izaguirre. A comparison of weighted ensemble and Markov state model methodologies. *J. Chem. Phys.*, 142(21):214113, 2015.

[87] G. Huber and S. Kim. Weighted-ensemble Brownian dynamics simulations for protein association reactions. *Biophys. J.*, 70(1):97–110, 1996.

[88] D. Bhatt and D. M. Zuckerman. Heterogeneous path ensembles for conformational transitions in semiatomistic models of adenylate kinase. *J. Chem. Theory Comput.*, 6(11):3527–3539, 2010.

[89] C. Shen, A. Welsh, and L. Wang. PSDBoost: Matrix-generation linear programming for positive semidefinite matrices learning. In *Adv. Neural Inf. Process. Syst. 21*, pages 1473–1480, 2009.

[90] N. Djurdjevac, M. Sarich, and C. Schütte. Estimating the eigenvalue error of Markov state models. *Multiscale Model. Simul.*, 10(1):61–81, 2012.

[91] D. E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R. O. Dror, M. P. Eastwood, J. A. Bank, J. M. Jumper, J. K. Salmon, Y. Shan, and W. Wriggers. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330(6002):341–346, 2010.

[92] R. T. McGibbon, B. Ramsundar, M. M. Sultan, G. Kiss, and V. S. Pande. Understanding protein dynamics with l_1-regularized reversible hidden Markov models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1197–1205, Beijing, China, 2014.

[93] C. F. Wong and J. A. McCammon. Protein flexibility and computer-aided drug design. *Annu. Rev. Pharmacol. Toxicol.*, 43(1):31–45, 2003.

[94] K. A. Dill, S. Bromberg, K. Yue, H. S. Chan, K. M. Ftebig, D. P. Yee, and P. D. Thomas. Principles of protein folding – a perspective from simple exact models. *Protein Sci.*, 4(4):561–602, 1995.

[95] K. A. Beauchamp, Y.-S. Lin, R. Das, and V. S. Pande. Are protein force fields getting better? A systematic benchmark on 524 diverse NMR measurements. *J. Chem. Theory Comput.*, 8(4):1409–1414, 2012.

[96] K. J. Kohlhoff, D. Shukla, M. Lawrenz, G. R. Bowman, D. E. Konerding, D. Belov, R. B. Altman, and V. S. Pande. Cloud-based simulations on google exacycle reveal ligand modulation of gpcr activation pathways. *Nature Chem.*, 6(1):15–21, 2014.

[97] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé. Markov models of molecular kinetics: Generation and validation. *J. Chem. Phys.*, 134(17):174105, 2011.

[98] H. Maity, M. Maity, M. M. G. Krishna, L. Mayne, and S. W. Englander. Protein folding: The stepwise assembly of foldon units. *Proc. Natl. Acad. Sci. U.S.A.*, 102(13):4741–4746, 2005.

[99] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. R. Statistic. Soc. B*, 67(1):91–108, 2005.

[100] D. Sontag, T. Meltzer, A. Globerson, T. S. Jaakkola, and Y. Weiss. Tightening LP relaxations for MAP using message passing. *arXiv preprint arXiv:1206.3288*, 2012.

[101] P. Di Lena, P. Baldi, and K. Nagata. Deep spatio-temporal architectures and learning for protein structure prediction. In *Adv. Neural Inf. Process. Syst. 25*, pages 521–529, 2012.

[102] W. Chu, Z. Ghahramani, A. Podtelezhnikov, and D. L. Wild. Bayesian segmental models with multiple sequence alignment profiles for protein secondary structure and contact map prediction. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 3(2):98–113, 2006.

[103] P. Baldi and G. Pollastri. The principled design of large-scale recursive neural network architectures–DAG-RNNs and the protein structure prediction problem. *J. Mach. Learn. Res.*, 4:575–602, 2003.

[104] W. Humphrey, A. Dalke, and K. Schulten. VMD: Visual molecular dynamics. *J. Mol. Graphics*, 14(1):33–38, 1996.

[105] M. Karplus and J. Kuriyan. Molecular dynamics and protein function. *Proc. Natl. Acad. Sci. U.S.A.*, 102(19):6679–6685, 2005.

[106] F. Noé, H. Wu, J.-H. Prinz, and N. Plattner. Projected and hidden Markov models for calculating kinetics and metastable states of complex molecules. *J. Chem. Phys.*, 139(18):184114, 2013.

[107] S. K. Sadiq, F. Noé, and G. De Fabritiis. Kinetic characterization of the critical step in HIV-1 protease maturation. *Proc. Natl. Acad. Sci. U.S.A.*, 109(50):20449–20454, 2012.

[108] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, 1989.

[109] J. Guo, E. Levina, G. Michailidis, and J. Zhu. Pairwise variable selection for high-dimensional model-based clustering. *Biometrics*, 66(3):793–804, 2010.

[110] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Stat. Assoc.*, 96(456):1348–1360, 2001.

[111] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2006.

[112] J. K. Weber and V. S. Pande. Protein folding is mechanistically robust. *Biophys. J.*, 102(4):859–867, 2012.

[113] D. J. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Rev.*, 43(3):525–546, 2001.

[114] A. Hershko and A. Ciechanover. The ubiquitin system. *Annu. Rev. Biochem.*, 67(1):425–479, 1998.

[115] Y. Zhang, L. Zhou, L. Rouge, A. H. Phillips, C. Lam, P. Liu, W. Sandoval, E. Helgason, J. M. Murray, I. E. Wertz, et al. Conformational stabilization of ubiquitin yields potent and selective inhibitors of USP7. *Nat. Chem. Biol.*, 9(1):51–58, 2012.

[116] D. Komander, M. J. Clague, and S. Urbé. Breaking the chains: structure and function of the deubiquitinases. *Nat. Rev. Mol. Cell Biol.*, 10(8):550–563, 2009.

[117] A. Aleshin and R. S. Finn. SRC: a century of science brought to the clinic. *Neoplasia*, 12(8):599–607, 2010.

[118] D. Shukla, Y. Meng, B. Roux, and V. S. Pande. Activation pathway of Src kinase reveals intermediate states as targets for drug design. *Nat. Commun.*, 5, 2014.

[119] Z. Fang, C. Grütter, and D. Rauh. Strategies for the selective regulation of kinases with allosteric modulators: Exploiting exclusive structural features. *ACS Chem. Biol.*, 8(1):58–70, 2013.

[120] R. T. McGibbon and V. S. Pande. Efficient maximum likelihood parameterization of continuous-time Markov processes. *J. Chem. Phys.*, 143(3):034109, 2015.

[121] B. Singer and S. Spilerman. The representation of social processes by Markov models. *Am. J. Sociol.*, 82(1):1, 1976.

[122] H. Madsen, H. Spliid, and P. Thyregod. Markov models in discrete and continuous time for hourly observations of cloud cover. *J. Appl. Meteorol.*, 24(7):629–639, 1985.

[123] C. M. Turner, R. Startz, and C. R. Nelson. A Markov model of heteroskedasticity, risk, and learning in the stock market. *J. Financ. Econ.*, 25(1):3–22, 1989.

[124] V. N. Minin and M. A. Suchard. Counting labeled transitions in continuous-time Markov models of evolution. *J. Math. Biol.*, 56(3):391–412, 2008.

[125] R. A. Jarrow, D. Lando, and S. M. Turnbull. A Markov model for the term structure of credit risk spreads. *Rev. Financ. Stud.*, 10(2):481–523, 1997.

[126] S. Spilerman. Extensions of the mover-stayer model. *Am. J. Sociol.*, 78(3):599–626, 1972.

[127] M. Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.*, 16(2):111–120, 1980.

[128] D. F. Anderson and T. G. Kurtz. Continuous time Markov chain models for chemical reaction networks. In *Design and analysis of biomolecular circuits*, pages 3–42. Springer, 2011.

[129] A. Ikai and C. Tanford. Kinetic evidence for incorrectly folded intermediate states in the refolding of denatured proteins. *Nature*, 230:100–102, 1971.

[130] R. Zwanzig. Two-state models of protein folding kinetics. *Proc. Natl. Acad. Sci. U.S.A.*, 94(1):148, 1997.

[131] I. E. Sánchez and T. Kiefhaber. Evidence for sequential barriers and obligatory intermediates in apparent two-state protein folding. *J. Mol. Biol.*, 325(2):367–376, 2003.

[132] H. S. Chan and K. A. Dill. Protein folding in the landscape perspective: Chevron plots and non-arrhenius kinetics. *Proteins Struct. Funct. Bioinf.*, 30(1):2–33, 1998.

[133] M. Pirchi, G. Ziv, I. Riven, S. S. Cohen, N. Zohar, Y. Barak, and G. Haran. Single-molecule fluorescence spectroscopy maps the folding landscape of a large protein. *Nat. Commun.*, 2:493, 2011.

[134] M. Senne, B. Trendelkamp-Schroer, A. S. Mey, C. Schütte, and F. Noé. EMMA: A software package for Markov model building and analysis. *J. Chem. Theory Comput.*, 8(7):2223–2238, 2012.

[135] R. Banerjee and R. I. Cukier. Transition paths of met-enkephalin from Markov state modeling of a molecular dynamics trajectory. *J. Phys. Chem. B*, 118(11):2883–2895, 2014.

[136] D. Shukla, C. X. Hernández, J. K. Weber, and V. S. Pande. Markov state models provide insights into dynamic modulation of protein function. *Acc. Chem. Res.*, 48(2):414–422, 2015.

[137] J. D. Chodera and F. F. Noé. Markov state models of biomolecular conformational dynamics. *Curr. Opin. Struct. Biol.*, 25(0):135–144, 2014.

[138] P. Metzner, E. Dittmer, T. Jahnke, and C. Schütte. Generator estimation of Markov jump processes. *J. Comput. Phys.*, 227(1):353–375, 2007.

[139] I. Holmes and G. Rubin. An expectation maximization algorithm for training hidden substitution models. *J. Mol. Biol.*, 317(5):753–764, 2002.

[140] M. Bladt and M. Sørensen. Statistical inference for discretely observed Markov jump processes. *J. R. Statistic. Soc. B*, 67(3):395–410, 2005.

[141] J. F. C. Kingman. The imbedding problem for finite Markov chains. *Probab. Theory Related Fields*, 1(1):14–24, 1962.

[142] E. Davies. Embeddable Markov matrices. *Electron. J. Probab.*, 15:1474–1486, 2010.

[143] J. Adams and J. Doll. Dynamical corrections to transition state theory adsorption rates: Effect of a precursor state. *Surf. Sci.*, 103(2):472–481, 1981.

[144] A. F. Voter and J. D. Doll. Dynamical corrections to transition state theory for multistate systems: Surface self-diffusion in the rare-event regime. *J. Chem. Phys.*, 82(1):80–92, 1985.

[145] H. Mori. Transport, collective motion, and Brownian motion. *Prog. Theor. Phys.*, 33(3):423–455, 1965.

[146] D. Crommelin and E. Vanden-Eijnden. Fitting timeseries by continuous-time Markov chains: A quadratic programming approach. *J. Comput. Phys.*, 217(2):782–805, 2006.

[147] D. Crommelin and E. Vanden-Eijnden. Data-based inference of generators for Markov jump processes using convex optimization. *Multiscale Model. Simul.*, 7(4):1751–1778, 2009.

[148] R. B. Israel, J. S. Rosenthal, and J. Z. Wei. Finding generators for Markov chains via empirical transition matrices, with applications to credit ratings. *Math. Finance*, 11(2):245–265, 2001.

[149] J. D. Kalbfleisch and J. F. Lawless. The analysis of panel data under a Markov assumption. *J. Am. Stat. Assoc.*, 80(392):863871, 1985.

[150] S. Asmussen, O. Nerman, and M. Olsson. Fitting phase-type distributions via the em algorithm. *Scand. J. Stat.*, 23(4):419–441, 1996.

[151] A. Hobolth and J. L. Jensen. Statistical inference in evolutionary models of DNA sequences via the EM algorithm. *Stat. Appl. Genet. Mol. Biol.*, 4(1):1–12, 2005.

[152] R. I. Jennrich and P. B. Bright. Fitting systems of linear differential equations using computer generated exact derivatives. *Technometrics*, 18(4):385–392, 1976.

[153] R. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16(5):1190–1208, 1995.

[154] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Software*, 23(4):550–560, 1997.

[155] A. Al-Mohy and N. Higham. Improved inverse scaling and squaring algorithms for the matrix logarithm. *SIAM J. Sci. Comput.*, 34(4):C153–C169, 2012.

[156] D. Bakry, I. Gentil, and M. Ledoux. *Analysis and Geometry of Markov Diffusion Operators*, volume 348 of *Grundlehren der mathematischen Wissenschaften*. Springer International Publishing, 2014.

[157] W. H. Vandevender and K. H. Haskell. The SLATEC mathematical subroutine library. *SIGNUM Newsl.*, 17(3):16–21, 1982.

[158] B. Gough. *GNU scientific library reference manual*. Network Theory Ltd., 2009.

[159] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

[160] C. R. Rao. *Linear Statistical Inference and its Applications*. John Wiley & Sons, Inc., 1973.

[161] D. V. Murthy and R. T. Haftka. Derivatives of eigenvalues and eigenvectors of a general complex matrix. *Int. J. Numer. Methods Eng.*, 26(2):293–311, 1988.

[162] R. T. McGibbon, C. R. Schwantes, and V. S. Pande. Statistical model selection for Markov models of biomolecular dynamics. *J. Phys. Chem. B*, 118(24):6475–6481, 2014.

[163] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[164] F. Liu, D. Du, A. A. Fuller, J. E. Davoren, P. Wipf, J. W. Kelly, and M. Gruebele. An experimental survey of the transition between two-state and downhill protein folding scenarios. *Proc. Natl. Acad. Sci. U.S.A.*, 105(7):2369–2374, 2008.

[165] R. T. McGibbon, K. A. Beauchamp, M. P. Harrigan, C. Klein, J. M. Swails, C. X. Hernández, C. R. Schwantes, L.-P. Wang, T. J. Lane, and V. S. Pande. MDTraj: A modern open library for the analysis of molecular dynamics trajectories. *Biophys. J.*, 109(8):1528–1532, 2015.

[166] C. R. Schwantes and V. S. Pande. Improvements in Markov state model construction reveal many non-native interactions in the folding of NTL9. *J. Chem. Theory Comput.*, 9(4):2000–2009, 2013.

[167] G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. Noé. Identification of slow molecular order parameters for Markov model construction. *J. Chem. Phys.*, 139(1):015102, 2013.

[168] S. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inform. Theory*, 28(2):129–137, 1982.

[169] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms*, pages 1027–1035. SIAM, 2007.

[170] W. C. Swope, J. W. Pitera, F. Suits, M. Pitman, M. Eleftheriou, B. G. Fitch, R. S. Germain, A. Rayshubski, T. J. C. Ward, Y. Zhestkov, and R. 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(21):6582–6594, 2004.

[171] G. R. Bowman, X. Huang, and V. S. Pande. Using generalized ensemble simulations and Markov state models to identify conformational states. *Methods*, 49(2):197–201, 2009.

[172] D. Chandler. Statistical mechanics of isomerization dynamics in liquids and the transition state approximation. *J. Chem. Phys.*, 68(6):2959–2970, 1978.

[173] D. Chandler. Barrier crossings: classical theory of rare but important events. In B. J. Berne, G. Ciccotti, and D. J. Coker, editors, *Classical and Quantum Dynamics in Condensed Phase Simulations*, pages 3–23. World Scientific, Singapore, 1998.

[174] R. M. Neal. MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*, pages 113–162. Chapman & Hall / CRC, 2011.

[175] B. Trendelkamp-Schroer and F. Noé. Efficient bayesian estimation of Markov model transition matrices with given stationary distribution. *J. Chem. Phys.*, 138(16):164113, 2013.

[176] C. M. Dobson. Protein folding and misfolding. *Nature*, 426(6968):884–890, 2003.

[177] S. J. Kim, B. Born, M. Havenith, and M. Gruebele. Real-time detection of protein–water dynamics upon protein folding by terahertz absorption spectroscopy. *Angew. Chem. Int. Ed.*, 47(34):6486–6489, 2008.

[178] R. H. Austin, K. W. Beeson, L. Eisenstein, H. Frauenfelder, and I. C. Gunsalus. Dynamics of ligand binding to myoglobin. *Biochemistry*, 14(24):5355–5373, 1975.

[179] I. Bahar, C. Chennubhotla, and D. Tobi. Intrinsic dynamics of enzymes in the unbound state and relation to allosteric regulation. *Curr. Opin. Struct. Biol.*, 17(6):633–640, 2007.

[180] G. Cosa, Y. Zeng, H.-W. Liu, C. F. Landes, D. E. Makarov, K. Musier-Forsyth, and P. F. Barbara. Evidence for non-two-state kinetics in the nucleocapsid protein chaperoned opening of DNA hairpins. *J. Phys. Chem. B*, 110(5):2419–2426, 2006.

[181] X. Zhang, V. Q. Lam, Y. Mou, T. Kimura, J. Chung, S. Chandrasekar, J. R. Winkler, S. L. Mayo, and S. Shan. Direct visualization reveals dynamics of a transient intermediate during protein assembly. *Proc. Natl. Acad. Sci. U.S.A.*, 108(16):6450–6455, 2011.

[182] E. A. Lipman, B. Schuler, O. Bakajin, and W. A. Eaton. Single-molecule measurement of protein folding kinetics. *Science*, 301(5637):1233–1235, 2003.

[183] H. D. Mertens and D. I. Svergun. Structural characterization of proteins and complexes using small-angle X-ray solution scattering. *J. Struct. Biol.*, 172(1):128–141, 2010.

[184] S.-R. Tzeng and C. G. Kalodimos. Protein dynamics and allostery: an NMR view. *Curr. Opin. Struct. Biol.*, 21(1):62–67, 2011.

[185] A. W. Götz, M. J. Williamson, D. Xu, D. Poole, S. Le Grand, and R. C. Walker. Routine microsecond molecular dynamics simulations with AMBER on GPUs. 1. generalized Born. *J. Chem. Theory Comput.*, 8(5):1542–1555, 2012.

[186] P. Eastman, M. S. Friedrichs, J. D. Chodera, R. J. Radmer, C. M. Bruns, J. P. Ku, K. A. Beauchamp, T. J. Lane, L.-P. Wang, D. Shukla, T. Tye, M. Houston, T. Stich, C. Klein, M. R. Shirts, and V. S. Pande. OpenMM 4: A reusable, extensible, hardware independent library for high performance molecular simulation. *J. Chem. Theory Comput.*, 9(1):461–469, 2013.

[187] D. Shaw, R. Dror, J. Salmon, J. Grossman, K. Mackenzie, J. Bank, C. Young, M. Deneroff, B. Batson, K. Bowers, E. Chow, M. Eastwood, D. Ierardi, J. Klepeis, J. Kuskin, R. Larson, K. Lindorff-Larsen, P. Maragakis, M. Moraes, S. Piana, Y. Shan, and B. Towles. Millisecond-scale molecular dynamics simulations on Anton. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis – SC'09*, pages 1–11. ACM, 2009.

[188] B. Hess. P-lincs: A parallel linear constraint solver for molecular simulation. *J. Chem. Theory Comput.*, 4(1):116–122, 2008.

[189] R. B. Best and G. Hummer. Coordinate-dependent diffusion in protein folding. *Proc. Natl. Acad. Sci. U.S.A.*, 107(3):1088–1093, 2010.

[190] J. D. Chodera, N. Singhal, V. S. Pande, K. A. Dill, and W. C. Swope. Automatic discovery of metastable states for the construction of Markov models of macromolecular conformational dynamics. *J. Chem. Phys.*, 126(15):155101, 2007.

[191] G. R. Bowman, L. Meng, and X. Huang. Quantitative comparison of alternative methods for coarse-graining biological networks. *J. Chem. Phys.*, 139(12):121905, 2013.

[192] W. E and E. Vanden-Eijnden. Towards a theory of transition paths. *J. Stat. Phys.*, 123(3):503–523, 2006.

[193] C. Sammut and G. I. Webb. *Encyclopedia of Machine Learning*. Springer, 2010.

[194] F. Noé and F. Nüske. A variational approach to modeling slow processes in stochastic dynamical systems. *Multiscale Model. Simul.*, 11(2):635–655, 2013.

[195] G. R. Bowman and P. L. Geissler. Equilibrium fluctuations of a single folded protein reveal a multitude of potential cryptic allosteric sites. *Proc. Natl. Acad. Sci. U.S.A.*, 109(29):11681–11686, 2012.

[196] G. R. Bowman, K. A. Beauchamp, G. Boxer, and V. S. Pande. Progress and challenges in the automated construction of Markov state models for full protein systems. *J. Chem. Phys.*, 131(12):124101, 2009.

[197] H. L. Gordon and R. L. Somorjai. Fuzzy cluster analysis of molecular dynamics trajectories. *Proteins Struct. Funct. Bioinf.*, 14(2):249–264, 1992.

[198] E. H. Kellogg, O. F. Lange, and D. Baker. Evaluation and optimization of discrete state models of protein folding. *J. Phys. Chem. B*, 116(37):11405–11413, 2012.

[199] A. R. Liddle. Information Criteria for Astrophysical Model Selection. *Mon. Not. R. Astron. Soc. Lett.*, 377(1):L74–L78, 2007.

[200] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, 2001.

[201] A. E. Gelfand and D. K. Dey. Bayesian model choice: Asymptotics and exact calculations. *J. R. Statistic. Soc. B*, 56(3):501–514, 1994.

[202] G. Schwarz. Estimating the dimension of a model. *Ann. Stat.*, 6(2):461–464, 1978.

[203] H. Akaike. A new look at the statistical model identification. *IEEE Trans. Autom. Control*, 19(6):716–723, 1974.

[204] R. E. Kass and A. E. Raftery. Bayes factors. *J. Am. Stat. Assoc.*, 90(430):773–795, 1995.

[205] A. R. Liddle. How many cosmological parameters? *Mon. Not. R. Astron. Soc. Lett.*, 351(3):L49–L53, 2004.

[206] J. Kuha. AIC and BIC: Comparisons of assumptions and performance. *Sociol. Method Res.*, 33(2):188–229, 2004.

[207] R. Kannan, L. Lovász, and M. Simonovits. Random walks and an $o^*(n^5)$ volume algorithm for convex bodies. *Random Structures Algorithms*, 11(1):1–50, 1997.

[208] M. Simonovits. How to compute the volume in high dimension? *Math. Program.*, 97(1-2):337–374, 2003.

[209] L. Lovász and S. Vempala. Simulated annealing in convex bodies and an $o^*(n^4)$ volume algorithm. *J. Comput. System Sci.*, 72(2):392–417, 2006.

[210] K. Müller and L. D. Brown. Location of saddle points and minimum energy paths by a constrained simplex optimization procedure. *Theor. Chim. Acta*, 53(1):75–93, 1979.

[211] K. Lindorff-Larsen, S. Piana, K. Palmo, P. Maragakis, J. L. Klepeis, R. O. Dror, and D. E. Shaw. Improved side-chain torsion potentials for the amber ff99SB protein force field. *Proteins Struct. Funct. Bioinf.*, 78(8):1950–1958, 2010.

[212] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein. Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.*, 79(2):926–935, 1983.

[213] T.-H. Chiang, D. Hsu, and J.-C. Latombe. Markov dynamic models for long-timescale protein motion. *Bioinformatics*, 26(12):i269–i277, 2010.

[214] R. T. McGibbon and V. S. Pande. Variational cross-validation of slow dynamical modes in molecular kinetics. *J. Chem. Phys.*, 142(12):124105, 2015.

[215] K. A. Dill, S. B. Ozkan, M. S. Shell, and T. R. Weikl. The protein folding problem. *Annu. Rev. Biophys.*, 37:289, 2008.

[216] M. Huse and J. Kuriyan. The conformational plasticity of protein kinases. *Cell*, 109(3):275–282, 2002.

[217] E. Vargas, V. Yarov-Yarovoy, F. Khalili-Araghi, W. A. Catterall, M. L. Klein, M. Tarek, E. Lindahl, K. Schulten, E. Perozo, F. Bezanilla, et al. An emerging consensus on voltage-dependent gating from computational modeling and molecular dynamics simulations. *J. Gen. Physiol.*, 140(6):587–594, 2012.

[218] A. H. Phillips, Y. Zhang, C. N. Cunningham, L. Zhou, W. F. Forrest, P. S. Liu, M. Steffek, J. Lee, C. Tam, E. Helgason, et al. Conformational dynamics control ubiquitin-deubiquitinase interactions and influence in vivo signaling. *Proc. Natl. Acad. Sci. U.S.A.*, 110(28):11379–11384, 2013.

[219] L.-P. Wang, J. Chen, and T. Van Voorhis. Systematic parametrization of polarizable force fields from quantum chemistry data. *J. Chem. Theory Comput.*, 9(1):452–460, 2012.

[220] L. Huang and B. Roux. Automated force field parameterization for nonpolarizable and polarizable atomic models based on ab initio target data. *J. Chem. Theory Comput.*, 9(8):3543–3556, 2013.

[221] J. W. Ponder, C. Wu, P. Ren, V. S. Pande, J. D. Chodera, M. J. Schnieders, I. Haque, D. L. Mobley, D. S. Lambrecht, J. Robert A. DiStasio, M. Head-Gordon, G. N. I. Clark, M. E. Johnson, and T. Head-Gordon. Current status of the AMOEBA polarizable force field. *J. Phys. Chem. B*, 114(8):2549–2564, 2010.

[222] R. B. Best, X. Zhu, J. Shim, P. E. Lopes, J. Mittal, M. Feig, and A. D. MacKerell Jr. Optimization of the additive charmm all-atom protein force field targeting improved sampling of the backbone ϕ, ψ and side-chain $\chi1$ and $\chi2$ dihedral angles. *J. Chem. Theory Comput.*, 8(9):3257–3273, 2012.

[223] P. E. Lopes, J. Huang, J. Shim, Y. Luo, H. Li, B. Roux, and A. D. MacKerell Jr. Polarizable force field for peptides and proteins based on the classical Drude oscillator. *J. Chem. Theory Comput.*, 9(12):5430–5449, 2013.

[224] M. A. Rohrdanz, W. Zheng, M. Maggioni, and C. Clementi. Determination of reaction coordinates via locally scaled diffusion map. *J. Chem. Phys.*, 134(12):124116, 2011.

[225] A. Altis, P. H. Nguyen, R. Hegger, and G. Stock. Dihedral angle principal component analysis of molecular dynamics simulations. *J. Chem. Phys.*, 126(24):244111, 2007.

[226] P. Das, M. Moll, H. Stamati, L. E. Kavraki, and C. Clementi. Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. *Proc. Natl. Acad. Sci. U.S.A.*, 103(26):9885–9890, 2006.

[227] V. N. Vapnik and V. Vapnik. *Statistical learning theory*. Wiley New York, 1998.

[228] S. C. Larson. The shrinkage of the coefficient of multiple correlation. *J. Educ. Psychol.*, 22(1):45, 1931.

[229] O. Z. Maimon and L. Rokach. *Data mining and knowledge discovery handbook*. Springer, 2005.

[230] C. Schütte. *Conformational dynamics: modelling, theory, algorithm, and application to biomolecules*. PhD thesis, Habilitation thesis, Department of Mathematics and Computer Science, Freie Universität Berlin, 1999.

[231] E. Schmidt. Zur theorie der linearen und nichtlinearen integralgleichungen. *Math. Ann.*, 63(4):433–476, 1907.

[232] R. Courant and D. Hilbert. *Methods of Mathematical Physics*. Number v. 1 in Methods of Mathematical Physics. Wiley, 2008.

[233] C. A. Micchelli and A. Pinkus. Some problems in the approximation of functions of two variables and n-widths of integral operators. *J. Approx. Theory*, 24(1):51–77, 1978.

[234] F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, and F. Noé. Variational approach to molecular kinetics. *J. Chem. Theory Comput.*, 10(4):1739–1752, 2014.

[235] K. Fan. On a theorem of Weyl concerning eigenvalues of linear transformations I. *Proc. Natl. Acad. Sci. U.S.A.*, 35(11):652, 1949.

[236] M. L. Overton and R. S. Womersley. On the sum of the largest eigenvalues of a symmetric matrix. *SIAM J. Matrix Anal. Appl.*, 13(1):41–45, 1992.

[237] P.-A. Absil, R. Mahony, R. Sepulchre, and P. Van Dooren. A Grassmann–Rayleigh quotient iteration for computing invariant subspaces. *SIAM Rev.*, 44(1):57–73, 2002.

[238] D. Sezer, J. H. Freed, and B. Roux. Using Markov models to simulate electron spin resonance spectra from molecular dynamics trajectories. *J. Phys. Chem. B*, 112(35):11014–11027, 2008.

[239] S. Muff and A. Caflisch. Identification of the protein folding transition state from molecular dynamics trajectories. *J. Chem. Phys.*, 130(12):125104, 2009.

[240] I. Buch, T. Giorgino, and G. De Fabritiis. Complete reconstruction of an enzyme-inhibitor binding process by molecular dynamics simulations. *Proc. Natl. Acad. Sci. U.S.A.*, 108(25):10184–10189, 2011.

[241] O. P. Choudhary, A. Paz, J. L. Adelman, J.-P. Colletier, J. Abramson, and M. Grabe. Structure-guided simulations illuminate the mechanism of ATP transport through VDAC1. *Nat. Struct. Mol. Biol.*, 21(7):626–632, 2014.

[242] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.

[243] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, 1998.

[244] U. Kjems, L. K. Hansen, and S. C. Strother. Generalizable singular value decomposition for ill-posed datasets. In *Adv. Neural Inf. Process. Syst. 13*, pages 549–555. 2000.

[245] T. J. Abrahamsen and L. K. Hansen. A cure for variance inflation in high dimensional kernel principal component analysis. *J. Mach. Learn. Res.*, 12:2027–2044, 2011.

[246] M. Cornec. Concentration inequalities of the cross-validation estimator for empirical risk minimiser. *arXiv preprint arXiv:1011.0096*, 2010.

[247] D. L. Theobald. Rapid calculation of RMSDs using a quaternion-based characteristic polynomial. *Acta Crystallogr., Sect. A: Found. Crystallogr.*, 61(4):478–480, 2005.

[248] A. K. Theophilou. The energy density functional formalism for excited states. *J. Phys. C*, 12(24):5419, 1979.

[249] E. K. U. Gross, L. N. Oliveira, and W. Kohn. Rayleigh-Ritz variational principle for ensembles of fractionally occupied states. *Phys. Rev. A*, 37:2805–2808, 1988.

[250] N. Gidopoulos, P. Papaconstantinou, and E. Gross. Ensemble-Hartree–Fock scheme for excited states. the optimized effective potential method. *Physica B*, 318(4):328–332, 2002.

[251] R. Lai, J. Lu, and S. Osher. Density matrix minimization with ℓ_1 regularization. *arXiv preprint arXiv:1403.1525*, 2014.

[252] C. R. Rao. The utilization of multiple measurements in problems of biological classification. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 10(2):159–203, 1948.

[253] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Comput.*, 12(10):2385–2404, 2000.

[254] L. Clemmensen, T. Hastie, D. Witten, and B. Ersbøll. Sparse discriminant analysis. *Technometrics*, 53(4):406–413, 2011.

[255] B. K. Sriperumbudur, D. Torres, and G. Lanckriet. A majorization-minimization approach to the sparse generalized eigenvalue problem. *Mach. Learn.*, 85(1-2):3–39, 2011.

[256] E. K. Rains and H. C. Andersen. A Bayesian method for construction of Markov models to describe dynamics on various time-scales. *J. Chem. Phys.*, 133(14):144113, 2010.

[257] S. Bacallado, J. D. Chodera, and V. Pande. Bayesian comparison of Markov models of molecular dynamics with detailed balance constraint. *J. Chem. Phys.*, 131(4):045106, 2009.

[258] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Adv. Neural Inf. Process. Syst. 25*, pages 2951–2959. Lake Tahoe, USA, 2012.

[259] D. Müllner. fastcluster: Fast hierarchical, agglomerative clustering routines for r and python. *J. Stat. Softw.*, 53(9):1–18, 2013.

[260] D.-W. Li and R. Brüschweiler. NMR-based protein potentials. *Angew. Chem.*, 122(38):6930–6932, 2010.

[261] G. Bussi, D. Donadio, and M. Parrinello. Canonical sampling through velocity rescaling. *J. Chem. Phys.*, 126(1):014101, 2007.

[262] M. Parrinello and A. Rahman. Polymorphic transitions in single crystals: A new molecular dynamics method. *J. Appl. Phys.*, 52(12):7182–7190, 1981.

[263] H. Eyring. The activated complex in chemical reactions. *J. Chem. Phys.*, 3(2):107–115, 1935.

[264] H. Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(4):284 – 304, 1940.

[265] D. G. Truhlar, B. C. Garrett, and S. J. Klippenstein. Current status of transition-state theory. *J. Phys. Chem.*, 100(31):12771–12800, 1996.

[266] S. Yang, J. N. Onuchic, and H. Levine. Effective stochastic dynamics on a protein folding energy landscape. *J. Chem. Phys.*, 125(5):054910, 2006.

[267] R. C. Bernardi, M. C. Melo, and K. Schulten. Enhanced sampling techniques in molecular dynamics simulations of biological systems. *Biochim. Biophys. Acta, Gen. Subj.*, 1850(5):872 – 877, 2015.

[268] A. Laio and M. Parrinello. Escaping free-energy minima. *Proc. Natl. Acad. Sci. U.S.A.*, 99(20):12562–12566, sep 2002.

[269] J. Kästner. Umbrella sampling. *Wiley Interdiscip. Rev. Mol. Sci.*, 1(6):932–942, 2011.

[270] J. L. Knight and C. L. Brooks. λ-dynamics free energy simulation methods. *J. Comput. Chem.*, 30(11):1692–1700, 2009.

[271] G. Torrie and J. Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *J. Comput. Phys.*, 23(2):187–199, 1977.

[272] J. I. Steinfeld, J. S. Francisco, and W. L. Hase. *Chemical kinetics and dynamics*. Prentice Hall, 1999.

[273] P. Hänggi, P. Talkner, and M. Borkovec. Reaction-rate theory: fifty years after kramers. *Rev. Mod. Phys.*, 62:251–341, 1990.

[274] B. Peters. Common features of extraordinary rate theories. *J. Phys. Chem. B*, 119(21):6349–6356, 2015.

[275] R. Zhou, B. J. Berne, and R. Germain. The free energy landscape for β hairpin folding in explicit water. *Proc. Natl. Acad. Sci. U.S.A.*, 98(26):14931–14936, 2001.

[276] F. B. Sheinerman and C. L. Brooks. Molecular picture of folding of a small α/β protein. *Proc. Natl. Acad. Sci. U.S.A.*, 95(4):1562–1567, 1998.

[277] J. Gsponer and A. Caflisch. Molecular dynamics simulations of protein folding from the transition state. *Proc. Natl. Acad. Sci. U.S.A.*, 99(10):6719–6724, 2002.

[278] R. R. Coifman, I. G. Kevrekidis, S. Lafon, M. Maggioni, and B. Nadler. Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems. *Multiscale Modeling & Simulation*, 7(2):842–864, 2008.

[279] K. Fukui. Formulation of the reaction coordinate. *J. Chem. Phys.*, 74(23):4161–4163, 1970.

[280] A. Tachibana and K. Fukui. Novel variational principles of chemical reaction. *Theor. Chim. Acta*, 57(1):81–94, 1980.

[281] W. Quapp and D. Heidrich. Analysis of the concept of minimum energy path on the potential energy surface of chemically reacting systems. *Theor. Chim. Acta*, 66(3):245–260, 1984.

[282] K. Yamashita, T. Yamabe, and K. Fukui. IRC approach to chemical dynamics: toward mode-selective chemical reactions. *Chem. Phys. Lett.*, 84(1):123–126, 1981.

[283] R. Olender and R. Elber. Yet another look at the steepest descent path. *J. Mol. Struct. (Theochem.)*, 398:63–71, 1997.

[284] M. Heymann and E. Vanden-Eijnden. The geometric minimum action method: A least action principle on the space of curves. *Commun. Pure Appl. Math.*, 61(8):1052–1117, 2008.

[285] P. Eastman, N. Grønbech-Jensen, and S. Doniach. Simulation of protein folding by reaction path annealing. *J. Chem. Phys.*, 114(8):3823–3841, 2001.

[286] W. E, W. Ren, and E. Vanden-Eijnden. Minimum action method for the study of rare events. *Commun. Pure Appl. Math.*, 57(5):637–656, 2004.

[287] J. Lipfert, J. Franklin, F. Wu, and S. Doniach. Protein misfolding and amyloid formation for the peptide GNNQQNY from yeast prion protein Sup35: Simulation by reaction path annealing. *J. Mol. Biol.*, 349(3):648 – 658, 2005.

[288] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler. Transition path sampling: Throwing ropes over rough mountain passes, in the dark. *Annu. Rev. Phys. Chem.*, 53(1):291–318, 2002.

[289] C. Dellago, P. Bolhuis, and P. L. Geissler. Transition path sampling. *Adv. Chem. Phys.*, 123(1), 2002.

[290] G. A. Natanson, B. C. Garrett, T. N. Truong, T. Joseph, and D. G. Truhlar. The definition of reaction coordinates for reaction-path dynamics. *J. Chem. Phys.*, 94(12):7875–7892, 1991.

[291] G. Hummer. From transition paths to transition states and rate coefficients. *J. Chem. Phys.*, 120(2):516–523, 2004.

[292] J. E. Guyer, D. Wheeler, and J. A. Warren. FiPy: Partial differential equations with Python. *Comput. Sci. Eng.*, 11(3):6–15, 2009.

[293] A. L. Gibbs and F. E. Su. On choosing and bounding probability metrics. *Int. Stat. Rev.*, 70(3):419–435, 2002.

[294] A. Berezhkovskii and A. Szabo. One-dimensional reaction coordinates for diffusive activated rate processes in many dimensions. *J. Chem. Phys.*, 122(1):014503, 2005.

[295] Y. M. Rhee, , and V. S. Pande*. One-dimensional reaction coordinate and the corresponding potential of mean force from commitment probability distribution. *J. Phys. Chem. B*, 109(14):6780–6786, 2005.

[296] A. M. Berezhkovskii and A. Szabo. Diffusion along the splitting/commitment probability reaction coordinate. *J. Phys. Chem. B*, 117(42):13115–13119, 2013.

[297] L. Onsager. Initial recombination of ions. *Phys. Rev.*, 54:554–557, Oct 1938.

[298] R. Du, V. S. Pande, A. Y. Grosberg, T. Tanaka, and E. S. Shakhnovich. On the transition coordinate for protein folding. *J. Chem. Phys.*, 108(1):334–350, 1998.

[299] V. S. Pande, A. Y. Grosberg, T. Tanaka, and D. S. Rokhsar. Pathways for protein folding: is a new view needed? *Curr. Opin. Struct. Biol.*, 8(1):68 – 79, 1998.

[300] T. S. van Erp and P. G. Bolhuis. Elaborating transition interface sampling methods. *J. Comput. Phys.*, 205(1):157 – 181, 2005.

[301] W. E, W. Ren, and E. Vanden-Eijnden. Transition pathways in complex systems: Reaction coordinates, isocommittor surfaces, and transition tubes. *Chem. Phys. Lett.*, 413(1):242–247, 2005.

[302] W. E, W. Ren, and E. Vanden-Eijnden. Finite temperature string method for the study of rare events. *J. Phys. Chem. B*, 109(14):6688–6693, 2005.

[303] R. B. Best and G. Hummer. Reaction coordinates and rates from transition paths. *Proc. Natl. Acad. Sci. U.S.A.*, 102(19):6732–6737, 2005.

[304] A. Ma and A. R. Dinner. Automatic method for identifying reaction coordinates in complex systems. *J. Phys. Chem. B*, 109(14):6769–6779, 2005.

[305] B. Peters and B. L. Trout. Obtaining reaction coordinates by likelihood maximization. *J. Chem. Phys.*, 125(5):054108, 2006.

[306] B. Peters, G. T. Beckham, and B. L. Trout. Extensions to the likelihood maximization approach for finding reaction coordinates. *J. Chem. Phys.*, 127(3):034109, 2007.

[307] E. E. Borrero and F. A. Escobedo. Reaction coordinates and transition pathways of rare events via forward flux sampling. *J. Chem. Phys.*, 127(16):164101, 2007.

[308] B. Peters. Inertial likelihood maximization for reaction coordinates with high transmission coefficients. *Chemical Physics Letters*, 554:248 – 253, 2012.

[309] B. Peters, P. G. Bolhuis, R. G. Mullen, and J.-E. Shea. Reaction coordinates, one-dimensional smoluchowski equations, and a test for dynamical self-consistency. *J. Chem. Phys.*, 138(5):054106, 2013.

[310] B. Peters. $p(tp|q)$ peak maximization: Necessary but not sufficient for reaction coordinate accuracy. *Chem. Phys. Lett.*, 494(13-3):100–103, 2010.

[311] B. Peters. Using the histogram test to quantify reaction coordinate error. *J. Chem. Phys.*, 125(24):241101, 2006.

[312] J. Rogal and P. G. Bolhuis. Multiple state transition path sampling. *J. Chem. Phys.*, 129(22):224107, 2008.

[313] M. Grünwald and C. Dellago. Transition state analysis of solid-solid transformations in nanocrystals. *J. Chem. Phys.*, 131(16):164116, 2009.

[314] S. V. Krivov. Numerical construction of the p-fold (committor) reaction coordinate for a markov process. *J. Phys. Chem. B*, 115(39):11382–11388, 2011.

[315] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Appl. Comput. Harmon. Anal.*, 21(1):113 – 127, 2006.

[316] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[317] C. R. Schwantes and V. S. Pande. Modeling molecular kinetics with tica and the kernel trick. *J. Chem. Theory Comput.*, 11(2):600–608, 2015.

[318] S. B. Kim, C. J. Dsilva, I. G. Kevrekidis, and P. G. Debenedetti. Systematic characterization of protein folding pathways using diffusion maps: Application to trp-cage miniprotein. *J. Chem. Phys.*, 142(8):085101, 2015.

[319] B. N. Parlett. *The symmetric eigenvalue problem*. SIAM, 1980.

[320] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Statistic. Soc. B*, pages 267–288, 1996.

[321] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–451, 2004.

[322] J. A. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Trans. Inf. Theory*, 52(3):1030–1051, 2006.

[323] B. K. Sriperumbudur, D. A. Torres, and G. R. Lanckriet. A majorization-minimization approach to the sparse generalized eigenvalue problem. *Mach. Learn.*, 85(1-2):3–39, 2011.

[324] R. Horst and N. V. Thoai. Dc programming: overview. *J. Optim. Theory Appl.*, 103(1):1–43, 1999.

[325] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, Jan. 2011.

[326] Y. Kiseliov. Algorithms of projection of a point onto an ellipsoid. *Lithuanian Math. J.*, 34(2):141–159, 1994.

[327] L. W. Mackey. Deflation methods for sparse pca. In *Adv. Neural Inf. Process. Syst. 21*, pages 1017–1024, 2009.

[328] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, and D. A. Case. Development and testing of a general amber force field. *J. Comput. Chem.*, 25(9):1157–1174, 2004.

[329] T. W. Anderson. Asymptotic theory for principal component analysis. *Ann. Math. Statist.*, 34(1):122–148, 03 1963.

[330] I. M. Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *Ann. Statist.*, 29(2):295–327, 04 2001.

[331] O. Ledoit and M. Wolf. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *J. Empirical Finance*, 10(5):603–621, 2003.

[332] O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *J. Multivariate Anal.*, 88(2):365–411, 2004.

[333] H. Markowitz. Portfolio selection. *J. Finance*, 7(1):77–91, 1952.

[334] J. Schäfer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Stat. Appl. Genet. Mol. Biol.*, 4(1):1–30, 2005.

[335] Y. Chen, A. Wiesel, and A. O. Hero III. Shrinkage estimation of high dimensional covariance matrices. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009).*, pages 2937–2940. IEEE, 2009.

[336] G. Casella and C. P. Robert. Rao-blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

[337] Y.-H. Dai. Fast algorithms for projection on an ellipsoid. *SIAM J. Optimiz.*, 16(4):986–1006, 2006.

[338] S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *J. Mach. Learn. Res.*, 2016. To appear.

[339] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. Operator splitting for conic optimization via homogeneous self-dual embedding. 2013.

[340] F. Pérez and B. E. Granger. IPython: a System for Interactive Scientific Computing. *Comput. Sci. Eng.*, 9(3):21–29, 2007.

[341] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011.

[342] J. D. Hunter. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.*, 9(3):90–95, 2007.

[343] D. R. Roe and T. E. Cheatham. PTRAJ and CPPTRAJ: Software for Processing and Analysis of Molecular Dynamics Trajectory Data. *J. Chem. Theory Comput.*, 9(7):3084–3095, 2013.

[344] K. Hinsen. The molecular modeling toolkit: A new approach to molecular simulations. *J. Comput. Chem.*, 21(2):79–85, 2000.

[345] N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. Beckstein. MDAnalysis: a toolkit for the analysis of molecular dynamics simulations. *J. Comput. Chem.*, 32(10):2319–2327, 2011.

[346] B. J. Grant, A. P. C. Rodrigues, K. M. ElSawy, J. A. McCammon, and L. S. D. Caves. Bio3d: an R package for the comparative analysis of protein structures. *Bioinformatics*, 22(21):2695–2696, 2006.

[347] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.*, 4(2):187–217, 1983.

[348] J. W. Ponder and F. M. Richards. An efficient Newton-like method for molecular mechanics energy minimization of large molecules. *J. Comput. Chem.*, 8(7):1016–1024, 1987.

[349] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.*, 117(1):1–19, 1995.

[350] J. A. Anderson, C. D. Lorenz, and A. Travesset. General purpose molecular dynamics simulations fully implemented on graphics processing units. *J. Comput. Phys.*, 227(10):5342–5359, 2008.

[351] S. M. Larson, C. D. Snow, M. Shirts, and V. S. Pande. Folding@Home and Genome@Home: Using distributed computing to tackle previously intractable problems in computational biology. In R. Grant, editor, *Computational genomics: Theory and Applications*. Horizon Scientific, 2004.

[352] D. L. Parton, P. B. Grinaway, S. M. Hanson, K. A. Beauchamp, and J. D. Chodera. Ensembler: Enabling high-throughput molecular simulations at the superfamily scale. *bioRxiv*, 2015.

[353] C. Klein. mBuild: a component based molecule builder tool that relies on equivalence relations for component composition. 2014–. Available at `http://imodels.github.io/mbuild/`.

[354] I. S. Haque, K. A. Beauchamp, and V. S. Pande. A fast 3 x N matrix multiply routine for calculation of protein RMSD. *bioRxiv*, 2014.

[355] W. Kabsch and C. Sander. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.

[356] A. Shrake and J. A. Rupley. Environment and exposure to solvent of protein atoms. Lysozyme and insulin. *J. Mol. Biol.*, 79(2):351–71, 1973.

[357] E. N. Baker and R. E. Hubbard. Hydrogen bonding in globular proteins. *Prog. Biophys. Mol. Biol.*, 44(2):97–179, 1984.

[358] B. Vögeli, J. Ying, A. Grishaev, and A. Bax. Limits on variations in protein backbone dynamics from precise measurements of scalar couplings. *J. Am. Chem. Soc.*, 129(30):9377–85, 2007.

[359] M. P. Allen and D. J. Tildesley. Liquid Crystals. In *Computer Simulation of Liquids*, chapter 11.5, pages 300–305. Clarendon Press, 1989.

[360] H. Li, K.-S. Leung, T. Nakane, and M.-H. Wong. iview: an interactive WebGL visualizer for protein-ligand complex. *BMC Bioinf.*, 15(1):56, 2014.

[361] G. Wilson, D. A. Aruliah, C. T. Brown, N. P. Chue Hong, M. Davis, R. T. Guy, S. H. D. Haddock, K. D. Huff, I. M. Mitchell, M. D. Plumbley, B. Waugh, E. P. White, and P. Wilson. Best practices for scientific computing. *PLoS Biol.*, 12(1):e1001745, 2014.

[362] G. Gousios, M. Pinzger, and A. van Deursen. An exploratory study of the pull-based software development model. In *Proceedings of the 36th International Conference on Software Engineering*, ICSE 2014, pages 345–355, New York, NY, USA, 2014. ACM.

[363] C. Burges. *Dimension Reduction: A Guided Tour*. Foundations and trends in machine learning. Now Publishers, 2010.