TOWARDS ROBUST DYNAMICAL MODELS OF BIOMOLECULES

A DISSERTATION
SUBMITTED TO THE 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

Matthew P. Harrigan
September 2017

This dissertation is online at: http://purl.stanford.edu/kb491ws1717

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

Biology is the ultimate emergent phenomenon, and we largely lack a full picture of its function at the smallest scales. Molecular dynamics purports to model biomolecules like proteins with all-atom resolution. Among other challenges, merely analyzing the large quantities of data that result from a simulation has become a bottleneck. In this dissertation, I present my work towards building reduced-complexity models that faithfully capture the relevant functional dynamics of biomolecular simulations. In chapter 1, I introduce a mathematical language for dealing with stochastic processes and show the connection to established modeling methods like Markov modeling and tICA. Chapter 2 develops and characterizes a method for including solvent degrees of freedom in Markov state models. In chapter 3, we apply state-of-the-art MSM modeling to understand multi-scale conformational dynamics of a potassium ion channel. Chapter 4 provides an overview of a curated selection of modeling building blocks accessible through our carefully designed software package. Chapter 5 introduces a new non-linear basis which unites the MSM and tICA approaches. Finally, in chapter 6, I introduce parameterized sets of basis functions and use the variational principle directly to optimize the basis set itself. It is my hope that these novel algorithms aided by well-engineered software implementations and validated by characterization on real biomolecular systems will lead the field closer towards truly robust dynamical models of biomolecules.

Acknowledgments

No amount of text would be sufficient to express my gratitude for the love, support, and mentorship of the great many people who made this work possible. Nevertheless, I would like to give special thanks to my whole family, especially: my parents Mary-Beth and Marty who have always supported me in all my endeavors; my grandparents Mae, Papa, Grandma, and Poppy for their love and wisdom; and my brother Daniel for sharing a bond only brothers can. Thanks to my high school friends, college friends, and grad school friends for great inside jokes and fun memories. Special shout-out to Doris Tang for her friendship, love, and support.

I would like to extend thanks to my collaborators, especially Tim MacKenzie, Veerabahu Shanmugasundaram, and Rajiah Aldrin Denny and all co-authors who contributed directly to this thesis.

The camaraderie and mentorship of lab-mates was instrumental to developing the ideas presented here and elsewhere. I extend my thanks to all members of Pande Group, but especially former grad students Christian Schwantes, Robert McGibbon, TJ Lane, and Jeff Weber; post-docs Gert Kiss, Diwakar Shukla, Nate Stanley, and Joe Gomes; and current graduate students Muneeb Sultan, Carlos Hernandez, Keri McKiernan, Evan Feinberg, Bharath Ramsundar, Brooke Husic, and Arianna Peck.

These words merely scratch the surface of my gratitude for everyone's contributions. Nowhere is this more true than with regards to the mentorship and leadership of Vijay Pande, who turned me from a student into a scientist. The lab environment he created is one of intellectual curiosity, rigor, and freedom. As an advisor, he is second to none.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

"It is very easy to answer many of these fundamental biological questions; you just look at the thing."

- Richard Feynman [1]

"It is an unwritten rule ... that Richard Feynman's famous 1959 lecture 'There's plenty of room at the bottom' should not be referred to at the start of articles unless absolutely necessary."

- Nature Nanotech Editorial [2]

1.1 Overview

Molecular dynamics and the modeling thereof concerns itself with probing functional conformational transitions of biomolecules. Important transitions can include protein folding [3–8], protein conformational change [9–12], and ligand binding [13–18]. Understanding these processes through computation come with major challenges [19–21] which we group into three main categories:

1. **Inaccurate forcefields**. Nature is inherently quantum mechanical, but simulating at this level of detail is not computationally tractable. Researchers have

devoted considerable time and effort developing classical potential energy functions, or forcefields, to approximate the quantum potential energy surface classically. Specific functions may be designed for proteins [22–26], waters [27–29], lipids [30], small molecules [31], or coarse-grained models [32]. These functions may be incorrectly parameterized and thus may fail to reproduce certain experimental properties. More fundamentally, they may miss important physics by restricting themselves to a classical functional form. Polarizable forcefields are being actively developed to increase accuracy [33, 34] at the expense of computational time. This trade-off reappears often with respect to forcefields. Recent work towards automated forcefield parameterization [35, 36] may improve our catalog of potential energy functions.

2. **Inadequate sampling**. Relevant functional conformational changes (folding, activation, . . .) are extremely rare relative to the molecular dynamics timestep. Biological function may happen on timescales of milliseconds to minutes, but fast atomic motions constrain the integration timestep to a small number of femtoseconds (10^{-15} s). Approaches for generating enough sampling have varied from writing high performance simulation codes [37–43] and sophisticated algorithms [44–47] to exploiting distributed infrastructure [48–51] and special purpose hardware [52, 53] including GPUs [54–57].

3. **Analysis**. The problem of analysis has recently been called into focus [19–21]. As the above two challenges are addressed—particularly number two—the amount of data collected in a given computational study of a biomolecule increases dramatically. First, faster hardware and software has led to simulating an individual trajectory for much longer periods of time. At the genesis of the field, simulating for several hundred picoseconds was considered state of the art. With commodity hardware producing hundreds of nanoseconds per day (on benchmark proteins) and bespoke ASICs producing microseconds per day, the protein trajectories have become much longer. Second, the availability of distributed and/or highly-parallel compute resources from university clusters or cloud resources means researchers have the ability to run many trajectories

in parallel. The naïve analysis of "watching a movie" certainly does not scale well to hundreds or thousands of movies. Finally, the sizes of systems has increased from the folding of small peptides to intricate conformational change in large kinases, GPCRs, or ion channels. Identifying by inspection the critical regions of the protein responsible for dynamics or function is becoming ever more impossible. A considerable amount of effort has gone into the analysis problem [19–21, 58]. Of particular note to this dissertation, Markov state models (MSMs) [59–62], variational methods [63–66], and dimensionality reduction methods [67–75] have given rise to sophisticated error analysis [76, 77], hyperparameter optimization [78–80], and feature improvements [81–83] as well as extensions beyond vanilla MSMs including hidden Markov models (HMMs) [84, 85] and rate-based approaches [86, 87].

These three challenges are inexorably linked; improving one area impacts another, both positively and negatively. Some of these links are summarized in fig. 1.1. Improving forcefields can yield more accurate analyses but hinder sampling if complex functional forms are used. Greater sampling can cause problems in forcefields to manifest and overwhelm simple forms of analysis, but can in turn improve forcefields through automated parameterization and give a more converged dataset for analysis. Better analysis can be used for adaptive sampling and fittable observables to improve forcefields.

In this dissertation, I will cover my efforts to (a) further develop algorithms for analysis, (b) apply state of the art modeling to novel systems, and (c) provide high-quality software packages to enable all researchers to take advantage of advances in analysis of molecular dynamics.

1.2 Sampling and Modeling Share a Common Theme

The challenges of sampling and modeling share a common idea: The dimensionality of the system is too high, so we have to be considerate about what manifold we "care" about.

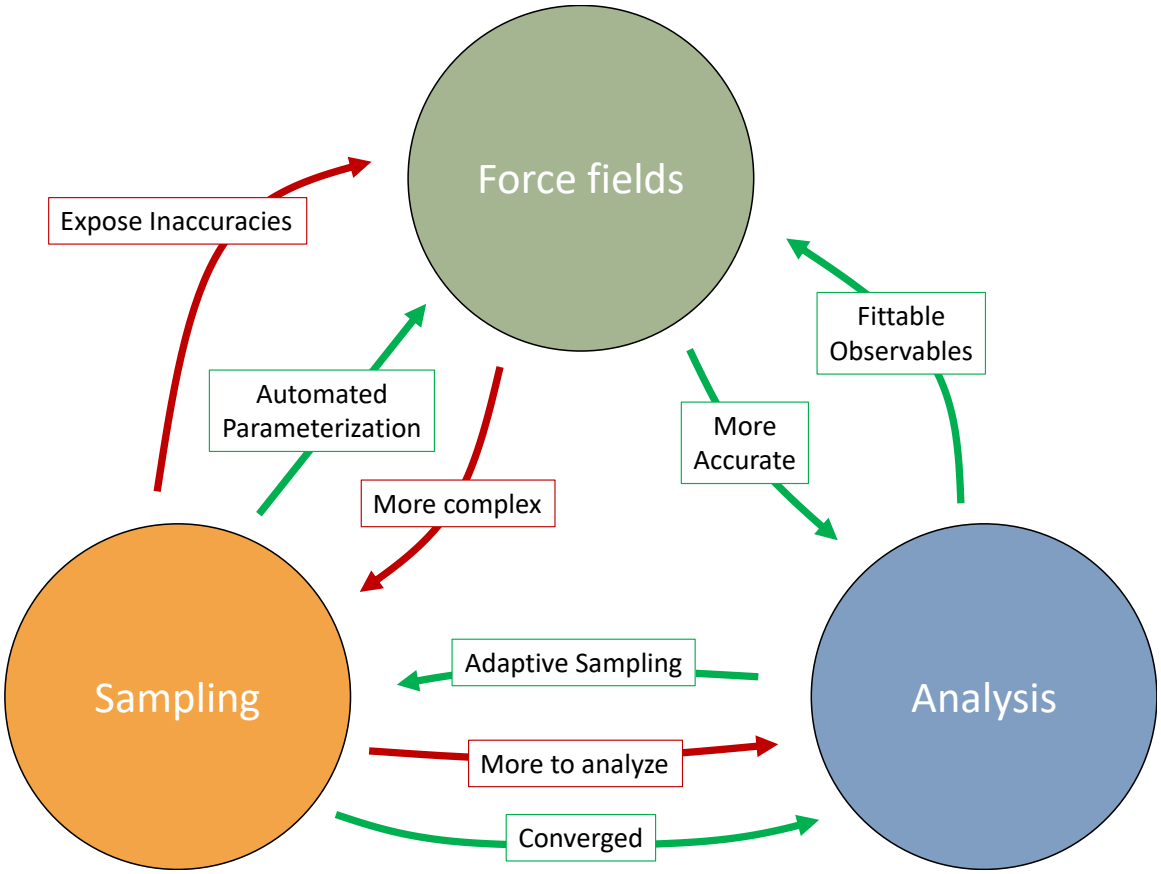

Figure 1.1: Challenges in molecular dynamics can be categorized into those of forcefields, sampling, or analysis. Improvements in one area can negatively (red) or positively (green) impact the other areas.

Consider the Leventhal paradox [88, 89]. For simplification, let's move from the continuous case of positions and velocities and construct a potential energy function (forcefield) as follows: Each protein residue can be in one of three residue states, alpha helix, beta sheet, or loop. The energy of an N-residue protein is a function of the state of each residue. To fully sample the energy landscape, we would need to evaluate the energy function for each protein state (i.e. the length-N vector of per-residue states). The number of evaluations is therefore 3^N. As a back of the envelope calculation, consider a 1000-residue protein. Optimistically, we assume that each energy function evaluation takes one CPU clock cycle and we have 1000 CPUs available, each running at a clock speed of 1 GHz. How long would it take to sample the energy landscape?

$$3^{1000}\text{evaluations} \times \left(\frac{1\text{ cycle}}{\text{evaluation}}\right)\left(\frac{\text{CPU}\cdot\text{second}}{10^9\text{cycles}}\right)\left(\frac{1}{10^3\text{ CPUs}}\right)\left(\frac{1\text{ year}}{3\times 10^7\text{seconds}}\right)$$
$$\approx 10^{477-9-3-7} \approx 10^{447} \times \text{universe age}$$

Leventhal noted that it takes much less time than the age of the universe for biomolecules to find their native conformation (i.e. the folded state of a protein).

An immediate insight is that many of the protein states "make no sense". In our brute-force approach, no prejudice is given against conformations with horrific clashes and correspondingly high energies, even though this directly implies an exponentially low probability of observing that particular state. Already we are developing a notion that the things we "care" about lie on some manifold in this subspace.

Molecular dynamics is a potential energy sampling strategy that takes its inspiration from nature. Instead of exhaustively searching every possible position and momentum quantum, we iteratively draw new positions and momentums by updating the current velocities according to a thermalized version of forces acting on the system. These forces are the negative derivative of the potential energy function. We update the positions according to the new velocities. By construction, this protocol will keep us on (or close to) our sub-manifold of low-energy conformations. This is at the expense of restricting exploration of the space. Namely, the iterative update

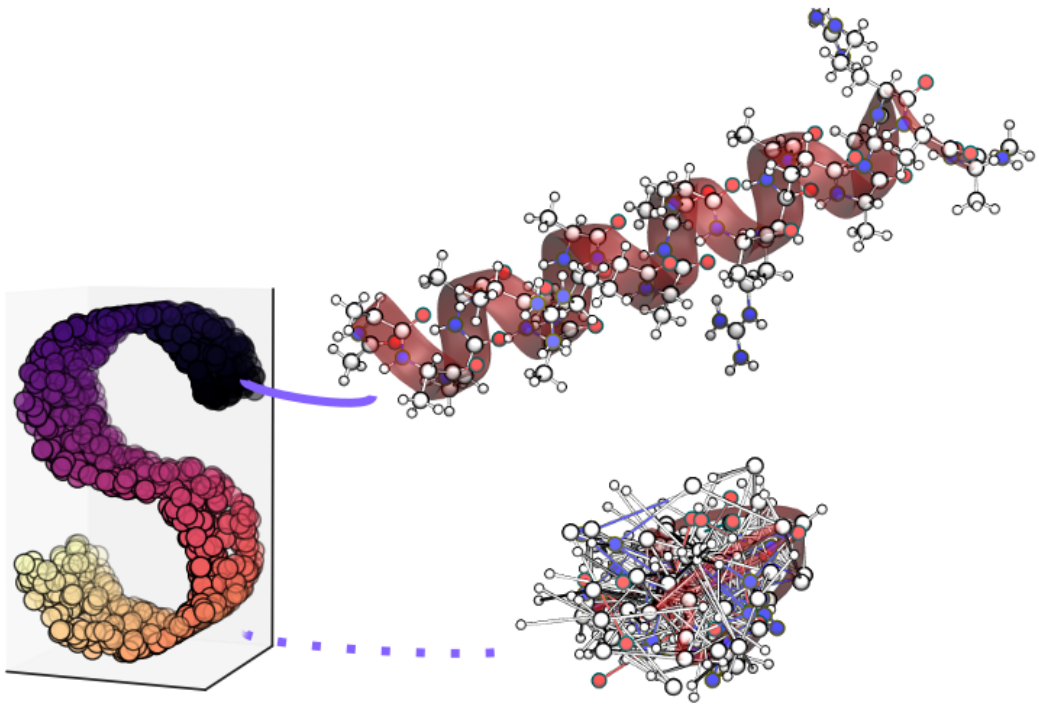

Figure 1.2: By analogy to this "S-curve", relevant protein conformations lie on a manifold much smaller than the full position-momentum space $\mathbb{R}^{6N}$. In this analogy, the folded helical peptide lies on the S whereas points far from the S correspond to "nonsense" conformations with high-energy clashes.

procedure will select with high probability new values that are close to the previous values for positions (and velocities).

With a sufficiently long simulation or sufficiently many simulation trajectories, the researcher is afforded a set of conformations drawn from the equilibrium distribution of the particular thermodynamic ensemble that was constructed. However, the sampled conformations are each still in a high dimensional space of all positions and velocities. In the *sampling* problem, we produced a set of low energy conformations drawn from some manifold much smaller than the full phase space. In the *analysis* problem, we seek to construct an explicit and physically interpretable representation of a manifold onto which we can project our simulations. Ideally, this projection should preserve all salient features of the biophysical system while discarding as many irrelevant features as possible. As an example, for a collection of folding trajectories the RMSD distance to the native structure can be used as a 1D projection which may preserve all the salient features of folding. However, hand picked or otherwise arbitrary projections can be insidious: a projection can easily hide important degrees of freedom and lull the researcher into a false sense of knowedness. In the following chapters, I present my work to develop relevant, unbiased, and automated projections for use in the analysis of biomolecule kinetics. In the following section, I introduce the transfer operator formalism on which most of the following work is based.

1.3 Stochastic Dynamics

Classical molecular dynamics simulations propagate atomic positions and velocities forwards in time according to Newton's equations of motion, usually subject to thermal noise. Following the derivation in ref. [90] and [64], our simulations yield a discrete-time stochastic process X_t (taking t to be an index 1, 2, ...) through a space Ω such that $X_t \in \Omega$. A particular X_t might be the positions and velocities of a protein, and Ω may be $\mathbb{R}^{6N}$.

1.3.1 Markovianity

A particular feature of classical physics (and even more complicated descriptions of nature like quantum mechanics) is that the distribution of future states of the system only depend on a complete description of the current state of the system. Concretely, if we know all atoms' positions and velocities at time t—as well as the potential energy function $U(x,t)$ and type of dynamics—we can confidently describe the new distribution of positions and velocities at a later time $t+1$. A system that lacks a dependence on past states of the system in this way is called **Markovian**

$$\mathbb{P}\left(X_t, T_t | X_1, T_1; X_2, T_2; \ldots; X_{t-1}, T_{t-1}\right) = \mathbb{P}\left(X_t, T_t | X_{t-1}, T_{t-1}\right)$$

For typical molecular dynamics simulations, the potential energy function (force-field) and method of integration is fixed for the duration of the simulation. Following this realization, we'll restrict ourselves to time-homogeneous dynamics where we drop the time variables

$$\mathbb{P}\left(X_t | X_1; X_2; \ldots; X_{t-1}\right) = \mathbb{P}\left(X_t | X_{t-1}\right)$$

Following the notation of ref. [90], we introduce a shorthand for describing the transition kernel

$$p\left(x, y\right) = \mathbb{P}\left(X_t = y | X_{t-1} = x\right)$$

which is the probability of moving from x to y in one time increment.

1.3.2 Ergodicity

In addition to Markovianity, we make some more assumptions about our dynamics. First, we assume that the dynamics are **ergodic**. In infinite time, every state will be visited infinitely many times. This time-connectivity of our states implies there is a unique stationary distribution of our dynamics, $\mu(x)$. This distribution describes the proportion of time spent in a state x in an infinitely long trajectory. For constant temperature NVT ensembles, this is the familiar Boltzmann distribution.

$$\mu(x) = \frac{1}{Z} e^{-\beta H(x)}$$
$$Z = \sum_x e^{-\beta H(x)} \tag{1.1}$$

with Hamiltonian $H(x)$ and inverse temperature $\beta = (k_b T)^{-1}$.

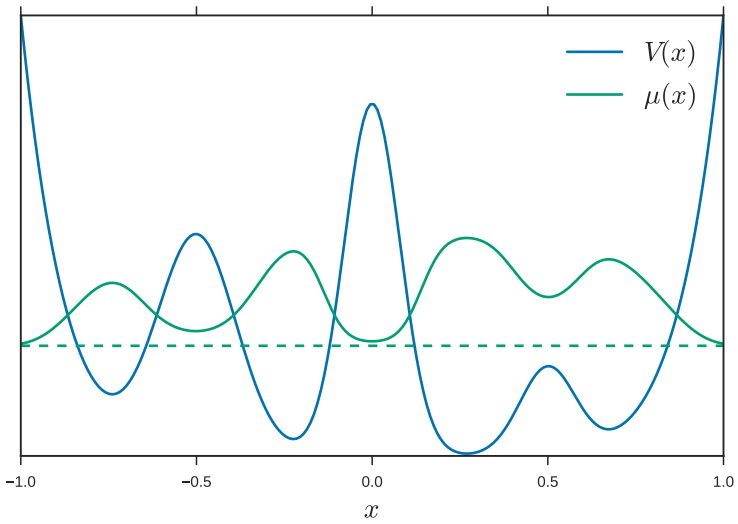

Figure 1.3: For illustration, we introduce a four-well, one-dimensional potential $V(x)$. Brownian dynamics on this potential are ergodic and Markovian. Therefore, we can compute a unique stationary distribution μ.

1.3.3 Reversibility

Finally, we assume our dynamics are **reversible** in time. This assumption is once again valid for equilibrium dynamics such as those simulated by conventional molecular dynamics. Mathematically, the concept of reversibility says a process is equally likely to happen in the forward direction as the reverse direction.

$$\mu(x)\, p(x, y) = \mu(y)\, p(y, x) \tag{1.2}$$

The factors of μ encode the probability of being in the starting state, and the factors of $p(\cdot,\ \cdot)$ are the probability of making the transition. It is important to note

that reversibility does not imply $p(x, y) = p(y, x)$.

1.3.4 Markovianity of Projections

Newtonian dynamics are Markovian if a full specification of the positions and velocities is given. Usually, we do not carry this full specification into the modeling regime. The modeler will drop velocities and save positions with finite precision. In addition, modelers may remove hydrogens or waters (see chapter 2). As soon as these "projections" are applied, the system is no longer technically Markovian. But it is the case that at a sufficiently coarse spatial or temporal resolution, the dynamics *appear* Markovian to good approximation.

1.4 The Propagator

In our thermal, stochastic process, even a small step forward in time will result in a proliferation of potential outcomes. We therefore move from the consideration of a single processes moving through time to an ensemble of processes described by a probability distribution $p_t(x)$ that is propagated through time. As expected, this probability distribution is normalized over our state space Ω.

$$\int_{\Omega} p_t(x)\,\mathrm{d}x = 1 \tag{1.3}$$

We can use the transition kernel to move forwards in time.

$$p_{t+1}(x) = \int_{\Omega} p_t(y) \cdot p(y, x)\,\mathrm{d}y \tag{1.4}$$

This operator is known as the propagator and is defined such that

$$p_{t+1} = \mathcal{Q} \circ p_t \tag{1.5}$$

Following our assumption of reversibility, we can show that the propagator is self-adjoint with respect to a particular inner product

$$\langle f|\mathcal{Q}\circ g\rangle_{?} = \langle Q\circ f|g\rangle_{?} \tag{1.6}$$

Where $\langle\cdot|\cdot\rangle_{?}$ Is an as-yet unspecified inner product. Self-adjoint operators have real eigenvectors and eigenvalues and a complete orthonormal basis set, which are useful for interpreting model timescales and processes. For this particular operator, the inner product we're interested in is with respect to the equilibrium distribution

$$\langle f|g\rangle_{\mu^{-1}} = \int_{\Omega} \frac{f(x)\cdot g(x)}{\mu(x)}\,\mathrm{d}x \tag{1.7}$$

We check the self-adjoint condition by showing the equivalence of the the left- and right-hand side of eq. (1.6) using the norm given in eq. (1.7)

$$\begin{aligned}\langle f|\mathcal{Q}\circ g\rangle_{\mu^{-1}} &= \left\langle f\middle|\int_{\Omega} g(x)p(x,y)\,\mathrm{d}x\right\rangle_{\mu^{-1}}\\ &= \int_{\Omega}\frac{f(y)\cdot\left(\int_{\Omega} g(x)p(x,y)\,\mathrm{d}x\right)}{\mu(y)}\,\mathrm{d}y\\ &= \int_{\Omega\times\Omega}\frac{f(y)g(x)p(x,y)}{\mu(y)}\,\mathrm{d}x\mathrm{d}y\end{aligned}$$

And similarly for the other side of the equation:

$$\begin{aligned}\langle\mathcal{Q}\circ f|g\rangle_{\mu^{-1}} &= \left\langle\int_{\Omega} f(y)p(y,x)\,\mathrm{d}y\middle|g\right\rangle_{\mu^{-1}}\\ &= \int_{\Omega\times\Omega}\frac{f(y)g(x)p(y,x)}{\mu(x)}\,\mathrm{d}x\,\mathrm{d}y\end{aligned}$$

Using the reversibility relationship, we can perform the "flip" from $p(x,y)$ to $p(y,x)$

$$\frac{p(x,y)}{\mu(y)} = \frac{p(y,x)}{\mu(x)} \tag{1.8}$$

1.5 The Transfer Operator

Instead of dealing with the μ-weighted inner product and the propagator operator, we can instead work with a pre-processed distribution

$$u_t(x) = \frac{p_t(x)}{\mu(x)}$$

And move this new quantity forwards in time

$$u_{t+1} = \mathcal{T} \circ u_t \tag{1.9}$$

Where we simply scale u_t to p_t, apply the propagator, and reverse the μ-scaling.

$$\mathcal{T} \circ u_t = \mu^{-1} [\mathcal{Q} \circ (\mu \cdot u_t)] \tag{1.10}$$

$$\mu \cdot \mathcal{T} \circ u_t = [\mathcal{Q} \circ p_t] \tag{1.11}$$

We name this the transfer operator and will use its nicer mathematical properties later. Esp. the second form Figure 1.4 shows the difference between propagator and transfer operator eigenfunctions.

The transfer operator is also self-adjoint, but with respect to a different inner product.

$$\langle \mathcal{T} \circ f | g \rangle_\mu = \langle f | \mathcal{T} \circ g \rangle_\mu \tag{1.12}$$

To prove this, we switch the inner product from μ-weighted to μ^{-1}-weighted where we can exploit the previously proved self-adjointness of $\mathcal{Q}$.

$$\begin{aligned} \langle \mathcal{T} \circ f | g \rangle_\mu &= \langle \mu^{-1} [\mathcal{Q} \circ (\mu \cdot f)] | g \rangle_\mu \\ &= \langle \mathcal{Q} \circ (\mu \cdot f) | \mu \cdot g \rangle_{\mu^{-1}} \\ &= \langle \mu \cdot f | \mathcal{Q} \circ (\mu \cdot g) \rangle_{\mu^{-1}} \\ &= \langle f | \mu^{-1} [\mathcal{Q} \circ (\mu \cdot g)] \rangle_\mu \\ &= \langle f | \mathcal{T} \circ g \rangle_\mu \end{aligned} \tag{1.13}$$

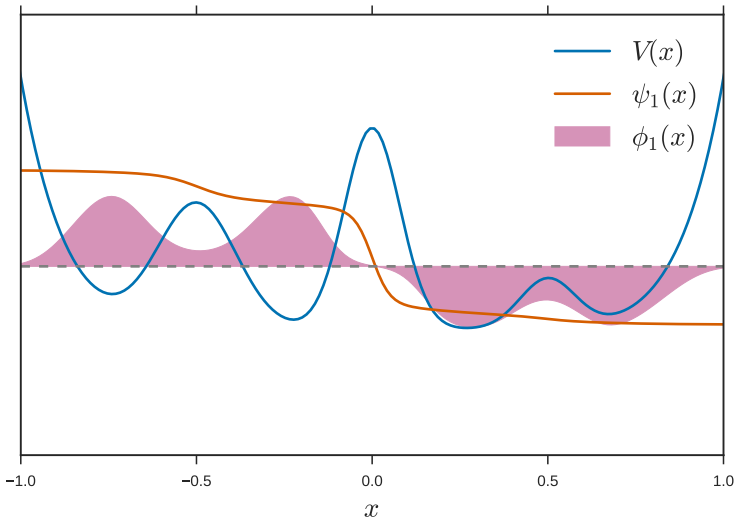

Figure 1.4: Equilibrium dynamics on a potential $V(x)$ can be described in terms of the propagator eq. (1.5) or equivalently by the transfer operator eq. (1.9). The distributions differ only by a factor of $\mu(x)$.

1.6 Dynamical Models

The propagator—or equivalently the transfer operator—describes all the essential dynamics of a system. We have described its operation in terms of molecular dynamics on a classical potential energy surface, but that doesn't give us any additional insights. Instead, we would like to build a model of a *simpler* propagator that describes the dynamics of interest. Specifically, we're interested in identifying a hierarchy of long-lived metastable states. Proteins live on complex energy landscapes with many metastable states accessed through diverse timescales [91]. Folding, conformational change, ligand binding, et al. (see section 1.1) are all *slow* (microseconds to seconds) relative to the molecular dynamics integration timestep (femtoseconds). Because the propagator is self-adjoint (see eq. (1.6)), it has a complete basis of orthonormal eigenfunctions.

$$\begin{aligned} \mathcal{Q} \circ \phi_i &= \lambda_i \phi_i \\ \langle \phi_i | \phi_j \rangle_{\mu^{-1}} &= \delta_{\mathrm{ij}} \end{aligned} \tag{1.14}$$

The simple proof is as follows:

$$\begin{aligned}\left\langle \phi_i \middle| \mathcal{Q} \circ \phi_j \right\rangle_{\mu^{-1}} &= \lambda_j \left\langle \phi_i | \phi_j \right\rangle_{\mu^{-1}} \\ \left\langle \mathcal{Q} \circ \phi_i | \phi_j \right\rangle_{\mu^{-1}} &= \lambda_i \left\langle \phi_i | \phi_j \right\rangle_{\mu^{-1}} \\ \Rightarrow (\lambda_i - \lambda_j) \left\langle \phi_i | \phi_j \right\rangle_{\mu^{-1}} &= 0\end{aligned}$$

One eigenfunction is trivially the equilibrium distribution $\mu(x)$.

$$\mathcal{Q} \circ \mu = \int_\Omega \mu(x) p(x, y) \, \mathrm{d}x \tag{1.15}$$

We use our reversibility relationship to pull the μ outside of the integral

$$\begin{aligned}&= \int_\Omega \mu(y) p(y, x) \, \mathrm{d}x \\ &= \mu(y) \int_\Omega p(y, x) \, \mathrm{d}x \\ &= \mu(y) \cdot 1\end{aligned} \tag{1.16}$$

Therefore, the equilibrium distribution is the eigenfunction of the propagator with eigenvalue 1. Relatedly, the transfer operator also has an eigenfunction with eigenvalue 1 and it is the constant function.

$$\psi_1(x) = \frac{\phi_1(x)}{\mu(x)} = \mathbf{1} \tag{1.17}$$

Due to the Perron-Frobenius theorem, all eigenvalues must be between -1 and 1. We will see that the remainder of the eigenvalues less than one correspond to eigenfunctions that cause the distribution to decay towards the equilibrium distribution. Without loss of generality, we'll sort the eigenvalues (and associated eigenfunctions) in descending order.

$$1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \dots \tag{1.18}$$

We note that we can decompose any distribution $p(x)$ into the eigenbasis of the propagator

$$p(x) = \sum_{i=1}^{\infty} a_i \phi_i(x) \tag{1.19}$$

with numerical coefficients $\{a_i\}$. We can determine these coefficients by projecting onto the eigenbasis.

$$\langle p|\phi_j\rangle_{\mu_{-1}} = \left\langle \sum_{i=1}^{\infty} a_i \phi_i \middle| \phi_j \right\rangle_{\mu^{-1}} = a_j \tag{1.20}$$

We can apply the propagator to the decomposed form of an initial distribution p.

$$\begin{aligned} \mathcal{Q} \circ p &= \mathcal{Q} \circ \sum_{i=1}^{\infty} a_i \phi_i(x) \\ &= \sum_{i=1}^{\infty} a_i \mathcal{Q} \circ \phi_i(x) \\ &= \sum_{i=1}^{\infty} a_i \lambda_i \phi_i(x) \end{aligned} \tag{1.21}$$

1.7 Timescales

With the above machinery in place, let's consider an initial distribution that is close to the equilibrium distribution.

$$p_0(x) = \mu(x) + \phi_j(x) \tag{1.22}$$

If we apply the propagator t times to move p_0 forward in time to p_t, the equilibrium term is unaffected ($\lambda_1 = 1$) and the ϕ_j term is multiplied by λ_j a total of t times.

$$p_t(x) = \mu(x) + \lambda_j^t \phi(x) \tag{1.23}$$

By introducing a suggestive new variable τ_j such that

$$\lambda_j = e^{-1/\tau_j} \tag{1.24}$$

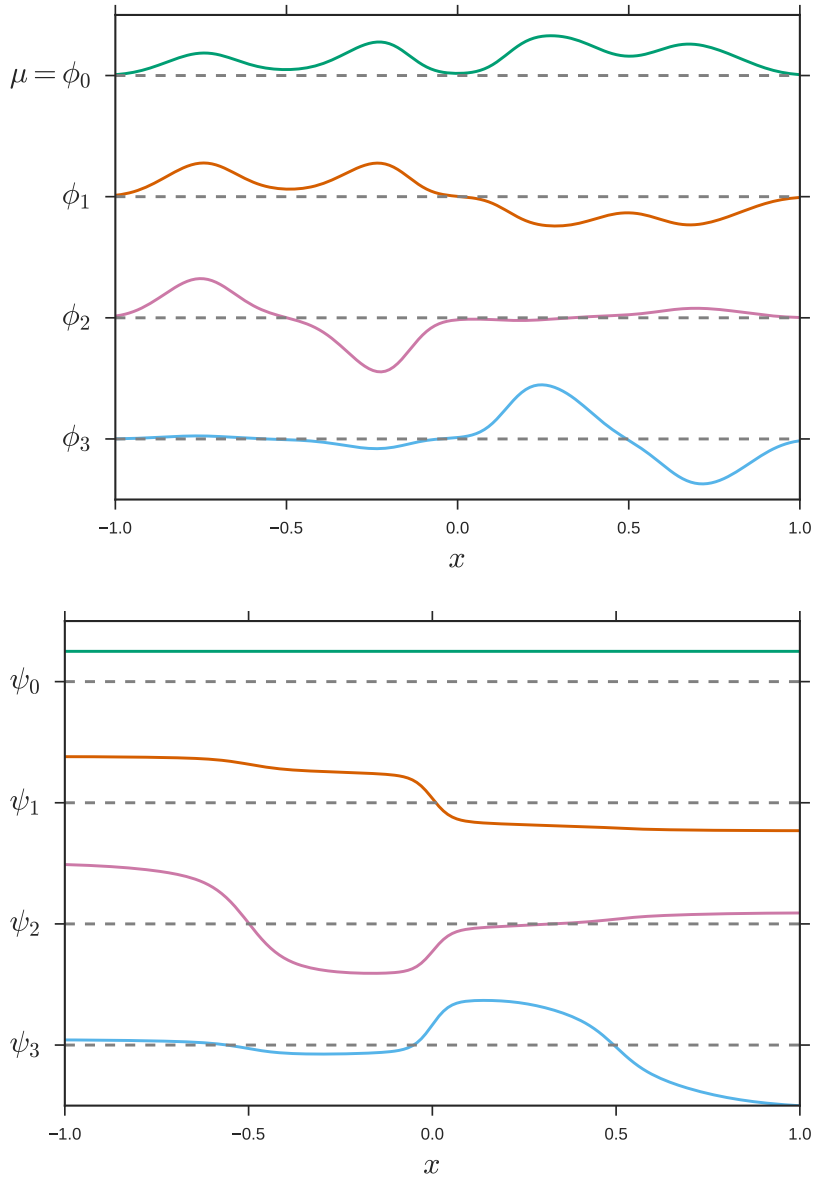

Figure 1.5: Eigenfunctions of the propagator (top, ϕ_i) and transfer operator (bottom, ψ_i) for the one dimensional potential introduced in fig. 1.3. This potential has four wells, with the highest barrier being between the left two and right two wells. Therefore, we observe three slow processes ($i = \{1, 2, 3\}$) in addition to the equilibrium distribution ($i = 0$). The first process ($i = 1$) corresponds to a transfer of flux from the left two and right two wells. The next two processes ($i = \{2, 3\}$) correspond to transitions *within* the left or right two wells. The propagator eigenfunctions differ from those of the transfer operator by a factor of the equilibrium distribution μ.

we can re-write eq. (1.23) as

$$p_t(x) = \mu(x) + e^{-t/\tau_j}\phi(x) \tag{1.25}$$

and immediately see that all eigenprocesses other than the equilibrium distribution decay exponentially (towards equilibrium) with time constant $\tau_j = -\log^{-1}(\lambda_j)$. Because we can decompose *any* distribution into the eigenbasis of the propagator (eq. (1.19)), this simple example generalizes such that *any* initial distribution will experience a multi-exponential decay towards equilibrium.

1.8 Variational theorem

Suppose we have a signal or observable that reports on our dynamical process

$$f(X_t) : \Omega \to \mathbb{R}$$

and we would like to measure how slowly the signal varies in time. This is given by the autocorrelation function. For simplicity, we mean-subtract and scale the signal data to have unit variance. The autocorrelation function is then

$$\text{acf}(f) = \mathbb{E}\left[f(x_t) \cdot f(x_{t+1})\right] \tag{1.26}$$

which can be computed from the propagator

$$\begin{aligned}\mathbb{E}\left[f(x_t) \cdot f(x_{t+1})\right] &= \int_{\Omega\times\Omega} \left[f(x)\mu(x)\ \mathrm{d}x\right]\ \left[f(y)p(x,y)\ \mathrm{d}y\right] \\ &= \int_{\Omega\times\Omega} \left[f(x)\mu(x)\ \mathrm{d}x\right]\ \left[f(y)\frac{\mu(y)p(y,x)}{\mu(x)}\ \mathrm{d}y\right]\end{aligned} \tag{1.27}$$

where on the second line we use the reversibility relationship eq. (1.2). Using eq. (1.5), we can express this with the propagator and following eq. (1.9), we can substitute in the transfer operator. Finally we can use the special inner product $\langle\cdot|\cdot\rangle_\mu$ to absorb the factor of μ

$$\begin{aligned}\mathbb{E} &= \langle \mu \cdot f | \mu^{-1} \left[\mathcal{Q} \circ (\mu \cdot f) \right] \rangle \\ &= \langle \mu \cdot f | \mathcal{T} \circ f \rangle \\ &= \langle f | \mathcal{T} \circ f \rangle_{\mu}\end{aligned} \tag{1.28}$$

If we expand f in terms of the transfer operator eigenfunctions

$$f = \sum_i a_i \psi_i$$

then—by taking into account the operation of $\mathcal{T}$ on its eigenfunctions and their orthonormality—the autocorrelation value is

$$\text{acf} = \sum_i a_i^2 \lambda_i$$

But we assumed the signal f was scaled to unit variance, so we can derive a constraint on the $\{a_i\}$.

$$\begin{aligned}1 &= \text{Var}[f(x_t)] \\ &= \int_{\Omega} \mu(x) f(x)^2 \, \mathrm{d}x \\ &= \langle f | f \rangle_{\mu} \\ &= \sum_i a_i^2\end{aligned} \tag{1.29}$$

again using the orthonormality of the eigendecomposition.

Our other assumption was that the data had its mean subtracted. From eq. (1.17) we know that the first eigenfunction is the constant function $\psi_1(x) = 1$. Therefore, we can safely assume $a_1 = 0$. Taken together, we can derive an upper bound on the autocorrelation function

$$\begin{aligned} \text{acf} = \mathbb{E}[f_t \cdot f_{t+1}] &= \sum_{i=2}^{\infty} \lambda_i a_i^2 \\ &\leq \sum_{i=2}^{\infty} \lambda_2 a_i^2 \\ &\leq \lambda_2 \left(\sum_{i=2}^{\infty} a_i^2 \right) \\ &\leq \lambda_2 \end{aligned} \tag{1.30}$$

This variational bound underpins the construction and evaluation of dynamical models. In particular, tICA and MSMs each try to maximize the autocorrelation of a signal for a certain choice of basis ansatz functions. Assume we have a collection of basis functions $\{\chi_i\}$ and coefficients such that

$$f(x) = \sum_i a_i \chi_i(x) \tag{1.31}$$

Then we want to find the set of coefficients to maximize

$$\begin{aligned} \mathbb{E}[f_t \cdot f_{t+1}] &= \langle f | \mathcal{T} | f \rangle_\mu \\ &= \sum_{ij} a_i a_j \langle \chi_i | \mathcal{T} | \chi_j \rangle_\mu \\ &= a^T \mathbf{C}^{(\tau)} a \end{aligned} \tag{1.32}$$

subject to the constraint that

$$\begin{aligned} 1 = \text{Var}[f] &= \langle f | f \rangle_\mu \\ &= \sum_{ij} a_i a_j \langle \chi_i | \chi_j \rangle \\ &= a^T \mathbf{C}^{(0)} a \end{aligned} \tag{1.33}$$

where we have introduced the two matrices $\mathbf{C}^{(0)}$ and $\mathbf{C}^{(\tau)}$ whose elements are given by $\langle \chi_i | \chi_j \rangle_\mu$ and $\langle \chi_i | \mathcal{T} | \chi_j \rangle_\mu$ respectively. In practice, we can't operate the transfer operator $\mathcal{T}$ on the basis functions directly. However, by sampling sufficiently many tuples of transitions from molecular dynamics simulation trajectories, we can estimate

$\langle \chi_i | \mathcal{T} | \chi_j \rangle$ by projecting the real ("all atom") trajectories onto the basis functions and evaluating the average product across all transition tuples. $\mathbf{C}^{(0)}$ can similarly be estimated from the sample covariance of the basis-projected trajectories.

Putting together eq. (1.32) and eq. (1.33), we aim to optimize the generalized Rayleigh quotient given by

$$\frac{a^T \mathbf{C}^{(\tau)} a}{a^T \mathbf{C}^{(0)} a} \tag{1.34}$$

which is achieved by choosing the eigenvector corresponding to λ_2 of the generalized eigenproblem

$$\mathbf{C}^{(\tau)} a = \mathbf{C}^{(0)} \lambda a \tag{1.35}$$

McGibbon and Pande [79] expand this optimization to find the optimal rank-m decomposition. Although the full mathematical details are beyond the scope of this introductory chapter, we present the salient results. Instead of a signal f, we consider a collection of m signals $\{f_n\}_{n=1}^m$ such that

$$f_n(x) = \sum_i a_{in} \chi_i(x) \tag{1.36}$$

In comparison with eq. (1.31), we now have a *matrix* of coefficients $\mathbf{A}$ in place of the vector a. The variational principle is generalized from the autocorrelation value

$$\langle f | \mathcal{T} \circ f \rangle \leq \lambda_2 \tag{1.37}$$

to the trace of a normalized autocorrelation matrix

$$\mathrm{Tr}\left(\langle f_l | \mathcal{T} \circ f_n \rangle \left(\langle f_l | f_n \rangle \right)^{-1} \right) \leq \sum_{i=2}^{m} \lambda_i \tag{1.38}$$

which is optimized by the generalized *matrix* Rayleigh quotient

$$\frac{\mathbf{A}^T \mathbf{C}^{(\tau)} \mathbf{A}}{\mathbf{A}^T \mathbf{C}^{(0)} \mathbf{A}} \tag{1.39}$$

1.8.1 tICA

In tICA, the basis functions are linear functions in the input coordinates. If we represent a conformation by a vector of internal coordinates (such as dihedral angles) $\mathbf{x} = \{x_1, x_2, \dots\}$ then $\chi_i = x_i$

1.8.2 MSMs

Markov state models (MSMs) use indicator basis functions. Given a complete partitioning of the space Ω such that $\cup_i S_i = \Omega$ and $S_i \cap S_j = \delta_{ij}$, the basis functions are

$$\chi_i(x) = \begin{cases} 1 & x \in S_i \\ 0 & x \notin S_i \end{cases} \tag{1.40}$$

1.8.3 Other

The variational principle and generalized matrix quotient applies to any set of basis functions. Linear (as in tICA) or state-based (as in MSMs) have been used successfully to build dynamical models of biomolecules. Other choices of basis functions may prove to match the intrinsic dynamics of a particular simulation better. In this thesis, I investigate this idea further.

Chapter 2

A Method for the Analysis of Solvent in Molecular Dynamics

This chapter is adapted with permission from Matthew P. Harrigan, Diwakar Shukla, and Vijay S. Pande. Conserve water: A method for the analysis of solvent in molecular dynamics. *J. Chem. Theory Comput.*, 11(3):1094–1101, 2015. doi: 10.1021/ct5010017 [82].

Abstract

Molecular dynamics with explicit solvent is favored for its ability to more correctly simulate aqueous biological processes and has become routine thanks to increasingly powerful computational resources. However, analysis techniques including Markov state models (MSMs) ignore solvent atoms and focus solely on solute coordinates despite solvent being implicated in myriad biological phenomena. We present a unified framework called the "solvent-shells featurization" for including solvent degrees of freedom in analysis, and show that this method produces better models. We apply this method to simulations of dewetting in the two-domain protein BphC to generate a predictive MSM and identify functional water molecules. Furthermore, the proposed methodology could be easily extended for building MSMs of any systems with indistinguishable components.

2.1 Introduction

Changes in conformations of proteins and nucleic acids underlie the majority of emergent biological phenomena in daily life. Life, death, and disease are the result of molecules changing shape in dynamical processes such as protein folding, kinase activation, and signaling [10, 92, 93]. Understanding these dynamical processes is fundamental to our understanding of biology. Experimental probes such as X-ray crystallography and NMR can provide static pictures of macromolecules, and certain specialized methods can give limited information about dynamics [93]. For systems ill-suited to experimental characterization, molecular dynamics (MD) offers unparalleled atom-level detail of the dynamics of microscopic systems. Recent advances in computing including the use of GPUs [41, 57], specialized hardware [52], and distributed computing [13, 48, 51] have enabled simulations to probe biologically-relevant macromolecules at biologically-relevant timescales. Additionally, increasing computational power has enabled simulations to probe molecules in biologically-relevant solvent environments: explicit representation of water molecules [29] and lipid membranes [30] has become routine. With simulation times reaching milliseconds and the number of

atoms approaching hundreds of thousands, some sort of dimensionality reduction is needed to make sense of this huge amount of data [19, 20].

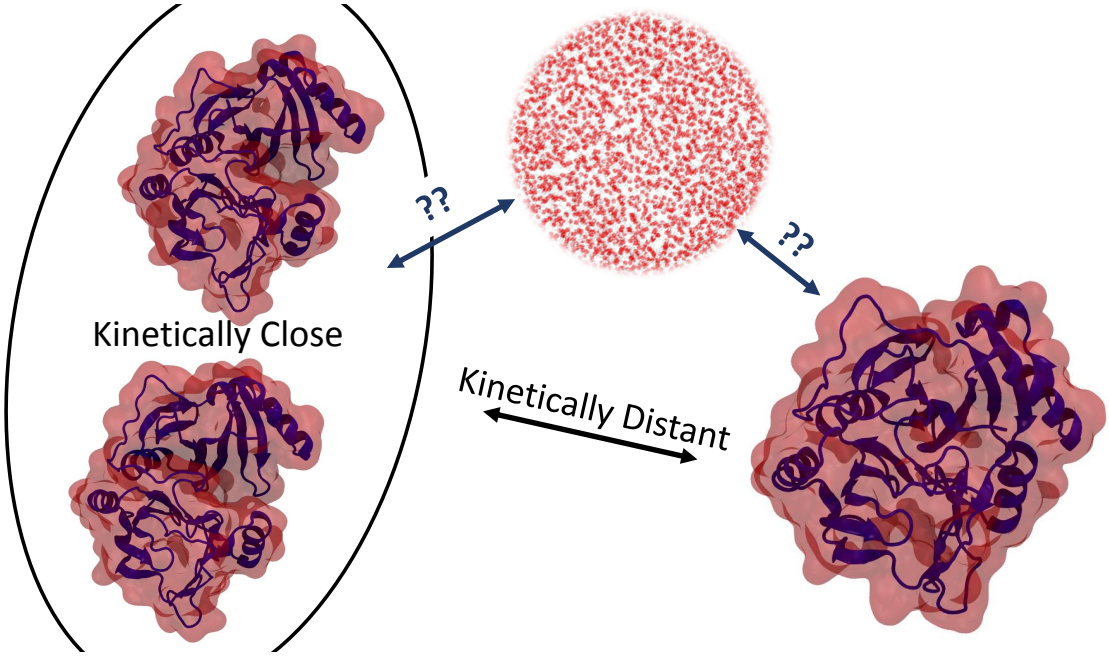

Figure 2.1: Building an MSM requires a distance metric to cluster kinetically-close conformations. The enzyme BphC is depicted. Left: two extended conformations with a "wet" interfacial cavity. We estimate that these similar structures interconvert rapidly. Right: The collapsed, dewetted structure. This is kinetically distant from the extended conformations. Top: A sample of the water box solvating BphC. We lack a method for estimating kinetic distance for solvent degrees of freedom due to the large number and indistinguishability of solvent molecules.

One dimensionality reduction technique involves construction of a Markov state model (MSM) from the time-series of atomic coordinates from molecular dynamics [94]. MSMs parameterize a system by a set of states and rates. Snapshots from MD trajectories are grouped or clustered into k states. Some information is necessarily lost by lumping conformations, but with a sufficiently fine partitioning, we can resolve states with sufficient detail [64]. The ideal clustering for MSM construction groups conformations that interconvert rapidly (fig. 2.1). Lacking an *a priori* measure of the interconversion rate of two given conformations (e.g. two frames of an MD trajectory),

estimation of this kinetic closeness is approximated by a conformational distance metric. For example, root-mean-square deviation (RMSD) or euclidean distance between features such as dihedral angles or contact distances have been used for state definition [60]. With the states defined and MD snapshots assigned to their proper states, we estimate k^2 transition probabilities among the k states. "Markovianity" means we idealize the dynamics of the biological system as a set of memoryless jumps between states. MSMs have been used successfully to reveal important biological structure and function in diverse biological applications such as folding [49, 95–97], kinase and protease activation [98, 99], GPCR signaling [13], protein-ligand binding [14], and self-assembly [100].

A second, routine "dimensionality reduction" has been to discard all solvent atoms prior to analysis [4, 8, 96], despite water and membranes being crucial to protein function and biology as a whole. In fact, solvent has been implicated in hydrophobic collapse [101], protein stability [102], protein-ligand binding [16], modulation of ion channel function [103], antifreeze proteins [104], and aggregation [105, 106]. There is understandable reason to discard solvent atoms. First, the number of atoms is large: there are often 50 times as many water molecules as protein residues (and 10,000 times as many water molecules as protein *molecules*) in an explicit-solvent MD simulation box. Second, solvent molecules—unlike proteins or nucleic acids—are indistinguishable. It does not matter if water #1525 or #19832 is solvating a particular side-chain; conformations in which these solvent atoms are exchanged should be treated as identical. Methods tuned for analysis of solutes are ill-suited to considering both the indistinguishability and large number of solvent molecules in a typical simulation. Markov state models are no exception: specifically, we lack an estimation of kinetic closeness in solvent degrees of freedom. Traditional treatments of solvent rely on aligning solute conformations and laying down a grid of voxels for which properties like density can be calculated and visualized [14, 15]. These methods (1) fail for large conformational changes or folding when alignment of the solute is poor, and (2) fail to interface with statistical tools (e.g. principal component analysis (PCA), MSMs) due to an over-abundance of resultant features. If we consider a cubic simulation box of side-length 80 Å and voxels of side-length 3 Å, grid-based approaches

would yield nineteen thousand features. Other work has focused on grouping solvent conformations based on truncated hydrogen bonding networks [107]. This approach fails for general indistinguishable particles other than H_2O for which bonding criteria is not known or not relevant. This method does not afford a distance metric to relate conformations and, as such, requires one state per enumerated hydrogen bonding network ($k = 50,000$ when only considering the first and second solvation shells). The lack of an Euclidean distance metric once again hinders interface with statistical tools like PCA or K-means clustering. A suitable transformation of solvent positions into tractable features ("featurization") like those available for solutes would yield a solvent distance metric that could be used during the clustering stage of MSM construction.

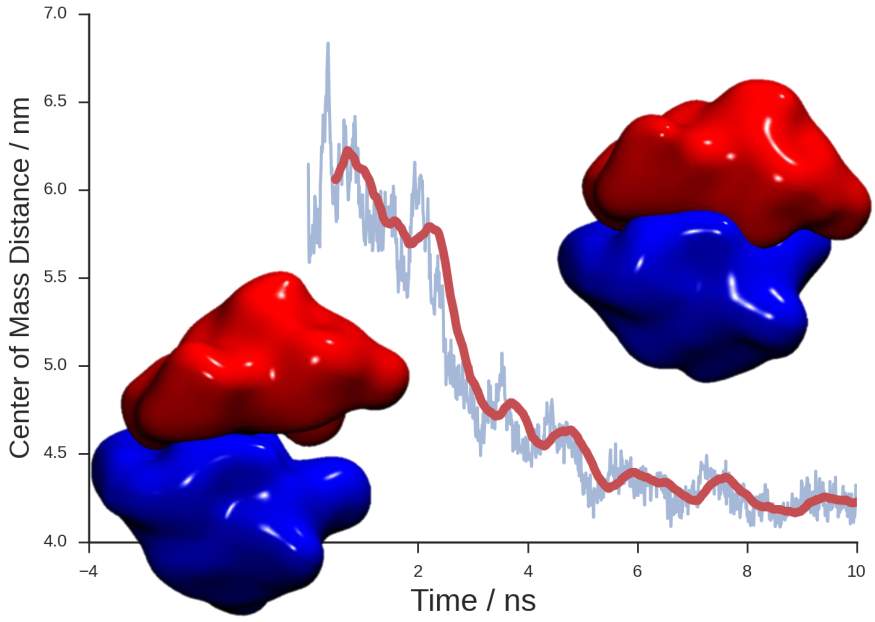

Figure 2.2: The two-domain enzyme BphC is started from an extended conformation and is allowed to dewet. The two domains are shown in a low-resolution surface representation at $t = 0.7\,\text{ns}$ (left) and $t = 8.8\,\text{ns}$ (right). The center of mass distance is plotted over time. We use this system as a model to test the new method presented. This molecule can be seen in "cartoon" representation in fig. 2.1

A model system for solvent dynamics is that of hydrophobic collapse in the BphC

enzyme (1dhy) [108, 109]. This two-domain protein functions in degrading toxic polychlorinated biphenyls. Hydrophobic residues on the interface of the two domains promote both dewetting of the interfacial cavity and structural collapse from extended conformations. By starting simulations from artificially extended conformations, we can observe dewetting transitions. Figure 2.2 shows one such transition. We stress that the focus of this study is to demonstrate how to include solvent degrees of freedom in MSM analysis and not to provide novel insight into the function of BphC.

In this chapter, we introduce a new method called the solvent-shells featurization for transforming solvent positions into suitable solvent features. We characterize and parameterize this method on 100 (10 ns each) molecular dynamics simulations of the BphC enzyme, each initialized from an extended conformation. Through the use of an appropriate scoring function under cross-validation, we examine the hyperparameters in model construction and show that including solvent degrees of freedom in MSM construction gives better models. Finally, we interpret the resulting model by taking advantage of state-of-the-art MSM techniques and the new solvent features.

2.2 Solvent-Shells Featurization

In contrast to traditional conformational distance metrics used for clustering of solute (protein) states, we seek a distance metric suitable for solvents that (1) treats solvent molecules as indistinguishable, and (2) is invariant under translation and rotation of the solvent box relative to the solute molecule. A solvent metric would be particularly desirable if it (3) can identify solvent molecules of interest, and (4) is fast to compute.

Gu et al. defined a "solvent fingerprint" [110] which uses a sum of weighted solute–solvent distances for each solute atom to define a vector representation of the solvent configuration.

$$\mathrm{FP}(x \in \text{solute}) = \sum_{y \in \text{solvent}} \exp\left(-\frac{||x-y||^2}{2\sigma^2}\right) \tag{2.1}$$

where $||x-y||$ is the Euclidean distance between solute atom position x and solvent atom position y, and σ is a free parameter that defines a distance scale. The resulting

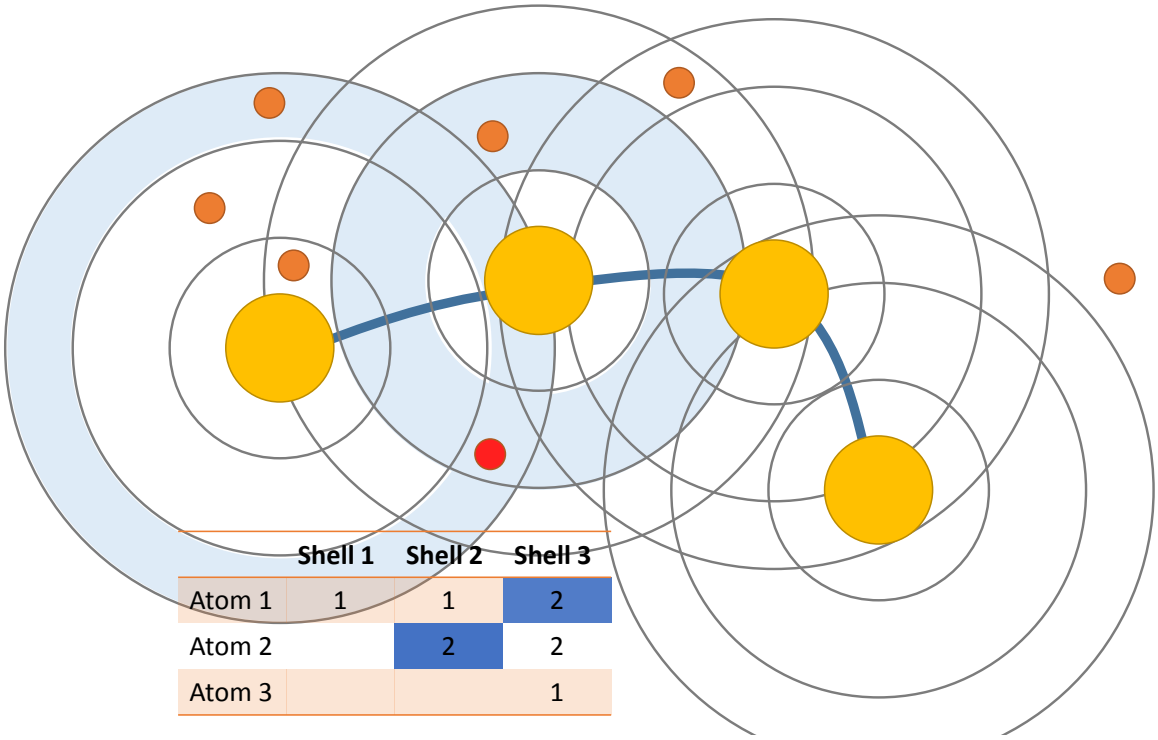

Figure 2.3: Schematic representation of the solvent-shells featurization. Solvent atoms (small, filled circles) are binned based on radial distance (concentric rings) from each solute atom (large, filled circles). Training an intermediate kinetic model such as tICA, "important" solvent-shell features can be identified (highlighted table entries). We can exploit overlapping shells (e.g. two shaded rings corresponding to highlighted table entries) to provide non-radially-symmetric identification of regions of solvent (small, red circle).

feature vector is of length N_{solute} and can be used with an ordinary l^2 norm for clustering. Physically, we can interpret the feature values as the degree of solvation of each solute atom.

We propose an extension of this fingerprint where we seek to preserve spatial resolution of the solvent that is destroyed in the summation. We define the solvent-shell featurization:

$$\text{SS}(x \in \text{solute}; r) = (4\pi r_{\text{mid}} dr)^{-1} \sum_{y \in \text{solvent}} \mathbf{I}(r \leq ||x - y|| < r + dr) \tag{2.2}$$

parameterized by a set of spherical shells specified by distance r and width dr, and where $r_{\text{mid}} = r + dr/2$. The solvent-shell featurization gives the instantaneous solvent density in each of the shells. The resulting feature vector is of length $N_{\text{solute}} \cdot N_{\text{shells}}$ and can be used with an ordinary l^2 norm for clustering. For computational and cognitive convenience, we use an integer number of equal-width shells. The featurization is shown schematically in fig. 2.3.

These features bin solvent atoms without regard for their identity, satisfying (1), and only consider relative solute-solvent distances, satisfying (2). By recording the assignment of solvent atoms to shells, we can back-out individual solvent atoms corresponding to each feature with resolution at least as good as dr. This satisfies criterion (3) and is a powerful way to extract biophysical understanding by identifying functional waters, see section 2.5.

This featurization is implemented as a plug-in for the open-source software package MSMBuilder. Computation of the features is performed with SSE4.1 vectorized operations, thus satisfying (4). The resulting feature vector enables the use of fast clustering methods such as Mini-batch K-means [111], further enhancing computational speed in contrast to the traditional RMSD metric.

2.3 Unified Framework for Solvent and Conformational Dynamics

Having a protocol for computing solvent features, we wish to construct a model that captures both solvent and solute dynamics. To that end, we choose a set of solute conformational features (e.g. dihedral angles, raw Cartesian coordinates) to be used in conjunction with the solvent-shells features.

State-of-the-art MSM construction methods suggest training a kinetic model prior to the clustering step [19, 61]. One such model is time-independent component analysis (tICA). tICA is similar to principal component analysis (PCA) in that it produces a set of linear combinations of input features which define "components" to serve as a new basis set for the data. Whereas PCA finds components which maximize variance among input degrees of freedom, tICA finds components which maximize autocorrelation of the time-series input [70, 71]. We effectively find the slowest degrees of freedom for the system (subject to the constraint that the degrees of freedom be linear combinations of the input features). By projecting our input features on the top n slowest time-independent components (tICs), we introduce an intermediate dimensionality reduction which aligns our conformation-based estimate of kinetic interconversion rates even closer to the actual kinetics of the model. Because of the linearity constraint of tICA, we generally still need to build an MSM to capture the non-linear dynamics of the system under study. This intermediate processing with tICA or PCA permits a unified framework for treating solvent and conformational degrees of freedom in which all features are fed as input, and the component analysis model select those features deemed "relevant".

Special care should be taken when building an MSM directly from a union of conformational and solvent features without an intermediate model such as tICA. Spherical clustering algorithms (e.g. k-centers) are sensitive to scaling of input features; the two sets of features should be normalized to have equal variance to ensure meaningful clustering. PCA includes normalization by variance. tICA includes normalization by either variance or autocorrelation timescale (i.e. slowness); in this study, the autocorrelation timescales were used for normalization.

2.4 Evaluation on BphC Enzyme

With this new method, we seek to create better MSMs by capturing the slow, biologically-relevant dynamical processes with a generalizable model that uses a unified framework to include solute, solvent, membrane, and any other key degrees of freedom. Recently, McGibbon et al. introduced a scoring function based on the Generalized Matrix Rayleigh Quotient (GMRQ) that quantifies the goodness-of-fit for an MSM [79]. The GMRQ is a scalar functional which measures the ability of a rank-m projection operator (in this case, the top m eigenvectors of the MSM) to capture the slow dynamics of a system. [65]. In theory, the GMRQ is bounded by the sum of the first m eigenvalues of the true dynamical propagator: a non-perfect dimensionality reduction will always model dynamics which are too fast [66]. However, McGibbon et al. showed that this bound can be violated when the model was parameterized from statistically-noisy inputs (e.g. a MD simulation with less than infinite sampling). This overconfidence in the model is a result of overfitting. It can be eliminated by evaluating the model on a different dataset than the one on which it was trained, i.e. via cross-validation. Due to its generality, the GMRQ permits direct comparison among MSMs built with methods that could differ in hyperparameters, intermediate processing steps, and/or featurization.

We performed 3-fold cross validation (trajectories kept whole across folds) over the grid of hyperparameters specified in table 2.1 (Grid Search 1) on 100 (10 ns each) molecular dynamics simulations of the BphC enzyme, each initialized from an extended conformation. Conformational degrees of freedom were included using distribution of reciprocal of interatomic distances (DRID) features [81]. For each solute atom, the reciprocal distance to every other solute atom is computed, forming a distribution. DRID characterizes this distribution by its first three moments. DRID is a translationally and rotationally invariant way to featurize solute molecules with no *a priori* knowledge of the system. Solvent degrees of freedom were included using the solvent-shells features introduced in this chapter. Solute atoms were defined to be the alpha-carbons of the protein residues, and solvent atoms were defined to be the water oxygens. Pruning redundant parameter configurations (solvent-specific parameters do

not matter when including only DRID features) yielded 255 models and associated scores over five dimensions. The models were evaluated using GMRQ based on fidelity to the two slowest dynamical processes and the equilibrium distribution (rank $m = 3$) at a lag time of 0.5 ns.

Hyperparameter	Grid Search 1	Grid Search 2
Features	DRID, Solvent, Both	
Shell widths (dr)	1, 2, 3, 4 Å	3 Å
Total extent ($dr \cdot N_{\text{shells}}$)	5, 10 Å	
Number of shells (N_{shells})		1, 2, 3, 4
tICA components	1, 2, 3	1
MSM microstates	50, 100, 200, 400, 800	
# Models	255	45

Table 2.1: Hyperparameters were investigated and tuned by performing two grid searches over the values given in this table. For each parameter configuration, models were scored using 3-fold cross validation using the GMRQ scoring functional. In grid search 1, the spacial extent of the featurization was explicitly specified. In cases where this would dictate a fractional number of shells, the nearest integer number was used. In grid search 2, the number of shells was explicitly specified which implicitly determined the spacial extent.

Due to the generality of the GMRQ, we can simply select the set of hyperparameters that yield the highest mean (over folds) test set score. This suggests using both solute (via DRID) and solvent (via solvent-shells) features, 4 shells each of width 3 Å, 1 tIC, and 100 MSM states. Further investigation of the marginal effects of specific hyperparameters offer insight into the new method.

Figure 2.4a shows scores as a function of the number of tICA components included in MSM construction for each of the three input-feature configurations. These scores were taken from the optimal solvent-shell parameters (given above) and marginalized over number of MSM microstates by taking the maximum score. Error bars represent standard deviation over folds. For configurations which include the solvent-shells metric, the score decreases with increasing number of components included in MSM construction. This is most likely due to overfitting to the extra degrees of freedom. These observations are consistent with these simulations, which are dominated by

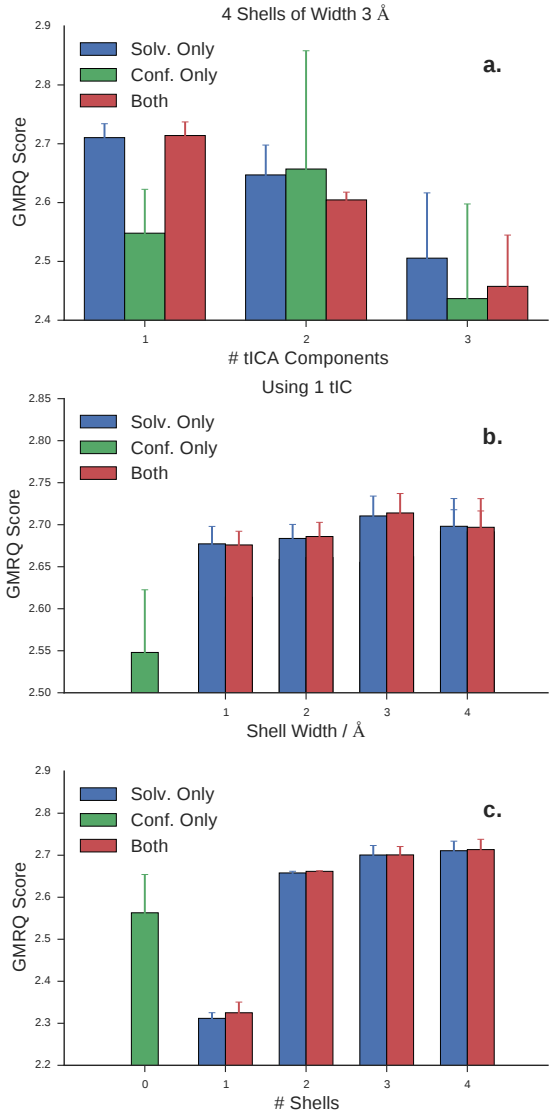

Figure 2.4: (a) Increasing the number of tICs biases the model towards over fitting. In this simple system, one coordinate is sufficient to provide a generalizable model. Using only the conformational features requires two tICs to maximize the score, whereas inclusion of solvent features which match the physics of the simulations maximize score with only one tIC. The score is maximized with the inclusion of the solvent-shells features introduced in this chapter. (b) A shell width of 3 Å balances statistical variance with spatial resolution. (c) Increasing number of solvent shells results in a better score. By not extending the solvent featurization far enough (i.e. using only 1 shell), the model performs significantly worse than one fit only on conformational features.

one coordinate (the dewetting conformational change). Similarly, including conformational degrees of freedom in addition to the solvent degrees of freedom does not appreciably increase the score of the MSM. The conformational degrees of freedom do not capture anything in the first tIC that the solvent metric does not. The conformational metric (DRID) behaves differently: its score peaks at 2 tICs (albeit with high variance across folds) before succumbing to overfitting. It makes sense that the solvent features reproduce this one coordinate better than a general conformational metric in this system where solvent change is the dominating characteristic. We expect more complicated systems to benefit from multiple tICs and a combination of solvent and conformational degrees of freedom.

The dependence of score on solvent shell width (dr) was investigated for 1 tIC and total extent $N_{\text{shells}} \cdot dr = 10\,\text{Å}$, again marginalizing over number of MSM microstates (fig. 2.4b). Choosing an appropriate shell width balances statistical variance with spatial resolution: A large number of skinny shells provides a higher resolution description of the solvent environment, but wider shells occupied by more molecules provide a lower-variance estimate of the local density. We observe that a shell width of 3 Å provides the best balance and maximizes the GMRQ score. This is physically reasonable as it corresponds to between one and two solvation shells in water.

A second grid search was run to investigate dependence of model score on the number of shells included in the featurization (fig. 2.4c). Shell widths and number of tICs were kept constant at optimal values from the first grid search (table 2.1, Grid Search 2). While we might expect solvation to be a local effect, including more shells seems to improve the model without introducing overfitting. In fact, including only the closest shell results in a significantly worse model than one with just conformational features. We postulate that increasing the spatial extent of the featurization allows non-spherically symmetric localization of important regions of solvent when used in conjunction with tICA. For example, consider a region of solvent that partially occupies shell 1 of residue 1, but shell 2 of neighboring residue 2. The overlap of these two occupancies breaks the spherical symmetry of the featurization around each individual residue as shown schematically in fig. 2.3. These findings suggest that only considering hyper-local solvation as in ref. 110 may be misguided.

2.5 Biophysical Interpretation

A powerful feature of the solvent-shells featurization is its interpretability. Solvent molecules can be assigned to the shells which they occupy. One convenient way to exploit this property is to use the coefficients of the tICA model's slowest components. Each time-independent component (tIC) is defined as a linear combination of input features. When the input features are the solvent-shell occupancies, the resulting coefficients can be used to assign "importance" (i.e. degree of contribution to slow dynamical processes) to the solvent shells, and by extension, individual solvent molecules which occupy those shells.

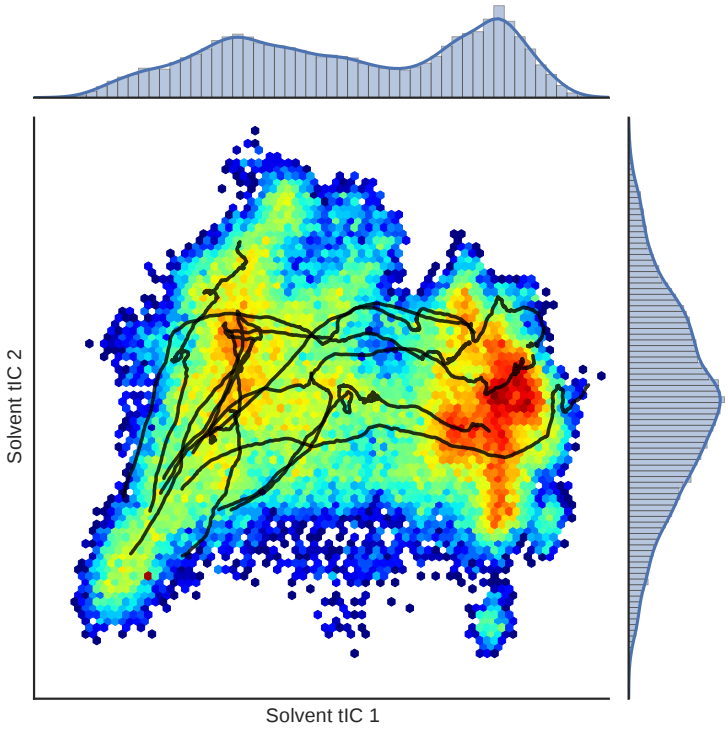

Figure 2.5: tICA analysis on solvent-shell features successfully identifies the slow degree of freedom corresponding to dewetting. Several trajectories (black lines) projected onto the first two tICs are overlaid on a 2D histogram of solvent conformations to show the general progression from wet (low tIC1, left) to dry (high tIC1, right)

We applied the solvent-shell featurization to the same ensemble of 100 (10 ns

each) simulations of the model two-domain protein BphC. Figure 2.5 shows that projection along the two slowest tICs provides separation into at least two regions of high population. The trajectory paths suggest that the first tIC is highly correlated with dewetting of the inter-domain cavity. We use the trained tICA model to enhance our biophysical understanding by visualizing each solvent molecule colored according to its tIC coefficient. We applied a logistic function as depicted in fig. 2.6 and summed the coefficients of water-oxygen atoms occupying overlapping solvent shells. Altering the logistic function parameters did not qualitatively affect the resulting visualization. With VMD [112], these values could be used to color, show, or hide solvent molecules of importance. As seen in fig. 2.7, the solvent-shells features allow the automated discovery of the interesting, slow-dynamical solvent features. The water molecules in BphC's hydrophobic cavity are discovered *a priori*.

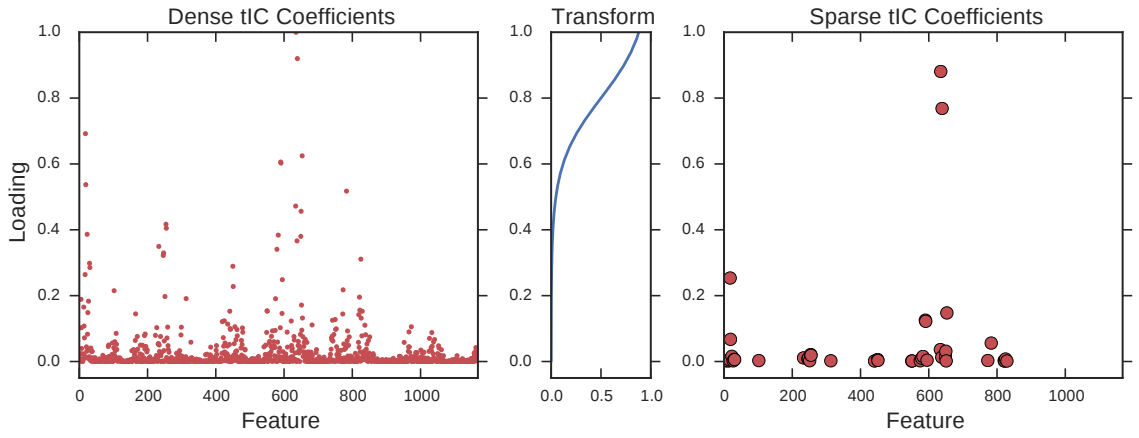

Figure 2.6: tICA independent component coefficients are dense with regards to input features. A logistic function is applied to up-weight the significant features while reducing noise. When performed on the solvent-shells features, the sparse coefficients can be used to visualize solvent molecules of importance.

2.6 Conclusions

The inclusion of solvent degrees of freedom in MD trajectory analysis has the potential to be a boon for biophysical understanding both by enhancing interpretability of the

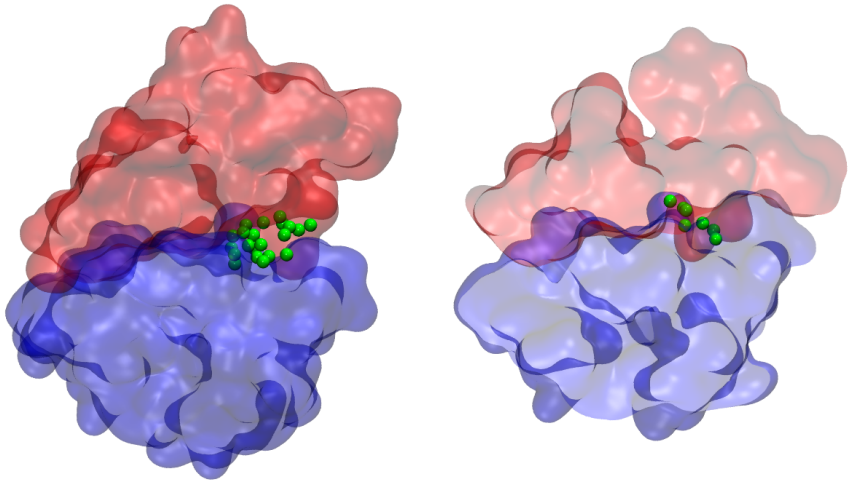

Figure 2.7: By including only the solvent molecules with tIC coefficients (learned by this method) above a cutoff value, individual water molecules that comprise the slowest degree of freedom are revealed. Here, the two domains of BphC (colored red and blue) are shown in a surface representation, and water molecules are represented by spheres centered on the water-oxygens. (Left) A wet structure at $t = 1\,\text{ns}$ contains many waters in the inter-domain cavity. (Right) The same trajectory at $t = 9\,\text{ns}$. The water has mostly been expelled from the cavity. The front half of the BphC molecule is not drawn to show the few trapped waters within the enzyme. The solvent molecules participating in the dewetting are automatically identified by the method due to their kinetic relevance.

models as well as improving the models themselves. As seen from the GMRQ scores for MSMs of the model protein BphC, the solvent-shells featurization in conjunction with a structural metric yields more generalizable dynamical models. With the aid of tICA, these models are also more interpretable. Projection onto solvent tICs allows visual inspection of candidate metastable states. Visualizing solvent molecules with high tIC coefficients shows important solvent features learned entirely from the MD data in their appropriate biological context. This enables the discovery of functional solvent molecules from simulations without prior knowledge. We anticipate that the ease of incorporating the solvent-shells metric into the standard MSM analysis paradigm combined with the benefits of treating *all* degrees of freedom in our analysis will have broad applications beyond water. Whereas the water-oxygen atoms were considered in this study, we emphasize that this method is general to any indistinguishable particles including ions and lipids. Including indistinguishable particles in dynamical analysis allows MSMs to more naturally model a much broader range of phenomena including protein-lipid interactions, membrane dynamics, colloidal systems, docking, and many-body protein simulations [100].

2.7 Simulation Details

The simulations were started from the crystal structure of BphC (PDB id: 1dhy.pdb) [108, 109]. The crystal structure contains two domains (residue 1-135 and 135-292) at a center of mass distance of 18.72 Å. These structures were solvated in a tip3p [27] water box containing ∼16,500 molecules such that the minimum distance between the boundary and the protein is 12 Å. The system was neutralized by adding 8 Na^+ ions. The Amber99sb-ildn [25] forcefield was used for protein and ions. The structures obtained after an initial equilibration for 1 ns at constant temperature and pressure and with constraints on the heavy atom positions were used as the starting conformation for the subsequent simulations. The interdomain distance of the crystal structure was increased by ∼6 Å along the direction of two domain centers of geometry to create a gap between the two domains using the Steered Molecular Dynamics [113] method. The constant velocity pulling method was used by restraining residues

1-135 and applying a force on residues 136-292. The resulting structure was then used for running 100 dewetting simulations of 10 ns each for a total simulation time of 1000 ns. The Gromacs [38] simulation package was used for running these simulations. Covalent bonds involving hydrogen atoms were constrained with LINCS [114] and particle mesh Ewald [44] was used to treat long-range electrostatic interactions. Production MD simulations were carried out at constant temperature and pressure of 300 K and 1 atm respectively, with a timestep of 2 fs.

The code used for computing the Solvent-Shells featurization is available at http://github.com/mpharrigan/wetmsm and depends on MDTraj (mdtraj.org) and MSMBuilder (msmbuilder.org).

Acknowledgments

For this chapter, we thank Robert T. McGibbon and Christian R. Schwantes for useful discussion throughout the preparation of this chapter. DS was supported by the Biomedical Data Science Initiative Fellowship from Stanford School of Medicine. DS and VSP acknowledge support from the SIMBIOS NIH National Center for Biomedical Computation through the NIH Roadmap for Medical Research Grant U54 GM07297. In addition, we acknowledge funding from NSF-MCB-0954714 and NIH-R01-GM062868.

Chapter 3

Markov modeling reveals novel intracellular modulation of the human TREK-2 selectivity filter

This chapter is adapted from Matthew P. Harrigan, Keri A. McKiernan, Veerabahu Shanmugasundaram, Rajiah Aldrin Denny, and Vijay S. Pande. Markov modeling reveals novel intracellular modulation of the human TREK-2 selectivity filter. *Sci. Rep.*, 7(1), 2017. doi: 10.1038/s41598-017-00256-y [115], which is licensed under a Creative Commons Attribution 4.0 International License. K. A. McKiernan and M. P. Harrigan contributed equally to this work.

Abstract

Two-pore domain potassium (K2P) channel ion conductance is regulated by diverse stimuli that directly or indirectly gate the channel selectivity filter (SF). Recent crystal structures for the TREK-2 member of the K2P family reveal distinct "up" and "down" states assumed during activation via mechanical stretch. We performed 195 µs of all-atom, unbiased molecular dynamics simulations of the TREK-2 channel to probe how membrane stretch regulates the SF gate. Markov modeling reveals a novel "pinched" SF configuration that stretch activation rapidly destabilizes. Free-energy barrier heights calculated for critical steps in the conduction pathway indicate that this pinched state impairs ion conduction. Our simulations predict that this low-conductance state is accessed exclusively in the compressed, "down" conformation in which the intracellular helix arrangement allosterically pinches the SF. By explicitly relating structure to function, we contribute a critical piece of understanding to the evolving K2P puzzle.

3.1 Introduction

TREK-2 is a member of the human K2P family of tandem-pore potassium channels. This protein is responsible for leak currents in nearly all cells. Its dysregulation has been linked to pain and depression, and it can be functionally regulated by mechanical stretch, heat, fatty acids, pH, secondary messengers of signaling proteins, and several drugs [116–118]. Recent evidence suggests that the regulatory processes acting on this channel modulate conductance through structurally distinct mechanisms [119]. However, the specific details of the conformational changes involved in these mechanisms remain unclear.

Potassium channel ion conductance is mediated by both an intracellular and interior gating process. The former involves large-scale rearrangements of the intracellular domains of the channel's transmembrane helices to sterically block passage of ions. Interior gating involves smaller conformational changes occurring directly in the channel's selectivity filter (SF), so named for its role in conferring potassium selectivity

to the channel. The SF is formed by the p-loop domains connecting the protein subunits. These p-loops are arranged parallel to one another and are radially symmetric about the ion conduction pathway such that the backbone carbonyls point inward to form 4 adjacent binding sites (fig. 3.1). Interior gating alters the stability of these binding sites.

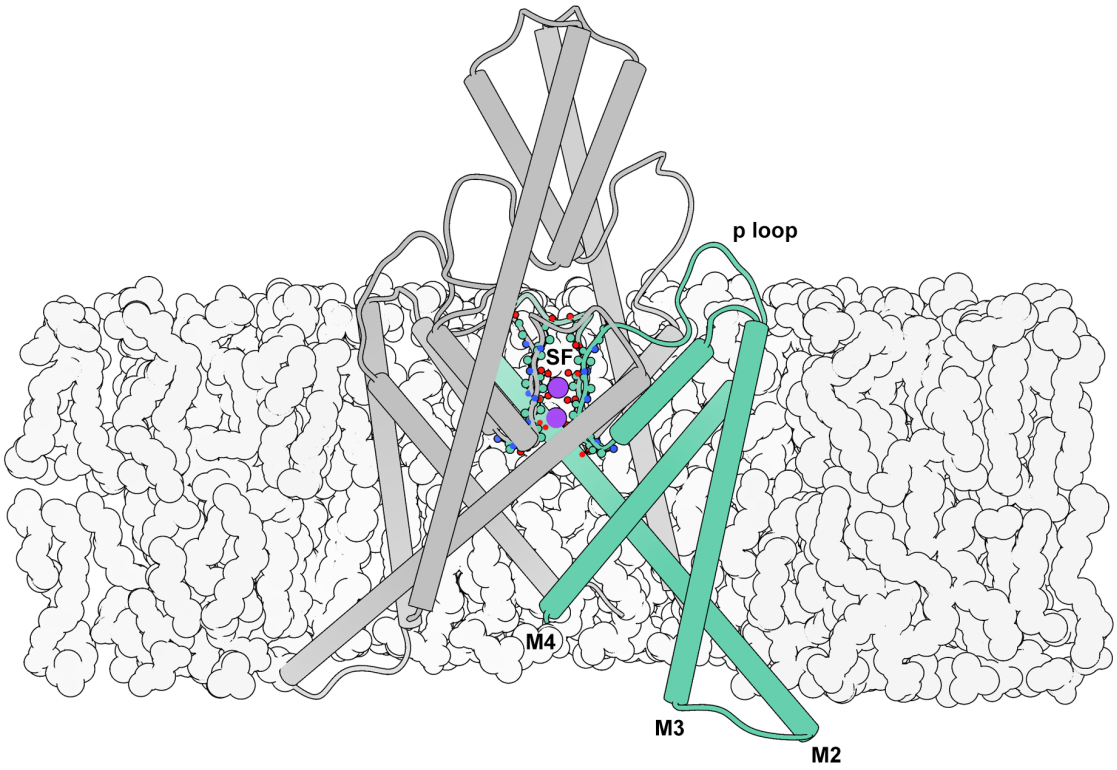

Figure 3.1: Illustration of the TREK-2 system. The selectivity filter (SF) is located at the center of the extracellular region of the membrane embedded pore, and is formed by the backbone of the four p loops. Here it is shown with occupancy corresponding to the simulated resting state. Intracellular gating involves movements of the M2, M3, and M4 helices, while interior gating involves conformational changes of the SF.

In contrast to other types of potassium channels, switching of the TREK-2 intracellular gate is not thought to "close" the channel directly. Rather, the helical conformational changes confer information to the SF, which acts as the primary gate

[120–122]. Recent crystal structures of TREK-2 provide structural insight into major conformational states involved in mechanical (stretch) activation by resolving distinct "up" (stretched) and "down" (compressed) conformations [123]. The number of bound ions observed for each state suggests differing levels of conductance, rather than closure of the channel. To understand how the channel switches between these conductance levels, we apply dynamical techniques to complement static crystal structures.

Molecular dynamics (MD) is a method for simulating biological systems with atom-level resolution. By using commodity GPUs [57] paired with a massively distributed architecture [48], simulations can reach biologically-relevant timescales. Advances in computing result in large quantities of high-dimensional time-series data. To draw interpretable conclusions from this information, further analysis is required. Markov state models (MSMs) have gained favor for their robust, statistical analysis of biophysical systems [19, 60, 61, 94]. Time-structure based independent component analysis (tICA) is a component of Markov state modeling that can aid in interpretability. tICA is a machine learning technique to find the slowly-decorrelating modes in high-dimensional time-series data. Biophysical processes of interest are often those that occur with the slowest timescales. By using tICA on a large set of input features, we avoid codifying our preconceived notions about the dynamics of the system into the analysis while still reducing the dimensionality of the system by projecting conformational features onto a small number (≈ 5) of kinetically-motivated independent components (tICs). Finally, Potential of mean force (PMF) computations can complement MSM analysis by offering quantification of free-energy profiles along paths or coordinates of interest [124].

In this chapter we investigate how the structural conformational changes involved in stretch activation influence the thermodynamics and kinetics of ion permeation. We sample along this pathway through unbiased MD simulation. We then examine the dynamical behavior of these simulations using Markov modeling. Specifically, we apply MSM analysis of the large MD dataset to survey equilibrium conformational dynamics. We detect a novel "pinched" configuration of the SF favored during compression. We survey SF equilibrium ion dynamics, and contrast structural macrostates'

unique SF ion occupancy preferences. We compute PMF curves to quantitatively analyze the energetics involved in SF ion occupancy state transitions which show a correlation with both the whole-molecule and SF conformational states. Notably, the states favored under membrane compression have the most disfavored outward ion rectification. Our results predict a specific structural motif that we implicate in the coupling of intracellular and interior gates. We propose the novel conformation as a basis for future study.

3.2 Results

3.2.1 Dynamics of conformational change

X-ray crystal structures offer an unrivaled atom-level view into static structures of proteins. The published crystal structures for TREK-2 identify two distinct stable conformations for the channel, presumed to be the result of membrane stretch and compression. The stretched, "up" state's transmembrane (M) helices are displaced upwards and outwards compared to the compressed, "down" state. The conformational differences are large compared to the size of the protein. In this chapter, we refer to the states distinguished by large-scale motions as "macrostates". These static structures lack dynamical information, and the relative populations or free-energies of each macrostate in physiological conditions is unknown. We ran 195 µs of unbiased, explicit-solvent, explicit-membrane, isobaric molecular dynamics (MD) initialized with protein coordinates from the four crystal structures aiming to dynamically connect the experimentally observed structures and discover metastable or intermediate conformations inaccessible to crystallography.

Due to the large amount of data among many distributed trajectories, we apply state-of-the-art MSM modeling methods to glean insight from the large dataset. We use tICA to automatically discover an "up–down" stretch activation coordinate from a large number of atom-pair distances. Building an MSM on these coordinates permits generation of a 500 µs representative trajectory. By inspection, we identify the primary tIC as an "up–down" measurement. The up–down trace for the MSM

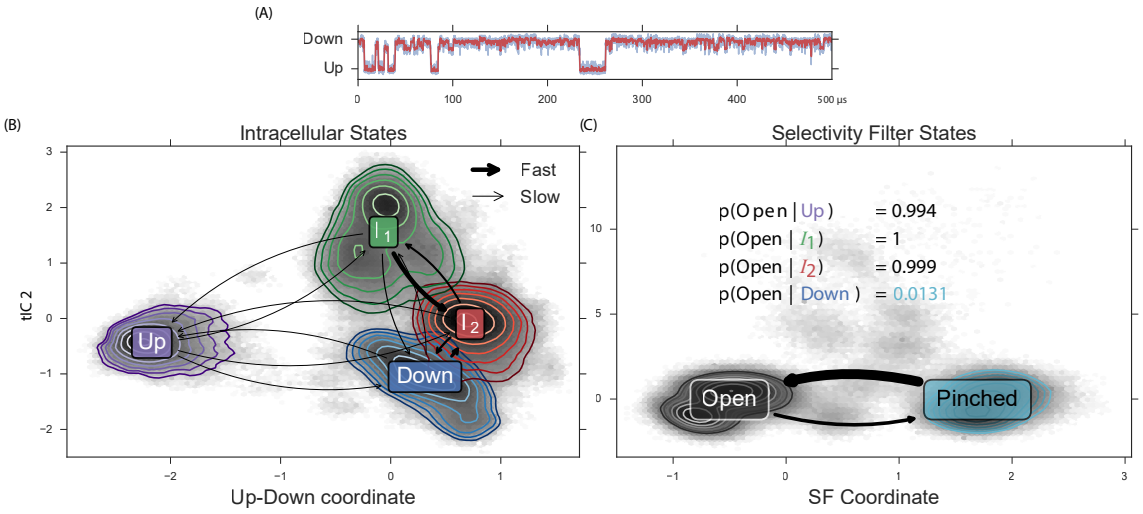

Figure 3.2: Automatic coordinate discovery using molecular dynamics data identifies an up-down coordinate from a set of high-dimensional input data. Modeling the observed dynamics by construction of an MSM permits rapid analysis of conformational dynamics. From our analysis, we discover a novel selectivity filter (SF) conformation. **(A)** MSM analysis permits generation of a 500 µs representative trajectory which contains transitions between up, down, and intermediate conformations, see movie in SI. **(B)** We find four metastable conformational macrostates, shown as contours in tIC space. Transition rates show a rapid relaxation from I_1 to I_2, and a fast exchange between I_2 and *Down*. **(C)** Automatic coordinate discovery on the selectivity filter finds a structurally distinct pinched state. Inset enumerates the SF open probability as a function of intracellular macrostate. The *Up* and intermediate intracellular states favor an open SF, while the *Down* intracellular state strongly favors a pinched SF.

trajectory is shown in fig. 3.2A and a movie is available (see Fig. S1 and supporting files). The kinetic information in the MD dataset can be exploited to assign each observed conformation to a macrostate (fig. 3.2B, contours). As a matter of notation, we refer to MSM-assigned macrostates corresponding to a specific ensemble of conformations with script-text (e.g. Up) to distinguish from abstract notions of states, which we quote (e.g. "up"). We investigate the kinetics and thermodynamics among the macrostates (fig. 3.2B, arrows). We find a highly-populated *Down* state in exchange with an intermediate I_2, differing by a partial unfolding of the M2-M3 loop (Fig. S2). We find a metastable intermediate I_1 that rapidly relaxes to I_2 (Fig. S3). We find a kinetically-distant Up state that can transition to and from each of the other three states (fig. 3.3A). The TREK-2 system is known to form fenestration sites between the membrane embedded helices wherein inhibitors may bind. The volume of these sites varies monotonically along the up–down coordinate (Fig. S1), with the largest corresponding to the *Down* states.

In addition to using tICA to discover an up–down coordinate, we focused the algorithm specifically on the selectivity filter region of the channel. Specifically, we use interatomic distances only among selectivity filter atoms as the input features to learn a second tICA model. In this case, the molecular dynamics reveals a metastable "pinched" conformation of the selectivity filter (fig. 3.3B). This SF conformation is not represented in the TREK-2 crystal structures, and could describe a conformation similar to that assumed during C-type inactivation [125]. In this conformation, residues at the top of the S_1 binding site rotate outwards, away from the ion conduction pathway, while residues at the top of the S_0 binding site move inward. This inward movement is most drastic for G176 (A) and G176 (B) (where (A) signifies the protein chain), which restricts the pore from approximately 8.5 to 4.5 angstroms. We model the dynamics between two selectivity filter states (*Open* and *Pinched*) with a Markov model. The model suggests a free-energy minimum *Open* conformation and a metastable *Pinched* conformation (fig. 3.2B). Comparing the structural macrostate and SF state for each conformation indicates that the *Pinched* SF conformation is strongly coupled to the channel's intracellular conformation. The *Pinched* SF state is more prevalent in the fully *Down* state by a ratio of 75:1 (fig. 3.2C, inset table).

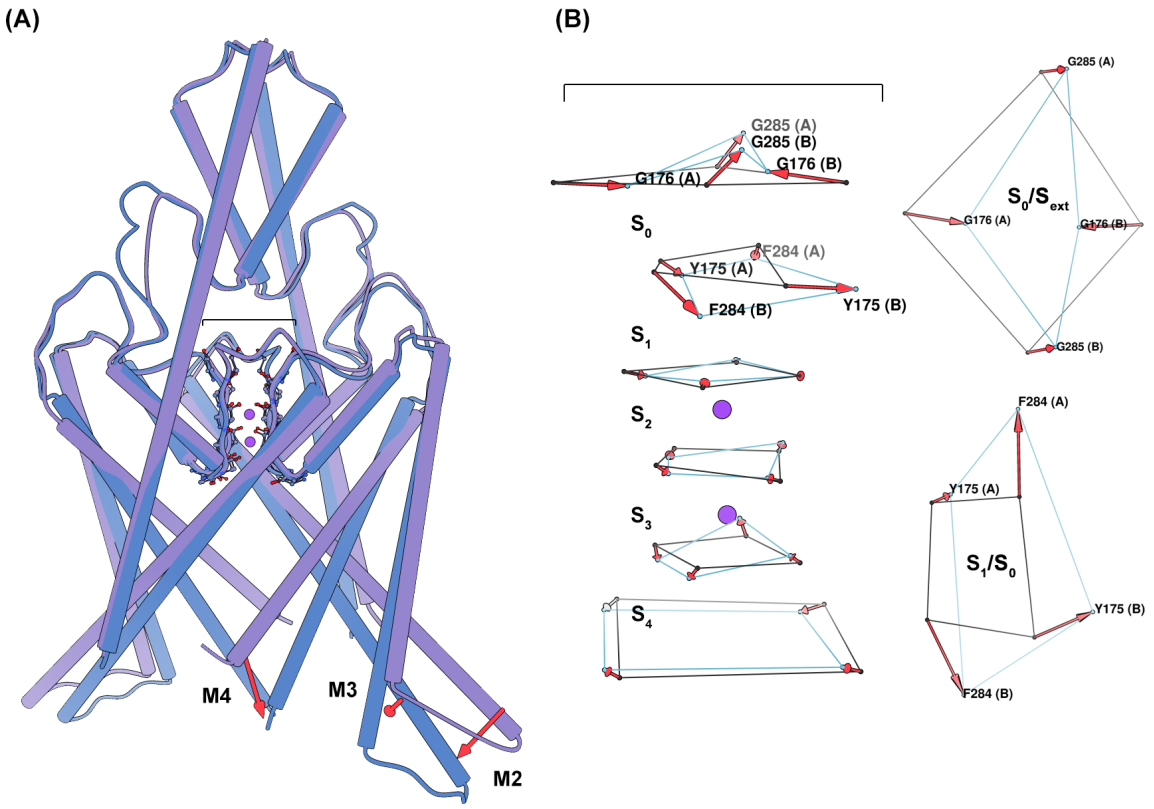

Figure 3.3: Conformational sampling of the MSM provides structural insight into the major states assumed during simulation. **(A)** A side-view of a superposition of the TREK-2 *Up* (purple) and *Down* (blue) intracellular macrostates, highlighting significant conformational differences. Transition from the *Up* to the *Down* state involves the large-scale inward movement of M2, M3, and M4. **(B)** A side-view (left) of a superposition of the TREK-2 SF *Open* (black) and *Pinched* (light blue) states, and aerial view (right) of the carbonyl oxygens at the interface of the S_1/S_0 and S_0/S_{ext} binding sites. Here the SF is represented only by the carbonyl oxygens constituting the SF binding sites. Transition to the *Pinched* filter involves rotations of the residues surrounding the S_1 and S_0 binding sites. These changes disrupt the symmetry of the filter such that the S_0/S_1 carbonyls no longer face the ion conduction pathway (reducing the ability of these sites to coordinate ions), and the extracellular mouth of the filter is reduced in diameter by nearly 4 angstroms (G176 (A) - G176 (B)).

In the *Up* and I_2 macrostates, the situation is reversed; the proportion of *Pinched* conformations observed is strongly disfavored and the *Open* configuration dominates. In the I_1 state, only the *Open* SF is seen. The disruption of symmetry and restriction in pore diameter caused by the rearrangements of upper-SF carbonyl oxygens in the *Pinched* state strongly suggests that this conformational difference will negatively affect ion conductance rates. In the following sections, we will provide evidence for this reduced conduction rate.

3.2.2 Dynamics of selectivity filter ion occupancy

TREK-2 not only exhibited large-scale conformational differences in the available crystal structures, but also subtle differences in ion binding. The "down" structures were crystallized with $3K^+$ while the "up" structure was crystallized with $4K^+$. Dong et al. [123] hypothesized that all four combinations of (up, down) $\times$ ($3K^+$, $4K^+$) states are accessible, despite only observing the two. By analogy to other channels, they hypothesized that $4K^+$ states are highly conductive and $3K^+$ states are less conductive.

We test these hypotheses with our MD dataset. Specifically, we investigate the effect of conformational macrostate on ion occupancy in the selectivity filter (SF). By first partitioning the data by conformational macrostate (*Up*, *Down*, I_1, and I_2), we construct four MSMs of ion dynamics to compare and contrast the observed ion microstates and rates (shown in fig. 3.4A and C). Thick arrows represent rapid transitions. fig. 3.4A is the kinetic model for SF ion transitions in the *Up* conformation while fig. 3.4C is for *Down* (see Fig. S7 and 8 for I_1 and I_2). The relative populations of each ion microstate can be related to the up–down coordinate and is shown as a 2D-histogram in fig. 3.4B.

Several trends are observed in these network graphs and population histogram. For all macrostates, OXXO [X = occupied; O = vacant; positions are given from S_1 to S_4 left-to-right] is found to be the dominant equilibrium ion microstate regardless of macrostate. The OXXO state serves as a sink with many observed transitions into it. This suggests that this state could represent the resting occupancy state. The

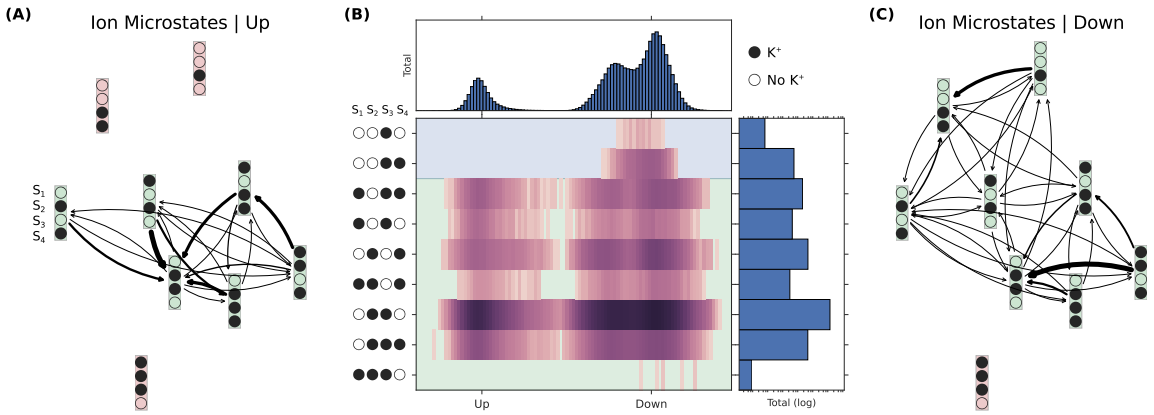

Figure 3.4: Ion conduction can be modeled as transitions among ion-occupancy states. **(A)** Transition rates between ion states in the *Up* macrostate conformation. **(B)** Populations of ion states as a function of the up–down coordinate identified by statistical analysis of the large-scale conformational dynamics. Relative free energies of the ion microstates are conformationally dependent. Additionally, some ion microstates are observed that exist solely in down conformations. **(C)** Transition rates between ion states in the *Down* macrostate conformations.

higher ion occupancies observed in the crystal structures may be an artifact of the cryogenic crystallography conditions (investigated in Ref. [126]. In particular, Fig. S2 of its Supplementary Information). An additional commonality lies in the solvation of the SF ions. The hydration of the ions in the SF during the conduction process is an area of recent contention [127]. The soft knock-on mechanism posits that the ions are never directly adjacent, and they must be padded by at least one water molecule. The hard knock-on mechanism asserts that ions are preferably adjacent to one another. Both theories are based on the knock-on mechanism first formulated by Hodgkin and Keynes in 1955 [128], wherein conduction is explained by the energetic perturbation of an ion entering the SF, and differ only with respect to ion solvation. We find that the full set of probable occupancy states determined from our simulations contains exclusively unsolvated ions. In fact, we observe very little water occupancy for any SF binding site, not just at the sites between ions, for all macroscopic conformations. We find the hard-knock mechanism described by Köpfer et al. [126] was supported by our simulations due to the degree of ion solvation and similarity between the set of most probable SF occupancy states. Note that TREK-2 is not canonically voltage

sensing [129], so we would not expect the protein conformation to change drastically in the presence of a membrane potential.

The dissimilarities among the macrostate-specific MSMs are striking. The *Down* macrostate displays more diffuse ion dynamics with a large number of possible transitions among an extended state space. The equilibrium OXXO behaves less as a network sink with comparatively fewer high flux transitions into it. One might hypothesize that the greater ion movement would imply lower kinetic barriers and higher conduction. This is counter-intuitive as the down state has been suggested to be the low-conductivity state. While it is not clear how or to what extent the large, slow changes in macrostate impact the fast microstate transition pathways and stabilities, this joint conformation–ion analysis highlights qualitative differences that merit further study. To investigate quantitatively the effects of conformation on ion conduction, we perform additional simulation to study specific transitions in more detail.

3.2.3 Impact of structure on function

Large-scale conformational changes among macrostates are suggested to impact the ion transition pathways in the selectivity filter, and therefore impact conduction. To quantify these differences, we partition the data by macrostate. For each macrostate, we compute PMF curves for key ion occupancy transitions. The OXXO $\rightarrow$ XXXO and XXOX $\rightarrow$ XOXX transitions were of particular interest because of their proposed role in the hard knock-on ion conduction mechanism.

The OXXO $\rightarrow$ XXXO transition describes the process of an ion moving from the intracellular channel cavity into the SF. This transition has been proposed to be most critical for the initiation of a conduction cycle. As can be seen in fig. 3.5A left, the forward and reverse rates for this process are all less than 3 kCal mol^{-1}. The *Up* state is found to have the highest forward kinetic barrier, followed by the two intermediate states, with the *Down* states having the lowest. The reverse rates for each macrostate are comparable, but the sample from the conformation with the alternative SF conformation, $Down_{\text{pinch}}$ displays a highly destabilized second binding well.

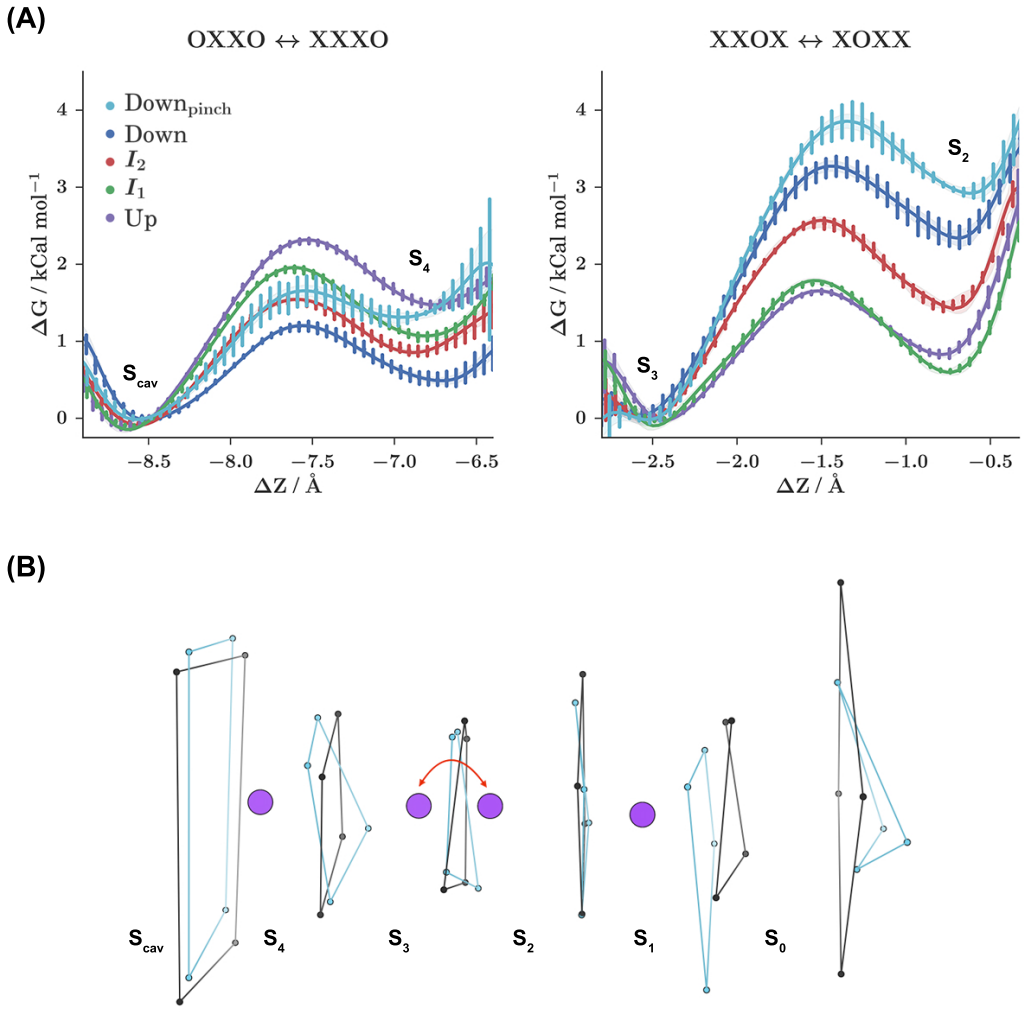

Figure 3.5: The energetics of ion occupancy transitions are conformationally dependent and can be analyzed using potential of mean force (PMF) calculations. For each conformational macrostate, PMF curves are computed for two key ion occupancy transitions. **(A)** Representative PMF curves for each macrostate. The *Down* macrostate favors the novel, pinched SF configuration and has been separated into *Down* (blue) and $Down_{\text{pinch}}$ (light blue) curves. ΔZ is progress along the ion transition path. **(B)** Illustration of the process sampled during the $XXOX \leftrightarrow XOXX$ PMF calculations for the Up_{Open} (black) and $Down_{Pinched}$ SF states. This process represents the oscillation of an ion between the S_3 and S_2 binding sites, while occupancy of S_4 and S_1 remains static.

The XXOX → XOXX transition describes the movement of an interior ion nearer to the SF exit, from S_3 to S_2. This transition was of interest because the conformations sampled from the MSM macrostates displayed variability in the structure of the SF on the extracellular side. This transition is closer to the region of SF structural variability. As shown by fig. 3.5A right, the forward and reverse rates for this process are all less than 4 kCal mol^{-1}. The size of the forward barrier is highest for the Down states, followed by I_2, while I_2 and the up state are the lowest. Once again, the $Down_{\text{pinch}}$ sample yields a highly destabilized second binding well.

Although the *Down* state represents the least conductive state, for the OXXO → XXXO transition it displays the smallest forward kinetic barrier, and for the XXOX → XOXX transition it displays the largest forward kinetic barrier. This illustrates how the conductance rate can be understood only through the consideration of multiple transition events. Barriers are not uniformly higher for the *Down* state. Instead, the effect is stronger for our analyzed downstream conduction transition, especially in the case where an conformational change in the SF has occurred.

For both transitions, conformational macrostate is found to have a marked impact on the forward kinetic barriers, while SF conformation is found to have a marked impact on the reverse kinetic barriers. Alternative SF conformations are observed with the highest probability for the *Down* state from the MD dataset (see fig. 3.2C and SI). The conformational changes associated with these alternative SF states are characterized by rotations of the backbone dihedral angles for the residues at the interface of the S_1/S_0 and S_0/S_{ext} binding sites (see fig. 3.3B). In the case that a backbone dihedral rotation did occur, the S_1 carbonyl oxygens no longer face the ion conduction pathway. Because the carbonyl oxygens of the residue backbone are responsible for coordinating the ions at each binding site, these rotations could render S_1 nonfunctional. Additionally, these rotations allow for the inward movement of the S_0 G176 (A) and G176 (B) residues. This movement appreciably restricts the diameter of the SF extracellular mouth, which could present a steric barrier to ion permeation.

Experimental studies show that TREK-2 exhibits a bidirectional allosteric coupling between the SF and the cytosolic proximal C-terminus [130], occurring via

movements of M4. This coupling has been proven, but the exact conformational changes of the SF assumed during this process remain unknown. Similar movements of M4 occur during mechanical activation. It is therefore reasonable to propose that there exists an analogous allosteric coupling between the SF and the *Up* and *Down* states. However, because transitioning between the *Up* and *Down* states also involves movements of the M2 and M3 helices, we do not claim that this coupling is precisely the same. Markov modeling reveals the conformation of a *Pinched* SF filter that is starkly favored by the *Down* intracellular conformation, and PMF calculations demonstrate that the *Pinched* SF disfavors extracellular ion rectification. We hypothesize that the intracellular rearrangements occurring as TREK-2 assumes the *Down* state transduces a conformational change of the SF residues at the interfaces of the S_1/S_0 and S_0/S_{ext} ion binding sites, resulting in reduced ion conduction.

3.3 Discussion

We applied molecular dynamics and MSM analysis to predict that a novel pinched state favored during membrane compression represents a low-conducting state of the TREK-2 K2P potassium channel. An initial hypothesis [123] suggested a direct connection between intracellular motion and channel activity. Membrane stretch and compression ("up" versus "down" macrostates) would beget high- and low-conducting functionality, respectively. This hypothesis has been suggested to be overly simplistic by experimental measurements [119] and the present study. After simulating the TREK-2 channel for 195 µs, we discover a novel selectivity filter conformation highly preferred by the *Down* state, but disfavored by the down-like I_1 and I_2 metastable intermediates and *Up* state. Steric consequences of this SF configuration suggest an impact on conductance. For further investigation, we use MSMs to model the behavior of ions in the selectivity filter conditioned on the conformational macrostate of the channel. Our ion dynamics models highlight similarities between the kinetic networks from the different conformations.In particular, the Köpfer [126] hard-knock mechanism is well supported by our simulations. The macrostate-conditioned models also suggest striking qualitative differences between ion microstate occupancies and

transition paths. To gain a quantitative understanding of these differences, we selected representative conformations from our unbiased MD calculations and computed PMF free energy profiles for crucial ion transitions, again partitioned by conformation. While the initial transition OXXO $\rightarrow$ XXXO shows the smallest barrier for down configurations (contrary to the idea that down conformations are less conductive), a later Köpfer pathway transition shows the highest barrier for down states, particularly the down state with pinched selectivity filter ($Down_{\text{pinch}}$). The $Down_{\text{pinch}}$ state is shown to have destabilized second wells in both important transitions studied, adding further evidence to the reduced activity of this conformation. Our findings agree with the growing consensus that intracellular motion is not directly coupled to conduction. Instead, conformational states affect selectivity filter conformations. We contribute a structural basis for this idea by the discovery of a novel pinched SF state involving rotations around Y175/F284. We encourage future structural studies to target this novel state.

3.4 Methods

3.4.1 Molecular Dynamics

The simulations were started from four crystal structures of TREK-2 (PDB codes: 4xdj, 4bw5, 4xdl, 4xdk) [123] Crystal structures were solvated in a 4:1 POPC:POPE lipid bilayer membrane using charmm-gui [131]. Simulations inputs were generated using the tip3p [27], lipid14 [30] and AMBER14SB [26] forcefields. Equilibration was performed with restrained proteins and lipids, by slowly increasing temperature from 0K to 300K in Amber 14 [132]. The canonical forcfield was generated as a prmtop file using Amber's tLeap program. Production simulations were run at 300K constant temperature and 1 bar constant pressure with constrained hydrogen bond lengths and rigid waters. Simulations were run on Folding@Home [48] using OpenMM 6.2 [41] and Gromacs 4.5.3 [42]. OpenMM simulations exploit GPUs while Gromacs is faster for CPUs, and the two codes were used simultaneously. Production simulations used a 2.0 fs timestep and the particle mesh Ewald (PME) method for

electrostatics. There is not a consistent choice of thermostat and barostat implemented in these two codes. Rather, we used the recommended NPT settings for each code. OpenMM simulations used a Langevin integrator with a 1.0 / ps friction coefficient and a (fully) anisotropic Monte Carlo barostat with a frequency of 25 steps. OpenMM natively reads the prepared forcefield prmtop files. For Gromacs, ParmEd (https://github.com/ParmEd/ParmEd) was used to convert the prmtop topology to the Gromacs "top" format. Gromacs simulations used the velocity-rescaling-with-stochastic-term temperature coupling with a time constant of 2.0 ps and a Parrinello-Rahman barostat with time constant of 2.0 ps.

PMF calculations were run with Gromacs 5.0.5 on Bluewaters and reweighted with WHAM [133]. Analysis was performed with MDTraj 1.5 [134] and MSMBuilder 3.4 [135]. Conformations were visualized with VMD [112] and Chimera [136].

3.4.2 Markov State Modeling

The pore region of the protein was featurized by taking all respective chain A to chain B distances between residues 1-24, 112-189, 230-256. These features were transformed into kinetic coordinates with tICA (lag-time = 1 ns). Conformations were clustered into 500 states using the first 3 dimensions of tIC coordinates via the Mini-batch KMeans algorithm. An MSM was fit using MLE at a lag-time of 20 ns. Conformational microstates were lumped into four macrostates with PCCA+. Ion occupancies were featurized by enumerating the 16 possible occupancy states (from four binding sites). An MSM was fit using MLE, again at a lag-time of 20 ns.

3.4.3 PMF Calculations

PMF initial conformations were obtained through PCA analysis on the protein dihedral angles for each MSM state. These conformations were aligned such that the SF was centered about the z-axis. Pulling simulations were performed in order to sampling the reaction coordinate for each ion transition. Five umbrella sampling simulations were run for each PMF, one in each ion binding well, one at the transition state, and one between each binding well and the transition state. This arrangement

resulted in a window spacing of approximately .05 nm. These simulations were run with a spring constant of 8000 kCal mol^{-1} for 6 ns, with the first 1 ns treated as equilibration. Convergence was determined by monitoring the overlap between simulation histograms. For each transition and macrostate, at least 8 conformations were analyzed. PMF error bars were computed over these conformations. The PMF reaction coordinate was defined as the projection of the distance from the SF center of mass to the transitioning ion onto the SF symmetric axis. All PMF related simulations were run using Gromacs 5.0.5 [42]. RMSD of protein conformation to initial structure was monitored during umbrella sampling simulations to ensure there were no artifacts caused by the umbrella potential.

Acknowledgments

For this chapter, we thank John Mathias and Mark Bunnage for their encouragement and support. We thank Arianna Peck and Brooke Husic for critical feedback on this chapter. We acknowledge funding from NIH grants U19 AI109662 and 2R01GM062868. This work was funded by Pfizer's Science and Technology budget. We thank the Folding at Home donors who contributed to this project (PROJ9712, 9761 and 9762). This work is part of the "Petascale Simulations of Biomolecular Function and Conformational Change" PRAC allocation support by the NSF (award number 1439982).

Author contributions for this chapter: All authors designed the research. MPH and KAM performed the research. RAD and VSP supervised the research. All authors edited this chapter.

Disclosure statement for this chapter: VSP is a consultant and SAB member of Schrodinger, LLC and Globavir, sits on the Board of Directors of Omada Health, and is a General Partner at Andreessen Horowitz. Other authors declare no competing financial interest.

Chapter 4

MSMBuilder: Statistical Models for Biomolecular Dynamics

This chapter is adapted from Matthew P. Harrigan, Mohammad M. Sultan, Carlos X. Hernández, Brooke E. Husic, Peter Eastman, Christian R. Schwantes, Kyle A. Beauchamp, Robert T. McGibbon, and Vijay S. Pande. MSMBuilder: statistical models for biomolecular dynamics. *Biophys. J.*, 112(1):10–15, 2017. doi: 10.1016/j.bpj.2016.10.042 [135], re-used with permission.

Abstract

MSMBuilder is a software package for building statistical models of high-dimensional time-series data. It is designed with a particular focus on the analysis of atomistic simulations of biomolecular dynamics such as protein folding and conformational change. MSMBuilder is named for its ability to construct Markov State Models (MSMs), a class of models that has gained favor among computational biophysicists. In addition to both well-established and newer MSM methods, the package includes complementary algorithms for understanding time-series data such as hidden Markov models (HMMs) and time-structure based independent component analysis (tICA). MSMBuilder boasts an easy to use command-line interface, as well as clear and consistent abstractions through its Python API (application programming interface). MSMBuilder is developed with careful consideration for compatibility with the broader machine-learning community by following the design of `scikit-learn`. The package is used primarily by practitioners of molecular dynamics but is just as applicable to other computational or experimental time-series measurements.

4.1 Introduction

Molecular dynamics (MD) is a powerful probe into atomistic dynamics. Recent advances in technology (specialized hardware [52] or commodity GPUs [57]) and strategies (massively distributed architectures [13, 48, 51]) enable simulations to reach larger size and longer timescales. Increasing quantities of raw data require novel and sophisticated analysis techniques [19]. Markov state models (MSMs) have gained favor for drawing interpretable conclusions from time-series data [19, 60, 61, 94]. Briefly, MSMs model dynamic systems using a set of discrete states and pairwise transition rates. From these models, the researcher can compute observables of interest and make predictions. These models are statistically rigorous and easy to interpret. Furthermore, MSMs are able to stitch together many independent simulation runs, allowing researchers to fully exploit distributed computing.

The idea of describing a system by its states and rates is natural for chemists and

biologists, but the estimation of states and rates from finite data (perhaps molecular dynamics) is not obvious. From the introduction of MSMs to the biophysics community, algorithmic improvements for constructing MSMs and computing observables have been the focus of intense study. The practical implementation of these algorithms has spawned several historical packages for MSM construction [137–139]. Each of these packages was tied strongly to the best practices in MSM construction of the time. Due to the fast-moving research around MSMs, software re-writes were common [43, 140].

We introduce MSMBuilder 3, a community-driven, open source software package for constructing MSMs. MSMBuilder offers a curated selection of MSM construction algorithms based on modern advances in the field. MSMBuilder is implemented in the Python programming language with performance-critical components written in C. It exposes an extensible API modeled after that of `scikit-learn`. The modular design ensures MSMBuilder 3 is adaptable to future improvements in MSM construction. The package can be invoked directly from Python or via the command line.

Through two instructive examples, we showcase the capabilities of MSMBuilder. In the first, we use MSMBuilder to analyze a biological system of interest from a dataset composed of more than 20,000 trajectories. This example builds a single MSM using methods unavailable in previous tools. Due to rapid advances in MSM methods, a variety of modeling choices are now available to researchers. In the second example, we demonstrate how MSMBuilder's implementation of scoring functionals can be used to choose among these methods.

4.2 Example: Constructing an MSM

MSMBuilder allows rapid analysis of large molecular dynamics datasets. In this example, we construct an MSM of a kinase molecule. Kinases are critical enzymes that control cellular pathways. Malfunctions of kinases have been linked to many different cancers [141]. Here, we use MSMBuilder to study the c-Src kinase, a regulator of cellular growth [98], and demonstrate that the resulting MSM can capture activation

dynamics. Understanding the activation process reveals atomistic, kinetic, and thermodynamic insights into the protein's conformational heterogeneity, which can help design better therapeutics.

Broadly, the procedure for constructing an MSM is to define a set of states and then estimate transition rates among those states. Before beginning model construction, researchers must obtain time-series data they wish to model. Usually, this is the output of a molecular dynamics engine (MSMBuilder supports nearly every MD trajectory file format [134]), but it could also be experimental time-series measurements. For this example, we use a previously-generated MD dataset of the c-Src kinase, publicly available from the Stanford Digital Repository (SDR) [1].

The first step of model construction is to transform the raw Cartesian coordinates into vector features that are invariant to translation and rotation (fig. 4.1, step 1). Here, we project our trajectory frames onto the dihedral angles created by each set of four consecutive alpha carbons (α angles) [9]. This reduces the dimensionality of the data from 12,693 Cartesian coordinates to 518 features. The appropriate featurization depends on the particular system under study (see section 4.3). MSMBuilder offers a collection of featurization strategies with a unified interface. Popular features include backbone and side chain dihedrals (through the `DihedralFeaturizer` class), heavy atom or C_α contact distances (`ContactFeaturizer`), distance of reciprocal interatomic distances (`DRIDFeaturizer`) [81], and root mean squared deviation to a set of structures (`RMSDFeaturizer`). There are additional utilities for concatenation of multiple choices of features and feature scaling.

The second step in MSM construction projects structural features onto a lower-dimensional subspace (fig. 4.1, step 2). This improves the statistical qualities of subsequent steps, but may discard important information if the projection is not carefully chosen. Time-structure based independent component analysis (tICA) finds a set of "slow" (high autocorrelation) coordinates. In practice, this dimensionality reduction has proven to be very useful for capturing slow, biophysical conformational change [70, 71]. In this example, we reduce the dimensionality of our kinase data from 518 dihedrals to 5 tICA coordinates. MSMBuilder includes support for similar algorithms

[1] Available here: https://goo.gl/LLchMT. For simulation details, see [98].

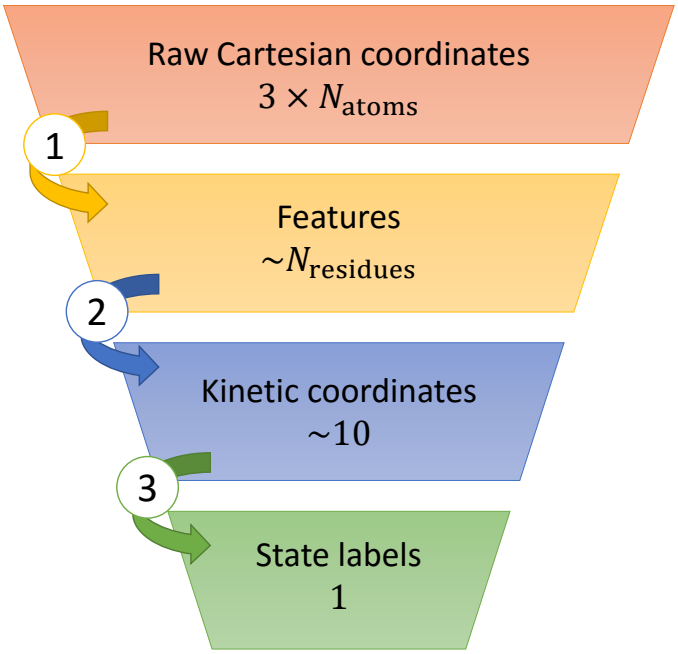

Figure 4.1: **Data transformations and their dimensionality.** Markov state models (MSMs) partition dynamical data into a set of states and estimate rates between them. A typical pipeline for state definition consists of a series of transformations (indexed by circled numbers) between representations of the data. Each step projects a higher dimensional representation onto a lower dimensional representation. The approximate dimension of each representation is reported below the representation name. Although not traditionally thought of as a dimensionality reduction, clustering (step 3) reduces each frame to a single integer cluster label.

(`SparseTICA` [74]) as well as general manifold learning algorithms like principal components analysis (`PCA`), `SparsePCA`, or `MiniBatchSparsePCA`. Prior to 2013, this step was not available for model construction. Accordingly, software available at the time could not easily be extended to accommodate tICA intermediate processing. The design of MSMBuilder 3 permits arbitrary addition, subtraction, and re-ordering of data transformation steps.

Next, we define the states of our MSM by grouping conformations which interconvert rapidly (fig. 4.1, step 3). For the c-Src kinase, we employ the `MiniBatchKMeans` [111] clustering algorithm to parition our data into 200 microstates. We note that our data has been reduced from 5 tICA coordinates to one integer cluster label per frame. The prior dimensionality reduction permits using off-the-shelf clustering algorithms. Accordingly, MSMBuilder supports K-Means like clustering algorithms (`KCenters`, `KMedoids`, and `MiniBatchKMedoids`), and hierarchical clustering.

With our states defined, we proceed to estimate the rates among them. As the final model construction step, we learn a continuous-time MSM [87] from our labeled trajectories. We have chosen to use a continuous-time MSM to directly estimate transition rates; we could have alternatively built a traditional MSM (to estimate transition probabilities) or a hidden Markov model (HMM). We direct interested readers to a more thorough application of HMM modeling to the c-Src dataset [142]. The relevant Python code for constructing this MSM is shown in fig. 4.2. Complete, executable code is available in the SI as an IPython [143] notebook.

To draw interpretable conclusions from our data via Markov modeling, we query the model. For c-Src, we use MSMBuilder to relate model behavior to biological function. We present a log-scaled 2D histogram (fig. 4.3a) of the trajectories projected onto the two dominant slow processes, or "tICs", from our tICA model. We then sample the centroids of states (shown as pink and black stars) in low free energy regions to visualize representative configurations in three dimensions [112] (fig. 4.3c and d). The dominant tIC (x-axis) highly correlates with the activation of the kinase. Kinase activation requires the unfolding of the activation loop (red) and an inward swing of the catalytic helix (C-helix). The inward rotation of the helix coincides with the switching of hydrogen bonding pair from Glu-Arg to Glu-Lys (licorice). We

```
feat = AlphaAngleFeaturizer(sincos=True)
ds = dataset('trajectories/*.lh5')
alphas = feat.fit_transform(ds)

tica = tICA(lag_time=500, n_components=10)
ticas = tica.fit_transform(alphas)

kmeans = MiniBatchKMeans(n_clusters=200)
assignments = kmeans.fit_transform(ticas)

msm = ContinuousTimeMSM(lag_time=400,
                        ergodic_cutoff='on')
msm.fit(assignments)
dump(msm, 'msm.pickl')
```

Figure 4.2: **Sample MSM code.** MSMBuilder balances a powerful API (application programming interface) with ease of use. A sample workflow is shown here using the Python API. Following the successful model of the broadly-applicable `scikit-learn` package, each modelling step is represented by an estimator object which operates on the data. Here, the `AlphaAngleFeaturizer` transforms raw coordinates into α angles. The output of this transformation is fed into the `tICA` dimensionality reduction, `MiniBatchKMeans` clustering algorithm, and finally into the `ContinuousTimeMSM` model. MSMBuilder provides a litany of utility functions for dealing with large molecular dynamics datasets for I/O. While this example shows the Python API, MSMBuilder is fully functional from the command line with an intuitive 1-to-1 correspondence between Python estimator objects and command-line commands.

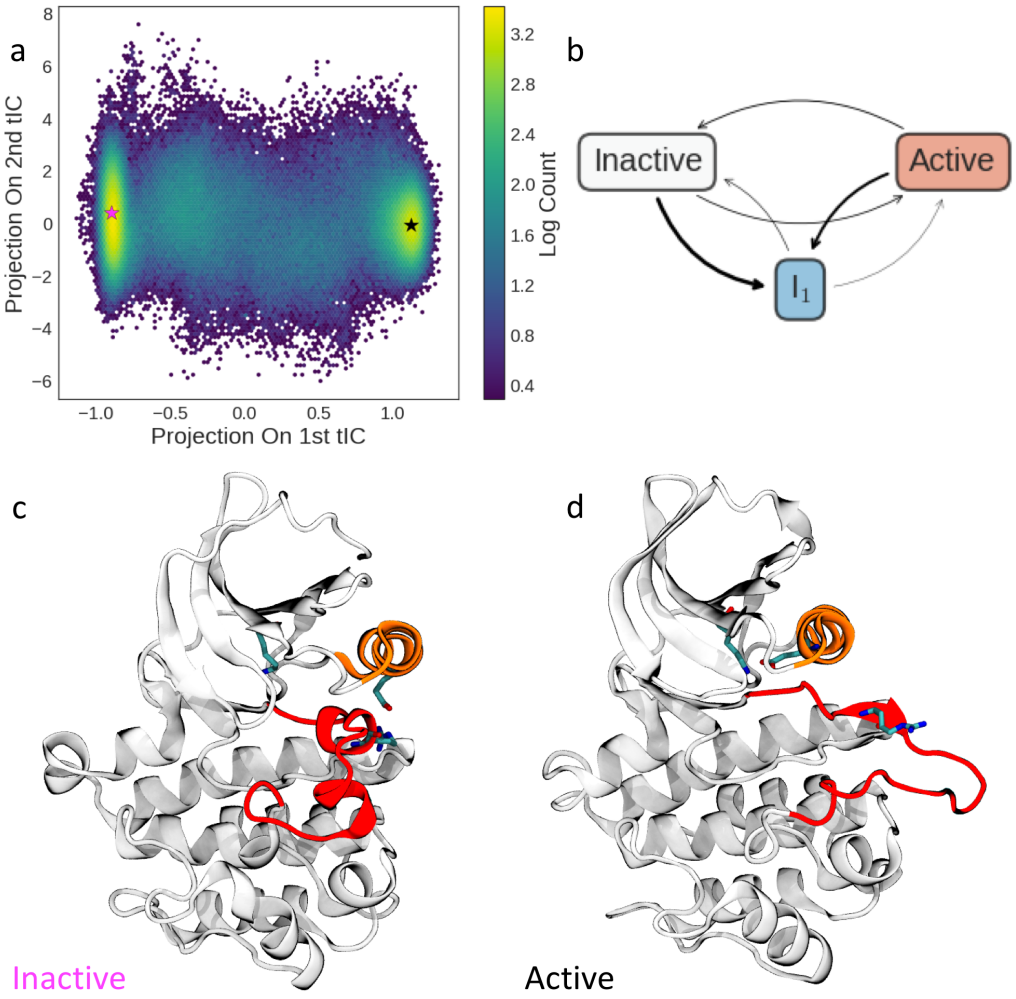

Figure 4.3: **c-Src kinase MSM.** MSMBuilder constructs interpretable models from large datasets. This figure shows a 2D-histogram for the Src kinase from tICA-MSM analysis projected onto the dominant modes of a tICA model (a). A simple macrostate model of the dynamics shows the presence of an intermediate state I_1 connecting the inactive and active states (b). The arrow thickness corresponds to the rate of transitions. The model indicates that the active state (red) is the most stable state followed by the inactive and intermediate states (gray and blue, resp.). The analysis discovers a coordinate (the first tIC) between the known active and inactive conformations. Representative structures are selected from MSM states and show the conformational differences between the two basins. The unfolding of the activation loop (red helix) forms a catalytically active Src capable of initiating and regulating downstream signaling pathways (c and d).

investigate the dynamics between the active, inactive, and intermediate macrostates by applying Robust Perron Clustering Analysis (PCCA+) to our MSM. PCCA+ is a spectral clustering method, which lumps MSM states into an arbitrary number of metastable macrostates, facilitating qualitative analysis of rates and populations among biologically-relevant macrostates [144]. The rates among three macrostates are shown by the thickness of arrows in fig. 4.3b. Further options of querying the model (not shown here but available in MSMBuilder) include computation of relaxation timescales, transition path theory analysis [145–147], and generation of synthetic trajectories for visual inspection.

The assortment of modeling options such as the choice of featurizer, the use of dimensionality reduction, and the selection of the clustering algorithm, along with any associated internal parameter choices, presents the modeler with a motley of modeling decisions and tunable parameters. In the next section, we show how a scoring metric for MSMs can provide the modeler with a unbiased protocol for determining which parameters are suitable given a set of MD trajectories.

4.3 Example: Selecting Hyperparameters

Historically, the heuristic choice of hyperparameters—choices of protocol—rendered MSM construction as much of an art as a science. It is clear from section 4.2 that there is an abundance of algorithms available in MSMBuilder. In this instructive example, we use a scoring functional to select the best models.

Noé and Nüske [65] introduced a variational principle that formalized the definition of a "good" MSM. In keeping with inspiration from the broader machine learning community, MSMBuilder extends this formalism in the context of cross-validation through the work of McGibbon and Pande [79]. The resulting generalized matrix Rayleigh quotient (GMRQ) score offers an objective way to pick the best model (i.e. the appropriate modeling choices) from the given data. Briefly, the GMRQ measures the ability of a model to capture the slowest dynamics of a system. The variational principle states that approximating the full phase space by discrete states will always yield dynamics that are too fast. The GMRQ score is a summation of the leading

eigenvalues of the model and therefore provides a measure of "slow-ness". A higher score means the model is closer to the variational bound, and therefore should be preferred over lower scoring models.

```
features = FeatureSelector([
    ('diheds', DihedralFeaturizer(types=['phi', 'psi']),
    ('contacts', ContactFeaturizer(scheme='ca'))])
pipeline = Pipeline([
    ('featurizer', features),
    ('tica', tICA(n_components=2)),
    ('cluster', MiniBatchKMeans(n_clusters=250)),
    ('msm', MarkovStateModel(n_timescales=3))])
 ss = ShuffleSplit(28, n_iter=50, test_size=0.5)
 cv = GridSearchCV(pipeline, cv=ss, param_grid={
    'featurizer__which_feat': ['contacts', 'diheds'],
    'tica__lag_time': [1, 4, 16, 64]})
 cv.fit(trajs)
```

Figure 4.4: **Sample GMRQ code.** MSMBuilder seamlessly interoperates with the broader Python ecosystem. In this code sample, we use `scikit-learn` for algorithm-agnostic data processing and MSMBuilder for biophysics-oriented time-series algorithms with the goal of selecting model hyperparameters. Our analysis pipeline is similar to that of section 4.2 but with a choice of features (between dihedrals and contact distances) and tICA lag times (among 1, 4, 16, and 64 steps). The `ShuffleSplit` cross-validation scheme runs 50 iterations of equal partitioning of the 28 trajectories between train and test sets, and we perform a full grid-search over parameter choices. We can plot the distribution of scores vs. parameters as in fig. 4.5.

In this example, we use the GMRQ score under cross-validation to evaluate the relative merit of enumerated hyperparameter values when constructing a model for the Fs peptide [148]. The relevant code in fig. 4.4 sets up a choice between two structural features (dihedral angles or contact distances) and a choice among tICA lag times. We perform shuffle-split cross-validation by randomly assigning the 28 trajectories to either the training set or test set. The MSM is learned on the training set and scored on the test set. By concealing the training data during scoring, cross-validation guards against overfitting (overconfidence in excessively complex models). The trajectories

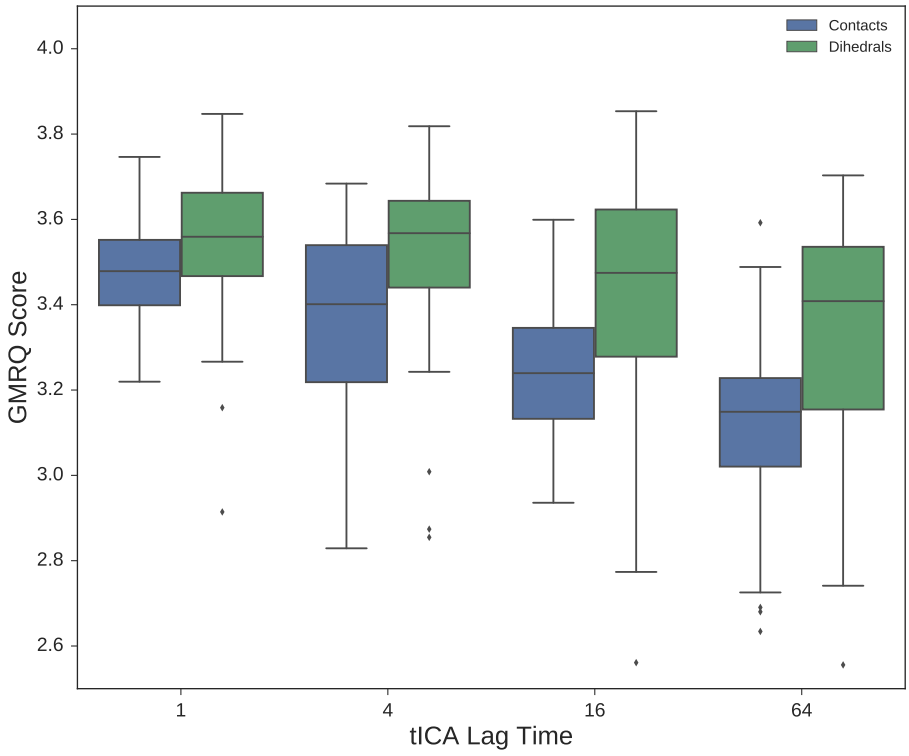

Figure 4.5: **GMRQ parameter selection.** MSMBuilder offers robust machinery for selecting hyperparameters that cannot *a priori* be learned from the data. Here, we perform shuffle-split cross validation over choices of featurization and tICA lag time. Historically, these parameters were chosen heuristically. With the advent of the GMRQ score and its implementation in MSMBuidler, we can choose these parameters in a statistically rigorous way. Here, we plot the distribution of scores for each set of of model parameters. Note that a higher score is generally an indication of a more predictive model. In this example, we find that featurization with dihedral angles at a lag time of 4 steps has highest median score and recommend this hyperparameter set to be chosen for the final model.

are re-shuffled and this process is repeated to compute an average score for a given set of hyperparameters. The scores for each of the 50 cross-validation splits are plotted in a box plot in fig. 4.5. The dihedral angle featurization with a lag-time of 4 steps gives the best model in this search space. A simple grid search as performed in this example can become intractable as the number of hyperparameters (i.e. the dimension of the search space) increases. We direct interested users to Osprey [149], a tool for hyperparameter optimization with a variety of search strategies and support for parallel computation. Osprey interoperates with any `scikit-learn` estimator including those in MSMBuilder.

This example leverages the `Pipeline`, `ShuffleSplit`, and `GridSearchCV` machinery from `scikit-learn`. Additionally, MSMBuilder uses this library internally for generic machine learning algorithms such as clustering or PCA. We note that such general algorithms do not need to be reinvented and re-programmed by the biophysics community. By delegating some development effort to this widely-used machine learning library, we ensure that the development of MSMBuilder is focused on biophysical algorithms and considerations. This advantage offers rapid adoption of the latest algorithms which have demonstrated improved ability to build MSMs (e.g. [79]) and a larger community for code maintenance and longevity.

4.4 Conclusions

MSMBuilder 3 is a powerful and accessible software package for drawing interpretable conclusions from time-series data. We used two examples to demonstrate how MSMBuilder can make sense of a molecular dynamics dataset consisting of thousands of trajectories in a highly automated and statistically robust way. In the first example, we construct a "vanilla" MSM and show how MSMBuilder enables the construction of interpretable models that expose the connection between biological function and structural dynamics. We highlight the breadth of relevant algorithmic choices for featurization, normalization, dimensionality reduction, clustering, and MSM modeling. In the second instructive example, we acknowledge that the explosion of choices in parameters

and protocol can be overwhelming. We use the GMRQ score and off-the-shelf cross-validation machinery to do a simple grid search over tunable parameters to evaluate the relative merit of many MSMs built on the same MD dataset of a small protein. Since cross-validation is not a technique unique to biophysics, we leverage the greater Python machine learning ecosystem for this example.

More broadly, MSMBuilder's power and clarity is derived from its integration with the machine learning community at large. Our power to focus on developing methods bespoke to biophysics and time-series analysis comes from exploiting general-purpose algorithms implemented by respective experts. The clarity of MSMBuilder's API is due in large part to the massive amount of effort and skill put into the design of `scikit-learn`'s API. As distributed computing and Markov modeling continue to become more prominent, MSMBuilder offers a sustainable, extensible, powerful, and easy-to-use set of Python and command-line tools to help researchers draw meaningful conclusions from their data.

4.5 Availability

MSMBuilder documentation and installation is available at http://msmbuilder.org. The source code is available under the open-source LGPL2.1 license and is accessible at http://github.com/msmbuilder/msmbuilder. The current release at time of writing is version 3.5 [135]. Complete examples can be found as IPython notebooks in the supporting information and at http://github.com/msmbuilder/paper.

Acknowledgments

For this chapter, we extend thanks to all our contributors including Stephen Liu, Patrick Riley, Steven Kearnes, Joshua Adelman, and Gert Kiss. We acknowledge funding from NIH grants U19 AI109662 and 2R01GM062868. MMS acknowledges support from NSF-MCB-0954714. CXH acknowledges support from NSF GRFP (DGE-114747). KAB acknowledges support from NIH grant P30CA008747, the Sloan

Kettering Institute, and Starr Foundation grant I8-A8-058. We acknowledge members of the Chodera, Pande, and Noë labs for helpful discussions. We thank Ariana Peck for invaluable feedback on this chapter.

Author contributions for this chapter: MPH, MMS, CXH, and BEH wrote the chapter. MPH, MMS, CXH, BEH, PE, CRS, KAB, RTM, and VSP edited the chapter. MPH, MMS, CXH, BEH, PE, CRS, KAB, and RTM wrote the software. VSP supervised the project.

Disclosure statement for this chapter: KAB is currently an employee of Counsyl, Inc. RTM is currently an employee of D.E. Shaw Research, LLC. VSP is a consultant of Schrodinger, LLC and a member of its scientific advisory board.

Chapter 5

Landmark Kernel tICA for Conformational Dynamics

This chapter is adapted from Matthew P Harrigan and Vijay S Pande. Landmark kernel tICA for conformational dynamics. bioRxiv:10.1101/123752 [150], which has been submitted for publication and is available as a pre-print on bioRxiv.

Abstract

Molecular dynamics simulations of biomolecules produce a very high dimensional time-series dataset. Performing analysis necessarily involves projection onto a lower dimensional space. *A priori* selection of projection coordinates requires (perhaps unavailable) prior information or intuition about the system. At best, such a projection can only confirm the intuition. At worst, a poor projection can obscure new features of the system absent from the intuition. Previous statistical methods such a time-structure based independent component analysis (tICA) and Markov state modeling (MSMs) have offered relatively unbiased means of projecting conformations onto coordinates or state labels, respectively. These analyses are underpinned by the propagator formalism and the assumption that slow dynamics are biologically interesting. Although arising from the same mathematics, tICA and MSMs have different strengths and weaknesses. We introduce a unifying method which we term "landmark kernel tICA" (lktICA) which uses a variant of the Nyström kernel approximation to permit approximate non-linear solutions to the tICA problem. We show that lktICA is analogous to MSMs with "soft" states. We demonstrate the advantages of this united method by finding improved projections of (a) a 1D potential surface (b) a peptide folding trajectory and (c) an ion channel conformational change.

5.1 Introduction

Protein dynamics are responsible for carrying out the functions of life. Molecular dynamics (MD) is a powerful tool to understand dynamics on an atomistic scale by modeling and simulating each atom as a classical particle. Thanks to increasing computer power, simulations have modeled larger systems at longer timescales [20]. Among other challenges, drawing interpretable conclusions from these increasingly high-dimensional and increasingly lengthy time-series data sets has been called in to focus [19]. Markov State Models (MSMs) [59, 60, 64] and time-structure based independent component analysis (tICA) [68, 70, 71] have been introduced to address

this challenge. These models are backed by a useful formalism: the transfer operator [63, 64]. This has given rise to a variational approach [65, 66] which can be used to select the best models (hyperparameters) for a given dataset [79]. Furthermore, it has been shown that tICA and MSMs solve the same problem in this formalism, namely numerical estimation of the transfer operator. The difference between the two is in choice of basis set [66, 73]. In particular, MSMs construct indicator-function basis functions over microstates, and tICA uses linear basis functions in the input coordinates. It is noteworthy that tICA was introduced primarily as an intermediate processing step for MSM construction, and only sometimes connected by this formalism [70].

Figure 5.1 shows an example potential energy function $V(x)$ yielding four wells and a large barrier between the leftmost and rightmost wells. For a simple, low-dimensional potential energy surface, we can analytically calculate the equilibrium distribution $\mu(x) = \frac{\exp[-\beta V(x)]}{Z}$, where Z is the partition function $\sum \exp[-\beta V(x)]$. For a high-dimensional potential energy surface where quadrature integration is impossible we must use a different approach. For example, we can use molecular dynamics to sample conformations of a protein in solvent subject to a forcefield. A natural way of estimating the distribution $\hat{\mu} \approx \mu$ is by constructing a histogram (fig. 5.1a). Specifically, we partition the data into bins and count the number of data points in each bin. In addition to thermodynamic properties, we may also be interested in kinetic properties. We expect Brownian dynamics on this potential to yield three slow processes corresponding to transitions among the four wells. The slowest process is transfer of flux from the left two wells to the right two wells (fig. 5.1b). The transfer operator (and related propagator) formalism sets up a framework for estimating kinetics from data. The MSM is the kinetic analog of the thermodynamic histogram: we partition the data into discrete states and count transitions between states. Both the histogram (thermodynamics) and MSM (kinetics) describe smooth data with jagged bins, making estimation highly sensitive to shot noise, poor statistics, and binning protocol. To overcome this limitation, one might consider using a "smooth" estimator. One smooth analogue to the histogram is the kernel density estimator (KDE). In a similar vein, we seek a more sophisticated technique for smoothly estimating

kinetics.

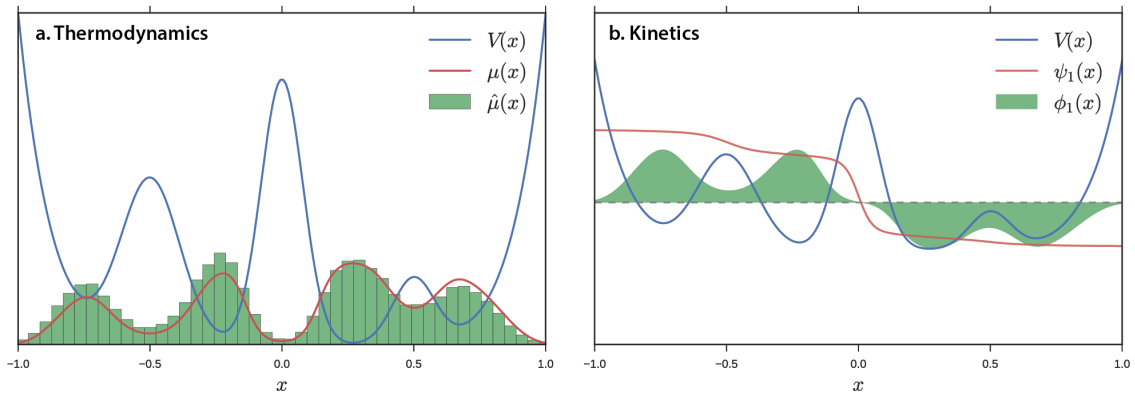

Figure 5.1: We can characterize a biophysical system by (a) its thermodynamics and (b) its kinetics. Thermodynamic properties can be estimated without regards to time. Here, we estimate the equilibrium distribution by computing a histogram. We model kinetics by estimating the eigenfunctions of the transfer operator (ψ) or μ-weighted propagator (ϕ). Here we plot the slowest dynamical eigenfunction, which represents moving from the left two basins to the right two basins. For this 1D potential energy function, the analytic eigenfunctions are tractable. In analogy to the histogram in (a), we seek a numerical method to evaluate ψ or ϕ from finite data.

tICA is an alternative method for modeling kinetics. Whereas MSMs use indicator functions (bins) to estimate ψ, tICA uses linear functions. This corresponds to using hyperplanes (for our 1D potential, a line) to estimate ψ, see fig. 5.2d. Typically for biomolecules, one extracts vector features such as dihedral angles or contact distances to serve as inputs to tICA. Although no longer subject to the limitations of discrete bins (as in MSMs), Schwantes and Pande [73] noted that the linearity of tICA harshly constrains the solutions. To introduce non-linearity, the authors borrowed a technique from machine learning and introduced a kernel trick. By re-writing the tICA problem only in terms of inner products, researchers can solve the tICA problem in an arbitrarily large, expanded space relative to the original representation *without* explicitly transforming the original representation into this space. By using an appropriate kernel function such as a Gaussian kernel (a member of the class of radial basis functions (RBFs) and sometimes used synonymously with RBF kernel),

the implicit representation is infinitely large, containing every power of input coordinates (per Taylor expansion of the exponential function). The authors showed that this method could capture the important, slow degrees of freedom from a simulation using many fewer, non-linear coordinates. There are drawbacks to kernel tICA that have precluded it from wide adoption: (1) it is sensitive to hyperparameters. (2) It scales very badly with amount of data. Presciently, Schwantes and Pande [73] noted that developments in the kernel learning community could be applied here. In that spirit, we introduce a variant of the Nyström approximation to address the problems identified above (summarized in Table 1). We also show that this method connects MSMs and tICA in a novel way.

Table 5.1: Problems with existing solutions motivating the present work

	MSM	tICA	Kernel tICA
Non-linear	Yes	No	Yes
Smooth	No	Yes	Yes
Tractable	Yes	Yes	No

5.2 Method

The Nyström method of approximation can be used to speed kernel-trick computations [151, 152]. Instead of computing the Gram matrix $K \in \mathbb{R}^{n \times n}$ (where n is the number of data points) in full, we approximate it by $\tilde{K} = K_{n,m} K_{m,m}^{-1} K_{m,n}$ where $K_{n,m}$ is constructed by randomly selecting m columns of the original matrix. Williams and Seeger [151] showed that we can choose $m \ll n$ in practice. Since the limiting step is diagonalizing the Gram matrix, the computational complexity can be reduced considerably from $O(n^3)$ in amount of data to $O(m^2 n) \approx O(n)$. We improve upon this approximation by selecting columns (corresponding to data points) of the original Gram matrix according to the result of a (perhaps rough) clustering of the data. This comes at the expense of slightly additional computational cost, but improved coverage of the observed data and added interpretability (see section 2.2). We name the m

data points corresponding to the selected columns "landmark points". In practice, we transform the input time-series data by explicitly computing the kernel function to each landmark point and then apply the ordinary linear tICA algorithm in this new space. In this way, the kernelized distance to the landmark points can be thought of as a set of features not dissimilar to dihedral angles or contact distances.

5.2.1 Kernel Function and Distance Metric

Any suitable kernel function can be used with this formalism. In this work, we have constructed our examples with the Gaussian RBF kernel, which is popular in the machine learning community due to (1) its infinite Taylor expansion, giving rise to an infinite dimensional latent feature space and (2) its interpretation as a similarity measure with range [0,1]. It admits one hyperparameter: the basis function width, σ.

$$k(x, y) = k(\|x - y\|) = \exp\left[-\frac{\|x - y\|^2}{2\sigma^2}\right]$$

For simple problems in an explicit Euclidean vector space, like the toy potential in section 3.1, the choice of distance metric is simple. A ℓ^2 norm will suffice. In protein simulations, however, Euclidean norm in the raw Cartesian coordinate space is rarely the best metric of distance. Instead, we require a distance metric that (1) respects translational and rotational degrees of freedom and (2) ignores highly varying but functionally irrelevant conformational changes like hydrogen atom vibrations and solvent motion. Popular distance metrics for MSM construction typically involve transforming the protein Cartesian coordinates into a set of internal coordinates like backbone dihedral angles or distance pairs, and then using the Euclidean norm in this space.

Alternatively, the root mean squared deviation (RMSD) distance metric is a very natural way of thinking about protein conformations. It has been used since the inception of atomistic simulation to describe the difference between conformations (roughly) as the average difference between atomistic coordinates. It takes into account the rotational and translational symmetry of conformations by centering and conformations and reporting the minimum value over rotation. It is a protein-agnostic

algorithm that can be applied to any system under study, in contrast with particular distance pairs or dihedral angles which vary in importance between systems. As a disadvantage: RMSD is *only* a distance metric and does not embed a Euclidean vector space. It is impossible to use off-the-shelf machine learning techniques like KMeans clustering or principal component analysis (PCA) or specialized techniques like linear tICA. For this reason, much of the algorithmic advances in MSM construction has focused on vector space features. Landmark kernel tICA does *not* require vector space features (although it works naturally with them as well). We can once again use RMSD in our analysis pipeline.

RMSD is a measure of local distance. With small values ($<3\,\text{Å}$) the researcher can be confident that the two conformations are similar. As RMSD values become large ($>10\,\text{Å}$) the researcher can say that the conformations are different, but the degree of difference is not sensitive to RMSD changes and the ways in which conformations are different cannot be deduced. This gives rise to a rule of thumb for landmark kernel tICA. can be chosen to be around the cutoff of RMSD's utility ($\sim 3\,\text{Å}$). Other bandwidth selection algorithms from histogram or KDE construction may be applicable here.

5.2.2 Connection to MSMs with soft states

The kernelized distances to landmark points vary between 0 (entirely dissimilar to landmark point) and 1 (equal to the landmark point). We introduce the notion of MSM "states" defined by a centroid conformation and interpret the kernelized distances as partial occupancies in those states. In practice, MSMs are constructed by filling in a "counts matrix" of transitions between labeled states. This is equivalent to computing the time-lagged correlation matrix of state-occupancy vectors of the form $|\mathbf{k}\rangle = \{0, \ldots, 0, 1, 0, \ldots, 0\}$ zero everywhere except at the kth position. Define $\mathbf{K}^{(t)}$ to be the set of these vectors over time starting at time t. Then

$$\text{Counts}\,(\tau) = |\mathbf{K}^{(0)}\rangle\,\langle\mathbf{K}^{(\tau)}|$$

Now take $|\mathbf{k}'\rangle$ to be soft occupancies between zero and one (e.g. the result of

kernel featurization). With sufficiently separated landmark points, the vector will take on a form $|\mathbf{k}'\rangle = \{\ldots, 0.1,\ 0.9,\ 0.1, \ldots\}$, i.e. all entries close to zero except at the kth position, where it is close to one. The analogous computation:

$$|\mathbf{K}'^{(0)}\rangle\langle\mathbf{K}'^{(\tau)}| = \text{time lagged correlation}$$

gives the time-lagged correlation matrix, which—when properly normalized by the covariance—gives the tICA eigenvalue problem. This mathematical relation lets us view the landmark kernel tICA model as a Markov state model with "soft" states less susceptible to shot noise and poorly positioned states.

The construction in this chapter can also be interpreted as using Gaussian basis functions to fit a variational model. This is similar to the work by Wu and Noé [85] who used Gaussian mixture models to construct Markov transition models (MTMs).

5.2.3 Drawbacks

The tICA components (i.e. eigenvectors) become harder to interpret in this method. When using molecular features like dihedral angles or atom pair distances, the magnitude of the individual values in a tICA eigenvector represents a "relative importance" of that feature to the chosen dynamical mode. Specific amino acids can be colored or otherwise visualized based on their contribution to a particular tICA component, guiding the researcher towards "interesting" regions of a large biomolecule. In this method, the eigenvector values relate to landmark conformations, which may be more difficult to interpret.

The choice of number of landmark points, m, as well as the choice of kernel function and associated kernel parameters adds additional tunable hyperparameters, which is never desirable. We address some of these issues in the following sections.

5.3 Results and Discussion

5.3.1 Model quality on a 1D potential

We performed a numerical experiment to determine the effect of hyperparameters on the full kernel tICA solution (without the Nyström approximation) and on the landmark kernel tICA solution. We simulated one hundred Brownian dynamics trajectories on the potential energy function from fig. 5.1. We learned a kernel tICA model at several values of σ and estimated the slowest timescale of the system, fig. 5.2a For this simple problem, we can compute the true value of the timescale analytically (dashed line). Interestingly, the true timescale (which should serve as a variational bound in the infinite data limit) is easily exceeded for particularly small values of σ. We plot the estimated propagator eigenfunctions at small (fig. 5.2b) and large (fig. 5.2c) values of σ. Whereas large basis function widths miss nuance and non-linearity in the data and approach the linear tICA limit (fig. 5.2d), small values result in overfitting to noise (and incorrectly slow timescales). In fact, kernel tICA is highly susceptible to overfitting because each data point is related to every other data point leading to a large number of parameters to fit.

We have demonstrated that relying on the variational principle in the context of finite data would lead us to an improper choice of sigma. We can control for overfitting by using cross validation and the GMRQ score [79]. By splitting our data into a separate train and test set, we can evaluate how well a model built on the training set can capture the slow dynamics of the test set. We performed this analysis over a variety of choices of σ for kernel tICA and landmark kernel tICA, see fig. 5.3. Note that the landmark approximation shows a marked increase in maximum model quality (as measured by the GMRQ score). It displays a high score over a large range of hyperparameter values, including small values of σ where full kernel tICA does especially poorly (c.f. fig. 5.2b). For large values of σ ($> 10^0$) the full solution performs slightly better. We remind the reader that in this limit, the kernel tICA solution loses its non-linearity and reverts to linear tICA (c.f. fig. 5.2 c and d). The landmark approximation inherently regularizes the model. By reducing the number of parameters (i.e. number of landmarks), we remove the ability of the model to overfit

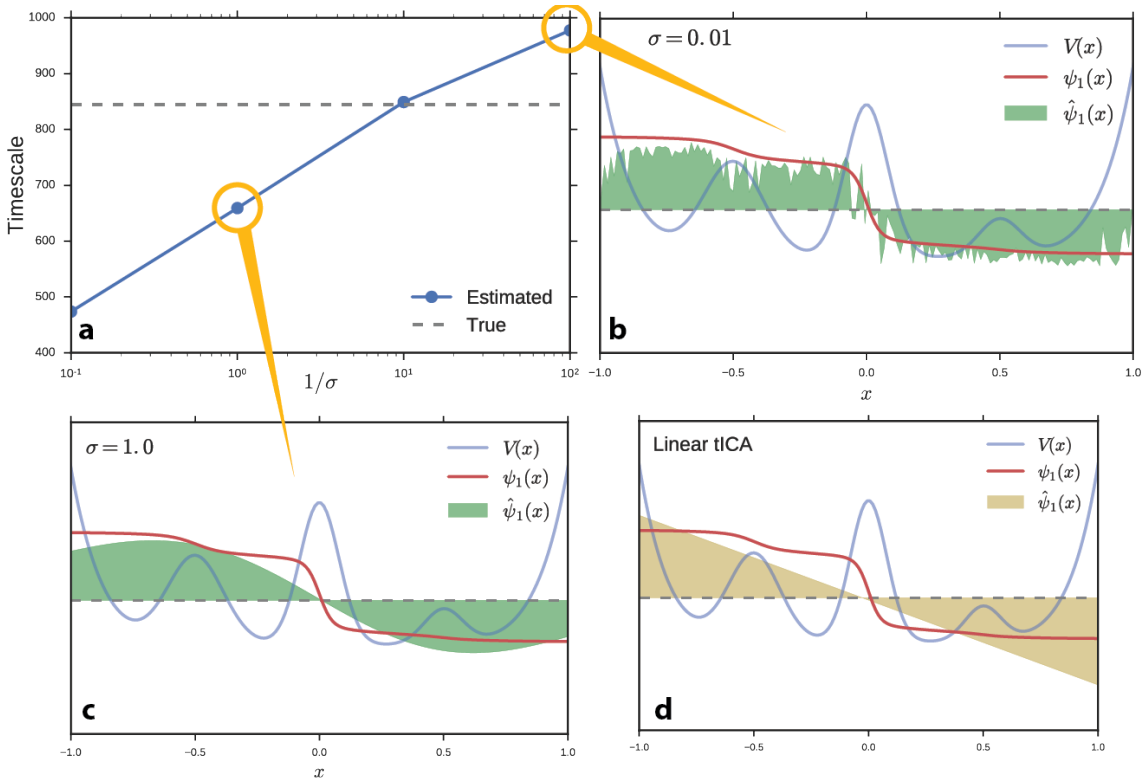

Figure 5.2: The previously-reported kernel tICA approach is highly sensitive to the choice of σ (basis function width). (a) The model timescales increase as σ decreases. Note that the variational bound on the model's slowest timescale is easily exceeded as σ shrinks. (b-d) The models numerically estimate (shaded region) the slowest dynamical eigenfunction of the system. (b) A small σ (width) will overfit to noise. (c) A large width misses nuance in the data and begins to resemble ordinary linear tICA. (d) Ordinary linear tICA.

to noise and spurious data points. In addition to the large computational savings, the landmark ktICA approach also produces better models.

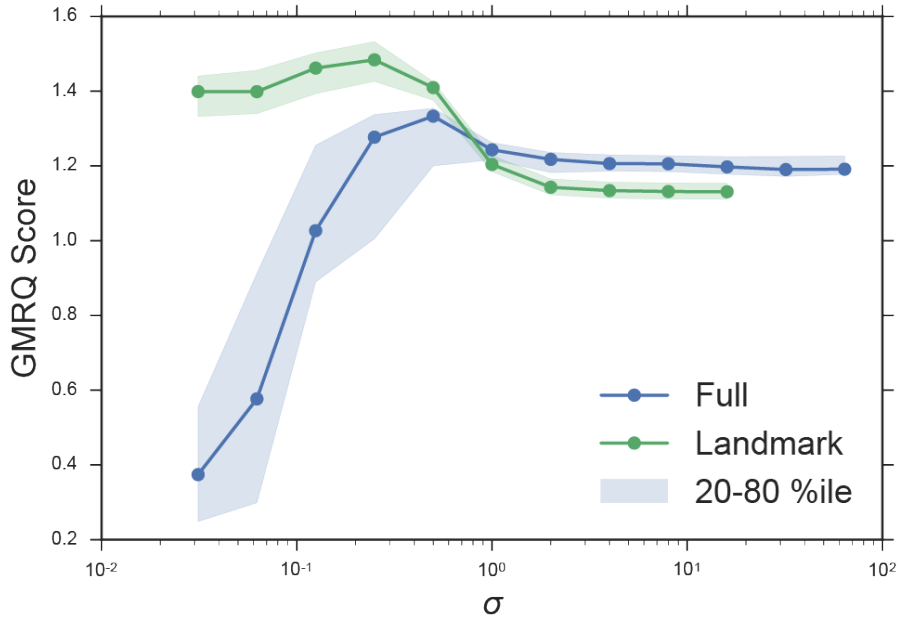

Figure 5.3: Landmark kernel tICA performs better than full kernel tICA over a large range of σ values. By employing landmarks, we inherently regularize the solution. Full kernel tICA essentially treats every point as a landmark point, which can give rise to solutions like fig. 5.2b. Additionally, full kernel tICA is computationally unfeasible with $O(n^3)$ dependence on the number of data points.

Unfortunately, the landmark approximation introduces a new hyperparameter m—the number of landmark points. From the perspective that the explicit kernel evaluations are features similar to dihedral angles or contact distances (see Section 2), the MD practitioner can reason that the degree of approximation m controls the resolution of the representation of the data and might be chosen to be of the same order of magnitude as the number of dihedrals or number of atoms. Moreover, it suggests that full kernel tICA ($m = n$) uses far too many landmark points for a typical MD dataset. As an example, consider a 100-residue peptide simulated for 1ms with frames saved every 1 ns. We might expect to describe this system with 10^2 or 10^3 features, whereas full kernel tICA would use 10^6. More stringently, one can use GMRQ cross validation to determine the best selection for this hyperparameter. We

performed GMRQ cross validation on the four-well potential dataset across a range of values for m and found a weak dependence of score on number of landmark points with a maximum around $m = 8$. This simple toy model does not require a large number of landmark points. The consistently high score across a range of values suggests heuristics may be sufficient for this parameter. In fig. 5.3, we fixed $m = 20$. In fig. 5.4, we fixed $\sigma = 0.25$. Cross validation was 10-fold shuffle-split among 100 trajectories. The two parameters could be simultaneously optimized on a two-dimensional grid search with added computational cost.

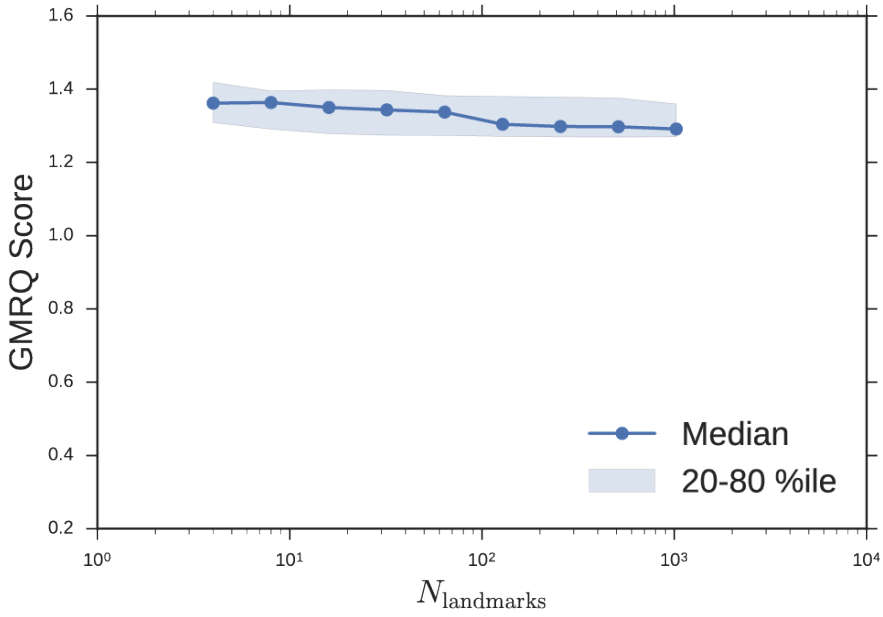

Figure 5.4: The quality of the model is weakly dependent on number of landmarks, m, for this simple system.

5.3.2 A folding coordinate for a small peptide

tICA has been used successfully as an intermediate step during MSM construction. By transforming the data into kinetically-oriented coordinates, defining the indicator-function basis functions via clustering of the data becomes more robust. Building an MSM is still necessary with linear tICA, because the linearity overly constrains the

solutions [71]. Kernel tICA transforms the data into coordinates that can comprise non-linear movements, and would therefore serve as a better intermediate step for MSM construction. However, by introducing non-linearity to the solutions, we can construct a kernel tICA model that is both accurate and interpretable without building an MSM.

In this example, we re-analyze the fip35 ww domain peptide folding simulations of Shaw et. al. [6] with landmark kernel tICA. We modeled 200 µs of trajectory data using 500 landmark conformations selected by all-atom RMSD mini-batch k-medoids clustering implemented in MSMBuilder [135]. Kernel features were computing using the Gaussian kernel with $\sigma = 3\,\text{Å}$. The tICA model was computed with a lag-time of 10 steps or 1 ns. The large spectral gap in tICA timescales (fig. 5.5, right) confirms that the slowest timescale is accurately captured by only one landmark kernel tIC. Although this is a loaded term [74], we declare this tIC to be the folding reaction coordinate. We can negative-log-histogram the data along this coordinate to estimate the energy profile for folding, fig. 5.5. The energy landscape shows a two-state behavior. The global minimum is the folded structure (fig. 5.5, inset, left). The metastable basin is a collection of unfolded conformations; we show the unphysical structural mean of a number of these conformations (fig. 5.5, inset, right).

We can generate a trajectory (movie) by selecting points along the tIC coordinate and drawing conformations from the raw data. See the SI for the fip35 kernel tIC folding trajectory. Note that these trajectories are often unphysical due to the naïve way we draw conformations along the tIC. In particular, the large structural changes that are all consistent with the entropically-dominated unfolded basin are often selected in adjacent time points. In the movie, we have smoothed the trajectory; in the unfolded region, this often results in atoms being averaged on top of each other. Further work can be done to produce more realistic and visually appealing movies, perhaps by selecting conformations consistent with the desired tIC value *and* similar to the previous frame.

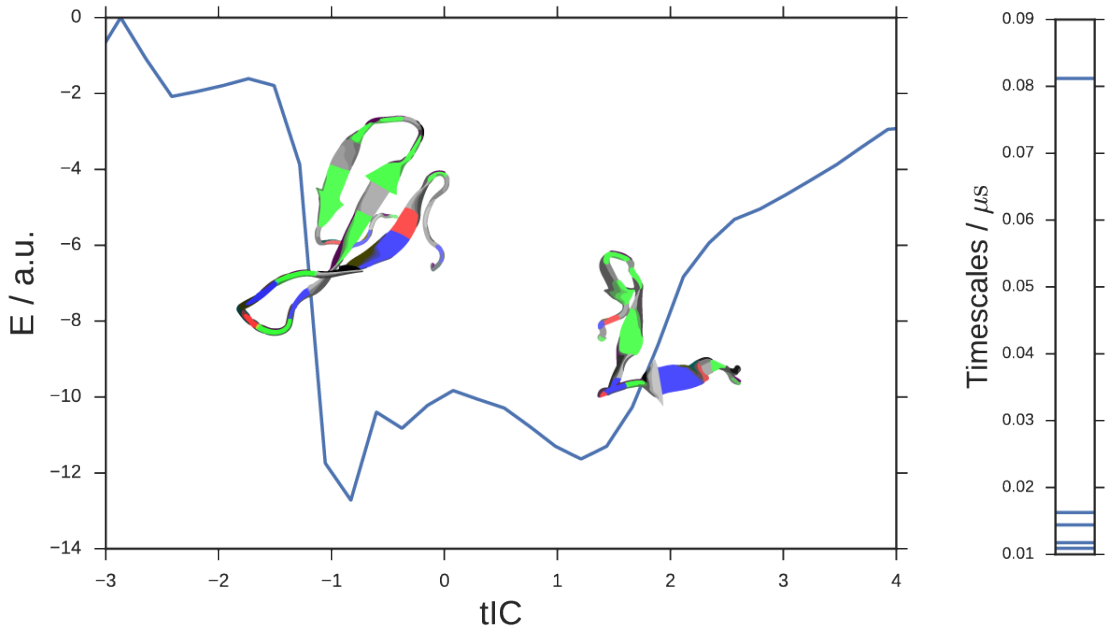

Figure 5.5: An energy landscape for the fip35 ww peptide. The analysis captures the minimum-energy folded state and an unfolded basin without hand-picking any coordinates or reference structures. Example conformations are overlaid on their tIC value. The learned kernel tICA model shows a large spectral gap (model timescales, right) which supports the projection of the dataset onto a single tIC coordinate.

5.3.3 An activation coordinate for conformational change

We apply the new method to a potassium ion channel that was previously shown to undergo a large conformational change between a compressed, down state and a stretched, up state. Similar to the peptide example, we recaptured the up-down dynamics with a single landmark kernel tIC. We used 400 µs of aggregate Folding at Home trajectory data to extract 500 alpha-carbon RMSD landmark points. Once again, we used the Gaussian kernel with $\sigma = 3$ Å. The tICA model was computed at a lag time of 10 steps or 9.6 ns. In fig. 5.6, we histogram the data along the learned coordinate to estimate a free-energy diagram for the up-down transition. The right basin is the stretched, up state with intracellular helices splayed outwards. The left well comprises the compressed, down state with rigid helices. For this system, there is an additional slow process within the compressed basin, identifying a metastable down-like state noted in the original study. Figure 5.6, bottom shows a detailed comparison of the two down-like states and the up state. By sampling representative conformations along this coordinate, we prepared a trajectory showing the up-down process, see the movie in the SI.

5.4 Conclusions

Landmark kernel tICA is the culmination of years of research into using MSMs, tICA, and the transfer operator formalism for understanding conformational dynamics. The first key insight was when Swope and Pitera [59] applied the theory of Markov chains to protein dynamics. As researchers applied this formalism, it became clear that state-space definition was crucial for constructing accurate MSMs. By applying the theory behind principal component analysis (PCA), several groups identified tICA as a useful dimensionality reduction for defining states in MSM construction. With the revelation that tICA and MSMs differ only by choice of basis, Schwantes and Pande [73] extended tICA using the kernel trick to build models of protein dynamics directly. We revisit MSMs by re-introducing the notion of "states", this time with fractional occupancy based on kernel distance to the state centroid. In so doing,

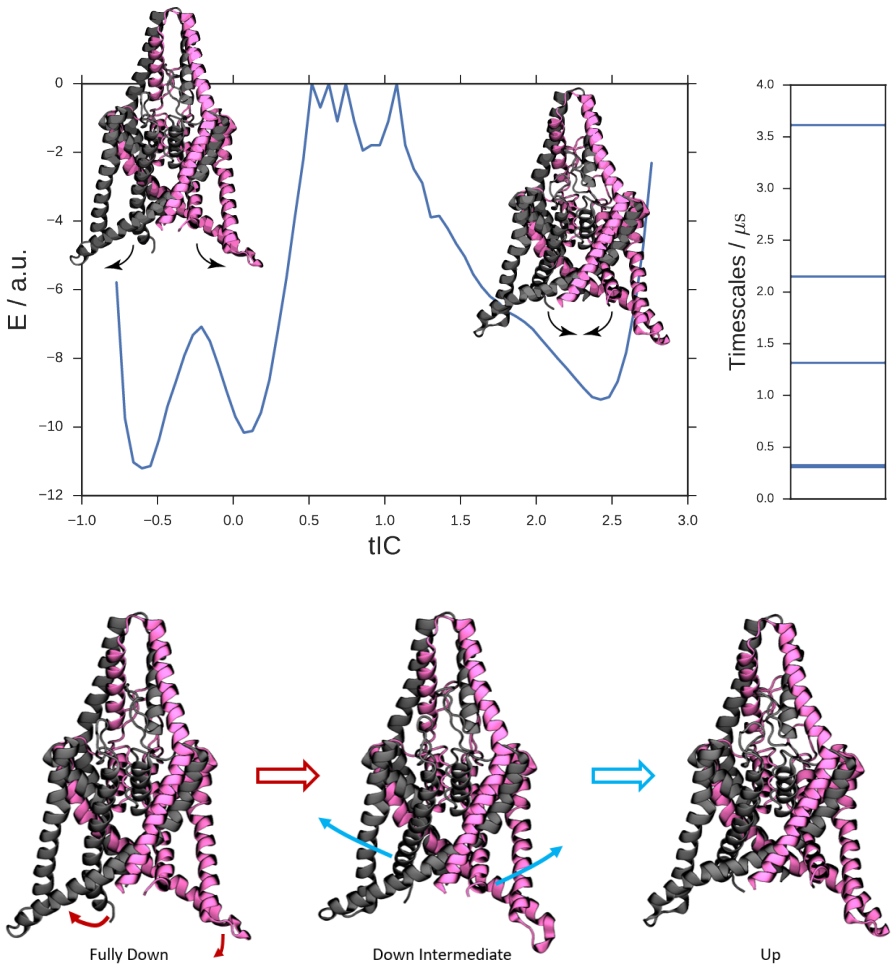

Figure 5.6: **(top)** Landmark ktICA learns the up-down coordinate from a simulation of the TREK-2 leak channel. The data is clearly separated into two large wells. The right well represents the "up" state resulting from membrane stretch. The left well represents the down, compressed state. The down region contains an intrastate barrier separating what was previously identified as a down-like intermediate. **(bottom)** Structural details between all three wells. The fully down state (left) has all helices in their down-most configuration. In the down-intermediate state (middle), one of the inner pore helices (grey) has moved up and there is local unfolding on the outer helix. In the up state (right), all helices are in their up-most configuration.

we introduced landmark kernel tICA which can be considered simultaneously (1) a regularized, computationally-tractable non-linear tICA model and (2) a Markov state model with "soft" states. This "best-of-both-worlds" approach yields highly accurate models that can be interpreted as MSMs *or* via projection onto reaction coordinates. We introduce this method with an eye towards improving existing methodologies that rely on robust projection of data. In particular, accelerated sampling schemes like umbrella sampling or metadynamics make strong assumptions about orthogonal degrees of freedom: the chosen reaction coordinates over which we accelerate are "slow" but every degree of freedom orthogonal to them must equilibrate must faster (e.g. within an umbrella window time frame). Without increasing the number of accelerated coordinates (which typically scale poorly), we anticipate that the present work can be used to choose coordinates for which this assumption is more likely to be true. Due to improved model quality, high interpretability, and computational tractability, the landmark kernel tICA approach can be widely applied to the analysis of dynamical biomolecules.

Acknowledgments

For this chapter, We thank Christian Schwantes, Robert McGibbon, Muneeb Sultan, and Carlos Hernandez for inspiration and helpful discussions. We thank Keri McKiernan, Muneeb Sultan, and Brooke Husic for critical feedback on this chapter. We thank the National Institutes of Health grant number NIH R01-GM62868 for funding. We graciously acknowledge D. E. Shaw Research for providing access to the fip35 folding trajectory datasets.

Author contributions for this chapter: MPH designed and performed the research and wrote the chapter. VSP supervised the research and edited the chapter.

Disclosure statement for this chapter: VSP is a consultant and SAB member of Schrodinger, LLC and Globavir, sits on the Board of Directors of Apeel Inc, Freenome Inc, Omada Health, Patient Ping, Rigetti Computing, and is a General Partner at Andreessen Horowitz.

Chapter 6

Learnable Soft Markov Models for Conformational Dynamics

This chapter is adapted from a manuscript which is under preparation.

Abstract

Building interpretable models of high-dimensional biomolecular simulations is currently a stepwise procedure with *a posteriori* hyperparameter optimization. This is largely due to the reliance on geometric proxies to the kinetic details we're actually interested in. Using "soft" versions of Markov state models (MSMs) and time-structure based independent component analysis (tICA) implemented in Tensorflow, we can estimate and optimize models directly based on kinetic considerations. We first optimize the landmark kernel tICA (lk-tICA) method introduced previously as a computationally-feasible version of non-linear tICA, but also considered a "soft" MSM. We show that the optimized landmarks produce better models. We then re-derive a set-free MSM method with softmax assignment of conformations to clusters, which we argue is better suited to estimate transfer operator eigenfunctions. Again, we optimize the softmax MSM parameters. We show that optimized softmax MSMs can build high-quality models directly from RMSD clusters with a small (~ 4) number of states, resulting in vastly improved interpretability.

6.1 Introduction

Markov state models (MSMs) have been used effectively to model the dynamics of biomolecules. Usually, the models are parameterized by simulated trajectories resulting from molecular dynamics calculations [61, 62, 64]. Since their original introduction [59, 63], the MSM method has been developed and improved. An incomplete overview of methodological advances is shown in fig. 6.1.

Although the MSM models kinetics, modelers have long relied on geometric proxies for kinetic properties. Originally, the RMSD distance metric was used to cluster conformations under the assumption that conformations that are geometrically similar in 3D space should interconvert rapidly. For a sufficiently fine partitioning of phase space, this is true. But with limited sampling, increasing the granularity of the clustering (increasing number of states) and thus increasing the number of parameters

to be estimated has statistical consequences like overfitting [78, 79]. Without drastically overparameterizing the model, researchers cannot be confident that the RMSD proxy for geometric closeness is a good one. Worryingly, there are many examples in biophysics where a small structural change causes profound functional differences (e.g. DFG flip in kinases) or slowly interconvert (register shift in beta sheets).

The introduction of tICA [68, 70, 71] was an important advance in the field of Markov modeling (fig. 6.1, blue). This linear dimensionality reduction technique finds an optimal rotation of the input data to maximize the "slowness" (autocorrelation) of the output features. The modern state of the art is to use this method as an intermediate processing step to transform internal coordinates (e.g. dihedral angles or contact distances) into a smaller vector space for clustering. Here again the field relies on machinery not designed with kinetics in mind. The clusters are defined through off-the-shelf classic unsupervised learning algorithms like k-means and Ward clustering. Various schemes for scaling the tICA coordinates [75, 153] have been introduced to make spherical clustering algorithms better respect kinetic distances.

Although the tICA method was introduced and has found great use as a dimensionality reduction technique, mathematically the tICA problem can be reduced to estimating the transfer operator [64] with functions linear in the input features [70, 73]. In practice, the use of linear functions harshly constrains the solutions and linear tICA is a poor model for conformational dynamics. How is the quality of a model quantified?

The variational principle of conformational dynamics [65, 66] sets up a clear definition of what makes an optimal model. In short, modeled dynamics will always be too "fast". The best models capture and report on the slowest dynamical processes of a system. However, the stepwise (e.g. xyz $\rightarrow$ dihedrals $\rightarrow$ tICs $\rightarrow$ MSMs) and geometric (e.g. RMSD clustering) approach to constructing models in practice prevents modelers from fully exploiting this variational principle. The state-of-the art is to perform GMRQ cross-validation [79] over hyperparameter choices (number of clusters, number of tICs) to build a collection of models and select the best. Often, hyperparameters are selected for testing via a grid-search [135] (which scales exponentially in the number of hyperparameters considered) or randomized selection from

a range of trial values [149].

Meanwhile, the use of "hard" msm clusters where a conformation is in one and only one state has been called into question [84, 142, 150, 154–158]. Particularly near the edges of the state, spurious transitions can be generated by fluctuations over the hard boundary. Figure 6.2 shows the first three dynamical eigenfunctions $\{\psi_i\}$ of the transfer operator for a 1 dimensional toy potential from ref [64] (see fig. 1.3). This potential has four wells, and therefore three slow processes. With stochastic dynamics, there are infinitely many fast processes. The slow processes can be rank ordered: the middle barrier is larger than the other two, so the slowest process is a transfer of flux from the right two wells to the left two wells. Functions ψ_2 and ψ_3 are nearly degenerate, and describe well hopping within the left or right half of the domain, respectively. The MSM plot of fig. 6.2 shows a general qualitative agreement with the true functions, but clear mis-approximations in transition regions due to their hard cut-offs. The tICA plot is smooth but the linearity harshly constrains the available solutions.

Gaussian MSMs [85], landmark kernel tICA (lktICA) [150], and so-called meshless or set-free MSMs [154–156, 158] have been introduced to provide something analogous to a soft MSM (see fig. 6.1 green). We stress that the name "soft MSM" is ambiguous, as there are multiple methods that resemble a softened form of the crisp, ordinary MSM.

6.1.1 Landmark kernel tICA

Landmark kernel tICA [150] transforms conformations into a set of $N = \#$ of landmarks occupancy values $x_i \in (0, 1]$ according to the Gaussian radial basis function (RBF) kernel $k(x, y_i)$ given by eq. (6.1).

$$k(x, y_i) = e^{\frac{-||x-y_i||^2}{2\sigma^2}} \tag{6.1}$$

Landmark kernel tICA with this kernel function can be thought of as a "soft" MSM in the sense that conformations have a non-zero occupancy in each RBF state. In this construction, a state is defined as closeness to the landmark points $\{y_i\}$.

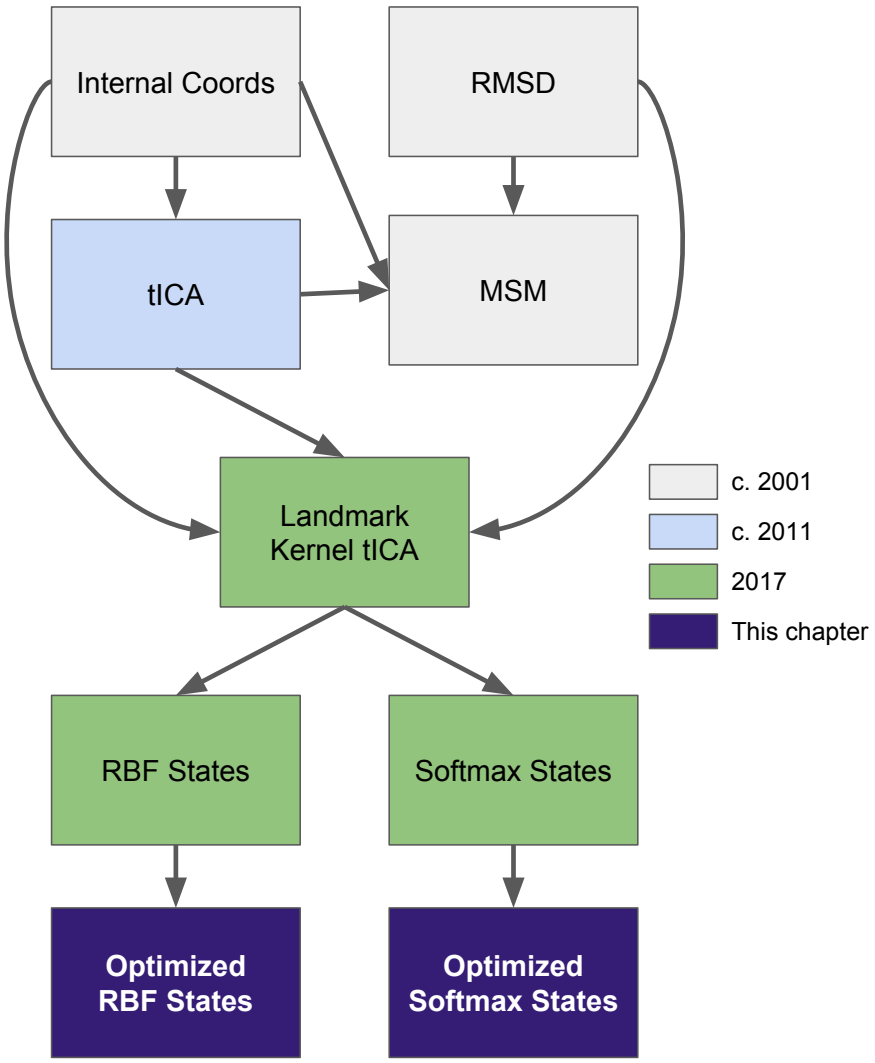

Figure 6.1: Historical overview of MSM-like methods. Arrows indicate potential flows of trajectory data. **(grey)** Markov state models were originally built via clustering of internal-coordinate representations or directly via RMSD. **(blue)** tICA was introduced as an intermediate dimensionality reduction technique but—like MSMs—models the transfer operator. **(green)** Landmark kernel tICA was introduced to unify the tICA and MSM approaches and can be estimated via internal coordinates, RMSD distance, or even linear tIC-transformed coordinates. Softmax MSMs were introduced with other motivations, but in this work we derive them as a particular choice of kernel for landmark kernel tICA. **(purple)** Optimization of RBF soft states is covered in section 6.2. Softmax states is covered in section 6.3 This diagram deliberately omits many important advances in dynamical modeling for clarity.

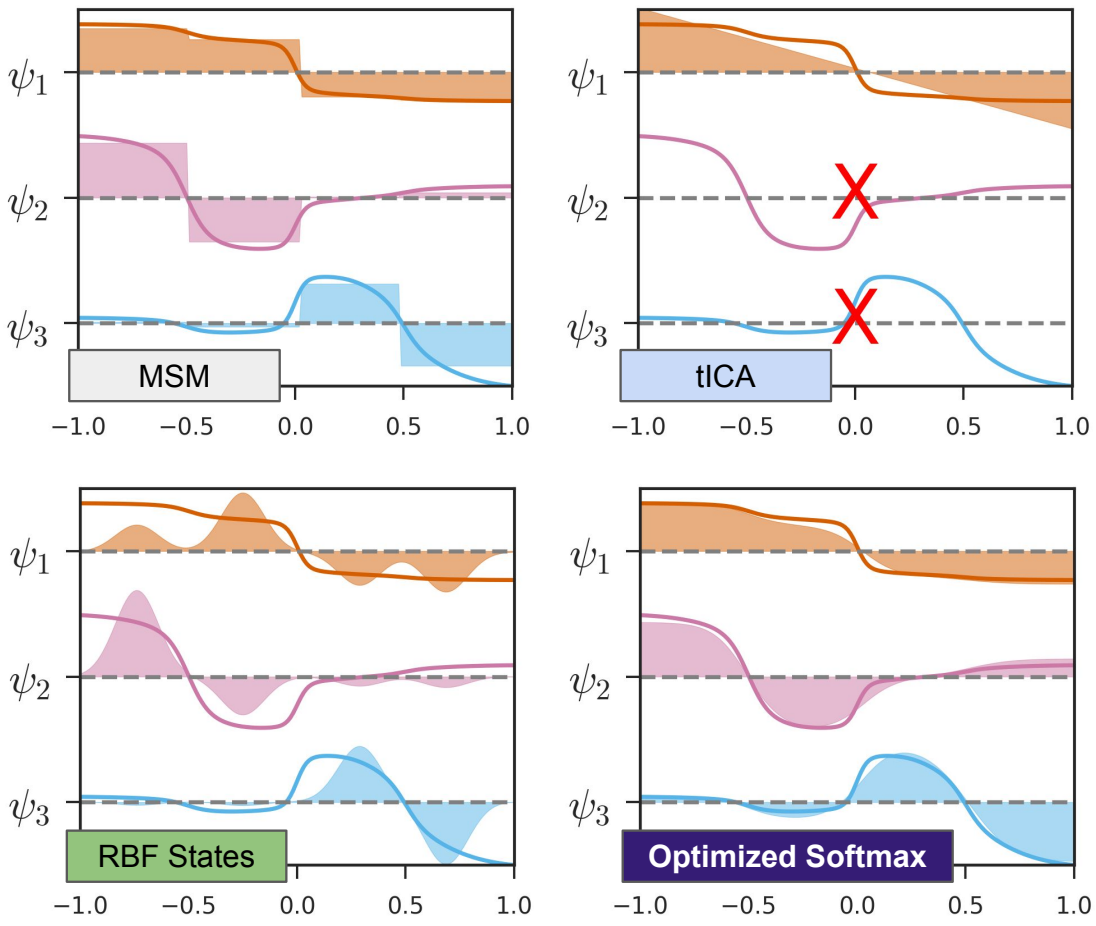

Figure 6.2: Examples of MSM-like methods (shaded areas) compared to the true eigenfunctions (solid lines) for a toy 1D, four-well potential from ref [64] (see fig. 1.3). Plot labels are colored in accordance with fig. 6.1. **MSM** indicator functions describe transition regions poorly. **tICA** is a linear algorithm; it is usually used merely as a dimensionality reduction step. For this 1D potential, tICA only estimates one function. The lack of $\psi_{i>1}$ is denoted by red Xs. **RBF states** are smooth, but may not be a good basis for transfer operator eigenfunctions. The **optimized softmax** MSM estimates the slow processes with great fidelity. For the MSM, RBF states, and Softmax model, four states were used.

Figure 6.2 shows RBF estimates to the four-well potential's eigenfunctions.

6.1.2 Set free MSMs

Originally called meshless or meshfree MSMs [154–157], this class of models has more recently been called set-free MSMs [158], perhaps under the realization that ordinary MSMs are already meshfree. If we consider a set of N basis functions $\{\theta_i\}$ supporting subspaces $\Omega_i = \text{supp}(\theta_i)$, then we can decompose the full space Ω. We assume all $|\Omega_i| > 0$ and the basis functions are normalizable

$$\sum \theta_i(q) = \mathbf{1}, q \in \Omega$$

A hard decomposition is one such that $|\Omega_i \cap \Omega_j| = 0$ if $i \neq j$ and results in a classic MSM. A soft decomposition is one such that $|\Omega_i \cap \Omega_j| \geq 0$ if $i \neq j$. Weber et. al. [158] chose an explicit set of basis functions defined with a radial dependence on a number of landmark points similar to the kernel function in landmark kernel tICA. They term the resulting model a set-free MSM. We anticipate future naming ambiguity, as other choices of basis functions could also be used to produce set-free MSMs. As such, in section 6.3 we provide an alternative derivation of the set-free MSM and term it a softmax MSM to prevent future ambiguity.

6.1.3 Differentiability

A notable disadvantage of hard MSMs is that the state assignment procedure is non-differentiable. In this work (fig. 6.1, purple), we use differentiable soft MSM constructs to optimize clustering and (hyper-)parameterization directly via the variational principle.

6.2 Learnable landmark points

The landmark kernel tICA algorithm is completely differentiable. Recall that the tICA problem is solved via

$$\begin{aligned} \mathbf{C}^{(\tau)} &= |X^{(-\tau)}\rangle \langle X^{(+\tau)}| \\ \mathbf{C}^{(0)} &= |X\rangle \langle X| \\ \mathbf{C}^{(\tau)}\mathbf{v} &= \lambda \mathbf{C}^{(0)}\mathbf{v} \end{aligned} \tag{6.2}$$

where τ represents some lag time, and $\mathbf{C}^{(\tau)}$ the time-lagged correlation matrix. Here, $X^{(-\tau)}$ are the time series data points from $t = 0$ to $t = T - \tau$ and $X^{(+\tau)}$ the data points from $t = \tau$ to $t = T$. These $X^{(\pm\tau)}$ when taken together form a collection of time-lagged tuples of observations.

Numerically, eq. (6.2) is implemented as two dot products and a generalized eigensolve. We implemented eq. (6.2) using the Tensorflow machine learning package [159], which can automatically generate gradients for the expression. In theory, we can now perform stochastic gradient descent on variables to maximize the leading eigenvalue $\lambda_{\max}$. It is not clear however, that in the current formulation of the problem we have any variables to optimize. We turn our attention to the trajectory data $|X\rangle$. Normally, this is a set of coordinates of a biomolecule that respects the translational and rotational symmetry of the problem. For instance, backbone dihedral angles or a subset of contact distance pairs can naturally be used. These protein features are all just functions of the original $\mathbb{R}^{3N}$ atomic coordinates. We can define our own function in this spirit, taking care to smoothly parameterize the function to permit optimization of the featurization itself.

It should not come as a surprise that the landmark kernel tICA method can be viewed as a featurization function specified by eq. (6.1). This function has a clear parameter: σ. In the paper introducing lktICA [150], this parameter was chosen either by rule of thumb or GMRQ cross-validation over a grid of reasonable values. Given a Tensorflow implementation with automatic differentiation, we can optimize the value of σ to maximize the leading tICA eigenvalue $\lambda_{\max}$. Since the GMRQ grid search scales exponentially in the number of hyperparameters, the original lktICA paper restricted itself to one shared σ parameter among all landmark points. This restriction is now lifted, so we use a different kernel width σ_i for each landmark point y_i. Furthermore, the landmark points themselves were originally selected by geometric clustering of the

data. As discussed in the introduction, this proxy to kinetic closeness is good but not great. Using a computationally-optimized and differentiable implementation of the RMSD algorithm, we allow the landmark conformations themselves to be optimized. In particular we allow the $\{x, y, z\}_j$ coordinates to vary as well as a per-atom weight w_j.

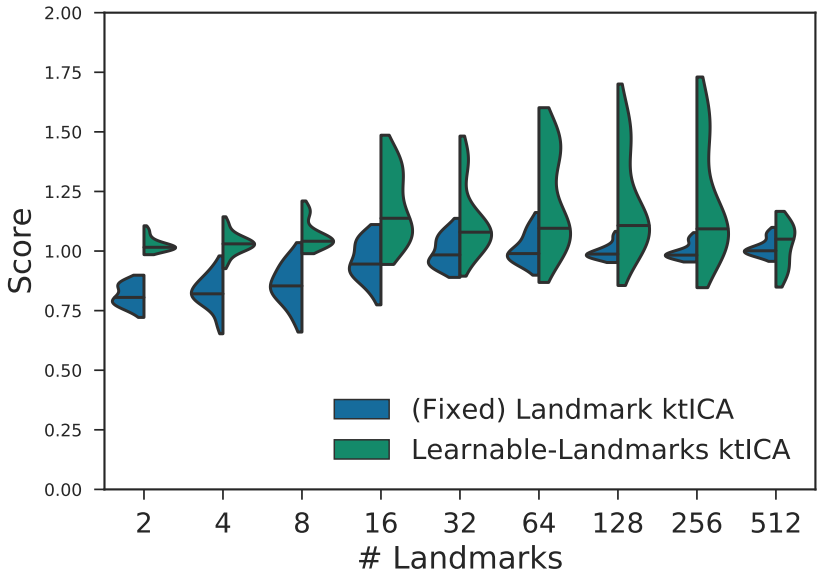

Figure 6.3: Learnable landmark points improve lktICA models relative to fixed landmark points. A distribution of GMRQ scores was calculated for differing number of landmark points. The median (inset solid lines) learnable landmark scores are higher than their fixed counterpart in all cases. The improvement is particularly stark in the low-N regime. Improvements in this regime are particularly useful for interpretability of the landmark conformations themselves.

With the introduction of these many new learnable parameters, the problem of overfitting is a real concern. It is essential to perform GMRQ cross validation to ensure the generalizability of these algorithms. We performed such a cross validation for two gold-standard datasets: The 200 µs folding and unfolding simulation of the fip35 WW domain and the 1 ms BPTI conformational change simulation [6]. Figure 6.3 shows the distribution of model score across different numbers of landmark conformations in the traditional fixed (blue) and new learnable landmarks (green) models for the

fip35 folding simulations. The scores were calculated under 10-fold shuffle-split cross validation.

According to the median, the learnable-landmarks models perform better over all numbers of landmarks. The best learnable-landmarks model ($N = 16$) shows marked improvement over the best fixed landmark model. The distribution of model scores is generally broader for the learned models, probably owing to the greater flexibility of parameters and stochastic nature of the gradient descent. Of particular note, fig. 6.3 shows particularly strong improvement in the low-N regime. With $N = 2$ landmarks, we can build a model of comparable quality to the best possible fixed landmark model. This is a boon to interpretability, as now the landmark conformations themselves can be a meaningful result of the model instead of just an intermediate result in the estimation of a final tICA model. We observe a turnover in model quality after $N = 16$ where the large number of parameters result in worse generalization.

The results for the BPTI simulation were even better, to the point of being uninteresting. A virtually perfect model was obtained at $N = 4$ upwards through all tested values of N. The BPTI results are plotted in fig. D.1.

6.3 Softmax MSMs

To practically construct a Markov state model, the modeler (1) clusters conformations into states, (2) counts transitions among states in a $N \times N$ counts matrix ($N = \#$ of states), and (3) normalizes the rows of this counts matrix to obtain transition probabilities. The clustering algorithm is usually k-means clustering (when working in a vector space like internal coordinates) or k-medoids (when using RMSD distances). In either case, a set of N cluster centers is obtained. The trajectory data is transformed from a time-series of conformations (RMSD clustering) or internal coordinate feature vectors (k-means) to a time-series of integer cluster indices $k_t \in 1, \ldots N$. The counts matrix is usually obtained by filling in a matrix using the k_t values to index the 2D array as in the following pseudocode:

```
counts = zeros(N, N)
for t in 1..T
    counts[k[t], k[t+1]] += 1
```

We now consider an alternative construction. Instead of transforming an input trajectory of length T to a "trajectory" of cluster indices $\in \mathbb{Z}^T$, instead transform it to a trajectory of one-hot encodings of the cluster index $\in \mathbb{Z}^{T\times N}$. For example:

$$\begin{aligned} k_t = 3 &\rightarrow x_t = \{0, 0, 1, 0, \dots\} \\ k_{t+1} = 2 &\rightarrow x_{t+1} = \{0, 1, 0, 0, \dots\} \end{aligned} \tag{6.3}$$

Since each conformation is placed into the cluster whose center is closest, this corresponds to performing an argmin over the vector of distances to cluster centroids. If we define $|X\rangle$ as the $T \times N$ trajectory of one-hot encodings, then the counts matrix at lag-time τ is

$$\mathbf{C}^{(\tau)} = |X^{(-\tau)}\rangle \langle X^{(+\tau)}| \tag{6.4}$$

completely analogous with eq. (6.2). Furthermore, normalization of the rows to turn the counts matrix into a transition matrix is equivalent to dividing each element of a given row by the total of that row. The total of the row is the number of times the state was exited. This can be expressed via matrix products

$$\mathbf{C}^{(0)} = |X\rangle \langle X| \tag{6.5}$$

$$\mathbb{T}(\tau) = (\mathbf{C}^{(0)})^{-1}\mathbf{C}^{(\tau)} \tag{6.6}$$

Re-arranging terms, once again the eigenprocesses of the transfer operator can be expressed as

$$\mathbf{C}^{(\tau)}\mathbf{v} = \lambda \mathbf{C}^{(0)}\mathbf{v} \tag{6.7}$$

Therefore, by using a one-hot encoding of state labels, we can transform the discrete state MSM problem into the tICA problem. Why is this useful? Now we can slightly modify the procedure to make it differentiable. The argmax function is not differentiable. Following the machine learning community, we replace a hard-cutoff classification task with a softmax function, eq. (6.8).

$$k'(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{6.8}$$

For our particular case, we have a set of distances to each cluster centroid $\{d_i\}$ and we want the minimum distance (not maximum). We scale the distances by a tunable parameter σ to both make the quantity dimensionless and tune the "softness". We take the negative distance to minimize the quantity.

$$k(d_i) = \frac{e^{-d_i^2/(2\sigma^2)}}{\sum_j e^{-d_j^2/(2\sigma^2)}} \tag{6.9}$$

Fans of statistical mechanics will immediately recognize this as the Boltzmann distribution. $2\sigma^2$ is analogous to a temperature. This is also the functional form of the partition of unity used in prior works building a set-free MSM [158]. If we explicitly write $d_i = ||x - y_i||^2$, we recover a kernel function c.f. eq. (6.1).

$$\begin{aligned} k(x, y_i) &= \frac{1}{Z} e^{\frac{-||x-y_i||^2}{2\sigma^2}} \\ Z &= \sum_j e^{\frac{-||x-y_j||^2}{2\sigma^2}} \end{aligned} \tag{6.10}$$

Now it should be clear that this formulation is analogous to landmark kernel tICA with a different kernel function. Figure 6.4 shows the slowest eigenprocess of the three-well, 2D Müller potential for four values of σ. As $\sigma \to 0$, we recover the Voronoi tessellation formed from hard-cutoff MSM states. As σ is increased, the states become softer and the transition region becomes smooth. To avoid all ambiguity, we term this particular choice of set-free construction the softmax MSM.

Similar to section 6.2, we can optimize the parameter σ. More importantly, we can optimize the positions of the cluster centers y_i as well as parameters introduced

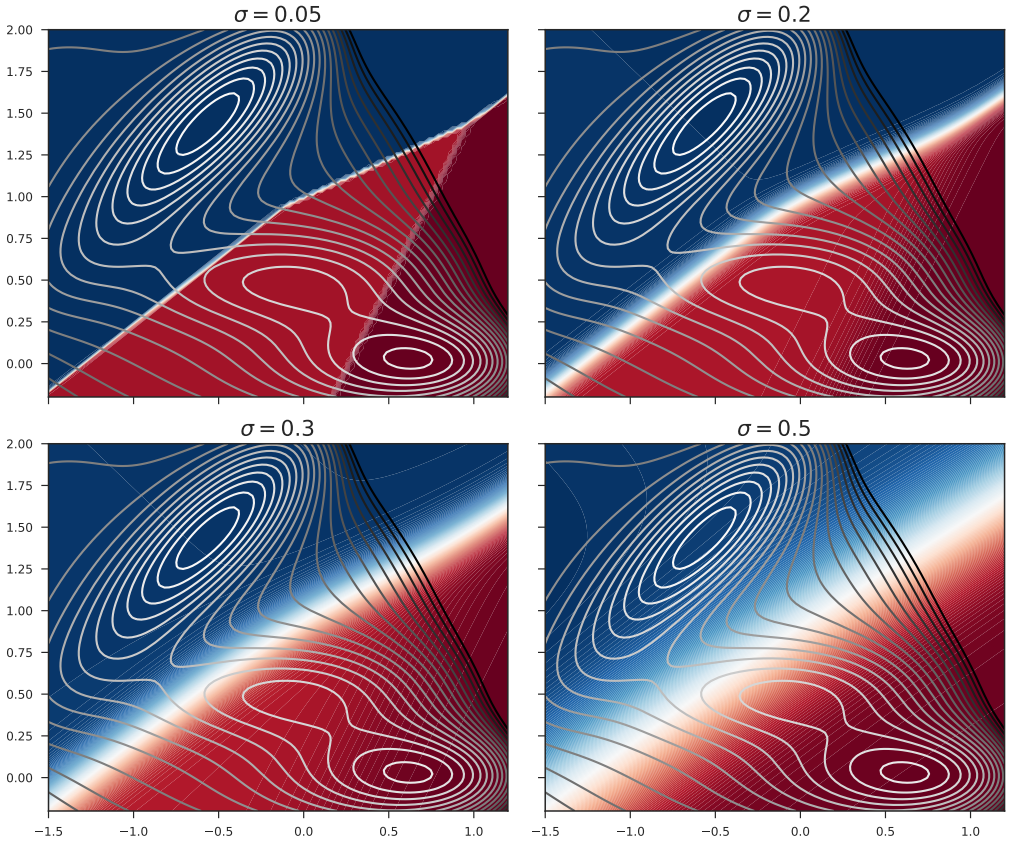

Figure 6.4: The σ parameter controls the softness of the Softmax MSM. Here, we plot the first dynamical eigenfunction of the transfer operator ψ_1 for the Müller potential for increasing values of σ as estimated by a four-state Softmax MSM. As $\sigma \to 0$, we recover an indicator-basis-function MSM.

to the distance metric. In this work, we parameterized a unique σ value for each landmark point (state centroid) and a per atom (or per feature) weight w_i when computing distances. For the first time, the clustering step in building an MSM directly optimizes the quality of the model (as measured by the variational principle) without using a geometric proxy.

The softmax MSM may be a better choice of basis compared to the Gaussian kernels used in lktICA. The softmax bases arguable more closely match the "shape" of transfer operator eigenfunctions: compare fig. 6.2 RBF vs. softmax states. For higher dimensional systems like biomolecules, we cannot directly visualize the eigenfunctions, but we argue that the toy potential used in fig. 6.2 captures the relevant properties. Namely, the transfer operator eigenfunctions are *not* weighted by the equilibrium distribution μ and therefore have non-zero support over *all* of phase space. This is in contrast to Gaussian kernels, which decay away from landmark points. Given a sufficiently few number of landmarks or a sufficiently small kernel width σ, large regions of phase space will have no basis support. Due to the normalization of state occupancies in eq. (6.10), the softmax MSM basis functions approximate a Voronoi tessellation and provide support for all of phase space.

Optimized softmax models have the power to drastically improve the quality of dynamical modeling similar to how the optimization in section 6.2 improved model fidelity. Once again, we turned to GMRQ cross validation to compare and contrast model quality. Figure 6.5 shows two comparisons between the new method and previous methods as a function of number of states. In the top plot, the RMSD-based softmax MSM (pink) is contrasted with the analogous RMSD-based hard-state MSM (blue). The softmax formulation shows markedly increased scores over the traditional MSM. Again, the improvement is most stark in the critically important low-N regime. Whereas the top plot compares analogous models (both built on RMSD clustering), the bottom plot compares the best traditional MSM and the best softmax MSM. As noted in the introduction, state-of-the-art modeling calls for the use of tICA dimensionality reduction on internal coordinates (here backbone dihedral angles). The so-called tICA-MSM performs much better than the RMSD MSM in the upper panel even for lower values of N, making it a worthy adversary. The best

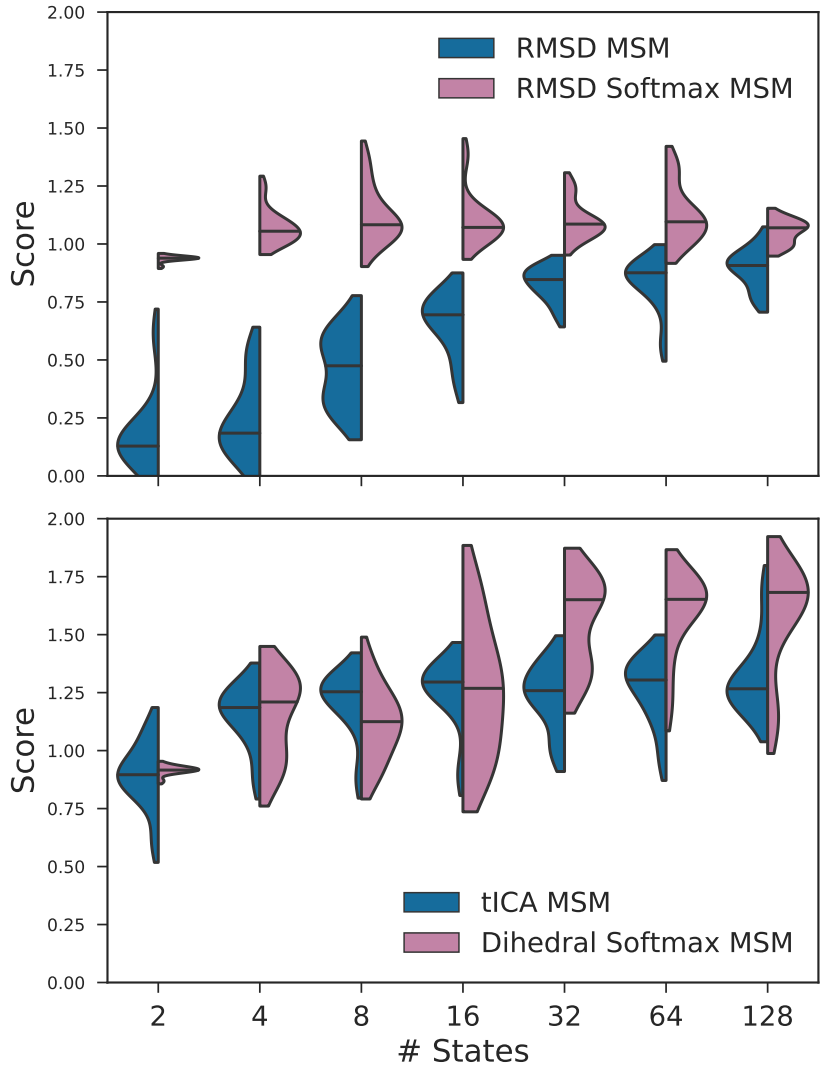

Figure 6.5: Comparison between hard MSMs (blue, left violins) and optimized softmax MSMs (pink, right violins) modeling the fip35 dataset. The upper plot compares models built directly on RMSD clustering to state centroids. The optimized softmax MSM shows a marked increase in scores, especially for low numbers of states. These models have no intermediate or pre-processing step. The lower plot compares the best models of each class. A hard MSM built on linear tICs fitted on dihedral angles performs quite well. However, a a softmax MSM fit directly on the dihedral angles surpasses the performance of a hard MSM for $N \geq 32$.

softmax MSM was one built on dihedral angles in place of RMSD clusters, so we plot this as the best softmax model. Here, the traditional modeling tactics fare better, but ultimately the highest scoring model is produced by the $N = 32$ state softmax MSM. We repeated the analysis on the BPTI conformational change system, see fig. D.2. For this system, the RMSD softmax MSM performed near-perfectly for all $N \geq 4$ and easily exceeded the state-of-the-art traditional methods. Figure D.3, fig. D.4, fig. D.5 show the full matrix of scores explored for models built on RMSD distances, dihedral angles, and linear tICs, respectively. For each input space, we scored a traditional MSM, learnable-landmarks ktICA model, and softmax MSM. Each plot was produced for the fip35 folding simulation and the BPTI conformational change simulation.

Low numbers of states offer a boon for interpretability. With normal MSM construction, the researcher must "lump" perhaps hundreds of "microstates" into a small number of "macrostates" for further analysis. We have shown that high quality models can be estimated even with a small number of states. The learned parameters can also be used for interpretability. Figure 6.6 shows the two centroid conformations from a $N = 2$ softmax MSM of the fip35 folding simulation. The researcher can use these conformations as a structural guide to elucidate the salient features of the slowly decorrelating modes in a protein. Note that the atomic positions are optimized for model quality, so the centroids may turn out to be "high-energy" configurations when viewed in the context of a molecular forcefield. In addition to the centroid coordinates, the optimization procedure learns a set of atomic weights w_i which weight the importance of individual atoms. Researchers can use these weights to focus or ignore certain regions of the protein which the modeling procedure has deemed unimportant for dynamics.

6.4 Conclusions

The long history of MSM estimation and construction has concerned itself with adding and tweaking steps to the procedure for modeling in the hopes that the new methods could more closely align our geometric proxies to the true dynamical operators. For the first time, we introduce a model where all steps are optimized with kinetic

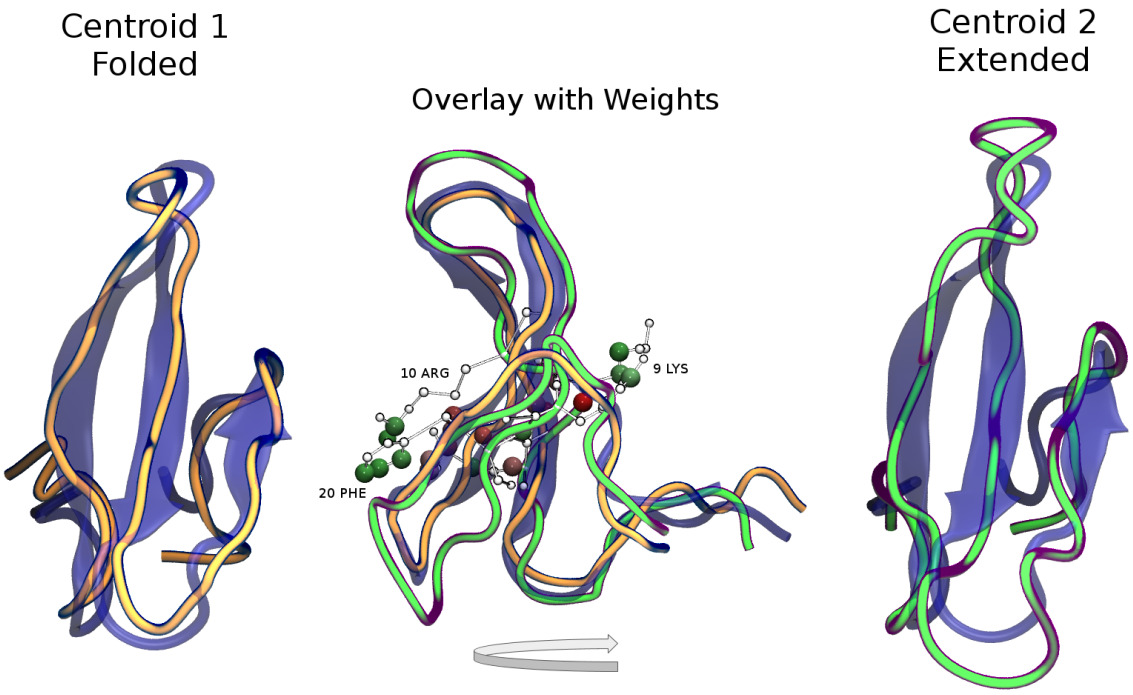

Figure 6.6: The two cluster centroids learned by a softmax model are shown as orange and green traces. MSMs with small numbers of states are inherently more interpretable as the cluster centroids can be inspected directly. We show the protein backbone superimposed on a cartoon representation of the crystal structure of the native state (blue, transparent). Centroid 1 is slightly compressed from the native state while centroid 2 is slightly extended. Per-atom weights are learned, and highly weighted atoms are shown in the center panel as spheres for both centroid 1 (red) and centroid 2 (green).

optimality as the goal. By developing and coding differentiable analogs of past algorithms, we can directly optimize the eigenvalues (timescales) of interest. In particular, we implemented the landmark-kernel tICA algorithm in Tensorflow. The optimized algorithms easily exceeded the quality of the fixed landmark approach of the original work. Developed as an approximation to the full non-linear tICA kernel method, it was shown that lktICA resembles an MSM with partial occupancy. However, in this chapter we note that the "soft" MSM is ambiguous as there are many basis sets that resemble softened indicator functions. In particular, we noted that the Gaussian RBF functions used for lktICA may not be the best match for transfer operator eigenfunctions. By starting from the MSM itself (instead of non-linear tICA), we re-derive the softmax MSM which reduces to a discrete state MSM as the parameter $\sigma \to 0$ but lends itself to optimization otherwise. We showed that optimized softmax MSMs produce higher scoring models of protein dynamics with markedly fewer states. The small number of learned centroids and learned RMSD weights can be used effectively for interpretation.

Acknowledgments

For this chapter, We thank Robert McGibbon, Muneeb Sultan, and Carlos Hernandez, Brooke Husic, and Nate Stanley for inspiration and helpful discussions. We thank the National Institutes of Health grant number NIH R01-GM62868 for funding. We graciously acknowledge D. E. Shaw Research for providing access to the fip35 and BPTI trajectory datasets.

Author contributions for this chapter: MPH designed and performed the research and wrote the chapter. VSP supervised the research and edited the chapter.

Disclosure statement for this chapter: VSP is a consultant and SAB member of Schrodinger, LLC and Globavir, sits on the Board of Directors of Apeel Inc, Freenome Inc, Omada Health, Patient Ping, Rigetti Computing, and is a General Partner at Andreessen Horowitz.

Chapter 7

Concluding Remarks

Building robust dynamical models of biomolecules is a challenging task under the best of circumstances. With increasing computational power and improved forcefields, the task of interpreting the deluge of data from a typical molecular dynamics simulation has become a bottleneck.

The work presented in this dissertation covers my efforts to (a) develop new algorithms for analysis, (b) apply the newest methods to challenging new biological systems and (c) distribute high-quality software to make the best methods readily accessible. I addressed point (a) in chapter 2 wherein I develop and characterize a method for including solvent degrees of freedom in our models. In chapter 3, I applied state-of-the-art MSM modeling to understand multi-scale conformational dynamics of the TREK-2 potassium ion channel in accordance with (b). In chapter 4, I document and provide examples of one particular piece of software developed collaboratively by myself and others to support point (c). In chapter 1, I introduced the variational principle for conformational dynamics and showed the connection to established modeling methods like Markov modeling and tICA. The mathematics presented here underpins the theory in all chapters 2 to 4. More suggestively, the introductory chapter introduced the question of whether new choices of basis might produce better models.

Chapter 5 and chapter 6 directly answer this question by introducing a unifying non-linear basis and a learnable basis, respectively. These chapters should not be the final word in basis set design and optimization for conformational dynamics. Generalized function approximation like deep neural nets could be used to learn kernel functions without being restricted by traditional thoughts about what a basis set *should* look like. Composable basis sets are another future avenue I would like to see investigated. Specifically, if a sufficiently well-parameterized basis set is developed for each protein amino acid you could conceivably compose them to generate a basis set for a whole protein. If residue-centric basis functions sufficiently capture local dynamics, an appropriate composable basis could allow the rapid construction of "zero-th" order models for seeding the simulation of new systems or 1^{st} order models for simulating protein mutants. It is my hope that the novel algorithms aided by well-engineered software implementations and supported by characterization on real biomolecular systems presented in this dissertation will lead the field ever closer towards truly robust dynamical models of biomolecules.

Appendix A

Supplementary information for Markov modeling of TREK-2

This chapter is supplementary information for chapter 3, which is adapted from Matthew P. Harrigan, Keri A. McKiernan, Veerabahu Shanmugasundaram, Rajiah Aldrin Denny, and Vijay S. Pande. Markov modeling reveals novel intracellular modulation of the human TREK-2 selectivity filter. *Sci. Rep.*, 7(1), 2017. doi: 10.1038/s41598-017-00256-y [115], licensed under a Creative Commons Attribution 4.0 International License. K. A. McKiernan and M. P. Harrigan contributed equally to this work. The referenced movie is available from the Nature Scientific Reports website.

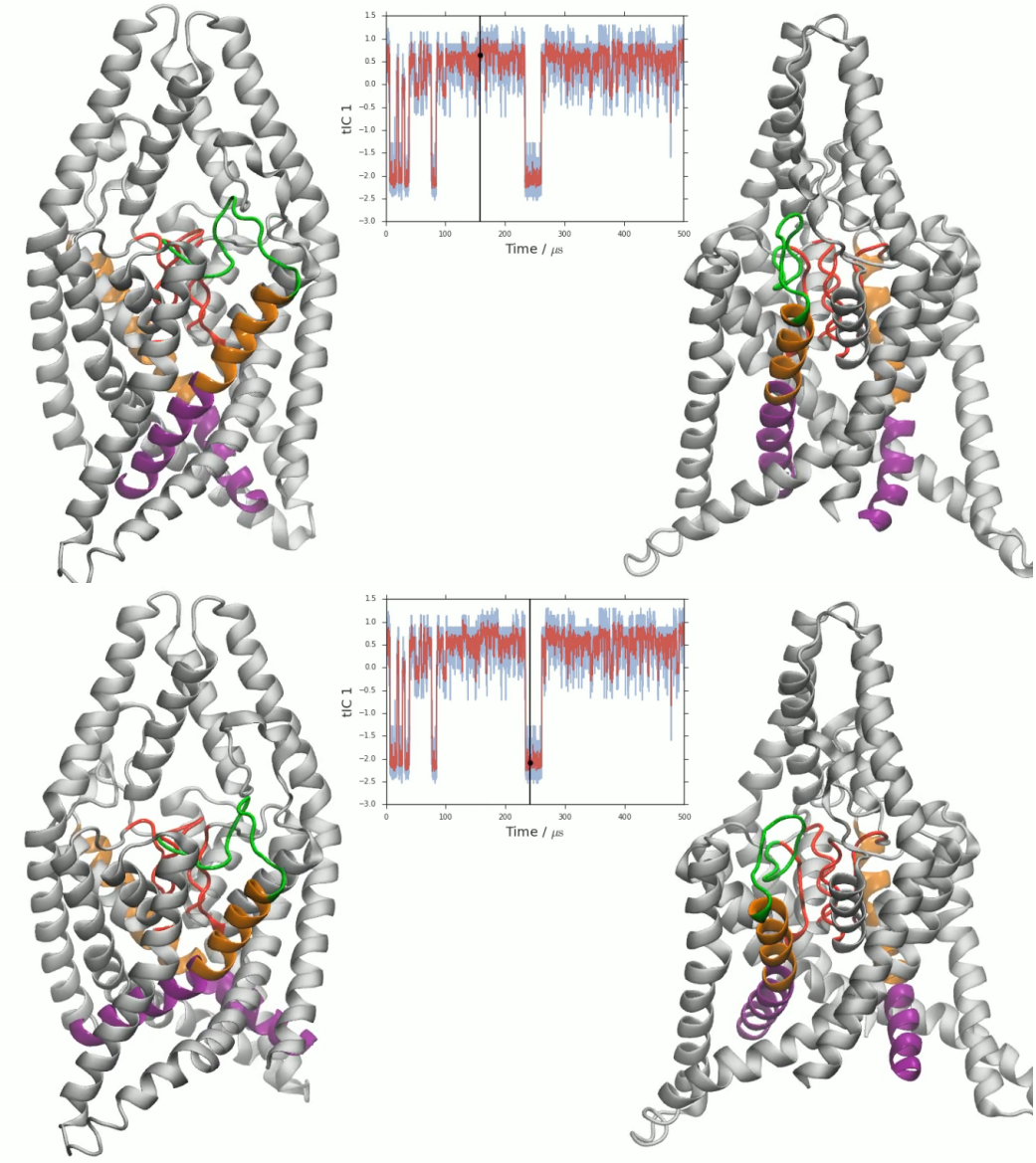

Figure A.1: We generated a 500 µs representative MSM trajectory by sampling among discrete states according to the modeled transition probabilities. The movie can be viewed in the supplementary file `trek.mp4`. Depicted here are two frames from the movie (top and bottom). Each frame consists of a view of the protein and its 90 degree rotation. We plot the projection of the conformation against its tIC 1 (Up-Down) coordinate. The black vertical line traces the current time in the trajectory.

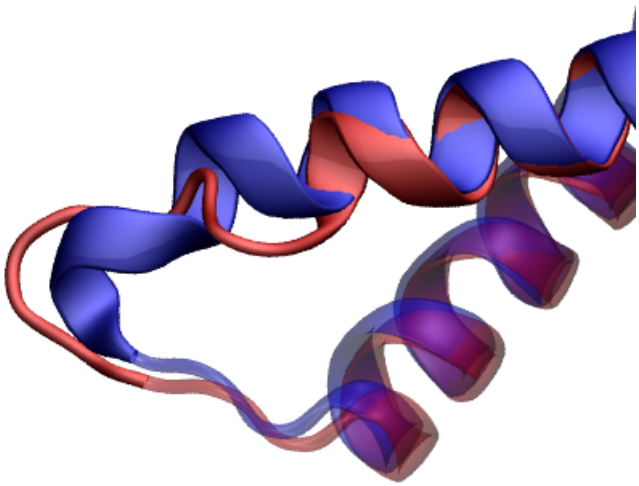

Figure A.2: Partial unfolding of the M2-M3 loop distinguishes I_2 from *Down*. Colors are as in fig. 3.2B.

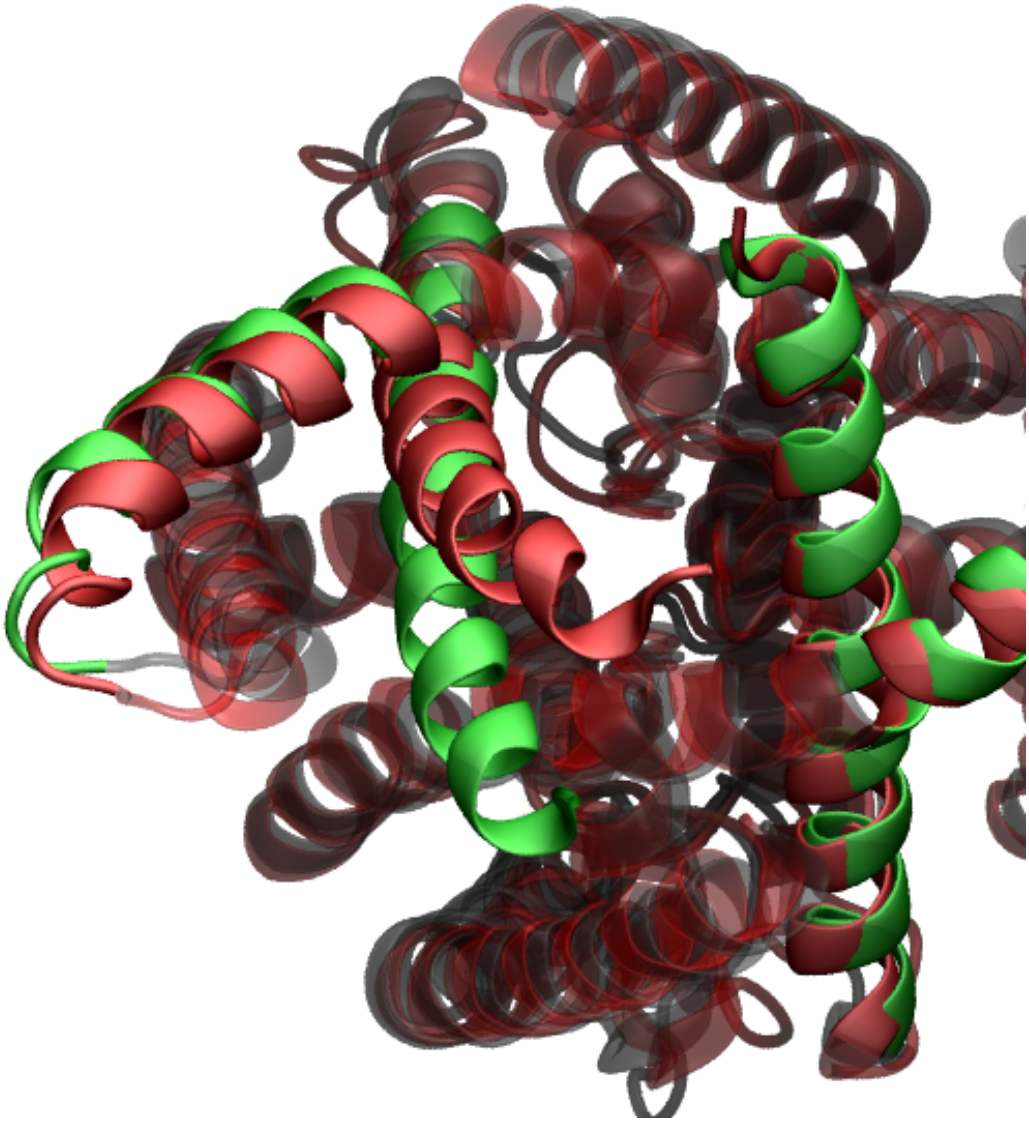

Figure A.3: I_1 and I_2 differ in positioning of the M4 helix. I_2 adopts the down-like configuration of this helix whereas I_1 adopts an an up-like configuration of this helix. Both intermediates adopt the down-like M2-M3 helix positions. From a purely structural standpoint, I_1 may be suggested to be "half down or half up". However, it is strongly kinetically related to the down conformations. Specifically, we observe a rapid relaxation from I_1 to I_2. Colors are as in fig. 3.2B.

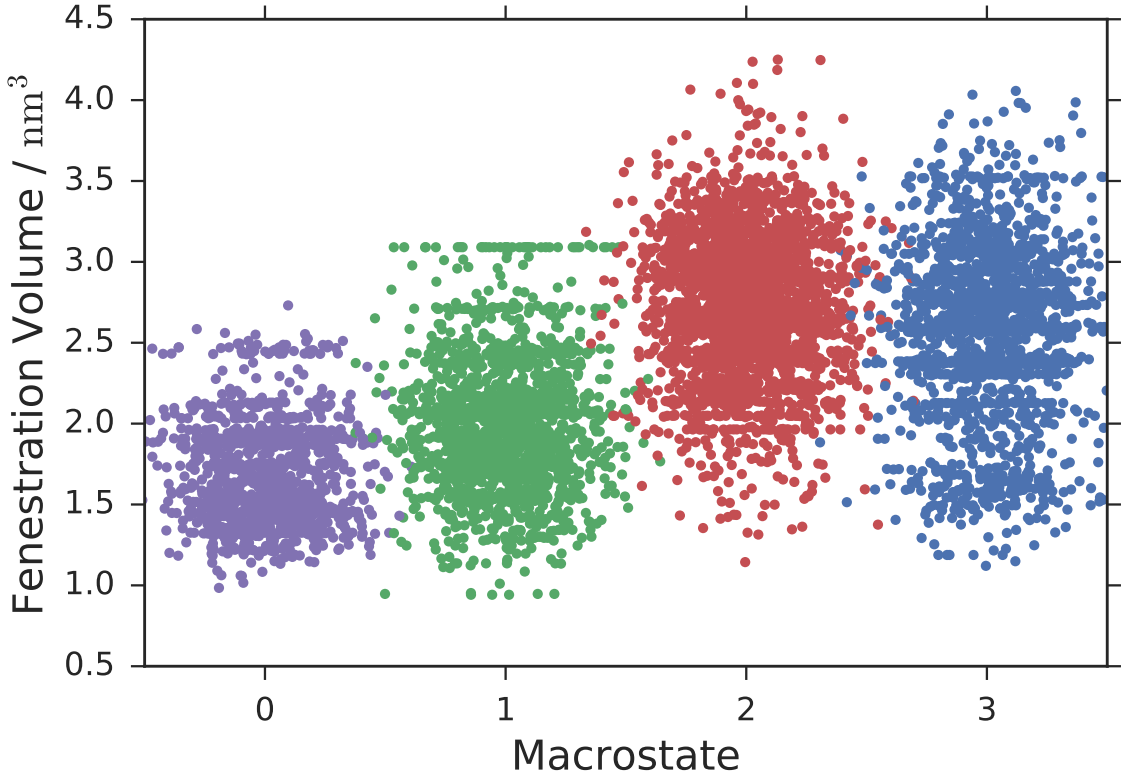

Figure A.4: Fenestration volume increases in macrostates I_2 and *Down*. Large fenestrations are created in the down conformations. Colors are as in fig. 3.2B.

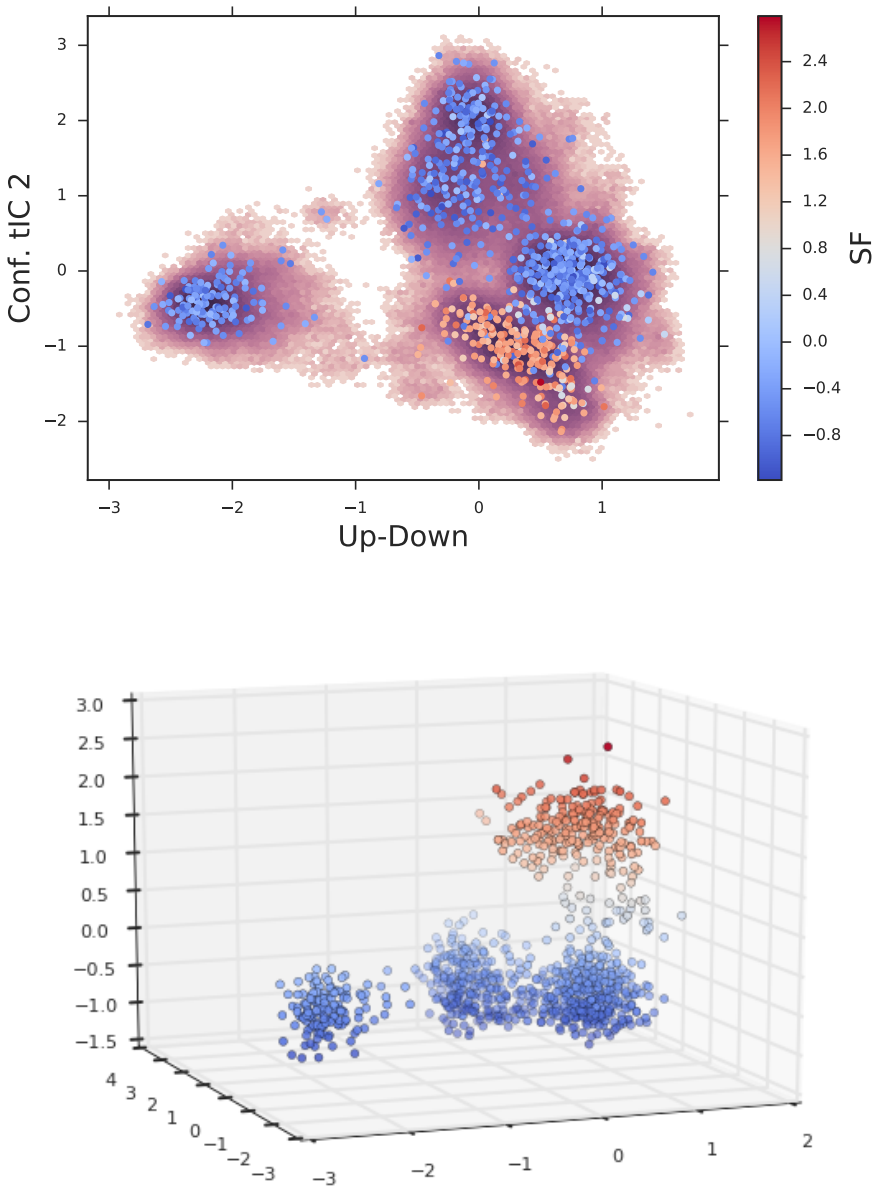

Figure A.5: We attempt to plot the three-dimensional histogram of observations versus the two conformational tICs and the SF tIC. (top) SF tIC value of cluster centers given by color. (bottom) 3D scatter plot of cluster centers, again colored by SF tIC value. These plots can be seen as a combination of fig. 3.2B and C. The coupling is shown by the inset table in fig. 3.2C.

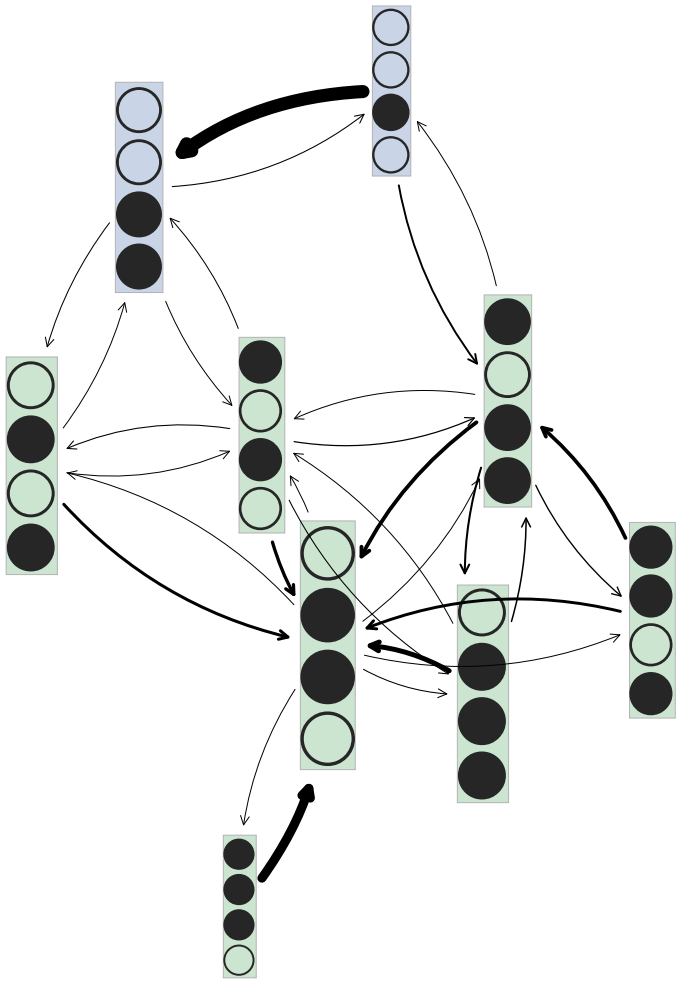

Figure A.6: Compare with fig. 3.4 A and C. Whereas those figures show ion microstate transition graphs partitioned by *Up* and *Down* macrostates (resp), here we plot the microstate transition graph for the whole dataset without partitioning by macrostate.

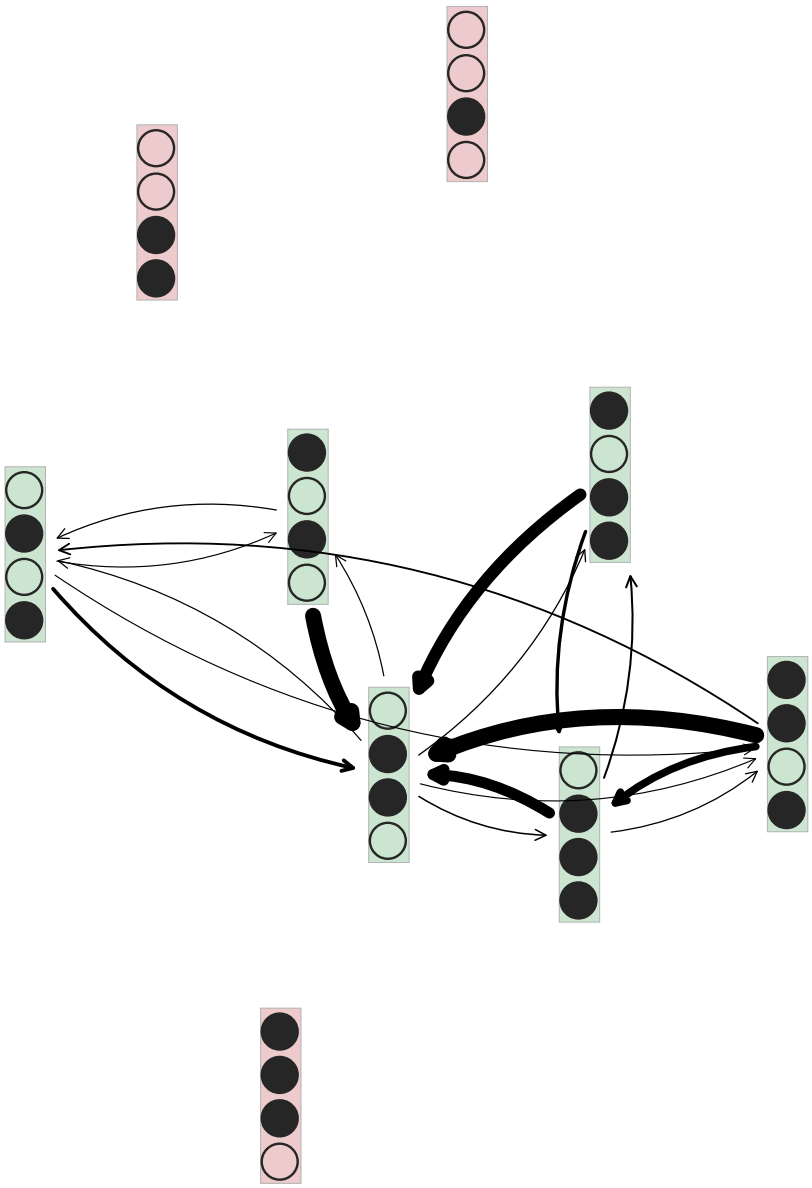

Figure A.7: Compare with fig. 3.4 A and C. Whereas those figures show ion microstate transition graphs partitioned by *Up* and *Down* macrostates (resp), here we plot the microstate transition graph for the I_1 macrostate.

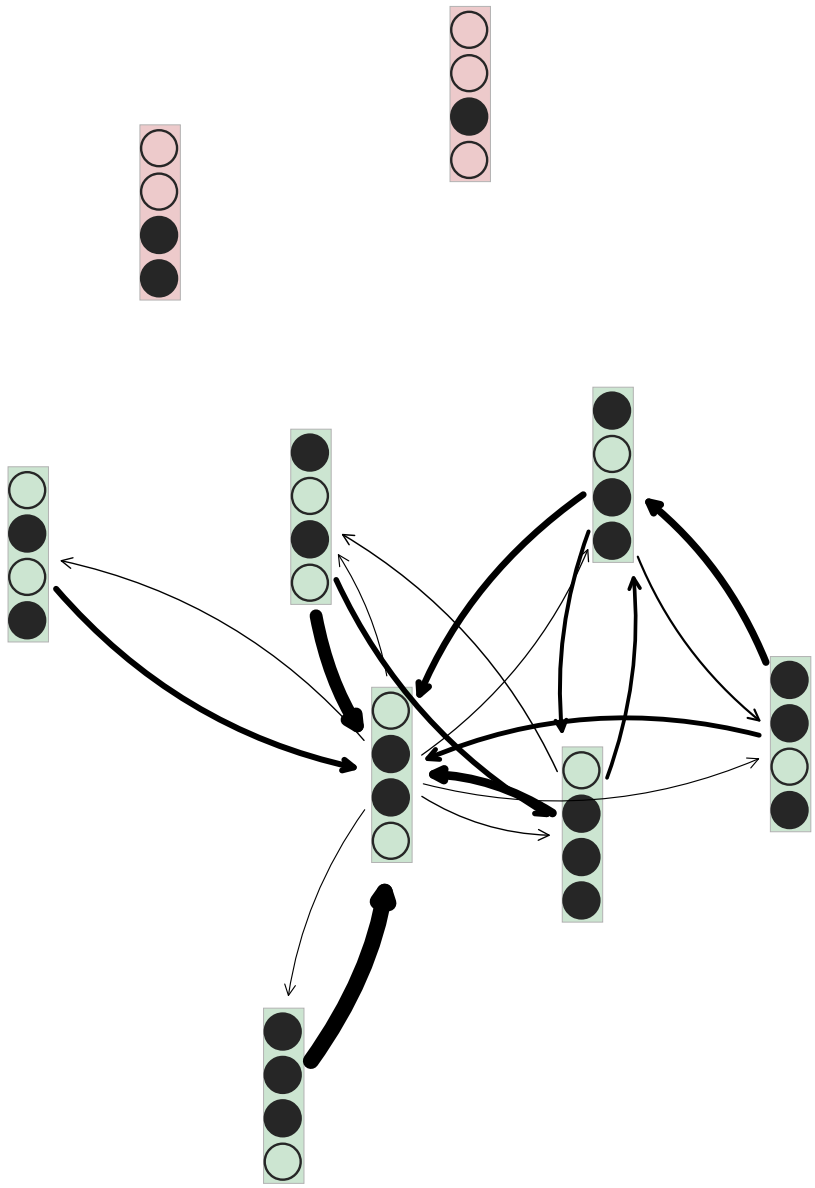

Figure A.8: Compare with fig. 3.4 A and C. Whereas those figures show ion microstate transition graphs partitioned by *Up* and *Down* macrostates (resp), here we plot the microstate transition graph for the I_2 macrostate.

Appendix B

Prior counts Quicken Convergence of MSMs

This appendix contains figures adapted from a poster presented at the Molecular and Chemical Kinetics conference, Freie Universität Berlin, Berlin, Germany in September 2015.

Markov state models (MSMs) model long-timescale dynamics by parameterizing k^2 transition probabilities, encoded in the transition matrix $\mathbb{T}$. This parameterization occurs at a discrete lag-time τ. The transition probabilities are in contrast to transition rates, which are reported per-unit-time and does not depend on the selection of a lag-time τ. Whereas a physical process characterized by rates (in rate matrix $\mathbf{K}$) may have sparse connectivity between state (i.e. only a subset of states have non-zero rates between them), at any $\tau > 0$, there should not be any transition probabilities which are zero. This can be seen mathematically in eq. (B.1) by the relationship between a matrix of rates and corresponding discrete time transition probabilities. The transition probability matrix can be understood as the decaying possibility of taking multiple jumps within a given lag-time window τ.

$$\mathbf{T} = \exp\left[\tau\mathbf{K}\right] = \sum_n \frac{1}{n!}\left(\tau\mathbf{K}\right)^n \tag{B.1}$$

$$= \mathbb{P}[1 \text{ step}] + \frac{1}{2}\mathbb{P}[2 \text{ steps}] + \frac{1}{6}\mathbb{P}[3 \text{ steps}] + \ldots \tag{B.2}$$

Although in practice our Markov state models' transition probability matrices are not completely dense, we can observe empirically in fig. B.1 that the probabilities become non-zero as we increase the sampling (improving the statistical accuracy of the model) or if we increase the lag time (increasing the change of a multi-step jump within one lag-time window).

The presence of probability-zero entries in the transition probability matrix breaks likelihood-based [78] and GMRQ [79] cross validation schemes. It renders information theoretical quantities like the KL-divergence between two transition matrices undefined.

Adding a small, fractional "count" to each entry in the counts matrix (from whence the transition matrix is estimated) speeds convergence of the model with respect to amount of sampling and does not negatively impact the timescale estimates of a model.

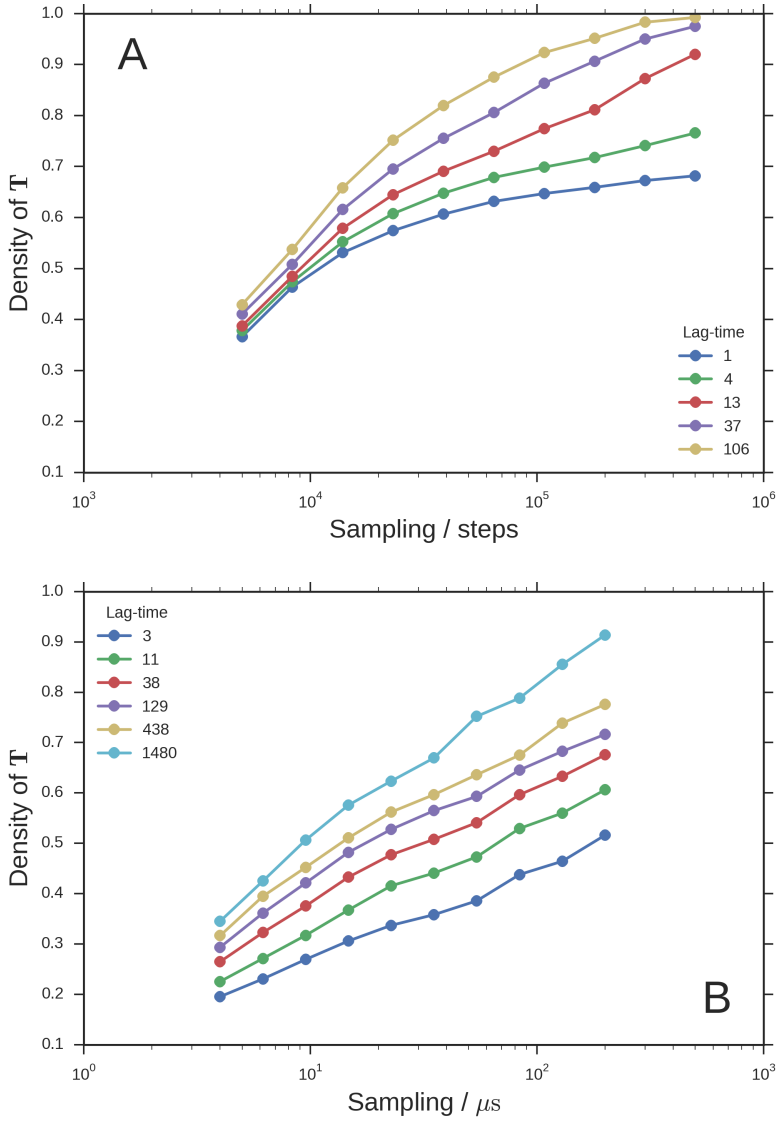

Figure B.1: The observed "density" of a transition matrix $\mathbf{T}$ increases with both (1) increased sampling and (2) increased lag-time τ. The density is the fraction of transition matrix elements that are non-zero. For a well-sampled system, the transition matrix entries should be fully dense. **(A)** density of transition probability matrix estimated from Muller potential Brownian dynamics. **(B)** likewise, estimated from fip35 WW domain simulations.

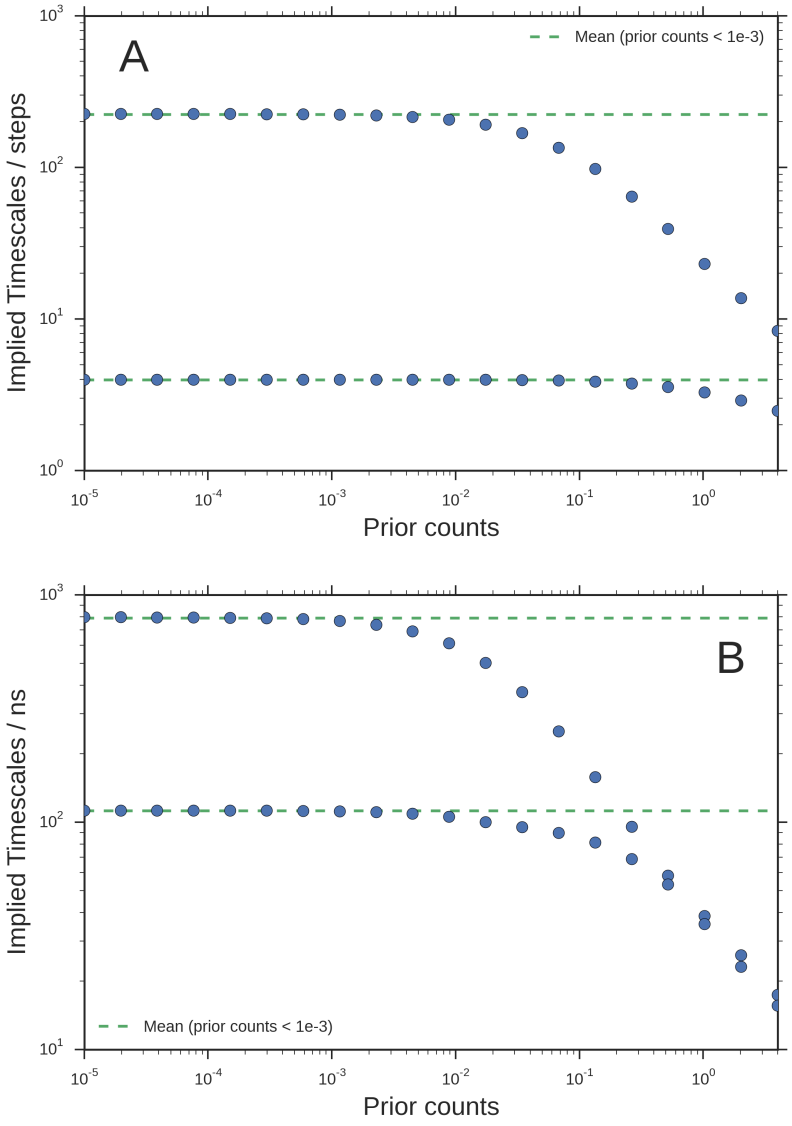

Figure B.2: Below a sufficiently small value, the prior count value does not affect the timescales of the model. The counts from data quickly overwhelm the prior counts during the estimation procedure. **(A)** The two slowest timescales as a function of prior count strength for Brownian dynamics on the Muller potential **(B)** likewise, for fip35 WW domain simulations.

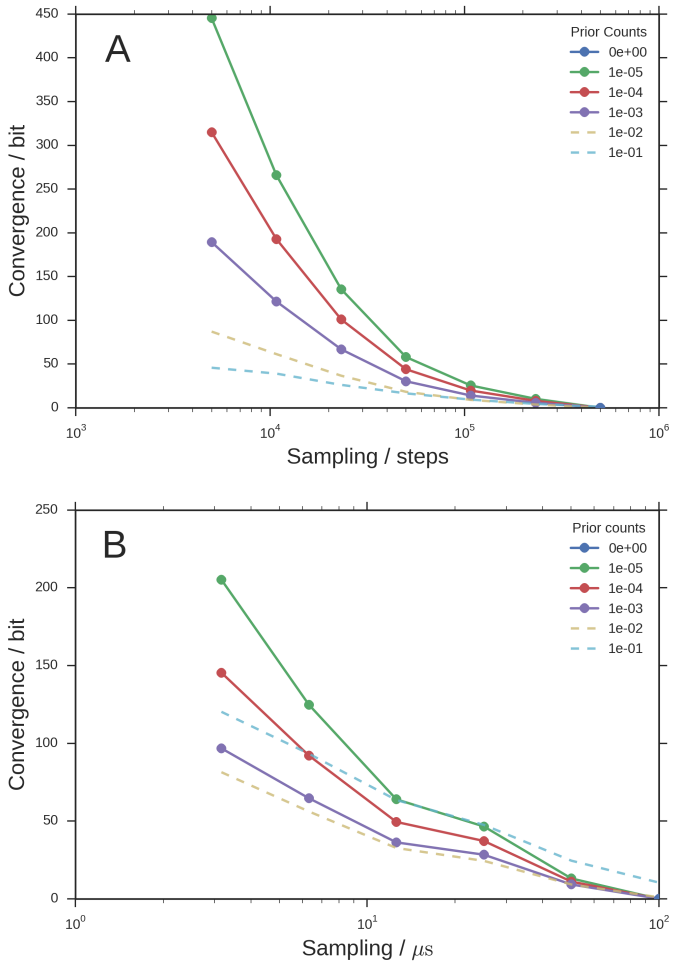

Figure B.3: Taking the model built with the entirety of the dataset as the gold standard, we present the convergence of models built with only part of the dataset. The convergence is measured as the KL divergence to the gold standard model. Increasing the strength of the prior count value speeds convergence as a function of sampling. Note that we cannot include the no-prior-count line on this plot, because the introduction of a zero probability in our candidate model is interpreted as infinitely unconverged according to the KL divergence. The dotted lines indicate prior count strength that was shown to negatively impact the timescales, see fig. B.2. **(A)** The convergence of models built on partial data of the Muller potential for varying prior count strengths. **(B)** likewise, for fip35 WW domain simulations.

Appendix C

Implementing Fast RMSD in Tensorflow

This chapter is adapted from a blog post written for the Pande Lab blog: "Tensorflow RMSD: Using Tensorflow for things it was not designed to do" available on Medium. Please forgive the more informal tone.

C.1 Introduction

Deep learning has revolutionized image and speech processing, allowing you to turn edges into cats. In our lab, we're applying these techniques to small molecule drug discovery.

A by-product of the revolution in deep learning has been the development of several high-quality open-source machine-learning frameworks that can compute gradients of arbitrary operations. Google's Tensorflow [159] may be the best known. My research has focused on understanding the results of large molecular dynamics simulations of proteins and other biomolecules. You can easily imagine framing the prediction of small molecule binding energies as a learning problem; can we leverage some of the deep learning advances for molecular dynamics for things in addition to artsy protein images?

A common operation in biophysics is computing the similarity of two protein

poses (conformations) with the RMSD distance metric. This metric is beloved for its respect of translational and rotational invariances. Roughly, it overlays two protein structures and reports the mean distance between an atom and its partner in the other structure.

C.2 RMSD and rotations

Satisfying translational symmetry is easy: you just center your proteins at the origin prior to doing a comparison

```
# traj = np.array(...) [shape (n_frames, n_atoms, 3)]
traj -= np.mean(traj, axis=1, keepdims=True)
```

Satisfying rotational symmetry is harder. You need to find the optimal rotation between each pair of conformations (optimal = minimizes the RMSD). Back in 1976, Kabsch [160] figured out that you could do an SVD of the 3x3 (xyz) correlation matrix to find the optimal rotation matrix.

```
# x1 = np.array(...) [shape (n_atoms, 3)]
correlation_matrix = np.dot(x1.T, x2)
V, S, W_tr = np.linalg.svd(correlation_matrix)
rotation = np.dot(V, W_tr)
x1_rot = np.dot(x1, rotation)
```

This isn't ideal because the SVD might give you a "rotoinversion", aka improper rotation, aka rotation followed by an inversion. We have to explicitly check for and fix this case:

```
correlation_matrix = np.dot(x1.T, x2)
V, S, W_tr = np.linalg.svd(correlation_matrix)
is_reflection = (np.linalg.det(V) * np.linalg.det(W_tr)) < 0.0
if is_reflection:
    V[:, -1] = -V[:, -1]
rotation = np.dot(V, W_tr)
x1_rot = np.dot(x1, rotation)
```

C.2.1 Quaternions to the rescue

In 1987, Horn [161] figured out that you can construct a 4x4 "key" matrix from combinations of elements of the correlation matrix. He derived this matrix from quaternion math (although the key matrix is a normal matrix). The leading eigenvalue of this matrix can be used to "rotationally correct" the naive squared difference between atomic coordinates.

```
correlation_matrix = np.dot(x1.T, x2)
F = key_matrix(correlation_matrix)
vals, vecs = np.linalg.eigh(F)
max_val = vals[-1] # numpy sorts ascending
sd = np.sum(traj1 ** 2 + traj2 ** 2) - 2 * max_val
msd = sd / n_atoms
rmsd = np.sqrt(msd)
```

Crucially, you don't need to explicitly construct a rotation matrix to find the RMSD value. If you want the rotation, you can reconstruct it from the leading eigen*vector* of the key matrix.

C.3 Tensorflow can do RMSD

We've formulated the problem as vector operations and one self-adjoint eigenvalue problem. All of these operations are implemented in Tensorflow!

```
def key_matrix(r):
    return [
        [r[0][0] + r[1][1] + r[2][2], r[1][2] - r[2][1], r[2][0] - r[0][2], r[0][1] - r[1][0]],
        [r[1][2] - r[2][1], r[0][0] - r[1][1] - r[2][2], r[0][1] + r[1][0], r[0][2] + r[2][0]],
        [r[2][0] - r[0][2], r[0][1] + r[1][0], -r[0][0] + r[1][1] - r[2][2], r[1][2] + r[2][1]],
        [r[0][1] - r[1][0], r[0][2] + r[2][0], r[1][2] + r[2][1], -r[0][0] - r[1][1] + r[2][2]],
    ]


def squared_deviation(frame, target):
    R = tf.matmul(frame, target, transpose_a=True)
    R_parts = [tf.unstack(t) for t in tf.unstack(R)]
    F_parts = key_matrix(R_parts)
```

```
    F = tf.stack(F_parts, name='F')
    vals, vecs = tf.self_adjoint_eig(F, name='eig')
    lmax = tf.unstack(vals)[-1] # tensorflow sorts ascending
    sd = tf.reduce_sum(frame ** 2 + target ** 2) - 2 * lmax
```

The benefit is now we get derivatives for free, so we can do interesting things. As a toy example, this shows finding a "consensus" structure that minimizes average RMSD to every frame in a molecular dynamics trajectory

This is the normal, Tensorflow code used to perform the optimization.

```
target = tf.Variable(tf.truncated_normal(
        (1, n_atoms, 3), stddev=0.3), name='target')
msd, rot = pairwise_msd(traj, target)
loss = tf.reduce_mean(msd, axis=0)

optimizer = tf.train.AdamOptimizer(1e-3)
train = optimizer.minimize(loss)

sess = tf.Session()
sess.run(tf.global_variables_initializer())
for step in range(2500):
    sess.run(train)
```

C.4 Tensorflow can't do RMSD very well

Doing an eigendecomposition for each data point gets expensive, especially since I want to be able to do pairwise (R)MSD calculations between a large trajectory and a sizable number of target structures. MDTraj can perform a huge number of RMSD calculations exceedingly quickly. It uses a better strategy for finding the leading eigenvalue of the 4x4 key matrix from above. The Theobald QCP [162, 163] method from 2005 explicitly writes out the characteristic polynomial for the key matrix. We use the fact that there is a bound for identical structures ((R)MSD = 0) to choose a starting point for an iterative, Newton method of finding the leading eigenvalue.

If we start from this point, we're guaranteed that the first root of the characteristic polynomial will be the largest eigenvalue. So let's code this up in Tensorflow! Not so fast (literally): you can't really do iteration in Tensorflow, and who knows how performant it would be if you could.

C.4.1 Custom Pairwise MSD Op

Instead, I implemented a custom Tensorflow "op". At first, I was intimidated by having to build and keep track of a custom Tensorflow installation. Luckily, Tensorflow will happily load shared libraries to register Ops at runtime. Even better, a Pande Group Alumn Imran Haque implemented a fast (R)MSD calculation implementation in C that I could wrap.

I implemented an Op that does pairwise MSD calculations where the double-for-loop is parallelized with OpenMP. In addition to the 10,000x speedup from the native Tensorflow implementation of the Horn method, we're slightly faster than MDTraj even though it's using the same implementation under the hood. For MDTraj, the looping over a trajectory is done with OpenMP in C, but the iteration over targets has to be done in Python with its associated overhead.

I ran a benchmark which performs a pairwise RMSD calculation among fs peptide trajectories. Specifically, between 2800 (stride = 100) frames and 28 targets (stride = 100 * 100).

Table C.1: RMSD speed benchmarks. The custom Op is more than 10,000 times faster than the naive Tensorflow implementation.

Implementation	Time / ms
TF Native Ops	22,843
MDTraj	33.3
TF Custom Op	0.9
TF Custom Op (w/rot)	1.6

C.5 What about gradients?

The reason why we wanted to use Tensorflow in the first place was to do fun things with the automatic differentiation. There's no free lunch, and Tensorflow will not auto-differentiate our custom Op. Coutsias et. al. [164] pointed out that the derivative of the MSD is simply the difference between the coordinates in the superposed pair of structures. We can code this.

The first problem is now we need the rotation matrix explicitly so we can use it to compute the gradients. Remember that Theobald came up with a smart method for finding the leading eigenvalue, but that only gives us the RMSD value, not the actual rotation (which requires the eigenvector). Luckily, in 2010 he extended the method to use the leading eigenvalue to quickly find the leading eigenvector.

I modified the `pairwise_msd` op to return a (n_frames, n_targets) pairwise MSD matrix *and* the (n_frames, n_targets, 3, 3) rotation matrices. Users should never use the rotation matrices for further calculations because I didn't implement the derivatives for that output. Instead, I use that output in the gradient calculation for the MSDs. If someone knows a better way to do this, please let me know.

In the benchmark table, this version of the Op is the "w/rot" variant, and is slower (because it has to do more work).

C.5.1 Gradient computation details

Most of the code in this subsection is just making tensors the right shape. We need to apply our n_frames * n_targets rotation matrices individually to each conformation, and we need to mix in the gradient `grad` from the previous Op in the compute graph, so we blow everything up to a rank 4 matrix and *explicitly tile* the conformations to be rotated because `matmul` doesn't do broadcasting.

```
rots = op.outputs[1]
N1 = int(confs1.get_shape()[0])
N2 = int(confs2.get_shape()[0])

# expand from (n_frames, n_targets) to (n_frame, n_targets, 1, 1)
```

```
grad = tf.expand_dims(tf.expand_dims(grad, axis=-1), axis=-1)

# expand from (n_frames OR n_targets, n_atoms, 3)
# to (n_frames OR 1, 1 OR n_targets, n_atoms, 3)
expand_confs1 = tf.expand_dims(confs1, axis=1)
expand_confs2 = tf.expand_dims(confs2, axis=0)

# Explicitly tile conformations for matmul
big_confs1 = tf.tile(expand_confs1, [1, N2, 1, 1])
big_confs2 = tf.tile(expand_confs2, [N1, 1, 1, 1])

# This is the gradient!
dxy = expand_confs1 - tf.matmul(big_confs2, rots, transpose_b=True)
dyx = expand_confs2 - tf.matmul(big_confs1, rots, transpose_b=False)
```

The actual form of the gradient has a couple factors which we must include:

```
n_atom = float(int(confs1.get_shape()[1]))
dxy = 2 * dxy / n_atom
dyx = 2 * dyx / n_atom
```

Finally, we sum over the axis that has the *other* conformations to make sure our gradient tensors match in shape to their variables.

```
dr_dc1 = tf.reduce_sum(grad * dxy, axis=1)
dr_dc2 = tf.reduce_sum(grad * dyx, axis=0)
```

Did you forget about translational symmetry after all this focus on rotation? I did originally! It's important to test your code on a variety of inputs *including* trajectories that aren't pre-centered. Let's use Tensorflow's automatic differentiation for this part.

Specifically, we set up the "forward" op and call `tf.gradients` on it. We pass in our gradients w.r.t. rotation as the `grad_ys` argument.

```
centered1 = confs1 - tf.reduce_mean(confs1, axis=1, keep_dims=True)
centered2 = confs2 - tf.reduce_mean(confs2, axis=1, keep_dims=True)
dc_dx1 = tf.gradients(centered1, [confs1], grad_ys=dr_dc1)[0]
dc_dx2 = tf.gradients(centered2, [confs2], grad_ys=dr_dc2)[0]
```

C.6 KMeans-inspired RMSD clustering

As an example of what we can do with our fast pairwise MSD op with gradients, let's find "optimal" cluster centers (centroids). For a trajectory of conformations, find centers that minimize the distance between each point and its closest centroid. To prevent it from finding the same centroid twice, we add a penalty to force the centroids apart. Be careful to make sure this penalty saturates at some point or your optimization will just make really different centroids with no respect for inter-cluster distances.

```
# Our inputs
n_clusters = 2
target = tf.Variable(tf.truncated_normal(
        (n_clusters, traj.xyz.shape[1], 3), stddev=0.3))

# Set up the compute graph
msd, rot = rmsd_op.pairwise_msd(traj.xyz, target)
nearest_cluster = msd * tf.nn.softmax(-msd)
cluster_dist = tf.reduce_mean(nearest_cluster, axis=(0, 1))
cluster_diff, _ = rmsd_op.pairwise_msd(target, target)
cluster_diff = cluster_diff[0, 1]
loss = cluster_dist - tf.tanh(cluster_diff*10)

# Train it in the normal way
optimizer = tf.train.AdamOptimizer(5e-3)
train = optimizer.minimize(loss)

sess = tf.Session()
sess.run(tf.global_variables_initializer())
for step in range(1000):
    sess.run(train)
```

Now you can do tICA or make an MSM in this nice space.

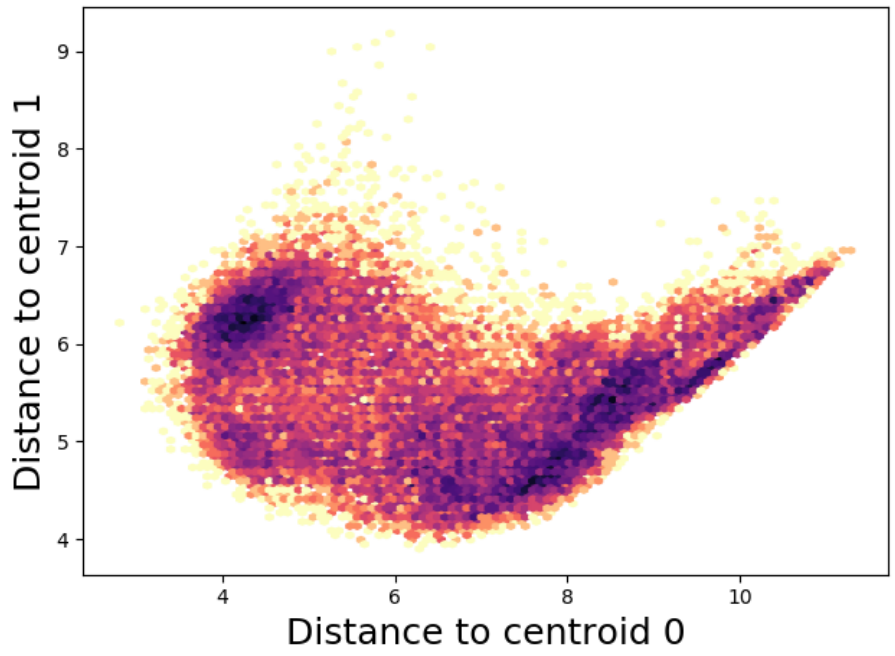

Figure C.1: In addition to being able to histogram / plot, there's no limit to the things you can do now that you're in a Euclidean vector space rather than having to deal with RMSDs. For example, you can do tICA or PCA now.

Code Availability

All code is available on Github. Make sure you check out the README for installation instructions, as the custom Op requires a working c++ compiler. The consensus example, clustering example, and profiling script are found in the examples folder and require the fs peptide dataset.

The native Tensorflow implementation lives in rmsd.py. The low level code for the custom Op lives in the rmsd/ subfolder, specifically rmsd.cpp. Finally, rmsd_op.py contains a convenience function for loading the shared object that registers the Op. It also implements the gradients (in Python).

Appendix D

Supplementary information for Learnable Soft MSMs

For each system, the top two timescales were optimized and scored using GRMQ. The number of discrete-state MSM states is technically one higher than reported. The discrete state MSM does not subtract mean feature values, and estimates the equilibrium distribution as the first eigenvector. For fair comparison, $n + 1$ hard states is considered equal to n lktICA landmark points. This favors the less-performant previous model; the conclusions of this manuscript would not be different (or would be strengthened) if this modification was not made. The number of shuffle-split folds was 10, and trajectories were kept whole (notwithstanding the splitting described in fig. D.6 and fig. D.7). A fixed shrinkage value of 10^{-3} was used instead of estimating this with the MSMBuilder default Rao-Blackwellized Ledoit-Wolf estimator to simplify the Tensorflow implementation of the algorithms. The Adam optimizer with learning rate set to 10^{-4} was used for optimization algorithms. 5000 steps of optimization were performed with a batch size of 1000. This means that 1000 randomized lagged-tuples were used during training of the model. For GMRQ scoring, all test/train data was used.

The sine and cosine transform of dihedral angles was computed to account for periodicity, and the resulting values were scaled per-feature by their inter-quartile range (the so-called "robust scaler").

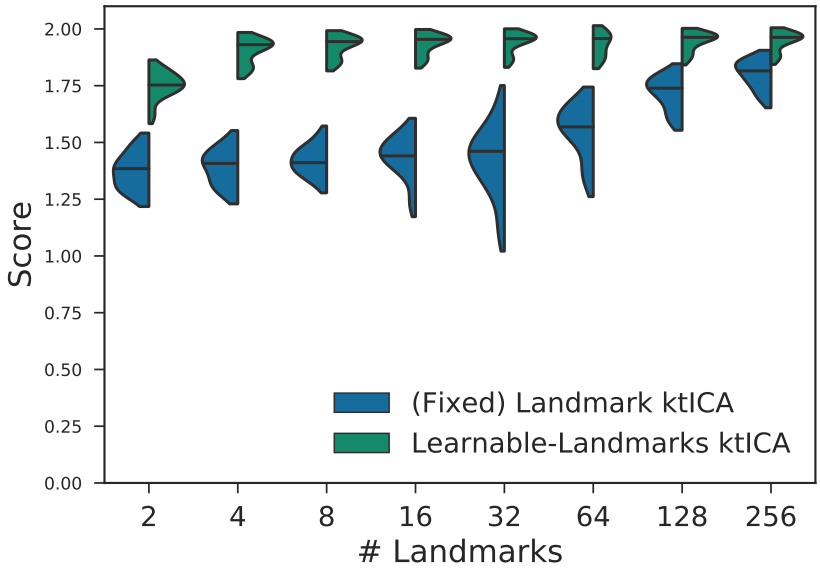

Figure D.1: Learnable landmark points improve ktICA models relative to fixed landmark points for BPTI as well. A distribution of GMRQ scores were calculated for differing number of landmark points. This is a companion plot to fig. 6.3.

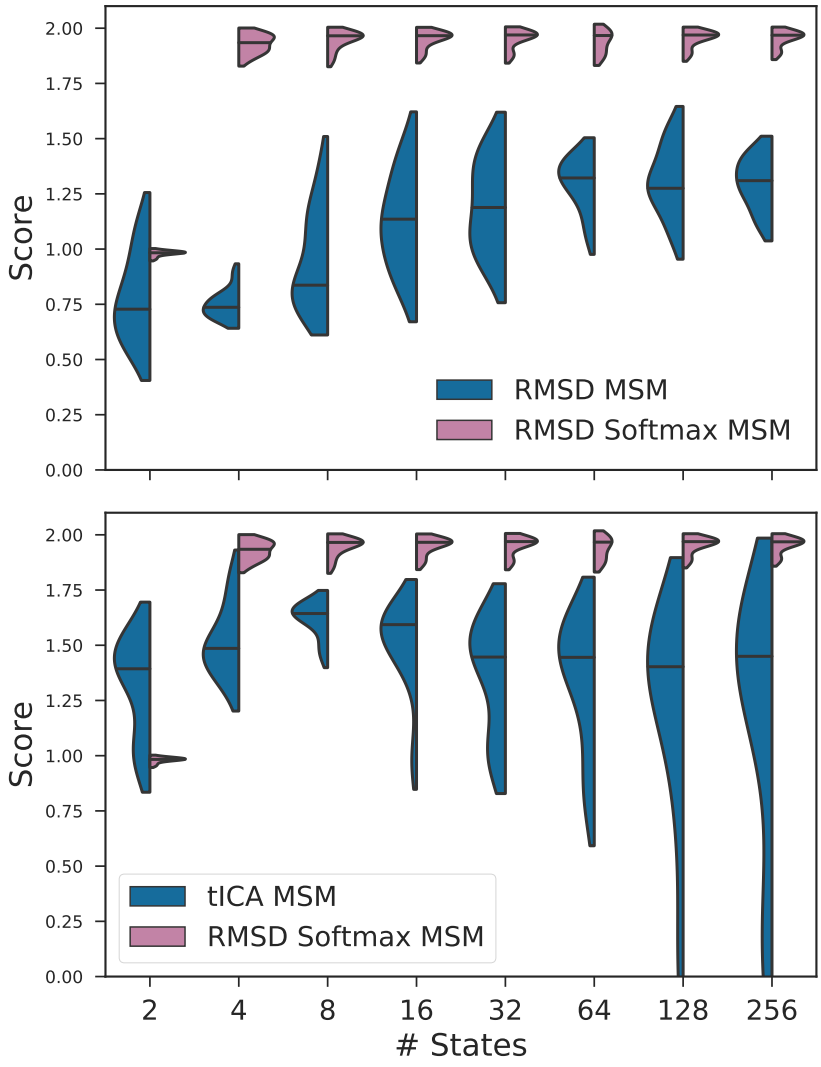

Figure D.2: This is the same comparison as fig. 6.5, but for BPTI. The softmax MSM is highly performant. For the bottom panel, we again show the RMSD softmax MSM instead of the dihedral softmax MSM because for this system the dihedral version was less performant, but see fig. D.4 for all the data.

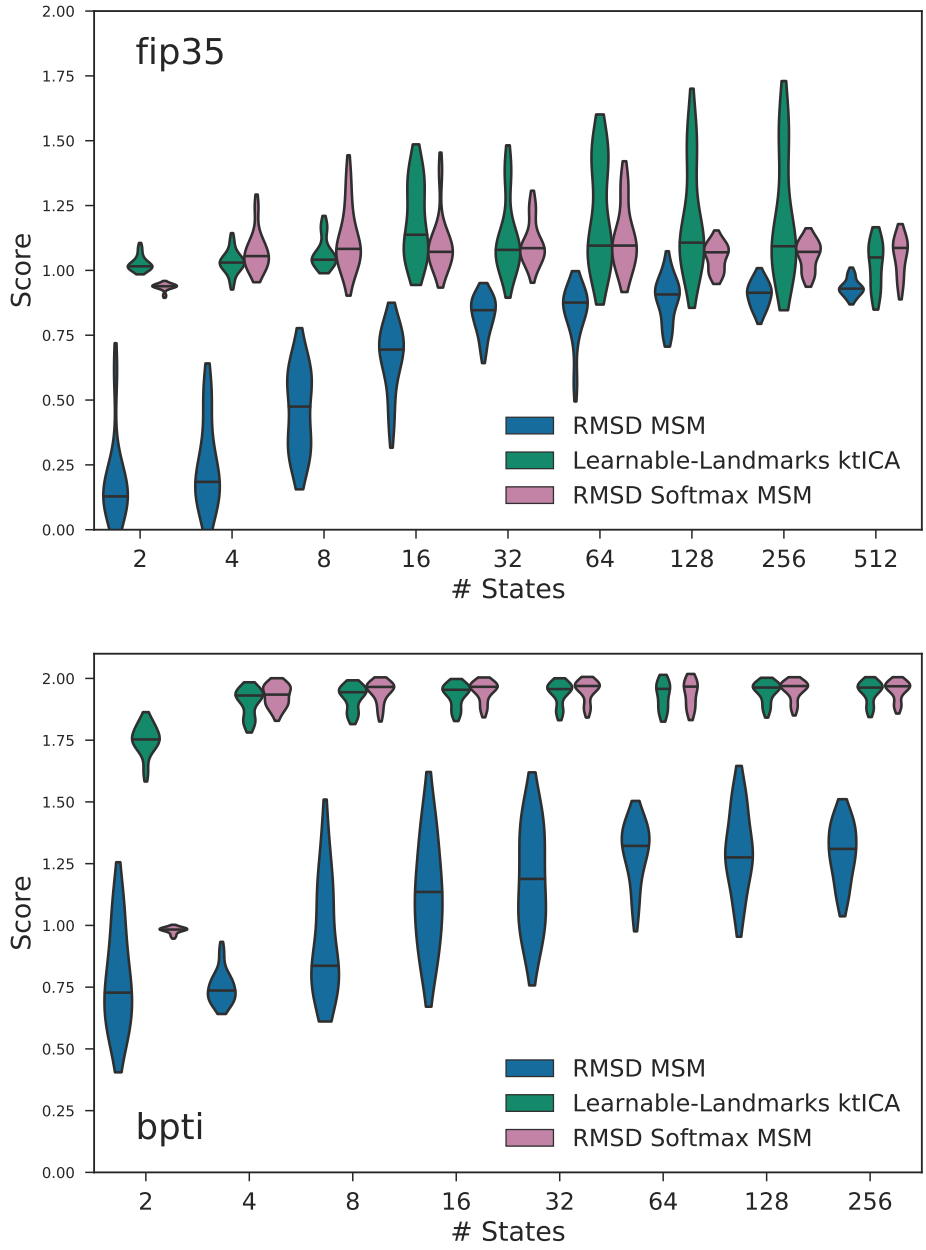

Figure D.3: Softmax MSMs built on RMSD distances are better than discrete MSMs. A distribution of GMRQ scores were calculated for differing number of states. **top:** fip35 WW domain. **bottom:** BPTI

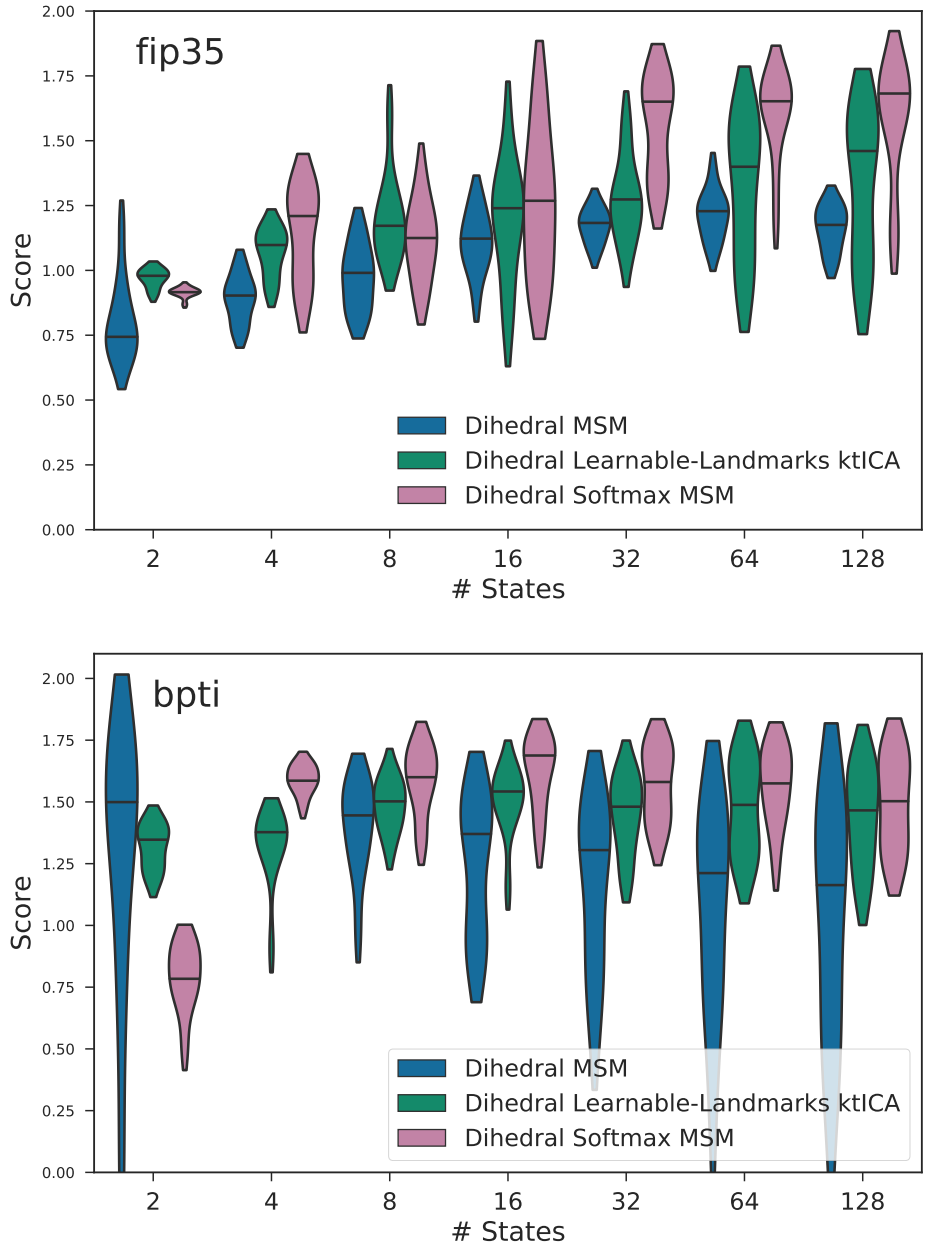

Figure D.4: Softmax MSMs built on dihedral angles are generally better than discrete MSMs. A distribution of GMRQ scores were calculated for differing number of states. **top:** fip35 WW domain. **bottom:** BPTI

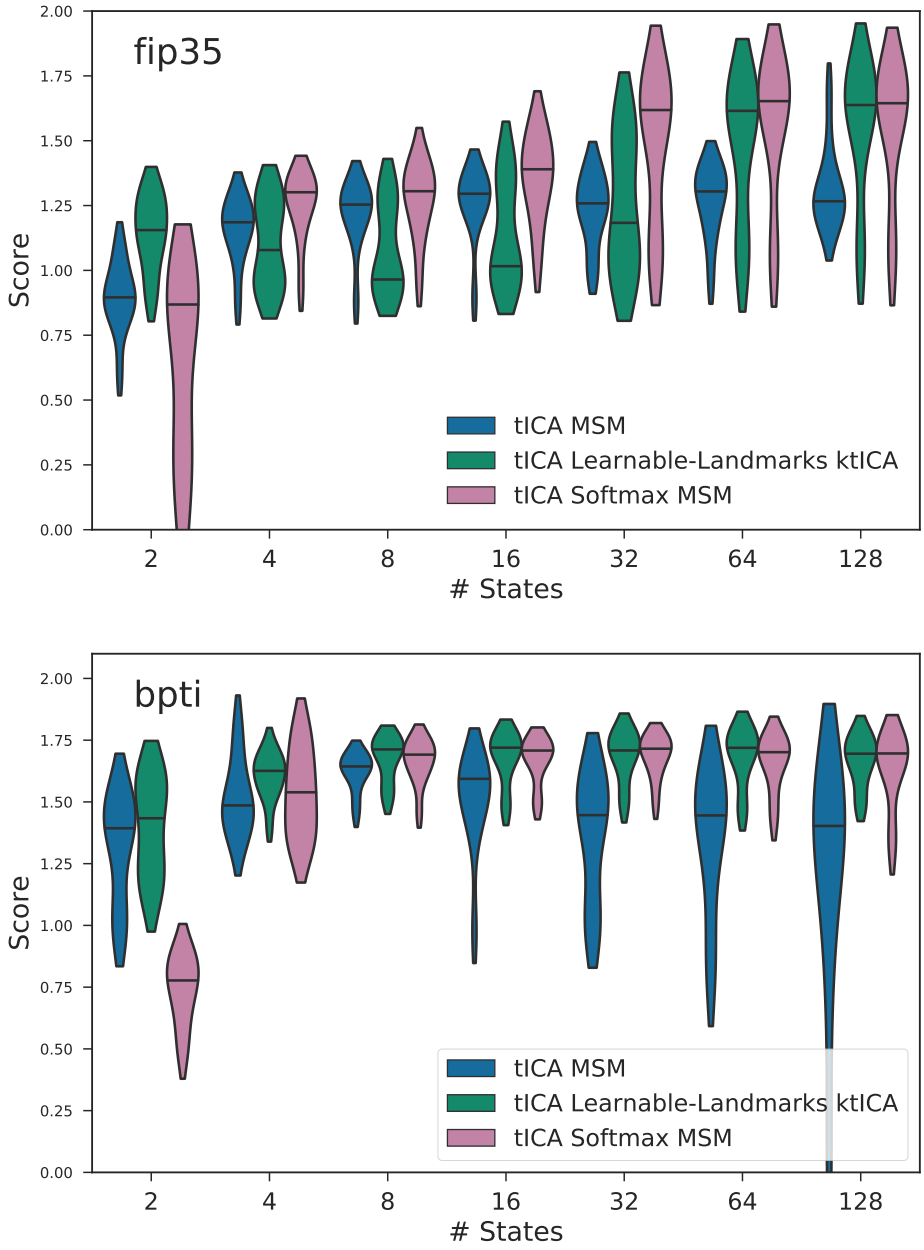

Figure D.5: Softmax MSMs built on linear tICs of dihedral angles are generally better than similarly estimated discrete MSMs A distribution of GMRQ scores were calculated for differing number of states. **top:** fip35 WW domain. **bottom:** BPTI

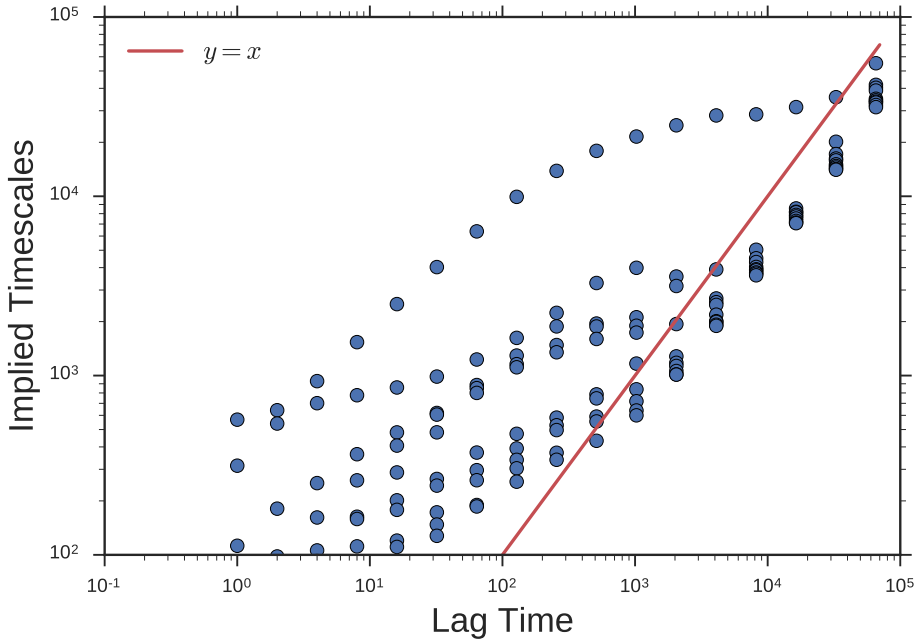

Figure D.6: The fip35 data was saved every 200 ps for a total of 200 µs across two equal-length trajectories. These two trajectories were split into five chunks for a total of 5 trajectories over which we performed the shuffle-split cross validation. During cross validation, the trajectories were strided by 10, so each frame was 2 ns. The lag-time was selected to be 10^3 steps (per this plot) (100 steps in the strided trajectories) or 200 ns.

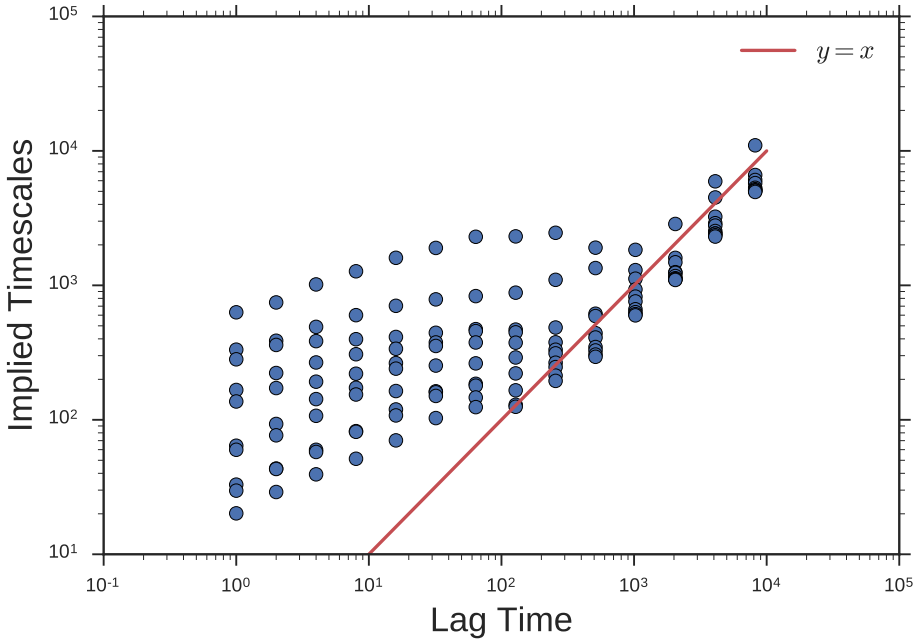

Figure D.7: The BPTI data was saved every 25 ns for a total of 1 ms across one trajectory. This trajectory was split into 10 chunks over which we performed the shuffle-split cross validation. The lag-time was selected to be 10^2 steps (per this plot) or 2.5 µs.

Bibliography

[1] Richard Feynman. There's plenty of room at the bottom. *Engineering and Science*, 23(5):22–36, 1960.

[2] 'Plenty of room' revisited. *Nat. Nanotechnol.*, 4(12):781–781, 2009. doi: 10.1038/nnano.2009.356.

[3] Christian R. Schwantes, Diwakar Shukla, and Vijay S. Pande. Markov state models and tICA reveal a nonnative folding nucleus in simulations of NuG2. *Biophys. J.*, 110(8):1716–1719, 2016. doi: 10.1016/j.bpj.2016.03.026.

[4] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011. doi: 10.1126/science.1208351.

[5] Vijay S. Pande. Simple theory of protein folding kinetics. *Phys. Rev. Lett.*, 105 (19), 2010. doi: 10.1103/PhysRevLett.105.198101.

[6] 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. doi: 10.1126/science.1187409.

[7] Ken A. Dill, S. Banu Ozkan, M. Scott Shell, and Thomas R. Weikl. The protein folding problem. *Annu. Rev. Biophys.*, 37(1):289–316, 2008. doi: 10.1146/annurev.biophys.37.092707.153558.

[8] Lisa J Lapidus, Srabasti Acharya, Christian R Schwantes, Ling Wu, Diwakar

Shukla, Michael King, Stephen J DeCamp, and Vijay S Pande. Complex pathways in folding of protein g explored by simulation and experiment. *Biophys. J.*, 107:947–955.

[9] Maria M. Flocco and Sherry L. Mowbray. C-based torsion angles: A simple tool to analyze protein conformational changes. *Protein Sci.*, 4(10):2118–2122, 1995. doi: 10.1002/pro.5560041017.

[10] Barry J Grant, Alemayehu A Gorfe, and J Andrew McCammon. Large conformational changes in proteins: signaling and other functions. *Curr. Opin. Struc. Bio.*, 20:142–147.

[11] S. Fischer, B. Windshugel, D. Horak, K. C. Holmes, and J. C. Smith. Structural mechanism of the recovery stroke in the myosin molecular motor. *Proc. Natl. Acad. Sci. U.S.A.*, 102(19):6873–6878, 2005. doi: 10.1073/pnas.0408784102.

[12] Frank Noé, Dieter Krachtus, Jeremy C. Smith, and Stefan Fischer. Transition networks for the comprehensive characterization of complex conformational change in proteins. *J. Chem. Theory Comput.*, 2(3):840–857, 2006. doi: 10.1021/ct050162r.

[13] Kai J. Kohlhoff, Diwakar Shukla, Morgan Lawrenz, Gregory R. Bowman, David E. Konerding, Dan Belov, Russ B. Altman, and Vijay S. Pande. Cloud-based simulations on google exacycle reveal ligand modulation of GPCR activation pathways. *Nature Chem.*, 6(1):15–21, 2013. doi: 10.1038/nchem.1821.

[14] 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. doi: 10.1073/pnas.1103547108.

[15] Gianni De Fabritiis, Sebastien Geroult, Peter V. Coveney, and Gabriel Waksman. Insights from the energetics of water binding at the domain-ligand interface of the src SH2 domain. *Proteins Struct. Funct. Bioinf.*, 72(4):1290–1297, 2008. doi: 10.1002/prot.22027.

[16] Tom Young, Robert Abel, Byungchan Kim, Bruce J. Berne, and Richard A. Friesner. Motifs for molecular recognition exploiting hydrophobic enclosure in protein–ligand binding. *Proc. Natl. Acad. Sci. U.S.A.*, 104(3):808–813, 2007. doi: 10.1073/pnas.0610202104.

[17] Eva-Maria Strauch, Steffen M Bernard, David La, Alan J Bohn, Peter S Lee, Caitlin E Anderson, Travis Nieusma, Carly A Holstein, Natalie K Garcia, Kathryn A Hooper, Rashmi Ravichandran, Jorgen W Nelson, William Sheffler, Jesse D Bloom, Kelly K Lee, Andrew B Ward, Paul Yager, Deborah H Fuller, Ian A Wilson, and David Baker. Computational design of trimeric influenza-neutralizing proteins targeting the hemagglutinin receptor binding site. *Nat. Biotechnol.*, 35(7):667–671, 2017. doi: 10.1038/nbt.3907.

[18] Kathryn M. Hart, Chris M. W. Ho, Supratik Dutta, Michael L. Gross, and Gregory R. Bowman. Modelling proteins' hidden conformations to predict antibiotic resistance. *Nat. Commun.*, 7:12965, 2016. doi: 10.1038/ncomms12965.

[19] 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. doi: 10.1063/1.4895044.

[20] Thomas J Lane, Diwakar Shukla, Kyle A Beauchamp, and Vijay S Pande. To milliseconds and beyond: challenges in the simulation of protein folding. *Curr. Opin. Struct. Biol.*, 23(1):58–65, 2013. doi: 10.1016/j.sbi.2012.11.002.

[21] Peter L. Freddolino, Christopher B. Harrison, Yanxin Liu, and Klaus Schulten. Challenges in protein-folding simulations. *Nat. Phys.*, 6(10):751–758, 2010. doi: 10.1038/nphys1713.

[22] Kyle A. Beauchamp, Yu-Shan Lin, Rhiju Das, and Vijay 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. doi: 10.1021/ct2007814.

[23] Alexander D. Mackerell, Michael Feig, and Charles L. Brooks. Extending the treatment of backbone energetics in protein force fields: Limitations of gas-phase quantum mechanics in reproducing protein conformational distributions in molecular dynamics simulations. *J. Comput. Chem.*, 25(11):1400–1415, 2004. doi: 10.1002/jcc.20065.

[24] Par Bjelkmar, Per Larsson, Michel A. Cuendet, Berk Hess, and Erik Lindahl. Implementation of the CHARMM force field in GROMACS: analysis of protein stability effects from correction maps, virtual interaction sites, and water models. *J. Chem. Theory Comput.*, 6(2):459–466, 2010. doi: 10.1021/ct900549r.

[25] Kresten Lindorff-Larsen, Stefano Piana, Kim Palmo, Paul Maragakis, John L Klepeis, Ron O Dror, and David E Shaw. Improved side-chain torsion potentials for the amber ff99SB protein force field. *Proteins: Struct., Funct., and Bioinf.*, 78:1950–1958.

[26] James A Maier, Carmenza Martinez, Koushik Kasavajhala, Lauren Wickstrom, Kevin E Hauser, and Carlos Simmerling. ff14SB: improving the accuracy of protein side chain and backbone parameters from ff99SB. *J. Chem. Theory Comput.*, 11:3696–3713.

[27] 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.*, 79:926.

[28] J. L. F. Abascal and C. Vega. A general purpose model for the condensed phases of water: TIP4P/2005. *J. Chem. Phys.*, 123(23):234505, 2005. doi: 10.1063/1.2121687.

[29] Lee-Ping Wang, Teresa Head-Gordon, Jay W. Ponder, Pengyu Ren, John D. Chodera, Peter K. Eastman, Todd J. Martinez, and Vijay S. Pande. Systematic improvement of a classical molecular model of water. *J. Phys. Chem. B*, 117 (34):9956–9972, 2013. doi: 10.1021/jp403802c.

[30] Callum J. Dickson, Benjamin D. Madej, Åge A. Skjevik, Robin M. Betz, Knut Teigen, Ian R. Gould, and Ross C. Walker. Lipid14: The amber lipid force field. *J. Chem. Theory Comput.*, 10(2):865–879, 2014. doi: 10.1021/ct4010307.

[31] Junmei Wang, Romain M. Wolf, James W. Caldwell, Peter A. Kollman, and David A. Case. Development and testing of a general amber force field. *J. Comput. Chem.*, 25(9):1157–1174, 2004. doi: 10.1002/jcc.20035.

[32] Timothy C. Moore, Christopher R. Iacovella, and Clare McCabe. Derivation of coarse-grained potentials via multistate iterative boltzmann inversion. *J. Chem. Phys.*, 140(22):224104, 2014. doi: 10.1063/1.4880555.

[33] Guillaume Lamoureux and Benoıt Roux. Modeling induced polarization with classical drude oscillators: Theory and molecular dynamics simulation algorithm. *J. Chem. Phys.*, 119(6):3025–3039, 2003. doi: 10.1063/1.1589749.

[34] Jay W. Ponder, Chuanjie Wu, Pengyu Ren, Vijay S. Pande, John D. Chodera, Michael J. Schnieders, Imran Haque, David L. Mobley, Daniel S. Lambrecht, Robert A. DiStasio, Martin Head-Gordon, Gary N. I. Clark, Margaret E. Johnson, and Teresa Head-Gordon. Current status of the AMOEBA polarizable force field. *J. Phys. Chem. B*, 114(8):2549–2564, 2010. doi: 10.1021/jp910674d.

[35] Lee-Ping Wang, Todd J. Martinez, and Vijay S. Pande. Building force fields: An automatic, systematic, and reproducible approach. *J. Phys. Chem. Lett.*, 5 (11):1885–1891, 2014. doi: 10.1021/jz500737m.

[36] Lee-Ping Wang, Keri A. McKiernan, Joseph Gomes, Kyle A. Beauchamp, Teresa Head-Gordon, Julia E. Rice, William C. Swope, Todd J. Martínez, and Vijay S. Pande. Building a more predictive protein force field: A systematic and reproducible route to AMBER-FB15. *J. Phys. Chem. B*, 121(16):4023–4039, 2017. doi: 10.1021/acs.jpcb.7b02320.

[37] Bernard R. Brooks, Robert E. Bruccoleri, Barry D. Olafson, David J. States,

S. Swaminathan, and Martin Karplus. CHARMM: a program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.*, 4(2): 187–217, 1983. doi: 10.1002/jcc.540040211.

[38] 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.*, 4:435–447, .

[39] James C. Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizabeth Villa, Christophe Chipot, Robert D. Skeel, Laxmikant Kalé, and Klaus Schulten. Scalable molecular dynamics with NAMD. *J. Comput. Chem.*, 26(16):1781–1802, 2005. doi: 10.1002/jcc.20289.

[40] Andreas W. Götz, Mark J. Williamson, Dong Xu, Duncan Poole, Scott Le Grand, and Ross C. Walker. Routine microsecond molecular dynamics simulations with AMBER on GPUs. 1. generalized born. *J. Chem. Theory Comput.*, 8(5):1542–1555, 2012. doi: 10.1021/ct200909j.

[41] Peter Eastman, Mark S. Friedrichs, John D. Chodera, Randall J. Radmer, Christopher M. Bruns, Joy P. Ku, Kyle A. Beauchamp, Thomas J. Lane, Lee-Ping Wang, Diwakar Shukla, Tony Tye, Mike Houston, Timo Stich, Christoph Klein, Michael R. Shirts, and Vijay S. Pande. OpenMM 4: A reusable, extensible, hardware independent library for high performance molecular simulation. *J. Chem. Theory Comput.*, 9(1):461–469, 2013. doi: 10.1021/ct300857j.

[42] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C. Smith, Berk Hess, and Erik Lindahl. GROMACS: high performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1-2:19–25, 2015. doi: 10.1016/j.softx.2015.06.001.

[43] S. Doerr, M. J. Harvey, Frank Noé, and G. De Fabritiis. HTMD: high-throughput molecular dynamics for molecular discovery. *J. Chem. Theory Comput.*, 12(4):1845–1852, 2016. doi: 10.1021/acs.jctc.6b00049.

[44] T. Darden, D. York, and L. Pedersen. Particle mesh ewald: An nlog (N) method for ewald sums in large systems. *J. Chem. Phys.*, 98:10089.

[45] Jay W. Ponder and Frederic M. Richards. An efficient newton-like method for molecular mechanics energy minimization of large molecules. *J. Comput. Chem.*, 8(7):1016–1024, 1987. doi: 10.1002/jcc.540080710.

[46] Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.*, 117(1):1–19, 1995. doi: 10.1006/jcph.1995.1039.

[47] Mohammad M. Sultan and Vijay S. Pande. tICA-metadynamics: Accelerating metadynamics by using kinetically selected collective variables. *J. Chem. Theory Comput.*, 13(6):2440–2447, 2017. doi: 10.1021/acs.jctc.7b00182.

[48] M. Shirts. COMPUTING: screen savers of the world unite! *Science*, 290(5498): 1903–1904, 2000. doi: 10.1126/science.290.5498.1903.

[49] Guha Jayachandran, V. Vishal, and Vijay S. Pande. Using massively parallel simulation and markovian models to study protein folding: Examining the dynamics of the villin headpiece. *J. Chem. Phys.*, 124(16):164902, 2006. doi: 10.1063/1.2186317.

[50] *Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters*, 2006. IEEE. doi: 10.1109/SC.2006.54.

[51] I. Buch, M. J. Harvey, T. Giorgino, D. P. Anderson, and G. De Fabritiis. High-throughput all-atom molecular dynamics simulations using distributed computing. *J. Chem. Inf. Model.*, 50(3):397–403, 2010. doi: 10.1021/ci900455r.

[52] David E. Shaw, Jack C. Chao, Michael P. Eastwood, Joseph Gagliardo, J. P. Grossman, C. Richard Ho, Douglas J. Lerardi, István Kolossváry, John L. Klepeis, Timothy Layman, Christine McLeavey, Martin M. Deneroff, Mark A. Moraes, Rolf Mueller, Edward C. Priest, Yibing Shan, Jochen Spengler, Michael Theobald, Brian Towles, Stanley C. Wang, Ron O. Dror, Jeffrey S. Kuskin, Richard H. Larson, John K. Salmon, Cliff Young, Brannon Batson, and Kevin J.

Bowers. Anton, a special-purpose machine for molecular dynamics simulation. *Commun. ACM*, 51(7):91, 2008. doi: 10.1145/1364782.1364802.

[53] *Millisecond-scale molecular dynamics simulations on Anton*, 2009. ACM Press. doi: 10.1145/1654059.1654126.

[54] John E. Stone, James C. Phillips, Peter L. Freddolino, David J. Hardy, Leonardo G. Trabuco, and Klaus Schulten. Accelerating molecular modeling applications with graphics processors. *J. Comput. Chem.*, 28(16):2618–2640, 2007. doi: 10.1002/jcc.20829.

[55] Joshua A. Anderson, Chris D. Lorenz, and A. Travesset. General purpose molecular dynamics simulations fully implemented on graphics processing units. *J. Comput. Phys.*, 227(10):5342–5359, 2008. doi: 10.1016/j.jcp.2008.01.047.

[56] Peter Eastman and Vijay S. Pande. Efficient nonbonded interactions for molecular dynamics on a graphics processing unit. *J. Comput. Chem.*, pages NA–NA, 2009. doi: 10.1002/jcc.21413.

[57] Mark S. Friedrichs, Peter Eastman, Vishal Vaidyanathan, Mike Houston, Scott Legrand, Adam L. Beberg, Daniel L. Ensign, Christopher M. Bruns, and Vijay S. Pande. Accelerating molecular dynamic simulation on graphics processing units. *J. Comput. Chem.*, 30(6):864–872, 2009. doi: 10.1002/jcc.21209.

[58] John L Klepeis, Kresten Lindorff-Larsen, Ron O Dror, and David E Shaw. Long-timescale molecular dynamics simulations of protein structure and function. *Curr. Opin. Struct. Biol.*, 19(2):120–127, 2009. doi: 10.1016/j.sbi.2009.03.004.

[59] William C. Swope, Jed W. Pitera, and Frank Suits. Describing protein folding kinetics by molecular dynamics simulations. 1. theory. *J. Phys. Chem. B*, 108 (21):6571–6581, 2004. doi: 10.1021/jp037421y.

[60] Vijay S. Pande, Kyle Beauchamp, and Gregory R. Bowman. Everything you wanted to know about markov state models but were afraid to ask. *Methods*, 52(1):99–105, 2010. doi: 10.1016/j.ymeth.2010.06.002.

[61] John D Chodera and Frank Noé. Markov state models of biomolecular conformational dynamics. *Curr. Opin. Struct. Biol.*, 25:135–144, 2014. doi: 10.1016/j.sbi.2014.04.002.

[62] Diwakar Shukla, Carlos X. Hernández, Jeffrey K. Weber, and Vijay S. Pande. Markov state models provide insights into dynamic modulation of protein function. *Acc. Chem. Res.*, 48(2):414–422, 2015. doi: 10.1021/ar5002999.

[63] Ch. Schütte, W. Huisinga, and P. Deuflhard. Transfer operator approach to conformational dynamics in biomolecular systems. pages 191–223. Springer Science + Business Media, 2001. doi: 10.1007/978-3-642-56589-2_9.

[64] Jan-Hendrik Prinz, Hao Wu, Marco Sarich, Bettina Keller, Martin Senne, Martin Held, John D. Chodera, Christof Schutte, and Frank Noe. Markov models of molecular kinetics: Generation and validation. *J. Chem. Phys.*, 134(17):174105, 2011. doi: 10.1063/1.3565032.

[65] Frank Noé and Feliks Nüske. A variational approach to modeling slow processes in stochastic dynamical systems. *Multiscale Model. Simul.*, 11(2):635–655, 2013. doi: 10.1137/110858616.

[66] Feliks Nüske, Bettina G. Keller, Guillermo Pérez-Hernández, Antonia S. J. S. Mey, and Frank Noé. Variational approach to molecular kinetics. *J. Chem. Theory Comput.*, 10(4):1739–1752, 2014. doi: 10.1021/ct4009156.

[67] Boaz Nadler, Stéphane Lafon, Ronald R. Coifman, and Ioannis G. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Appl. Comput. Harmon. Anal.*, 21(1):113–127, 2006. doi: 10.1016/j.acha.2005.07.004.

[68] Yusuke Naritomi and Sotaro Fuchigami. Slow dynamics in protein fluctuations revealed by time-structure based independent component analysis: The case of domain motions. *J. Chem. Phys.*, 134(6):065101, 2011. doi: 10.1063/1.3554380.

[69] Mary A. Rohrdanz, Wenwei Zheng, Mauro Maggioni, and Cecilia Clementi. Determination of reaction coordinates via locally scaled diffusion map. *J. Chem. Phys.*, 134(12):124116, 2011. doi: 10.1063/1.3569857.

[70] Guillermo Perez-Hernandez, Fabian Paul, Toni Giorgino, Gianni De Fabritiis, and Frank Noe. Identification of slow molecular order parameters for markov model construction. *J. Chem. Phys.*, 139(1):015102, 2013. doi: 10.1063/1.4811489.

[71] Christian R. Schwantes and Vijay 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. doi: 10.1021/ct300878a.

[72] Robert T. McGibbon and Vijay S. Pande. Learning kinetic distance metrics for markov state models of protein conformational dynamics. *J. Chem. Theory Comput.*, 9(7):2900–2906, 2013. doi: 10.1021/ct400132h.

[73] Christian R. Schwantes and Vijay S. Pande. Modeling molecular kinetics with tICA and the kernel trick. *J. Chem. Theory Comput.*, 11(2):600–608, 2015. doi: 10.1021/ct5007357.

[74] Robert T. McGibbon, Brooke E. Husic, and Vijay S. Pande. Identification of simple reaction coordinates from complex dynamics. *J. Chem. Phys.*, 146(4):044109, 2017. doi: 10.1063/1.4974306. arXiv:1602.08776.

[75] Frank Noé, Ralf Banisch, and Cecilia Clementi. Commute maps: Separating slowly mixing molecular configurations for kinetic modeling. *J. Chem. Theory Comput.*, 12(11):5620–5630, 2016. doi: 10.1021/acs.jctc.6b00762.

[76] Marco Sarich, Frank Noé, and Christof Schütte. On the approximation quality of markov state models. *Multiscale Model. Simul.*, 8(4):1154–1177, 2010. doi: 10.1137/090764049.

[77] Natasa Djurdjevac, Marco Sarich, and Christof Schütte. Estimating the eigenvalue error of markov state models. *Multiscale Model. Simul.*, 10(1):61–81, 2012. doi: 10.1137/100798910.

[78] Robert T. McGibbon, Christian R. Schwantes, and Vijay S. Pande. Statistical model selection for markov models of biomolecular dynamics. *J. Phys. Chem. B*, 118(24):6475–6481, 2014. doi: 10.1021/jp411822r.

[79] Robert T. McGibbon and Vijay S. Pande. Variational cross-validation of slow dynamical modes in molecular kinetics. *J. Chem. Phys.*, 142(12):124105, 2015. doi: 10.1063/1.4916292.

[80] Brooke E. Husic, Robert T. McGibbon, Mohammad M. Sultan, and Vijay S. Pande. Optimized parameter selection reveals trends in markov state models for protein folding. *J. Chem. Phys.*, 145(19):194103, 2016. doi: 10.1063/1.4967809.

[81] Ting Zhou and Amedeo Caflisch. Distribution of reciprocal of interatomic distances: A fast structural metric. *J. Chem. Theory Comput.*, 8(8):2930–2937, 2012. doi: 10.1021/ct3003145.

[82] Matthew P. Harrigan, Diwakar Shukla, and Vijay S. Pande. Conserve water: A method for the analysis of solvent in molecular dynamics. *J. Chem. Theory Comput.*, 11(3):1094–1101, 2015. doi: 10.1021/ct5010017.

[83] F. Vitalini, F. Noé, and B. G. Keller. A basis set for peptides for the variational approach to conformational kinetics. *J. Chem. Theory Comput.*, 11(9):3992–4004, 2015. doi: 10.1021/acs.jctc.5b00498.

[84] Frank Noé, Hao Wu, Jan-Hendrik Prinz, and Nuria Plattner. Projected and hidden markov models for calculating kinetics and metastable states of complex molecules. *J. Chem. Phys.*, 139(18):184114, 2013. doi: 10.1063/1.4828816.

[85] Hao Wu and Frank Noé. Gaussian markov transition models of molecular kinetics. *J. Chem. Phys.*, 142(8):084104, 2015. doi: 10.1063/1.4913214.

[86] Jan-Hendrik Prinz, John D. Chodera, and Frank Noé. Spectral rate theory for two-state kinetics. *Physical Review X*, 4(1), 2014. doi: 10.1103/PhysRevX.4.011020.

[87] Robert T. McGibbon and Vijay S. Pande. Efficient maximum likelihood parameterization of continuous-time markov processes. *J. Chem. Phys.*, 143(3): 034109, 2015. doi: 10.1063/1.4926516.

[88] C Levinthal. *J. Chim. Phys. Physico-Chim. Biol.*, 65:44–45, 1968.

[89] R. Zwanzig, A. Szabo, and B. Bagchi. Levinthal's paradox. *Proc. Natl. Acad. Sci. U.S.A.*, 89(1):20–22.

[90] Robert McGibbon. Notes on the theory of markov chains in a continuous state space, 2016. https://rmcgibbo.org/posts/notes-on-the-theory-of-markov-chains-in-a-continuous-state-space/.

[91] H Frauenfelder, S. Sligar, and P. Wolynes. The energy landscapes and motions of proteins. *Science*, 254(5038):1598–1603, 1991. doi: 10.1126/science.1749933.

[92] Susan S Taylor and Alexandr P Kornev. Protein kinases: evolution of dynamic regulatory proteins. *Trends Biochem. Sci.*, 36:65–77.

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

[94] G. R. Bowman, V. S. Pande, and F. Noé. *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*, volume 797. Springer.

[95] Thomas J. Lane, Gregory R. Bowman, Kyle Beauchamp, Vincent A. Voelz, and Vijay S. Pande. Markov state model reveals folding and functional dynamics in ultra-long MD trajectories. *JACS*, 133(45):18413–18419, 2011. doi: 10.1021/ja207470h.

[96] Vincent A. Voelz, Marcus Jäger, Shuhuai Yao, Yujie Chen, Li Zhu, Steven A. Waldauer, Gregory R. Bowman, Mark Friedrichs, Olgica Bakajin, Lisa J. Lapidus, Shimon Weiss, and Vijay S. Pande. Slow unfolded-state structuring in acyl-CoA binding protein folding revealed by simulation and experiment. *JACS*, 134(30):12565–12577, 2012. doi: 10.1021/ja302528z.

[97] Carlos R. Baiz, Yu-Shan Lin, Chunte Sam Peng, Kyle A. Beauchamp, Vincent A. Voelz, Vijay S. Pande, and Andrei Tokmakoff. A molecular interpretation of 2D IR protein folding experiments with markov state models. *Biophys. J.*, 106(6):1359–1370, 2014. doi: 10.1016/j.bpj.2014.02.008.

[98] Diwakar Shukla, Yilin Meng, Benoît Roux, and Vijay S. Pande. Activation pathway of src kinase reveals intermediate states as targets for drug design. *Nat. Commun.*, 5, 2014. doi: 10.1038/ncomms4397.

[99] S. K. Sadiq, F. Noe, 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. doi: 10.1073/pnas.1210983109.

[100] Matthew R. Perkett and Michael F. Hagan. Using markov state models to study self-assembly. *J. Chem. Phys.*, 140(21):214101, 2014. doi: 10.1063/1.4878494.

[101] David Chandler. Hydrophobicity: Two faces of water. *Nature*, 417(6888):491–491, 2002. doi: 10.1038/417491a.

[102] Eric J. Sorin and Vijay S. Pande. Nanotube confinement denatures protein helices. *JACS*, 128(19):6316–6317, 2006. doi: 10.1021/ja060917j.

[103] Élise Faure, Christine Thompson, and Rikard Blunck. Do lipids show state-dependent affinity to the voltage-gated potassium channel KvAP? *J. Biol. Chem.*, 289(23):16452–16461, 2014. doi: 10.1074/jbc.M113.537134.

[104] T. Sun, F.-H. Lin, R. L. Campbell, J. S. Allingham, and P. L. Davies. An antifreeze protein folds with an interior network of more than 400 semi-clathrate waters. *Science*, 343(6172):795–798, 2014. doi: 10.1126/science.1247407.

[105] Mary Griffin Krone, Lan Hua, Patricia Soto, Ruhong Zhou, B. J. Berne, and Joan-Emma Shea. Role of water in mediating the assembly of alzheimer amyloid- a1622 protofilaments. *JACS*, 130(33):11066–11072, 2008. doi: 10.1021/ja8017303.

[106] Guanghong Wei and Joan-Emma Shea. Effects of solvent on the structure of the alzheimer amyloid-(25–35) peptide. *Biophys. J.*, 91(5):1638–1647, 2006. doi: 10.1529/biophysj.105.079186.

[107] Francesco Rao, Sean Garrett-Roe, and Peter Hamm. Structural inhomogeneity of water by complex network analysis. *J. Phys. Chem. B*, 114(47):15598–15604, 2010. doi: 10.1021/jp1060792.

[108] R. Zhou. Hydrophobic collapse in multidomain protein folding. *Science*, 305 (5690):1605–1609, 2004. doi: 10.1126/science.1101176.

[109] Toshiya Senda, Kazuyuki Sugiyama, Hiroki Narita, Takeshi Yamamoto, Kazuhide Kimbara, Masao Fukuda, Mitsuo Sato, Keiji Yano, and Yukio Mitsui. Three-dimensional structures of free form and two substrate complexes of an extradiol ring-cleavage type dioxygenase, the BphC enzyme fromPseudomonassp. strain KKS102. *J. Mol. Biol.*, 255(5):735–752, 1996. doi: 10.1006/jmbi.1996.0060.

[110] Chen Gu, Huang-Wei Chang, Lutz Maibaum, Vijay S Pande, Gunnar E Carlsson, and Leonidas J Guibas. Building markov state models with solvent dynamics. *BMC Bioinf.*, 14(Suppl 2):S8, 2013. doi: 10.1186/1471-2105-14-S2-S8.

[111] *Web-scale k-means clustering*, 2010. Association for Computing Machinery (ACM). doi: 10.1145/1772690.1772862.

[112] William Humphrey, Andrew Dalke, and Klaus Schulten. VMD: visual molecular dynamics. *J. Mol. Graph.*, 14(1):33–38, 1996. doi: 10.1016/0263-7855(96)00018-5.

[113] Sergei Izrailev, Sergey Stepaniants, Barry Isralewitz, Dorina Kosztin, Hui Lu, Ferenc Molnar, Willy Wriggers, and Klaus Schulten. Steered molecular dynamics. pages 39–65.

[114] B. Hess, H. Bekker, H.J.C. Berendsen, and J.G.E.M. Fraaije. LINCS: a linear constraint solver for molecular simulations. *J. Comp. Chem.*, 18:1463–1472, .

[115] Matthew P. Harrigan, Keri A. McKiernan, Veerabahu Shanmugasundaram, Rajiah Aldrin Denny, and Vijay S. Pande. Markov modeling reveals novel intracellular modulation of the human TREK-2 selectivity filter. *Sci. Rep.*, 7(1), 2017. doi: 10.1038/s41598-017-00256-y.

[116] P. Enyedi and G. Czirjak. Molecular background of leak k+ currents: Two-pore domain potassium channels. *Physiol. Rev.*, 90(2):559–605, 2010. doi: 10.1152/physrev.00029.2009.

[117] Florian Lesage, Cécile Terrenoire, Georges Romey, and Michel Lazdunski. Human TREK2, a 2P domain mechano-sensitive K+Channel with multiple regulations by polyunsaturated fatty acids, lysophospholipids, and gs, gi, and GqProtein-coupled receptors. *J. Biol. Chem.*, 275(37):28398–28405, 2000. doi: 10.1074/jbc.m002822200.

[118] Jacques Noël, Guillaume Sandoz, and Florian Lesage. Molecular regulations governing TREK and TRAAK channel functions. *Channels*, 5(5):402–409, 2014. doi: 10.4161/chan.5.5.16469.

[119] Conor McClenaghan, Marcus Schewe, Prafulla Aryal, Elisabeth P. Carpenter, Thomas Baukrowitz, and Stephen J. Tucker. Polymodal activation of the TREK-2 K2P channel produces structurally distinct open states. *J. Gen. Physiol.*, 147(6):497–505, 2016. doi: 10.1085/jgp.201611601.

[120] D. Thomas and S.A.N. Goldstein. Two-p-domain (K2P) potassium channels: Leak conductance regulators of excitability. pages 1207–1220. Elsevier, 2009. doi: 10.1016/b978-008045046-9.01636-3.

[121] Paula L Piechotta, Markus Rapedius, Phillip J Stansfeld, Murali K Bollepalli, Gunter Erhlich, Isabelle Andres-Enguix, Hariolf Fritzenschaft, Niels Decher, Mark S P Sansom, Stephen J Tucker, and Thomas Baukrowitz. The pore structure and gating mechanism of K2P channels. *EMBO J.*, 30(17):3607–3619, 2011. doi: 10.1038/emboj.2011.268.

[122] Markus Rapedius, Matthias R. Schmidt, Chetan Sharma, Phillip J. Stansfeld, Mark S.P. Sansom, Thomas Baukrowitz, and Stephen J. Tucker. State-independent intracellular access of quaternary ammonium blockers to the pore of TREK-1. *Channels*, 6(6):473–478, 2014. doi: 10.4161/chan.22153.

[123] Y. Y. Dong, A. C. W. Pike, A. Mackenzie, C. McClenaghan, P. Aryal, L. Dong, A. Quigley, M. Grieben, S. Goubin, S. Mukhopadhyay, G. F. Ruda, M. V. Clausen, L. Cao, P. E. Brennan, N. A. Burgess-Brown, M. S. P. Sansom, S. J. Tucker, and E. P. Carpenter. K2P channel gating mechanisms revealed by structures of TREK-2 and a complex with prozac. *Science*, 347(6227):1256–1259, 2015. doi: 10.1126/science.1261512.

[124] Sanghyun Park and Klaus Schulten. Calculating potentials of mean force from steered molecular dynamics simulations. *J. Chem. Phys.*, 120(13):5946–5961, 2004. doi: 10.1063/1.1651473.

[125] Toshinori Hoshi, William N. Zagotta, and Richard W. Aldrich. Two types of inactivation in shaker k+ channels: Effects of alterations in the carboxy-terminal region. *Neuron*, 7(4):547–556, 1991. doi: 10.1016/0896-6273(91)90367-9.

[126] D. A. Kopfer, C. Song, T. Gruene, G. M. Sheldrick, U. Zachariae, and B. L. de Groot. Ion permeation in k+ channels occurs by direct coulomb knock-on. *Science*, 346(6207):352–355, 2014. doi: 10.1126/science.1254840.

[127] G. Hummer. Potassium ions line up. *Science*, 346(6207):303–303, 2014. doi: 10.1126/science.1260555.

[128] A. L. Hodgkin and R. D. Keynes. The potassium permeability of a giant nerve fibre. *J. Physiol. (Lond.)*, 128(1):61–88, 1955. doi: 10.1113/jphysiol.1955.sp005291.

[129] Marcus Schewe, Ehsan Nematian-Ardestani, Han Sun, Marianne Musinszki, Sönke Cordeiro, Giovanna Bucci, Bert L. de Groot, Stephen J. Tucker, Markus

Rapedius, and Thomas Baukrowitz. A non-canonical voltage-sensing mechanism controls gating in K2P k+ channels. *Cell*, 164(5):937–949, 2016. doi: 10.1016/j.cell.2016.02.002.

[130] Ren-Gong Zhuo, Peng Peng, Xiao-Yan Liu, Hai-Tao Yan, Jiang-Ping Xu, Jian-Quan Zheng, Xiao-Li Wei, and Xiao-Yun Ma. Allosteric coupling between proximal c-terminus and selectivity filter is facilitated by the movement of transmembrane segment 4 in TREK-2 channel. *Sci. Rep.*, 6(1), 2016. doi: 10.1038/srep21248.

[131] Jumin Lee, Xi Cheng, Jason M. Swails, Min Sun Yeom, Peter K. Eastman, Justin A. Lemkul, Shuai Wei, Joshua Buckner, Jong Cheol Jeong, Yifei Qi, Sunhwan Jo, Vijay S. Pande, David A. Case, Charles L. Brooks, Alexander D. MacKerell, Jeffery B. Klauda, and Wonpil Im. CHARMM-GUI input generator for NAMD, GROMACS, AMBER, OpenMM, and CHARMM/OpenMM simulations using the CHARMM36 additive force field. *J. Chem. Theory Comput.*, 12(1):405–413, 2016. doi: 10.1021/acs.jctc.5b00935.

[132] DA Case, V Babin, Josh Berryman, RM Betz, Q Cai, DS Cerutti, TE Cheatham Iii, TA Darden, RE Duke, and H Gohlke. Amber 14.

[133] Shankar Kumar, John M. Rosenberg, Djamal Bouzida, Robert H. Swendsen, and Peter A. Kollman. THE weighted histogram analysis method for free-energy calculations on biomolecules. i. the method. *J. Comput. Chem.*, 13(8): 1011–1021, 1992. doi: 10.1002/jcc.540130812.

[134] Robert T. McGibbon, Kyle A. Beauchamp, Matthew P. Harrigan, Christoph Klein, Jason M. Swails, Carlos X. Hernández, Christian R. Schwantes, Lee-Ping Wang, Thomas J. Lane, and Vijay S. Pande. MDTraj: a modern open library for the analysis of molecular dynamics trajectories. *Biophys. J.*, 109(8): 1528–1532, 2015. doi: 10.1016/j.bpj.2015.08.015.

[135] Matthew P. Harrigan, Mohammad M. Sultan, Carlos X. Hernández, Brooke E. Husic, Peter Eastman, Christian R. Schwantes, Kyle A. Beauchamp, Robert T.

McGibbon, and Vijay S. Pande. MSMBuilder: statistical models for biomolecular dynamics. *Biophys. J.*, 112(1):10–15, 2017. doi: 10.1016/j.bpj.2016.10.042.

[136] Eric F Pettersen, Thomas D Goddard, Conrad C Huang, Gregory S Couch, Daniel M Greenblatt, Elaine C Meng, and Thomas E Ferrin. UCSF chimera-a visualization system for exploratory research and analysis. *J. Comput. Chem.*, 25:1605–1612.

[137] Gregory R. Bowman, Xuhui Huang, and Vijay S. Pande. Using generalized ensemble simulations and markov state models to identify conformational states. *Methods*, 49(2):197–201, 2009. doi: 10.1016/j.ymeth.2009.04.013.

[138] Kyle A. Beauchamp, Gregory R. Bowman, Thomas J. Lane, Lutz Maibaum, Imran S. Haque, and Vijay S. Pande. MSMBuilder2: modeling conformational dynamics on the picosecond to millisecond scale. *J. Chem. Theory Comput.*, 7 (10):3412–3419, 2011. doi: 10.1021/ct200463m.

[139] Martin Senne, Benjamin Trendelkamp-Schroer, Antonia S.J.S. Mey, Christof Schütte, and Frank Noé. EMMA: a software package for markov model building and analysis. *J. Chem. Theory Comput.*, 8(7):2223–2238, 2012. doi: 10.1021/ct300274u.

[140] Martin K. Scherer, Benjamin Trendelkamp-Schroer, Fabian Paul, Guillermo Pérez-Hernández, Moritz Hoffmann, Nuria Plattner, Christoph Wehmeyer, Jan-Hendrik Prinz, and Frank Noé. PyEMMA 2: A software package for estimation, validation, and analysis of markov models. *J. Chem. Theory Comput.*, 11(11): 5525–5542, 2015. doi: 10.1021/acs.jctc.5b00743.

[141] Susan S. Taylor and Alexandr P. Kornev. Protein kinases: evolution of dynamic regulatory proteins. *Trends Biochem. Sci.*, 36(2):65–77, 2011. doi: 10.1016/j.tibs.2010.09.006.

[142] Robert McGibbon, Bharath Ramsundar, Mohammad Sultan, Gert Kiss, and Vijay Pande. Understanding protein dynamics with l1-regularized reversible hidden markov models. 32(2):1197–1205.

[143] Fernando Perez and Brian E. Granger. IPython: a system for interactive scientific computing. *Comput. Sci. Eng.*, 9(3):21–29, 2007. doi: 10.1109/MCSE.2007.53.

[144] Peter Deuflhard and Marcus Weber. Robust perron cluster analysis in conformation dynamics. *Linear Algebra Appl.*, 398:161–184, 2005. doi: 10.1016/j.laa.2004.10.026.

[145] Philipp Metzner, Christof Schütte, and Eric Vanden-Eijnden. Transition path theory for markov jump processes. *Multiscale Model. Simul.*, 7(3):1192–1219, 2009. doi: 10.1137/070699500.

[146] Alexander Berezhkovskii, Gerhard Hummer, and Attila Szabo. Reactive flux and folding pathways in network models of coarse-grained protein dynamics. *J. Chem. Phys.*, 130(20):205102, 2009. doi: 10.1063/1.3139063.

[147] F. Noe, C. Schutte, 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. doi: 10.1073/pnas.0905466106.

[148] Robert McGibbon. Fs MD trajectories, 2014.

[149] Robert T. McGibbon, Carlos X. Hernández, Matthew P. Harrigan, Steven Kearnes, Mohammad M. Sultan, Stanislaw Jastrzebski, Brooke E. Husic, and Vijay S. Pande. Osprey: Hyperparameter optimization for machine learning. *The Journal of Open Source Software*, 1(5), 2016. doi: 10.21105/joss.00034.

[150] Matthew P Harrigan and Vijay S Pande. Landmark kernel tICA for conformational dynamics. bioRxiv:10.1101/123752.

[151] Christopher K. I. Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, volume 13, pages 682–688, 2001.

[152] Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. In *Advances in neural information processing systems*, pages 476–484, 2012.

[153] Frank Noe and Cecilia Clementi. Kinetic distance and kinetic maps from molecular dynamics simulation. arXiv:1506.06259, 2015.

[154] Marcus Weber. *Meshless Methods in Conformational Dynamics*. PhD thesis, 2006.

[155] Konstantin Fackeldey, Susanna Röblitz, Olga Scharkoi, and Marcus Weber. Soft versus hard metastable conformations in molecular simulations. 2011.

[156] Konstantin Fackeldey, Alexander Bujotzek, and Marcus Weber. A meshless discretization method for markov state models applied to explicit water peptide folding simulations. pages 141–154. Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-32979-1_9.

[157] Ch. Schütte and M. Sarich. A critical appraisal of markov state models. *The European Physical Journal Special Topics*, 224(12):2445–2462, 2015. doi: 10.1140/epjst/e2015-02421-0.

[158] Marcus Weber, Konstantin Fackeldey, and Christof Schütte. Set-free markov state model building. *J. Chem. Phys.*, 146(12):124133, 2017. doi: 10.1063/1.4978501.

[159] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: a system for large-scale machine learning. arXiv:1605.08695, 2016.

[160] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallogr. A*, 32(5):922–923, 1976. doi: 10.1107/S0567739476001873.

[161] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Amer. A*, 4(4):629, 1987. doi: 10.1364/JOSAA.4.000629.

[162] Douglas L. Theobald. Rapid calculation of RMSDs using a quaternion-based characteristic polynomial. *Acta Crystallogr., Sect. A: Found. Crystallogr.*, 61 (4):478–480, 2005. doi: 10.1107/S0108767305015266.

[163] Pu Liu, Dimitris K. Agrafiotis, and Douglas L. Theobald. Fast determination of the optimal rotational matrix for macromolecular superpositions. *J. Comput. Chem.*, pages n/a–n/a, 2009. doi: 10.1002/jcc.21439.

[164] Evangelos A. Coutsias, Chaok Seok, and Ken A. Dill. Using quaternions to calculate RMSD. *J. Comput. Chem.*, 25(15):1849–1857, 2004. doi: 10.1002/jcc.20110.