Summer 8-15-2019

FAST-Forward Protein Folding and Design: Development, Analysis, and Applications of the FAST Sampling Algorithm

Maxwell Isaac Zimmerman
Washington University in St. Louis

WASHINGTON UNIVERSITY IN ST. LOUIS

Division of Biology and Biomedical Sciences
Computational and Molecular Biophysics

Dissertation Examination Committee:
Gregory Bowman, Chair
Alexander Barnes
Timothy Lohman
Rohit Pappu
Jay Ponder

FAST-Forward Protein Folding and Design: Development, Analysis, and Applications of the FAST Sampling Algorithm

by
Maxwell Zimmerman

A dissertation presented to
The Graduate School
of Washington University in
partial fulfillment of the
requirements for the degree
of Doctor of Philosophy

August 2019
St. Louis, Missouri

Table of Contents

List of Figures

List of Tables

Acknowledgments

Consider a spherical cow. Now consider a graduate student; one who prepared all of the work for their thesis in perfect isolation, ideas and all, without any help or input from others. What do both of these models have in common? They do not in any way, shape, or form, describe reality. As such, I dedicate this section to the many people who have helped me on my journey to collecting data and writing this thesis.

First and foremost, I would like to acknowledge my advisor, Gregory Bowman. Not only is he incredibly gifted, he has an awe-inspiring work ethic that stems from his insatiable drive to progress the frontiers of science. That is not to say that time in his lab was solely about generating data. On the contrary, Greg, has worked incredibly hard to create a stimulating, yet nurturing environment. Greg has spent many hours helping me succeed not only in science, but in life. Whether it be coaching me on talks, helping me navigate academic politics, or giving me advice (and many items) about being a new parent. To say Greg is supportive would be an understatement. It has been quite the experience to see the lab evolve and grow from the ground up.

Within the lab, there have been numerous people that made my experience superb. The lab mates have been my partners in science and helped generate so many thought-provoking discussions that helped to fuel and refine the research in this thesis. While this short paragraph cannot do everyone justice, I will attempt to acknowledge them to the best of my ability. Perhaps more than anyone at Washington University, I have spent time talking with Justin Porter, whom, for better or worse, I know will never shy away from a discussion. Since Justin joined the lab, I've had to up my computational skills to compete, though he's always been willing to share his

knowledge. He's also been a key sounding board for nearly all of my research and ideas, while also contributing to the work in chapter 4. Katie Hart and Carrie Sibbald, while never failing to have fresh opinions on science or politics, contributed immensely to the research in chapter 5, designing and collecting a majority of the experiments. Sukrit Singh and I joined the program and Greg's lab at the same time, for which I feel a brotherly bond. Tom Frederick, in addition to being our resident NMR spectroscopist and generating the spectra in chapter 5, has always helped to liven the lab mood with song and dance. Even when first rotating in the lab, Micky Ward has been a good friend, and generated insightful discussions about FAST and sampling in general. He has also helped to shape my future interests in deep learning, despite the possibility of a hostile AI takeover. Neha Vithani and Upasana Mallimadugula have been a constant source of joy, from encouraging the lab to celebrate Holi, teaching me to cook curries, or discussing the many intricacies of music. Others in the lab, Catie Knoverek, Matthew Cruz, and Katie Moeder, have been pivotal at shaping the lab dynamic and helping to encourage an open environment.

Outside of the Bowman lab, my friends within the department have been instrumental in helping me keep my sanity. Robb Welty, Drake Jensen, and Nicole Fazio have always been there to make me laugh. From dinner parties to musical concerts, our time was always well spent.

On a research note, John Jimah and Niraj Tolia contributed to the research in chapter 5 by solving the crystal structure of TEM-1 β-lactamase M182N structure. I should also mention that Carrie Sibbald spent tireless years to get the protein to eventually crystalize. Additionally, Pooch and Pumpkin, my cats, graciously donated their whiskers for streak-seeding the crystal structures.

Research is never completed without funding, for which I am immensely grateful for the following fellowships and awards: the center for biological and systems engineering scholarship,

the Bayer (formerly Monsanto) graduate research fellowship, and the Needleman prize in pharmacology.

I would also like to acknowledge my thesis committee members, Alexander Barnes, Timothy Lohman, Rohit Pappu, and Jay Ponder, for their advice and mentorship.

Lastly, I would like to thank my family: my partner Sarah Leonard and our son Avi. They have given me a life worth living. Sarah has supported me through my roughest moments and celebrated with me during my best. Words cannot express how appreciative I am, not only for her emotional support, but for being okay with me using her as a sounding board for every idea I've ever had and proofreading nearly everything I've written.

Maxwell Zimmerman

Washington University in St. Louis

August 2019

Dedicated to my brother.

ABSTRACT OF THE DISSERTATION

FAST-Forward Protein Folding and Design: Development, Analysis, and Applications of the FAST sampling algorithm

by

Maxwell Zimmerman

Doctor of Philosophy in Biology and Biomedical Sciences

Computational and Molecular Biophysics

Washington University in St. Louis, 2019

Professor Gregory R. Bowman, Chair

Molecular dynamics simulations are a powerful tool to explore conformational landscapes, though limitations in computational hardware commonly thwart observation of biologically relevant events. Since highly specialized or massively parallelized distributed supercomputers are not available to most scientists, there is a strong need for methods that can access long timescale phenomena using commodity hardware. In this thesis, I present the goal-oriented sampling method, Fluctuation Amplification of Specific Traits (FAST), that takes advantage of Markov state models (MSMs) to adaptively explore conformational space using equilibrium-based simulations. This method follows gradients in conformational space to quickly explore relevant conformational transitions with orders of magnitude less aggregate simulation time than traditional simulations. Since each of the individual simulations are at equilibrium, all of the thermodynamics and kinetics in the final MSM are preserved. Here, I first describe the FAST method then demonstrate that it can be used for a variety of tasks, from folding proteins to

finding cryptic pockets. Next, I validate that FAST discovers appropriate transition pathways between states. Lastly, I apply FAST in detailing the mechanism of stabilization for a clinically relevant mutation in TEM-1 β-lactamase. This mechanistic understanding is then used to design other stabilizing mutations, which are all supported experimentally.

Chapter 1

Introduction

1.1 Molecular Dynamics as a Tool for Accessing a Proteins' Conformational Ensemble

The development of structural biology in the middle of the 20th century transformed our understanding of biomolecules.[1] No longer were we blind to the underpinnings of cellular processes. Rather, all-atom conformational models have allowed us to *see* how amino acid sequences dictate the structural features important for function. Insights from structural models have increased our understanding of countless biological systems and influenced the way we perceive biological phenomenon, such as enzyme catalysis, protein folding, or cell signaling, to name a few. The ever-growing number of structures deposited into the protein data bank (PDB) each year highlights the importance we place on structural models. Nevertheless, there are a growing number of examples where structural models are insufficient in elucidating relevant molecular mechanisms: phenotypically distinct sequences oftentimes give rise to nearly identical crystal structures. This begs the question: how do these sequences behave so differently when they look so similar? The simple explanation is that the single structure obtained is not the whole story.

A proteins' dynamics and conformational ensemble are increasingly shown to be important for understanding their biological function.[2-7] Proteins are not static structures floating

in solution. Instead, they are in constant motion and are best characterized by their conformational ensemble, which might change in response to particular stimuli. Instead of any single conformation, it is the ensemble that is most representative of a proteins' role in and out of cells. It is no surprise then, that single snapshots of a particular conformation may not provide a complete picture. A method that could readily access a proteins' conformational ensemble would then bring about another revolution in our understanding of biological mechanisms. While there are some experimental methods that can provide insight into a proteins' conformation ensemble, namely spectroscopic methods such as fluorescence resonance energy transfer (FRET)[8] and nuclear magnetic resonance (NMR) spectroscopy[9], the only methods that can provide all-atom time-series descriptions of protein motions are computational.

Molecular dynamics (MD) simulations are very promising as a tool to access the conformational ensemble of a protein.[10] MD simulations propagate the position of each atom in a system by numerically solving Newton's equations of motion, where snapshots of the atomic coordinates are saved at discrete time intervals. The validity of the computed dynamics relies on the ability to represent the atomic energies of a system. Since the true energetics of a protein system—and even quantum approximations—are difficult to calculate, the energy landscape is most commonly an empirical model where atoms are represented as balls attached by springs. By repeatedly integrating the empirical model to obtain new coordinates, we obtain a time-series trajectory of the protein exploring conformational space. With a sufficiently long trajectory, we could theoretically know the thermodynamic and kinetic properties of the protein. However, current computational hardware severely limits the ability to access trajectories of sufficient length.

The timestep for numerically integrating atomic motions is constrained by the fastest motion, hydrogen bond vibrations, to be around 1 fs. This means that obtaining a 1 s trajectory—not an unreasonable timescale for many real protein systems—requires solving the equations of motion and propagating each atom a quadrillion (10^{15}) times. Generating such a trajectory, even with an empirical approximation of energies, where each timestep occurs within a small fraction of a second, could take a desktop computer over a million years.[11] Arguably, the greatest challenge prohibiting the use of MD simulations is capturing long time-scale phenomena without sacrificing the accuracy of thermodynamic and kinetic properties.

Since the limitation of reaching long timescales can be thought of as a hardware issue, there has been a large effort to expand computer power by orders of magnitude. The first approach is to simply make a faster computer that is better able to get a sufficiently long simulation. The ANTON supercomputer is a notable example of a special purpose hardware for running MD simulations.[12,13] This supercomputer is a triumph in computer engineering and can generate trajectories of small proteins into the millisecond regime[14], although their cost and use are out of reach for most researchers and academic institutions.

Another approach towards increasing computational abilities is to crowd-source the computation. This is the approach of Folding@Home, which utilizes the hardware of around 100,000 personal computers that are donated when not in use.[15] In this framework, many individual simulations are spawned across the commodity hardware that is donated. Instead of obtaining a single long simulation, the many independent simulations constitute a large aggregate simulation dataset. The aggregate of these simulations has the ability to capture long-timescale phenomena despite each simulation being orders of magnitude shorter than the said timescale. This can be seen in the case of a two-state system with a large energy barrier: each

short simulation has a chance to jump the large barrier, which in total matches the chance of the single long simulation.[16] Folding@Home has also been able to generate aggregate simulation times into the millisecond regime.[17-19]

While ANTON and Foldin@Home can generate impressive simulation datasets, there is a new challenge: how do we make sense of the atomic coordinates to answer specific biological questions? It is a non-trivial task to convert the time-series trajectories into a human interpretable characterization of conformational space. In the case of having multiple trajectories, there is an added task of stitching together the parallel simulations in a way that corrects for the fact that conformations may not be Boltzmann distributed. A powerful framework to analyze trajectories is the use of Markov State Models (MSMs). These models treat simulation datasets as a Markov chain, which has the power of stitching together many parallel simulations in a statistically rigorous manner. This ability has some significant implications for using molecular dynamics to explore a proteins' conformational landscape, which is explored in more detail in the following sections.

Considering the abovementioned, it is a central objective of this thesis to combat the sampling problem and regularly access a proteins' conformational ensemble. Work towards this goal is attempted in the following chapters with the use of MSMs, which allow for a more sophisticated sampling and analysis methods. The purpose of these methods is to connect the properties of protein ensembles with experimentally measurable quantities, such as stability and enzymatic activity.

1.2 Markov State Models

1.2.1 Introduction to Markov Chains

Markov state models (MSMs) are a network representation of a free-energy landscape. In this framework, each node represents a conformational "microstate" that corresponds to a free-energy minimum, and each edge represents the conditional transition probability of hopping between them.[20-23] The goal in using an MSM is to provide a framework for stitching together many parallel simulations in a thermodynamically meaningful way. This is accomplished by reframing the simulations as a Markov chain, which only tracks the probability of hopping between states; added sampling in any state will only serve to refine transition probabilities and not naively overestimate equilibrium distributions.

Markov chains were developed in the early 20th century as an elegant way to model stochastic processes. Markov chains can be described in the following way[24]: there are a set of N discrete states, $S = \{s_1, s_2, \dots, s_N\}$. Each of these states has a probability of transitioning to any of the other states within some specified unit of time. This unit of time is considered a step-size, or lag-time, and is represented as τ. We denote the probability of transitioning, from state i to state j, as T_{ij}. Sampling such a process for k-steps, we would obtain a trajectory, $\boldsymbol{X} = \{X_1, X_2, \dots, X_k\}$. A central postulate is that these transition probabilities are only dependent on the knowledge of being in the current state; how the process landed on this state does not influence where it will go next. This postulate is particularly valid for systems with sufficiently complex dynamics, where after some amount of time, a system will not remember how it arrived at its current state.[25]

The transition probability matrix, $\boldsymbol{T}$, has the ability to propagate a probability vector in the following way,

$$\mathbf{\nu}^{(1)} = \mathbf{\nu T}$$

where $\mathbf{\nu}$ is the initial probability of being in any state, and $\mathbf{\nu}^{(1)}$ is the probability of being in any state after one time-step. If we are interested in the probability for a second step, we can propagate the probability vector a second time,

$$\mathbf{\nu}^{(2)} = \mathbf{\nu}^{(1)}\mathbf{T} = (\mathbf{\nu T})\mathbf{T} = \mathbf{\nu T}^2$$

In this way, we can solve for the probability of being in any state at an arbitrary number of steps, n,

$$\mathbf{\nu}^{(n)} = \mathbf{\nu T}^{\mathrm{n}}$$

For a memoryless process, as the number of steps gets larger, the probability of being in any state becomes less dependent on the initial conditions. For an ergodic Markov chain, a process where every state has the ability to reach every other state, the probability distribution will approach the equilibrium distribution,

$$\mathbf{\pi} = \mathbf{\nu}^{(\infty)} = \mathbf{\nu T}^{(\infty)}$$

If the system is truly ergodic, the equilibrium distribution will converge independent of the starting distribution. Further propagation of this distribution by the transition probability matrix will return the same distribution,

$$\boldsymbol{\pi T = \pi}$$

From this, we can quickly determine the equilibrium populations from an ergodic Markov chain by calculating the eigenvectors of the transition probability matrix that have an eigenvalue of one.

As previously mentioned, the ability to capture the equilibrium populations solely from the transition probabilities is incredibly powerful. This means that we only need proper estimates of the transitions between states, and not necessarily their global population to gather sufficient thermodynamics. With this simple Markov chain framework, the main challenge to building a Markov state model is in defining a state space from all-atom conformation, and then estimating a transition probability matrix from sparse connections.

1.2.2 Defining a State-Space

One of the most important aspects in building an MSM is defining the state space. The number, size, and connectivity of the discrete states will completely dictate the thermodynamic properties within an MSM. Hence, there has been careful thought into how discrete states are generated, for which there are a couple dominant approaches: geometric clustering and kinetic clustering.

Geometric clustering aims to cluster conformations based on their structural similarity. This is appealing for a couple of reasons. First, structurally distinct states are a natural choice for defining energy minima. Additionally, geometric clustering is an excellent way to assess the

conformational heterogeneity in a particular dataset. Having many small states is often beneficial for describing the underlying kinetic network. In order to perform this type of clustering, we first have to define a structural metric for assessing the similarity between structures. Any metric may be used, so long as it can be framed as a distance that obeys the triangle inequality. A common metric that is used is the root-mean-squared deviation (RMSD) between atomic coordinates.[20] Another type of metric could be the Euclidean distance between a featurized representation of the protein—i.e. characterizing the secondary structure using ϕ and ψ angles, creating an internal distance matrix, or, as has been done previously, by computing the solvent accessible surface area (SASA).[26]

An important consideration in defining the state space is to determine which atoms to include from conformational frames. If one is characterizing large conformational rearrangements, it is found to be beneficial to only use the backbone heavy atoms (N, C_α, C_β, CO, O), as the sidechain degrees of freedom can create an incredibly rugged landscape with poor statistics in the MSM downstream. Conversely, there may be certain regions of a protein where dynamics are most important, and an all-atom clustering of solely this section may be crucial (as is shown to be the case in Chapter 5).

After the structural metric has been defined, structures are grouped using an unsupervised clustering algorithm. Common algorithms are the centroid-based clustering that include *k*-centers, *k*-means, and *k*-medoids.[27,28] These algorithms are optimization algorithms for the NP-hard problem of partitioning a set of points in a hyper-dimensional space. It should be noted that these methods are non-deterministic and serve to find a local optimum. In brief, the main difference between the three *k*-series clustering methods mentioned above is that *k*-centers minimizes the distance between cluster centers, while *k*-means and *k*-medoids minimizes an l_n

norm (or some type of distance) from each point to their assigned cluster center, which is either a hypothetical point or an actual data point, respectively. An implementational difference between the methods is that *k*-means and *k*-medoids requires the number of cluster centers as an input parameter, whereas *k*-centers requires either the number of cluster centers *or* a maximum distance to each cluster center. It should be noted that the number of states is less relevant than having small enough cluster centers that do not contain internal energy barriers. In practice, a recommended strategy for clustering data, when using an RMSD metric, is to use *k*-centers with a maximum distance cutoff to generate the initial assignments and cluster centers, and then refine the cluster centers with the *k*-medoids algorithm for some number of sweeps.

The *k*-centers algorithm, in short, proceeds as follows: 1) choose an initial cluster center, either as a predetermined data point or as a randomly chosen point, and assign all data points to this cluster. 2) Calculate all distances to their assigned cluster center. 3) Choose the point with the largest distance to its assigned cluster center as a new cluster center. 4) Calculate the distance between all points and the new cluster center. 5) If the new distance is smaller than the distance to its currently assigned center, reassign the data point to the new cluster center. 6) Repeat steps 3-5 until the specified number of cluster centers is reached or the maximum distance to any cluster center falls below some threshold.

The *k*-medoids algorithm that is typically used is the partitioning around medoids (PAM) variant. This version uses a greedy search to find the medoid at each sweep. Given an initial set of assignments, PAM proceeds by iterating through each cluster and choosing a new center from one of the points currently assigned. All states are then reassigned to the closest cluster center. A cost is calculated, as the sum of distances from each point to their respective cluster center, and the new center is accepted if the cost is minimized with the new assignments, otherwise the new

center is rejected. *K*-means works similarly, however the new cluster center is chosen as the average point within each state assignment and does not necessarily need to be a real (or even physical) data point.

An alternative to geometric clustering is a kinetic-based clustering, which attempts to group conformations based on the speed they exchange. This is thought to be particularly beneficial for systems that have very dynamic regions—a disordered tail to a protein could quickly produce many conformations with large RMSDs, though be kinetically very similar. Conversely, very similar conformations may have a large energy barrier separating them. The most common kinetic-based cluster algorithm is the time independent component analysis (tICA).[29-31] The independent component analysis finds the basis vectors that best separates independent signals, in this case on the time-domain. For molecular dynamics data sets, each frame is featurized in some way, either using a distance matrix or backbone dihedrals, and projected onto the first few independent components. This projection will group things that are kinetically similar in each dimension. With this reduced dimensional space, one of the *k*-series clustering algorithms can then be easily used. While this approach has proven well for describing events such as protein folding, where the longest timescale is of most interest, it performs poorly when slow degrees of freedom are not functionally relevant. This has a particular disadvantage when describing the active site of a protein.[32]

1.2.3 Estimating a Transition Probability Matrix

Once the state-space has been described, we are left with the task of estimating a transition probability matrix. While this may seem straightforward, defects in sampling and/or clustering can make the task very challenging. Specifically, statistics in certain regions may be very poor, and in many cases the resulting MSM will not be ergodic; a non-ergodic MSM will have sources

and sinks that will impede an eigenvalue decomposition and given unreliable equilibrium populations. On the other hand, knowledge of what we expect at equilibrium can aid us in reconstructing an appropriate transition probability matrix from simulations out of a global equilibrium. In this section, I review some of the basic ways to estimate a transition probability matrix given simulation data.

The first step in estimating a transition probability matrix, given a set of trajectories that have been clustered into a discrete state space, is to count the number of transitions between states. To ensure the Markov assumption in the resulting Markov model, we count transitions between frames that are a specified number apart. The simulation time of this transition is the lag time of the resultant MSM. With complex dynamics, the system should become memoryless after some amount of time, however if the lag time is too short, the state of the system will be influenced from its past and the Markovian assumption will not be valid. [25] Conversely, if the lag time is too long, the MSM loses resolution. More practically, a long lag time reduces the amount of available data. Due to the sliding window, the number of transitions counted in a single trajectory is computed as,

$$n_{frames} - l + 1$$

where n_{frames} is the number of frames in a trajectory and l is the number of frames between states to count as a transition. For many short simulations, a large lag time could severely reduce the number of observed transitions. In practice, an MSM is generated for a variety of lag times and selected as the smallest lag time that the Markov assumption is valid. Validity of the Markov assumption is measured by plotting the slowest timescale as a function of the lag time; if the

system is Markovian, the slowest motion should not be affected by choice of lag time. Once a lag time is chosen, transitions are counted and summed into the transition count matrix, C_{ij}, which counts the number of transitions observed between states i and j.

The simplest way to estimate a transition probability matrix is to row-normalize the transition count matrix,

$$T_{ij}^{normalize} = \frac{C_{ij}}{\sum_k C_{ik}}$$

As mentioned above, this is very likely to generate a state space that is not ergodic—i.e. there may be large number of observed transitions from state i to j but none from state j to i. A common way to ensure ergodicity, without perturbing the data too significantly, is to add a prior. This prior can take the form of a pseudocount, $\tilde{C}$, which serves as our estimate of the system in the absence of data. With this pseudocount, we assume that each state has a single observed transition equally distributed between all other states,

$$\tilde{C} = \frac{1}{N}$$

where N is the number of states in the model. The resultant transition probability matrix then becomes,

$$T_{ij}^{normalize} = \frac{C_{ij} + \tilde{C}}{\sum_k (C_{ik} + \tilde{C})}$$

While adding a pseudocount to the count matrix will help to condition it, and make calculation of the eigenspectrum behave properly, we know that the counts are not what are expected when at equilibrium. For a reversible Markov process at equilibrium, we know that,

$$\pi_i P_{ij} = \pi_j P_{ji}$$

or put another way,

$$C_{ij} = C_{ji}$$

Another way to think of this: if we run an infinitely long simulation (perfectly equilibrated) and build an MSM, it should be equivalent to an MSM built from the same simulation run in reverse. Knowing this, we can enforce this reversibility by averaging count matrix with the transpose of itself,

$$C_{ij}^{transpose} = \frac{C_{ij} + C_{ji}}{2}$$

From this, we can calculate the transition probability matrix by row normalizing. Since the count matrix is fully reversible, we can trivially calculate the equilibrium probabilities as,

$$\pi_i = \frac{\sum_j C_{ij}^{transpose}}{\sum_{k,j} C_{k,j}^{transpose}}$$

This method provides a very convenient way to calculate the transition probability matrix and equilibrium populations, however, as can be seen in the above equation, the equilibrium population of state i is determined by its number of observations. This is obviously not a desirable property when building an MSM, since one of the largest benefits should be its ability to stitch together simulations when the conformations are not Boltzmann distributed.

An alternative way to enforce reversibility is to leverage the information from the forward and reverse transitions to estimate the uncertainty of values in the transition probability matrix. From an information theoretic view, we know that the probability of a transition probability matrix generating the observed trajectory, **X**, is given by,

$$P(\mathbf{X}|T) = \prod_{ij} T_{ij}^{C_{ij}}$$

and from Bayes rule,

$$P(\mathbf{X}|T)P(T) = P(T|\mathbf{X})\mathrm{P}(\mathbf{X})$$

Therefore, we can assert that,

$$P(T|\mathbf{X}) \propto \prod_{ij} T_{ij}^{C_{ij}}$$

where $P(T|\mathbf{X})$ is the probability of a transition matrix given a set of data. Finding the transition matrix that maximizes the probability is termed the maximum likelihood estimation[33],

$$T_{ij}^{MLE} = \arg\max_{T_{ij}^*} P\left(T_{ij}^*|\mathbf{X}\right)$$

Trivially, this value will return the row normalized matrix. However, when the transition probability matrix is solved while simultaneously constrained to obey reversibility of the transition count matrix, the alterations to transition counts should be well balanced by sampling quality. Unfortunately, as is shown in chapter 4, large discrepancies between transition counts can lead to instability which tends to overpopulate a small subset of states.

1.3 Adaptive Sampling

There are a number of computational methods that aim to capture long timescale phenomena and enhance exploration of conformational space. Since MD simulations spend a majority of their time in energy minima—waiting to traverse some energy barrier—most methods attempt to alter the energy landscape to hasten transitions. If the transition is known *a priori*, energetic constraints can be added to pull the conformation over any barriers, which is known as steered molecular dynamics.[34,35] Alternatively, to rapidly explore a landscape in an undirected manner, well depths can be modulated to reduce all energy barriers, as is done in accelerated molecular dynamics simulations.[36,37] If exploration along some order parameter is desired, Metadynamics has gained popularity, which progressively adds gaussian penalty terms to previously explored regions of conformational space projected onto the order parameter.[38,39] In addition to these methods, there are a number of others that attempt to cleverly apply energetic constraints or other

alterations to the energy barriers to enhance exploration.[40-47] For most of these methods, there exist ways to undo the bias to the energetics after exploration, to reproduce accurate thermodynamics. Unfortunately, once the energetics of the landscape are altered, there is no way to obtain accurate kinetic information. Additionally, simulations have a significant chance of traversing unrealistic pathways between states, possibly crossing very large barriers; while there may be accurate free-energy differences between states that are discovered, if the set of states discovered are not realistic, the simulation results will be incredibly misleading. As such, there are serious advantages to using unbiased simulations for characterizing a conformational ensemble, especially if mechanistic details are desired.

Markov state models offer a promising solution to capture long timescale phenomena, while preserving both thermodynamic and kinetic information. As mentioned in section 1.2.1, an MSM provides a framework for stitching together many parallel simulations with structures that a not Boltzmann distributed. This means that we can have any distribution of starting structures and still make a meaningful model. Additionally, there is no reason that the simulations have to be run all at the same time—we can use knowledge of a current set of simulations to make an informed decision about where to sample from next in conformational space. This is the tactic of the set of strategies known as "adaptive sampling". This was first thought of as a great way to gather additional statistics from poorly sampled regions of conformational space and obtain an improved MSM.[48] In this first version, simulations were restarted from states that contributed the most to the uncertainty in eigenvectors and eigenvalues of the obtained transition probability matrix. After this method was developed, adaptive sampling was eventually thought of as a new way to enhance exploration of conformational space, by selecting for states that have a high probability of discovering new and exciting conformations.[49,50] Because each individual

simulation is run without any perturbation to the Hamiltonian, estimates of transition probabilities are unbiased, and proper thermodynamics and kinetics can be reconstructed.

Adaptive sampling schemes typically follow the same protocol: 1) run a swarm of simulations, 2) cluster and analyze the obtained conformations, 3) rank each state based on some ranking method, 4) restart simulations from a set of states that optimize the ranking, and 5) repeat steps 2-4 until sufficient sampling is obtained. Over the years, there have been many adaptive sampling schemes developed, which largely differ based on the way states are either clustered and/or ranked between each round.[51-57] Although the ranking functions differ between these methods, they all focus on statistical quantities and are "undirected" in terms of a direction in conformational space.

There is one major drawback to adaptive sampling: because the rankings are undirected, and conformational space is so unfathomably large, simulations often spend their time exploring large regions of space that are not interesting for a particular biological question. It is from this issue that a significant portion of this thesis was developed, to direct the sampling of conformational space and better explore protein dynamics.

1.4 Scope of Thesis

Despite their widespread use, the ability of molecular dynamics simulations to rigorously answer biologically interesting questions is still severely limited. As mentioned above, this is largely due to gathering sufficient data, although there is also the issue of analyzing the necessarily large data sets once obtained. Thus, the recurring theme for each chapter in this thesis is the need for developing methodologies and tools to gather meaningful data and make sense of it using commodity hardware.

Chapters 2-3 detail the development of the goal-oriented sampling algorithm, Fluctuation Amplification of Specific Traits (FAST). This algorithm differs from previously developed adaptive sampling algorithms in that it guides simulations based on structural metrics *in addition* to the traditional statistical metrics. Chapter 2 details the theoretical development of this algorithm. First, there is a formal justification for using counts-based adaptive sampling as part of the ranking to enhance state discovery. Next, the idea of conformational gradients in conformational space is explore. Then FAST is applied to three challenging problems in protein biophysics: finding cryptic pockets on proteins, finding transition pathways between two known states, and folding proteins. This method reduces the amount of aggregate simulation time required to make meaningful predictions of conformational ensembles. There is an added benefit, in that the smaller amount of data is easier to analyze. Chapter 3 details the practical considerations when running FAST simulation. In addition to providing a walk through, this chapter provides insights into the many hyperparameters that influence exploration and how to tune them for a particular use.

While Chapters 2-3 introduce FAST and show that it can discover interesting states with orders of magnitude less aggregate simulation time than other sampling methods, there remain some fundamental questions. Specifically, do FAST simulations provide realistic pathways between states and are the resulting MSMs, built using goal-oriented sampling, valid in terms of their kinetics and thermodynamics? Additionally, these questions can be raised for the other dominant equilibrium-based sampling methods: running a long simulation, parallel simulations, or adaptive sampling. Chapter 4 explores the relationship between a chosen equilibrium-based sampling scheme and its exploration of conformational space by focusing on state discovery. This relationship is incredibly important because the states discovered when simulations are out

of global equilibrium dictate the final computational predictions. With infinite sampling, all sampling algorithms should provide the same result, however, simulation time is always limited. Additionally, the benefit of adaptive sampling lies in the ability to make efficient use of limited simulations.

With theoretical and implementational considerations addressed in chapters 2-4, chapter 5 shows that the FAST algorithm can be applied to real systems for making meaningful predictions. As such, this chapter applies FAST to understand the difference in conformational ensembles between clinically relevant mutants of TEM-1 β-lactamase. TEM-1 β-lactamase is a protein found in bacteria that degrades β-lactam antibiotics and is a major contributor to the worldwide antibiotic resistance crisis.[58] To combat this scourge, new generations of antibiotics are developed that can evade this protein.[59] However, mutations appear in clinical isolates of β-lactamase that rescue its ability to degrade the new antibiotics faster than they can be developed. The TEM sequences with these rescuing mutations are known as extended spectrum β-lactamases (ESBLs) and are particularly difficult to predict, owing to our general lack of understanding in the proteins' conformational ensemble. As an example of our ignorance to its conformational landscape, when a small molecule was designed to target the β-lactamase active site, and was experimentally determined to bind to the protein, a crystal structure revealed the molecule to bind in a cryptic pocket, otherwise unknown to exist.[60] Without knowledge of β-lactamases' conformational ebb and flow, the factors that allow for mutations to generate ESBLs will remain elusive. This is the case for one particular mutation found in clinical isolates of ESBLs, the M182T mutation.[61,62] This mutation is found to be extraordinarily stabilizing, though there exist contradictory explanations for its mechanism.[63,64] In chapter 5, FAST is used to understand the conformational ensembles of TEM-1 β-lactamase, with and without the M182T

mutation, to develop a novel mechanistic model. This model is unique to previous models in that it is based on the conformational ensemble and is uniquely able to predict and explain new mutations on the protein.

While adaptive sampling reduces the amount of aggregate simulation required for any system, working on larger and more complicated proteins still necessitates tools for dealing with large amounts of data. On that front, Chapter 6 details tools developed for the handling of large MD datasets and the construction of MSMs. This intuitive python library is called enspara and is instrumental in generating FAST simulation data as well as performing subsequent analysis.

Lastly, chapter 7 concludes with summarizing the main advancements contained within this thesis. As science is never finished, future prospects are explored, both in terms of continual methods development, as well as what to do now that we have a suitable method to quickly explore a proteins' conformational landscape.

Bibliography

(1) Campbell, I. D. The March of Structural Biology. *Nature Reviews Molecular Cell Biology 2002 3:5* **2002**, *3* (5), 377–381.

(2) Henzler-Wildman, K.; Kern, D. Dynamic Personalities of Proteins. *Nature 2003 423:6936* **2007**, *450* (7172), 964–972.

(3) Motlagh, H. N.; Wrabl, J. O.; Li, J.; Hilser, V. J. The Ensemble Nature of Allostery. *Nature 2003 423:6936* **2014**, *508* (7496), 331–339.

(4) and, S. E.; Helms, V. Transient Pockets on Protein Surfaces Involved in Protein−Protein Interaction. *J. Med. Chem.* **2007**, *50* (15), 3457–3464.

(5) Nussinov, R. Introduction to Protein Ensembles and Allostery. *Chem. Rev.* **2016**, *116* (11), 6263–6266.

(6) Guo, J.; Zhou, H.-X. Protein Allostery and Conformational Dynamics. *Chem. Rev.* **2016**, *116* (11), 6503–6515.

(7) Boehr, D. D.; Nussinov, R.; Wright, P. E. The Role of Dynamic Conformational Ensembles in Biomolecular Recognition. *Nature Chemical Biology 2009 5:11* **2009**, *5* (11), 789–796.

(8) Heyduk, T. Measuring Protein Conformational Changes by FRET/LRET. *Current Opinion in Biotechnology* **2002**, *13* (4), 292–296.

(9) Ishima, R.; Torchia, D. A. Protein Dynamics From NMR. *Nature Structural & Molecular Biology 1997 4:1* **2000**, *7* (9), 740–743.

(10) Karplus, M.; McCammon, J. A. Molecular Dynamics Simulations of Biomolecules. *Nature Structural & Molecular Biology 1997 4:1* **2002**, *9* (9), 646–652.

(11) Zwier, M. C.; Chong, L. T. Reaching Biological Timescales with All-Atom Molecular Dynamics Simulations. *Curr Opin Pharmacol* **2010**, *10* (6), 745–752.

(12) Shaw, D. E.; Deneroff, M. M.; Dror, R. O.; Kuskin, J. S.; Larson, R. H.; Salmon, J. K.; Young, C.; Batson, B.; Bowers, K. J.; Chao, J. C.; Eastwood, M. P.; Gagliardo, J.; Grossman, J. P.; Ho, C. R.; Ierardi, D. J.; Kolossváry, I.; Klepeis, J. L.; Layman, T.; McLeavey, C.; Moraes, M. A.; Mueller, R.; Priest, E. C.; Shan, Y.; Spengler, J.; Theobald, M.; Towles, B.; Wang, S. C. Anton, a Special-Purpose Machine for Molecular Dynamics Simulation. *Communications of the ACM* **2008**, *51* (7), 91–97.

(13) Shaw, D. E.; Grossman, J. P.; Bank, J. A.; the, B. B. P. O.; 2014. Anton 2: Raising the Bar for Performance and Programmability in a Special-Purpose Molecular Dynamics Supercomputer. *dl.acm.org*.

(14) Shaw, D. E.; Bowers, K. J.; Chow, E.; Eastwood, M. P.; Ierardi, D. J.; Klepeis, J. L.; Kuskin, J. S.; Larson, R. H.; Lindorff-Larsen, K.; Maragakis, P.; Moraes, M. A.; Dror, R. O.; Piana, S.; Shan, Y.; Towles, B.; Salmon, J. K.; Grossman, J. P.; Mackenzie, K. M.; Bank, J. A.; Young, C.; Deneroff, M. M.; Batson, B. Millisecond-Scale Molecular Dynamics Simulations on Anton; ACM Press: New York, New York, USA, 2009; p 1.

(15) Shirts, M.; Pande, V. S. Screen Savers of the World Unite! *Science* **2000**, *290* (5498), 1903–1904.

(16) Shirts, M. R.; Pande, V. S. Mathematical Analysis of Coupled Parallel Simulations. *Physical Review Letters* **2001**, *86* (22), 4983–4987.

(17) Lane, T. J.; Shukla, D.; Beauchamp, K. A.; Pande, V. S. To Milliseconds and Beyond: Challenges in the Simulation of Protein Folding. *Current Opinion in Structural Biology* **2013**, *23* (1), 58–65.

(18) Voelz, V. A.; Bowman, G. R.; Beauchamp, K.; Pande, V. S. Molecular Simulation of Ab Initio Protein Folding for a Millisecond Folder NTL9(1−39). *Journal of the American Chemical Society* **2010**, *132* (5), 1526–1528.

(19) Bowman, G. R.; Voelz, V. A.; Pande, V. S. Atomistic Folding Simulations of the Five-Helix Bundle Protein Λ6−85. *Journal of the American Chemical Society* **2010**, *133* (4), 664–667.

(20) Bowman, G. R.; Pande, V. S.; Noé, F. Introduction and Overview of This Book. In *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*; Advances in Experimental Medicine and Biology; Springer Netherlands: Dordrecht, 2014; Vol. 797, pp 1–6.

(21) Pande, V. S.; Beauchamp, K.; Bowman, G. R. Everything You Wanted to Know About Markov State Models but Were Afraid to Ask. *Methods* **2010**, *52* (1), 99–105.

(22) Chodera, J. D.; Noé, F. Markov State Models of Biomolecular Conformational Dynamics. *Current Opinion in Structural Biology* **2014**, *25*, 135–144.

(23) Husic, B. E.; Pande, V. S. Markov State Models: From an Art to a Science. *Journal of the American Chemical Society* **2018**, *140* (7), 2386–2396.

(24) Grinstead, C. M.; Snell, J. L. *Introduction to Probability*; 2012.

(25) Zwanzig, R. From Classical Dynamics to Continuous Time Random Walks. *J Stat Phys* **1983**, *30* (2), 255–262.

(26) Porter, J. R.; Moeder, K. E.; Sibbald, C. A.; Zimmerman, M. I.; Hart, K. M.; Greenberg, M. J.; Bowman, G. R. Cooperative Changes in Solvent Exposure Identify Cryptic Pockets, Switches, and Allosteric Coupling. *Biophysical Journal* **2019**, *116* (5), 818–830.

(27) Gonzalez, T. F. Clustering to Minimize the Maximum Intercluster Distance. *Theoretical Computer Science* **1985**, *38*, 293–306.

(28) Park, H.-S.; Jun, C.-H. A Simple and Fast Algorithm for K-Medoids Clustering. *Expert Systems with Applications* **2009**, *36* (2), 3336–3341.

(29) Naritomi, Y.; Fuchigami, S. Slow Dynamics in Protein Fluctuations Revealed by Time-Structure Based Independent Component Analysis: the Case of Domain Motions. *The Journal of Chemical Physics* **2011**, *134* (6), 065101.

(30) Pérez-Hernández, G.; Paul, F.; Giorgino, T.; De Fabritiis, G.; Noé, F. Identification of Slow Molecular Order Parameters for Markov Model Construction. *The Journal of Chemical Physics* **2013**, *139* (1), 015102.

(31) Schwantes, C. R.; Pande, V. S. Modeling Molecular Kinetics with tICA and the Kernel Trick. *J. Chem. Theory Comput.* **2015**, *11* (2), 600–608.

(32) Hart, K. M.; Ho, C. M. W.; Dutta, S.; Gross, M. L.; Bowman, G. R. Modelling Proteins' Hidden Conformations to Predict Antibiotic Resistance. *Nature Communications* **2016**, *7*, 12965.

(33) Metzner, P.; Noé, F.; Schütte, C. Estimating the Sampling Error: Distribution of Transition Matrices and Functions of Transition Matrices for Given Trajectory Data. *Physical Review E* **2009**, *80* (2), 021106.

(34) Izrailev, S.; Stepaniants, S.; Isralewitz, B.; Kosztin, D.; Lu, H.; Molnar, F.; Wriggers, W.; Schulten, K. Steered Molecular Dynamics. In *Computational Molecular Dynamics: Challenges, Methods, Ideas*; Lecture Notes in Computational Science and Engineering; Springer Berlin Heidelberg: Berlin, Heidelberg, 1999; Vol. 4, pp 39–65.

(35) Isralewitz, B.; Gao, M.; Schulten, K. Steered Molecular Dynamics and Mechanical Functions of Proteins. *Current Opinion in Structural Biology* **2001**, *11* (2), 224–230.

(36) Voter, A. F. Hyperdynamics: Accelerated Molecular Dynamics of Infrequent Events. *Physical Review Letters* **1997**, *78* (20), 3908–3911.

(37) Hamelberg, D.; Mongan, J.; McCammon, J. A. Accelerated Molecular Dynamics: a Promising and Efficient Simulation Method for Biomolecules. *The Journal of Chemical Physics* **2004**, *120* (24), 11919–11929.

(38) Laio, A.; Parrinello, M. Escaping Free-Energy Minima. *Proc. Natl. Acad. Sci. U.S.A.* **2002**, *99* (20), 12562–12566.

(39) Laio, A.; Gervasio, F. L. Metadynamics: a Method to Simulate Rare Events and Reconstruct the Free Energy in Biophysics, Chemistry and Material Science. *Reports on Progress in Physics* **2008**, *71* (12), 126601.

(40) Go, N. Theoretical Studies of Protein Folding. *Annu. Rev. Biophys. Bioeng.* **1983**, *12* (1), 183–210.

(41) Takada, S. Go-Ing for the Prediction of Protein Folding Mechanisms. *Proc. Natl. Acad. Sci. U.S.A.* **1999**, *96* (21), 11698–11700.

(42) Sugita, Y.; Okamoto, Y. Replica-Exchange Molecular Dynamics Method for Protein Folding. *Chemical Physics Letters* **1999**, *314* (1-2), 141–151.

(43) Fukunishi, H.; Watanabe, O.; Takada, S. On the Hamiltonian Replica Exchange Method for Efficient Sampling of Biomolecular Systems: Application to Protein Structure Prediction. *The Journal of Chemical Physics* **2002**, *116* (20), 9058–9067.

(44) Faraldo-Gómez, J. D.; Roux, B. Characterization of Conformational Equilibria Through Hamiltonian and Temperature Replica-Exchange Simulations: Assessing Entropic and Environmental Effects. *J Comput Chem* **2007**, *28* (10), 1634–1647.

(45) Perez, A.; MacCallum, J.; Dill, K. A. Meld: Modeling Peptide-Protein Interactions. *Biophysical Journal* **2013**, *104* (2), 399a.

(46) Perez, A.; MacCallum, J. L.; Dill, K. A. Accelerating Molecular Simulations of Proteins Using Bayesian Inference on Weak Information. *Proc. Natl. Acad. Sci. U.S.A.* **2015**, *112* (38), 11846–11851.

(47) Zheng, L.; Chen, M.; Yang, W. Random Walk in Orthogonal Space to Achieve Efficient Free-Energy Simulation of Complex Systems. *Proc. Natl. Acad. Sci. U.S.A.* **2008**, *105* (51), 20227–20232.

(48) Hinrichs, N. S.; Pande, V. S. Calculation of the Distribution of Eigenvalues and Eigenvectors in Markovian State Models for Molecular Dynamics. *The Journal of Chemical Physics* **2007**, *126* (24), 244101.

(49) Bowman, G. R.; Ensign, D. L.; Pande, V. S. Enhanced Modeling via Network Theory: Adaptive Sampling of Markov State Models. *J. Chem. Theory Comput.* **2010**, *6* (3), 787–794.

(50) Weber, J. K.; Pande, V. S. Characterization and Rapid Sampling of Protein Folding Markov State Model Topologies. *J. Chem. Theory Comput.* **2011**, *7* (10), 3405–3411.

(51) Voelz, V. A.; Elman, B.; Razavi, A. M.; Zhou, G. Surprisal Metrics for Quantifying Perturbed Conformational Dynamics in Markov State Models. *J. Chem. Theory Comput.* **2014**, *10* (12), 5716–5728.

(52) Dickson, A.; Charles L Brooks, I. WExplore: Hierarchical Exploration of High-Dimensional Spaces Using the Weighted Ensemble Algorithm. *J. Phys. Chem. B* **2014**, *118* (13), 3532–3542.

(53) Doerr, S.; De Fabritiis, G. On-the-Fly Learning and Sampling of Ligand Binding by High-Throughput Molecular Simulations. *J. Chem. Theory Comput.* **2014**, *10* (5), 2064–2069.

(54) Bacci, M.; Vitalis, A.; Caflisch, A. A Molecular Simulation Protocol to Avoid Sampling Redundancy and Discover New States. *Biochimica et Biophysica Acta (BBA) - General Subjects* **2015**, *1850* (5), 889–902.

(55) Kukharenko, O.; Sawade, K.; Steuer, J.; Peter, C. Using Dimensionality Reduction to Systematically Expand Conformational Sampling of Intrinsically Disordered Peptides. *J. Chem. Theory Comput.* **2016**, *12* (10), 4726–4734.

(56) Sultan, M. M.; Pande, V. S. Decision Functions From Supervised Machine Learning Algorithms as Collective Variables for Accelerating Molecular Simulations. February 28, 2018.

(57) Noé, F.; Banisch, R.; Clementi, C. Commute Maps: Separating Slowly Mixing Molecular Configurations for Kinetic Modeling. *ACS Publications* **2016**, *12* (11), 5620–5630.

(58) Bush, K.; Jacoby, G. A. Updated Functional Classification of B-Lactamases. *Antimicrobial Agents and Chemotherapy* **2010**, *54* (3), 969–976.

(59) Drawz, S. M.; Bonomo, R. A. Three Decades of B-Lactamase Inhibitors. *Clinical Microbiology Reviews* **2010**, *23* (1), 160–201.

(60) Horn, J. R.; Shoichet, B. K. Allosteric Inhibition Through Core Disruption. *Journal of Molecular Biology* **2004**, *336* (5), 1283–1291.

(61) Huang, W.; Palzkill, T. A Natural Polymorphism in B-Lactamase Is a Global Suppressor. *Proc. Natl. Acad. Sci. U.S.A.* **1997**, *94* (16), 8801–8806.

(62) Kather, I.; Jakob, R. P.; Dobbek, H.; Schmid, F. X. Increased Folding Stability of TEM-1 B-Lactamase by in Vitro Selection. Journal of Molecular Biology 2008, 383 (1), 238–251.

(63) Wang, X.; Minasov, G.; Shoichet, B. K. Evolution of an Antibiotic Resistance Enzyme Constrained by Stability and Activity Trade-Offs. Journal of Molecular Biology 2002, 320 (1), 85–95.

(64) Orencia, M. C.; Yoon, J. S.; Ness, J. E.; Willem P. C. Stemmer; Stevens, R. C. Predicting the Emergence of Antibiotic Resistance by Directed Evolution and Structural Analysis. Nature Structural & Molecular Biology 1997 4:1 2001, 8 (3), 238–242.

Chapter 2

FAST Conformational Searches by Balancing Exploration/Exploitation Trade-Offs

2.1 Preamble

This chapter is adapted from the following article: Zimmerman, M.I. and Bowman, G.R., "FAST Conformational Searches by Balancing Exploration/Exploitation Trade-Offs", Journal of Chemical Theory and Computation, 11(12), 5747-5757

2.2 Introduction

Understanding the structural mechanisms of conformational changes, such as protein folding and allosteric communication, is a notoriously difficult problem. Molecular dynamics (MD) simulations can complement experimental studies of such problems by filling in information beyond their reach, such as an atomically-detailed picture of conformational heterogeneity. However, it is extremely difficult to simulate biologically relevant processes on millisecond and slower timescales with conventional molecular dynamics simulations.

Three broad classes of methods have been developed to capture longer timescale processes with computer simulations. The first class consists of directed methods that actively drive simulations towards some goal, such as steered molecular dynamics,[1] metadynamics,[2,3] the string method,[4,5] and methods for introducing restraints from experiments.[6,7] Unfortunately, these often go through unrealistically high-energy conformations (Figure 2.1, red path) and fail to

explore conformations orthogonal to the direction they're being driven in, though new methods are more capable of finding the energetically preferred paths.[8] The second class consists of undirected methods that attempt to accelerate the exploration of all conformations, such as replica exchange,[9] accelerated molecular dynamics,[10] weighted ensembles,[11-13] combinations of coarse-grained and all-atom simulations,[14] and adaptive sampling.[15-21] While these methods will eventually provide the correct result, conformational space is so enormous that researchers can easily expend all of their computing resources exploring structures that are not relevant to the problem they set out to solve (Figure 2.1 yellow enclosed space). Most of the approaches in these two classes also preclude the acquisition of kinetic information by introducing a biasing force or altering properties like the potential energy or temperature. While they still provide the proper thermodynamics, the lack of kinetic information makes it impossible to make quantitative connections with many experimental techniques. The third class of methods focuses on the development of a specialized supercomputer, such as a distributed computing platform[22,23] or purpose-built hardware,[24] that is capable of running enough simulation to discover the relevant conformational space. This approach has led to some of the most dramatic demonstrations of the power of simulations, including insights into protein folding[25,26] and allosteric communication[27-29] on up to millisecond timescales. However, there are still many processes beyond the reach of these computers. Moreover, very few researchers have access to these resources.

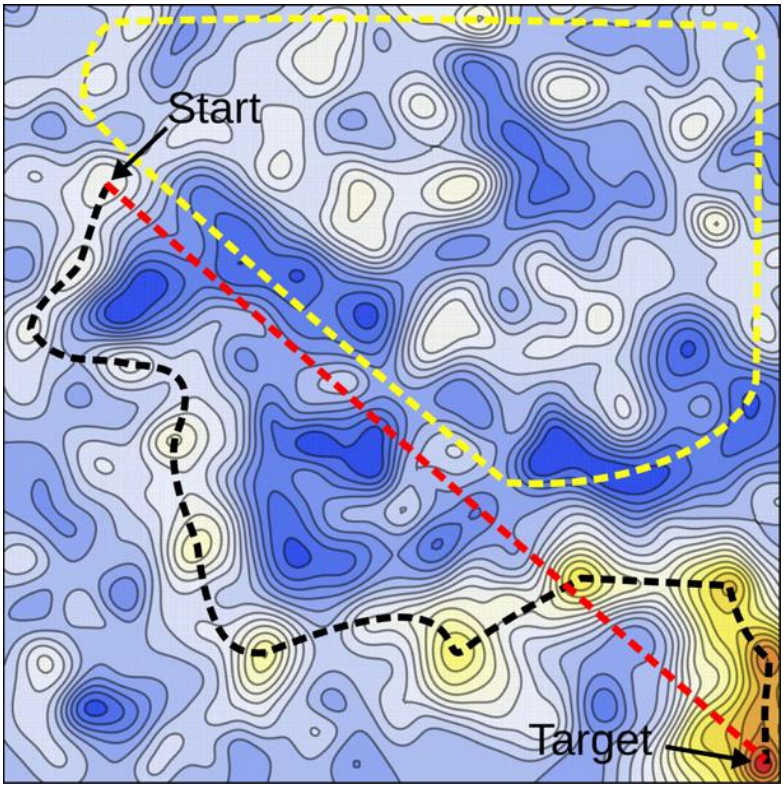

Figure 2.1: Contour plot of an energy landscape colored in blue, white, yellow, and red from highest to lowest energy. The black line is the optimal path from a starting state to a target. The red line is the path found by directed methods. The yellow line encompasses the area where undirected methods are likely to get lost.

Here, we propose a goal-oriented sampling method called fluctuation amplification of specific traits (FAST) that combines elements of a directed and undirected search to quickly explore regions of conformational space that are relevant to a given problem. This algorithm was inspired by the fact that a protein folds by following an energy gradient to its native state[30-32] but following such gradients is non-trivial because there are energy barriers and dead-ends along the way. We hypothesized that the correlation between structures and energies gives rise to similar gradients for many other physical properties—such as the root-mean-squared deviation (RMSD) to a target structure and the solvent-accessible surface area. For example, we expect that transitioning from a conformation with a small solvent accessible surface area to one with a large surface area will require passing through a series of conformations with steadily increasing

surface areas. If these gradients exist, then it should be possible to follow them to identify structures that maximize (or minimize) specific physical properties. Literature on optimization theory has dealt with related problems by balancing tradeoffs between focused searches around promising solutions (exploitation) and trying completely novel solutions (exploration).[33,34] The FAST algorithm leverages these ideas 1) to recognize and amplify structural fluctuations along gradients that optimize a selected physical property whenever possible, 2) to overcome barriers that interrupt these overall gradients, and 3) to re-route to discover alternative paths when faced with insurmountable barriers.

FAST achieves these objectives by drawing on work on the multi-armed bandit problem and particle-swarm optimization. The multi-armed bandit problem[33] is a classic exploration/exploitation trade-off problem in which a hypothetical gambler at a row of slot machines must decide when to 1) try a relatively untested slot machine that could easily yield enormous or meager returns and 2) when to exploit the expected rewards of a tried-and-true machine. A key result is that one can obtain outstanding performance by using estimates of the uncertainty in the expected rewards for each slot machine to select the one that has the highest probability of yielding the greatest rewards.[35] A simple means of achieving this objective is to always choose the slot machine with the highest probability of the greatest return, which can be assessed by a reward function of the form

$$r(i) = \mu(i) + \alpha\sigma(i) \qquad (1)$$

where i is a slot machine, $\mu(i)$ is its average return, $\sigma(i)$ is the standard deviation of the returns from that machine, and α is a constant that controls the importance of uncertainty.[36] Particle-

swarm optimization[34] is another means of addressing exploration/exploitation tradeoffs, but by using a swarm of walkers to explore parameter space. These walkers are designed to balance between spreading out to explore different potential solutions and converging on promising regions of parameter space.

Inspired by these ideas, FAST runs successive swarms of simulations where the starting points for each swarm are chosen from the set of all previously discovered conformations based on a reward function. This reward function quantifies the relative likelihood that simulations started from different structures will discover new conformations that maximize (or minimize) a selected physical property. It mimics the functional form of Equation 1 by including a directed component that parallels the mean return and an undirected component corresponding to the uncertainty in the return on investment, as described in the Methods section. The directed component allows FAST to follow gradients by searching near promising solutions for even better ones. Following such gradients alone is not an ideal search strategy because some regions of conformational space with a promising gradient may lead to dead ends. To avoid this pitfall, the undirected component favors poorly sampled regions of conformational space, allowing the algorithm to recognize dead-ends where simulations repeatedly fail to discover structures that better optimize the target function and to re-route to less explored regions of conformational space in search of new leads. Since no biasing force is applied to any individual simulation, the final dataset can be used to build a Markov state model (MSM) to extract the proper thermodynamics and kinetics despite the non-equilibrium distribution of starting points for the trajectories (see the Methods section for details).[37-39] This approach differs from existing adaptive sampling techniques[15-19] in that it seeks to prioritize what types of structures are explored rather than purely trying to minimize the statistical uncertainty in a model. This is an

important distinction because adaptive sampling can easily exhaust finite computational resources searching through irrelevant conformations, whereas we expect the goal-oriented method presented here to quickly focus in on regions of conformational space that are relevant to the problem at hand.

To test FAST, we have applied it to three challenging sampling problems 1) the discovery of unexpected pockets that might be valuable drug targets, 2) the identification of transition paths between specific conformations, and 3) protein folding. We begin by retrospectively analyzing existing MSMs to assess whether various physical properties have the gradients we hypothesize to exist in protein conformational space. Then we test FAST's ability to identify and follow gradients that are relevant to each of the problems considered.

2.3 Methods

2.3.1 FAST Algorithm

The FAST algorithm is intended to optimize any selected geometric function ϕ of a protein structure, including, but not limited to energies, RMSDs, and solvent accessible surface areas. For a given physical property ϕ, the FAST- ϕ algorithm is:

1. Start a swarm of simulations from a set of initial conformations, such as one or more known crystal structures.
2. Cluster all the simulation data collected so far into discrete conformational states.
3. Calculate a reward function for each state

$$r_{\phi}(i) = \bar{\phi}(i) + \alpha\bar{\psi}(i) \quad (2)$$

where i is a particular state, $\bar{\phi}(i)$ is a directed component that fosters exploitation by favoring states that optimize some structural metric of interest (such as the RMSD to a target) compared to other states, $\bar{\psi}(i)$ is an undirected component that fosters exploration by favoring states that are poorly sampled compared to other states, and α is a control parameter that determines the relative importance of the directed and undirected components of the reward function. The bars over each component of this reward function indicate that we feature-scale them (equations below) to highlight the differences between states and ensure that a variable with a greater dynamic range does not overshadow the other component. For example, when trying to maximize the solvent-accessible surface area, $\bar{\phi}(i)$ will range from zero for the state with the lowest solvent-accessible surface area to one for the state with the largest solvent-accessible surface area and $\bar{\psi}(i)$ will range from zero for the most sampled state to 1 for the least sampled state. Therefore, poorly sampled states that optimize the target function are expected to yield the highest reward while states that have been explored thoroughly and are far from the target are not expected to be rewarding.

4. Start a new swarm of simulations, where the number of simulations started from each state is proportional to the reward function for that state.
5. Repeat steps 2-4 until the target function has converged or until some predetermined amount of simulation has been conducted.
6. Build an MSM from the final dataset to capture the proper thermodynamics and kinetics, thereby correcting for any bias introduced by selecting starting conformations for each swarm of simulations according to our reward function instead of a Boltzmann distribution.[37,38]

It is important to note that a valid MSM does not need to be constructed for each round of FAST. This is an important feature since the algorithm needs to work properly even when there is not enough data to accurately estimate transition probabilities for parts of conformational space. The clustering simply needs to be at a resolution that is fine-grained enough to distinguish 1) structures with different values of the target geometric function and 2) regions of conformational space that are well-sampled versus those that are poorly sampled. In step 6, more care is required to build a valid MSM that satisfies the Markov assumption, has a reasonable lag time, and captures the phenomena of interest.

Feature-scaling transforms some quantity into a ranking that ranges from 0 to 1 from the least preferred to the most preferred value, respectively. For a quantity ϕ that one wishes to maximize

$$\bar{\phi}(i) = \frac{\phi(\mathrm{i}) - \phi_{min}}{\phi_{max} - \phi_{min}}$$

whereas for a quantity one wishes to minimize

$$\bar{\phi}(i) = \frac{\phi_{max} - \phi(\mathrm{i})}{\phi_{max} - \phi_{min}}$$

where ϕ_{min} and ϕ_{max} are the minimum and maximum values of $\phi(\mathrm{i})$.

For the undirected component of our reward function, $\bar{\psi}(i)$, we adopt a Bayesian perspective to devise a simple measure of how likely simulations started from a given state are to discover new states. We begin by assuming that the biomolecule under consideration has n

structural states and that $n = n_d + n_u$, where n_d is the number of states FAST has discovered so far and n_u is the number of undiscovered states. Following previous work,[15,16] we assume that, prior to observing any data, a simulation started from some initial state has an equal probability of transitioning to any possible final state. Formally, this is achieved by adding a pseudo-count $\tilde{C} = 1/n$ to every element of a transition count matrix (C) used to keep track of the number of transitions observed between every pair of states (C_{ij} is the number of transitions observed from state i to state j). Next we assume that the transition probabilities out of each state are Dirichlet distributed, which is a common way to enforce that they are properly normalized.[15,40,41] Given this assumption, the expected probability of transitioning from state i to any undiscovered state in the set u is

$$E(p_{iu}) = \sum_{j \in u} \left[\frac{1 + \tilde{C}}{\sum_{k=1}^{n} 1 + C_{ik} + \tilde{C}} \right]$$

This function reaches its maximum for the state i that was observed least, as captured by the total number of transitions from that state to any other state, $C_i = \sum_{k=1}^{n} C_{ik}$. Therefore, we can maximize our chances of discovering new states (e.g. transitioning to an as yet undiscovered state) by running simulations from the most poorly sampled states discovered so far. Feature-scaling the number of observations of each state to favor poorly sampled states and to put this undirected component of our reward function on the same scale as the directed component yields

$$\bar{\psi}(i) = \frac{C_{max} - C_i}{C_{max} - C_{min}}$$

where C_{min} and C_{max} are the minimum and maximum number of observations of any state, respectively. Favoring poorly sampled states parallels a previously reported heuristic for discovering new conformations.[42] However, we emphasize that balancing this with the directed component of our reward function provides a dramatic improvement in performance, as described in the Results section. The Results section also provides an explicit example of how this works in practice.

To determine how to set the balance between the directed and undirected components of FAST's reward function, the algorithm was run with different values of the α parameter using synthetic trajectories generated with existing MSMs, as has been done in previous work on adaptive sampling algorithms.[16] Values ranging from 0.5 to 1.5 gave very similar results, so $\alpha = 1$ was selected to place equal weight on the two components for this study. However, there is no guarantee that this value of α will be optimal for every application. Future work on how best to set this parameter may be valuable.

Simulation parameters for production runs with real molecular dynamics simulations are described below. For β-lactamase, 50 rounds of simulations were run. Each round consisted of a swarm of 30 simulations, each 10 ns in length. Therefore, a total of 15 μs of simulation were run for each variant of FAST performed for this study. For the variant of the villin headpiece, 20 rounds of simulations were run. Each round consisted of a swarm of 10 simulations, each 5 ns in length. Therefore, a total of 1 μs of simulation was run. These simulation lengths were chosen to balance a tradeoff between two competing factors: 1) needing simulations to be longer than the lag time used for the final model so that a reasonable MSM can be generated and so that each simulation has a reasonable chance of hopping to a new state and 2) favoring shorter simulations

so that each trajectory remains near the region of conformational space where more data is desired rather than drifting to less desirable structures.

2.3.2 MD Simulations

All simulations were run with Gromacs 4.6.5.[43,44] β-lactamase simulations were run at 300 K using the AMBER ff96 force field[45] with the OBC GBSA implicit solvent model.[46] Using implicit solvent is advantageous for these initial tests as we do not have to store water degrees of freedom or re-solvate/re-equilibrate protein conformations when spawning new swarms of simulations. The single starting conformation used for all of these simulations was generated by placing the crystallographic structure of β-lactamase (PDB ID: 1BTL[47]) in a cubic box that extended one nm beyond the protein in any dimension. This system was energy minimized with the steepest descent algorithm until the maximum force fell below 1,000 kJ/mol/min using a step size of 0.01 nm and a cut-off distance of 1.2 nm for the neighbor list, Coulomb interactions, and Van der Waals interactions. For production runs, all bonds were constrained with the LINCS algorithm[48] and virtual sites[49] were used to allow a 4 fs time step. Cut-offs of 1.0 nm were used for the neighbor list, Coulomb interactions, and Van der Waals interactions, respectively. The Verlet cut-off scheme was used for the neighbor list. The stochastic velocity rescaling (v-rescale) thermostat[50] was used to hold the temperature at 300 K. Conformations were stored every 10 ps. For the Villin headpiece (PDB ID: 2F4K[51]), the simulation settings and one of the extended starting structures from a previous study (structure 5) were employed.[52] Structures were drawn with PyMOL.[53]

2.3.3 Clustering and MSM Construction

All clustering and MSM construction were performed with MSMBuilder.[54,55] An MSM is a discrete-time Master equation model that models protein dynamics as stochastic hopping between discrete conformational states.[39] The states are identified by dividing conformational space up into discrete states, typically by clustering all the conformations sampled by some set of molecular dynamics simulations. Then a transition count matrix is constructed, where the element in row i and column j contains the number of transitions from state i to state j observed over the course of some observation interval, called the lag time of the model. The counts matrix is then used to infer a transition probability matrix that contains the probability of transitioning from every possible starting state i to every possible ending state j within a lag time. These matrices are typically estimated with an iterative procedure for identifying the maximum likelihood set of transition probabilities that satisfy microscopic reversibility.[56,57] Thermodynamic and kinetic properties can then be derived from the transition probability matrix rather than the raw simulation data. As a result, these properties are insensitive to the distribution of the starting points used for each simulation, as long as there is sufficient data to obtain a reasonable estimate of the transition probabilities out of each state.[37,38] While building an MSM from the final dataset is extremely important for obtaining the proper thermodynamics and kinetics, the clustering of each round of FAST simulations need not be a well-behaved MSM since our reward function does not depend on estimates of the transition probabilities between states. Therefore, these intermediate models just require a clustering with sufficient resolution to detect fluctuations that optimize the target function.

The same clustering procedure was used to analyze each round of simulations and to build an MSM for the final dataset. Following a standard protocol,[56] every 10th conformation

from the simulations for each protein were clustered with a k-centers algorithm based on the RMSD between protein conformations. The remaining 90% of the data was then assigned to these clusters and a lag time was selected based on an implied timescales plot.[58] FAST-SASA β-lactamase simulations were clustered based on the RMSD between all backbone heavy atoms and C_β atoms until every cluster had a radius—i.e. maximum distance between any data point in the cluster and the cluster center—less than 1.0 Å and a lag time of 30 ps was employed. FAST-RMSD β-lactamase simulations were clustered based on the RMSD between the helices and loops that move the most when comparing the starting and ending structures (all backbone heavy atoms and C_β atoms in helices 11 and 12 and the loops before and after helix 11, which include residues 215-227 and 270-290) until every cluster had a radius less than 1.0 Å and a lag time of 30 ps was employed. FAST-energy villin simulations were clustered based on the RMSD between all backbone heavy atoms and C_β atoms until every cluster had a radius less than 3.0 Å and a lag time of 2.5 ns was employed. Smaller clusters were employed for the β-lactamase simulations because the conformational changes we intended to capture were subtler than the folding process we targeted in the villin application. The same settings were also used for our retrospective analysis of existing β-lactamase[27] and villin simulations.[52]

2.3.4 Other Analyses

Pocket detection was performed with an implementation of LIGSITE.[27,59] RMSDs and solvent accessible surface areas were calculated with MDTraj.[60] The highest flux paths between specific starting and ending conformations were performed with transition path theory.[61,62]

2.4 Results

2.4.1 Many Physical Properties Have Gradients in Conformational Space

FAST will perform best if the physical property of interest has gradients in conformational space. We hypothesized that the correlation between structures and energies that gives rise to the energetic drive to fold might also give rise to similar gradients in conformational space for other physical properties of proteins. As a first test of this hypothesis, analysis of a number of existing MSMs was performed to determine if the highest flux paths from the crystallographic state to the states that optimize some geometric property do indeed have roughly monotonically increasing (or decreasing) values of that property. For example, Figure 2.2 shows the preferred pathways from the crystal structure of TEM-1 β-lactamase to the states with the highest solvent-accessible surface areas discovered in 81 microseconds of aggregate simulation conducted on the Folding@home distributed computing environment.[27] The solvent-accessible surface areas of structural states along these high flux pathways tend to increase monotonically, so it is reasonable to expect the directed component of FAST to help the algorithm move along these paths quickly. There are some backwards steps along these paths that require moving from states with larger solvent-accessible surface areas to states with lower surface areas but these steps are small enough that it is also reasonable to expect the undirected, statistical component of FAST to easily overcome these hurdles. Similar trends are also observed for properties like the energy and RMSD to a selected target structure in this model of β-lactamase, as well as models for proteins like a fast-folding variant of the villin headpiece (500 μs of simulation),[52] NTL9 (1.5 ms of simulation),[25] and lambda repressor (1.3 ms of simulation).[63] Taken together, this evidence supports the hypothesis that many physical properties have gradients in conformational space that the FAST algorithm is intended to identify and follow.

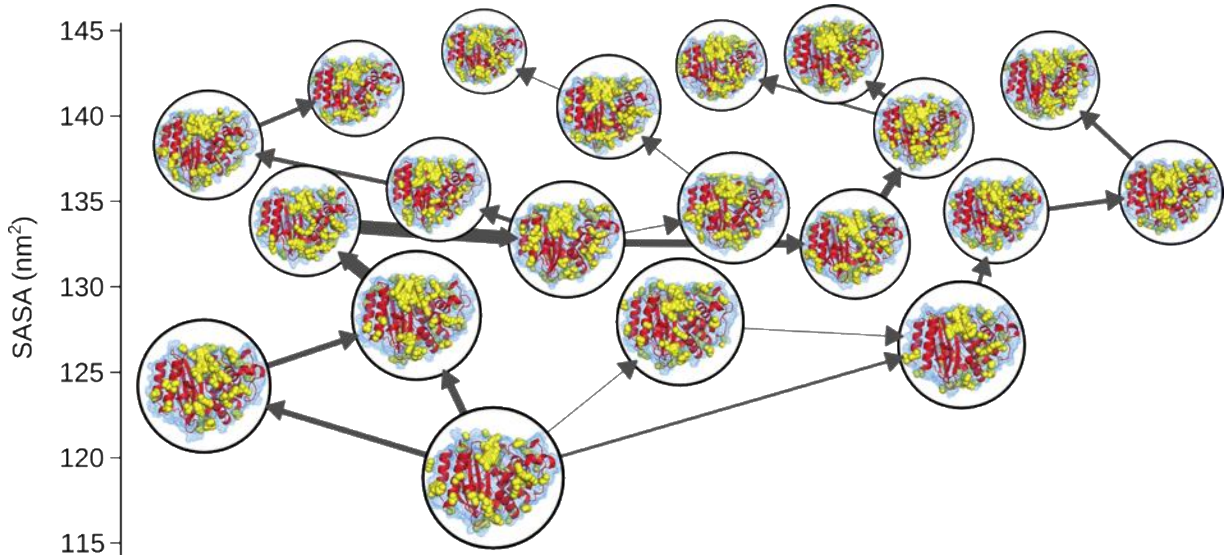

Figure 2.2: Transition pathways from the crystal structure of TEM-1 β-lactamase to the five states with the largest solvent accessible surface areas (SASAs) observed in our past work. β-lactamase is depicted with a red ribbon following the backbone, a blue mesh for the surface, and yellow spheres filling the observed pockets on the protein surface. State sizes are inversely proportional to their free-energies, so larger states have higher equilibrium probabilities. Line thickness is directly proportional to the relative flux observed between the start and end states.

2.4.2 FAST Accurately Identifies the Preferred Paths to Target Conformations

If FAST works as intended, then it should be capable of quickly following gradients in conformational space to find the preferred paths to structures that optimize a selected geometric function. As a first test of whether FAST successfully achieves this goal, we compared its performance to conventional simulations using an existing MSM to generate synthetic trajectories via kinetic Monte Carlo. To generate a synthetic trajectory, one first selects a starting state, then uses the transition probabilities out of that state to randomly select a new state, and repeats this procedure until a desired trajectory length is reached. Synthetic trajectories can then be used to estimate the transition probabilities between states to reconstruct the MSM they were generated with. Performing initial tests with such synthetic trajectories is advantageous because 1) it is much more computationally efficient than running real molecular dynamics simulations

and 2) the MSM used to generate the trajectories serves as a gold standard for assessing the performance of different methods.

We chose a previously reported relative entropy metric to assess the quality of MSMs reconstructed from synthetic trajectories.[16] The relative entropy between two MSMs is

$$D(P||Q) = \sum_{i,j}^{N} P_i \, P_{ij} \log \frac{P_{ij}}{Q_{ij}}$$

where P is the transition matrix for the reference MSM used to generate the synthetic trajectories, P_i is the equilibrium probability of state i in that MSM, and P_{ij} is the probability of hopping from state i to state j in the reference MSM. Q, Q_i, and Q_{ij} are the corresponding properties of the MSM reconstructed from synthetic trajectories. The relative entropy is zero if the two MSMs are identical and becomes increasingly large the more the two models differ. To ensure that every transition probability is non-zero and avoid infinite relative entropies, we used a pseudo-count of $1/n$, where n is the number of states in the model, as described in the methods section and our previous work.[16]

We used our existing MSM for β-lactamase to simulate how quickly FAST-RMSD finds structures resembling conformations bound to a surprising allosteric ligand compared to conventional simulations. First, we identified the five states with the lowest RMSD to the target structure and identified the three highest flux pathways from the state containing the ligand-free crystal structure to each of the five target states (15 paths total). Together, these paths contained 32 of the 3469 states in the MSM. Then we ran long conventional simulations and FAST-RMSD simulations started from the crystallographic state, constructed MSMs from each set of synthetic

trajectories, and employed the relative entropy metric to assess how well each method captured the transition probabilities for the 32 states along the highest-flux pathways to low RMSD states. Figure 2.3A and 2.3B show the results of repeating this analysis for varying numbers of simulations of different lengths. These results demonstrate that FAST-RMSD accurately captures this structural subspace with far less total simulation time than conventional simulations. Comparing the methods across all states (Figure 2.3C and 2.3D) also demonstrates that FAST yields models that are as accurate as conventional simulations on a global level.

Together, these results suggest it is possible to extract the proper thermodynamics and kinetics from FAST simulations despite the fact that starting points for simulations are not chosen according to a Boltzmann distribution. As a further test of the algorithm we also applied it to three real-world problems using real molecular dynamics simulations instead of synthetic trajectories, as described below.

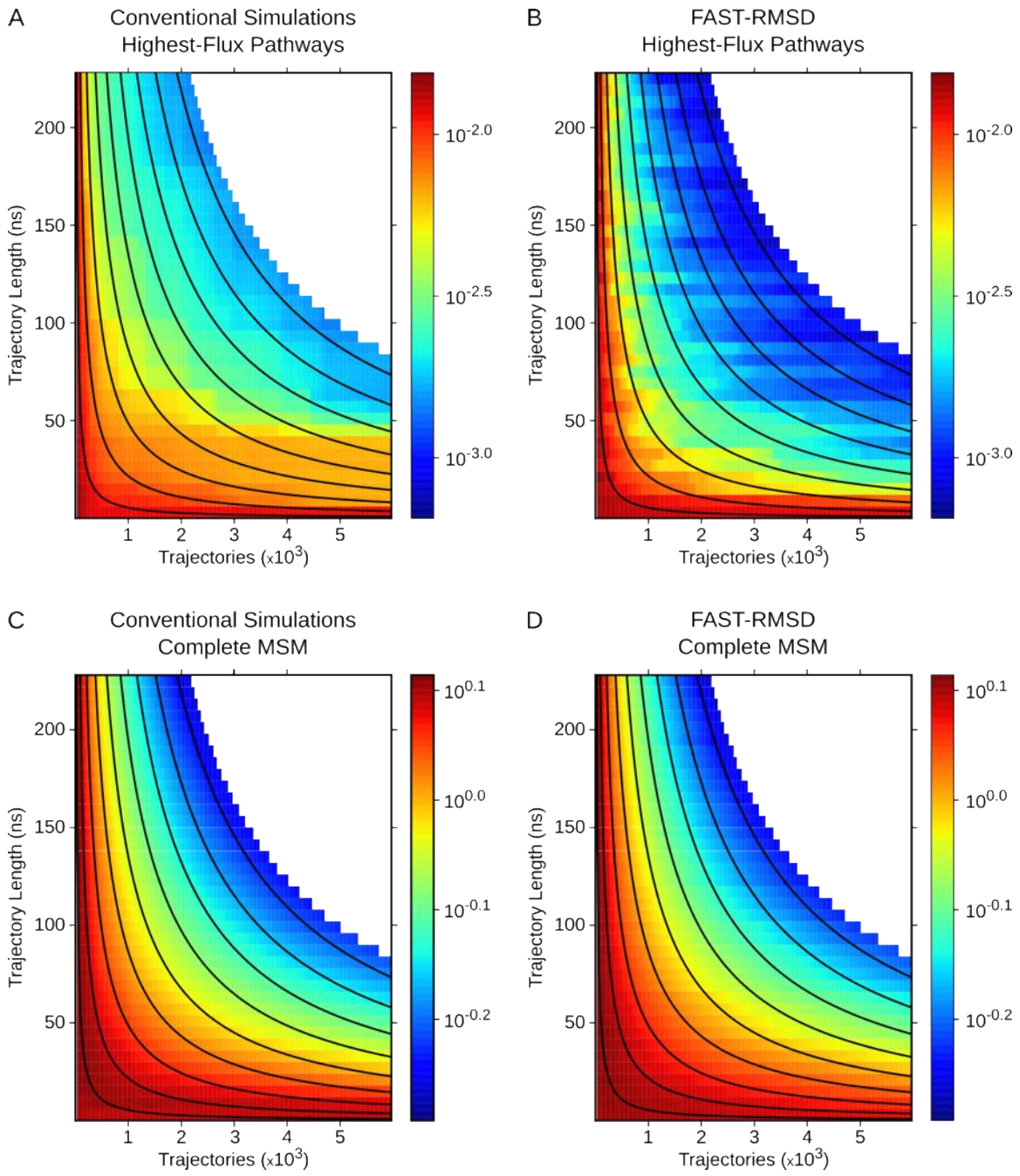

Figure 2.3: Relative entropies between the gold-standard MSM of β-lactamase and MSMs constructed with different sampling methods using varying numbers of kinetic Monte-Carlo simulations of different lengths. Panels A and B show the relative entropies for a subset of states along the highest flux pathways to the five states with the lowest RMSDs to a target structure for conventional and FAST-RMSD simulations, respectively. Panels C and D show the relative entropies for the entire MSMs from each sampling method. Black contours indicate equivalent aggregate simulation time. Calculations were not performed for the white regions.

2.4.3 FAST-SASA Discovers a Diversity of Pocket Structures

One use of molecular dynamics simulations is to discover unexpected pockets that open as a protein fluctuates away from its crystal structure that might serve as valuable drug targets. Since the opening of pockets will generally increase a protein structure's solvent accessible surface area,[64] we chose to maximize this property using FAST-SASA.

To understand how FAST works, the highest flux pathways from the initial (crystallographic) state to the five states with the largest solvent accessible surface areas discovered by FAST-SASA were identified and colored according to when they were first discovered, as shown in Figure 2.4. In the first few rounds of simulation, FAST-SASA finds a few states with somewhat higher solvent accessible surface areas, such as states A and B. At this point, these states have the highest solvent accessible surface areas and are poorly sampled since they were just discovered. Therefore, they are selected as starting conformations for the next round of simulations. Simulations spawned from these states then discover states C-E, which are selected as the starting points for the next swarm, again because they have large solvent accessible surface areas and are poorly sampled. Simulations that are spawned from state D, and those subsequently discovered, lead to the discovery of state F, one of the states with the largest solvent-accessible surface areas. When the sampling of state F fails to produce new states with larger solvent-accessible surface areas, its ranking decreases leading to the favoring of states that have been sampled less despite having a lower solvent-accessible surface area. Sampling from these lower-solvent accessible surface areas helps to discover a variety of new states, such as states G and H, that have the potential to elucidate new pathways to high solvent-accessible surface area states. These states are ranked highly due to their recent discovery and manage to discover independent pathways to some of the other states with the largest solvent-accessible

surface areas (I-L). The yellow spheres in Figure 2.4 fill in pockets that open in the protein structures, highlighting that there are distinct pockets forming in different states with equivalent solvent accessible surface areas.

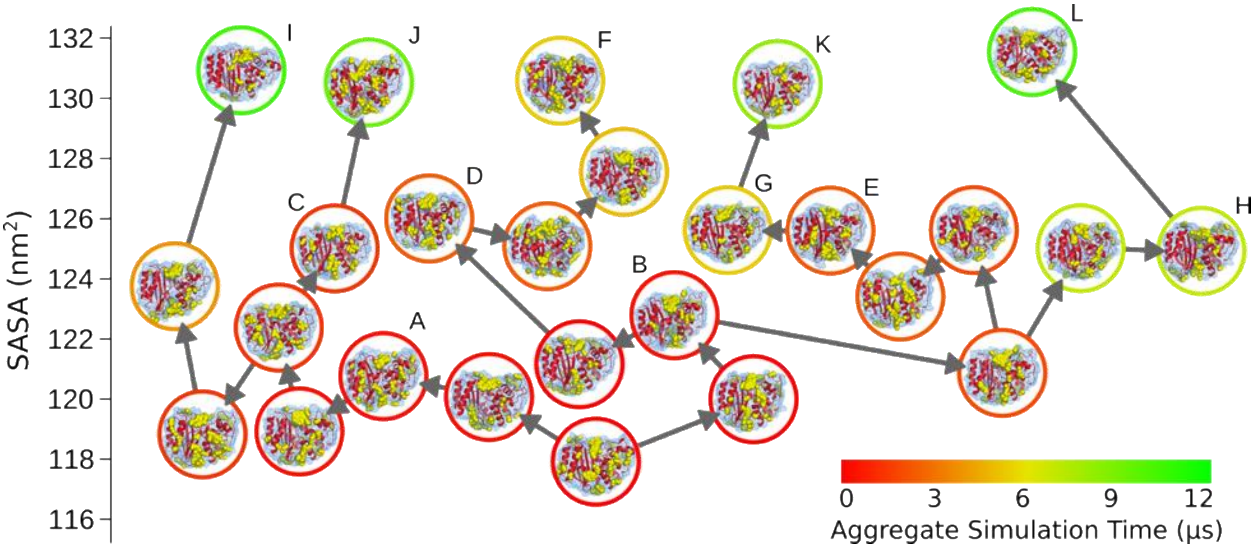

Figure 2.4: Transition pathways from the crystal structure of TEM-1 β-lactamase to the five states with the largest solvent accessible surface areas (SASAs) discovered using FAST-SASA. β-lactamase is depicted with a red ribbon following the backbone, a blue mesh for the surface, and yellow spheres filling the observed pockets on the protein surface. States are colored to indicate when they were discovered during the course of 12 μs of FAST-SASA sampling.

To assess the performance of FAST-SASA, we compared it to conventional molecular dynamics simulations, a purely SASA-based sampling scheme that just uses the directed component of FAST-SASA, and a variant of counts-based adaptive sampling that just uses the undirected component of FAST-SASA. An equivalent amount of conventional molecular dynamics simulations (ten 1.5 μs simulations) only explore conformations near the crystal structure, as shown in Figure 2.5A. The small increases in solvent accessible surface area that these simulations achieve make a quantitative comparison with FAST-SASA impossible, so we can only conclude that FAST-SASA is orders of magnitude more efficient.

Counts-based sampling is also significantly less efficient than FAST-SASA. The fact that this algorithm lacks a directed component prevents it from aggressively capitalizing on

promising structures. Instead, counts-based sampling tries to build out from every new state that it discovers. In doing so, it discovers more total states than FAST-SASA, as shown in Figure 2.5B, but most have small solvent accessible surface areas and do not have the sort of pockets that we set out to discover in this application. FAST-SASA finds states with equally large solvent-accessible surface areas at least eight times faster than counts-based sampling alone. Moreover, this is a conservative estimate of the improved performance of FAST-SASA because it finds at least 30-times as many conformations with surface areas greater than 125 nm^2. Finding equivalent diversity with counts-based sampling would likely take orders of magnitude more simulation than with FAST-SASA given the undirected nature of the purely counts-based algorithm.

SASA-based sampling finds states with much higher solvent accessible surface areas than the conventional simulations or counts-based sampling (Figure 2.5A). Indeed, SASA-based simulations find a few states with solvent accessible surface areas that are comparable to the best structures found by FAST-SASA. However, compared to FAST-SASA, it essentially finds a single a high solvent accessible surface area state and then persistently simulates that state because it lacks the undirected component that allows FAST-SASA to give up on a state and re-route to other potentially more fruitful starting conformations. Therefore, FAST-SASA discovers far more states (Figure 2.5B), including at least twice as many conformations with surface areas greater than 125 nm^2. Since SASA-based sampling persistently spawns new simulations from the single high surface area state that it finds, it is unlikely to ever discover the diversity of structures that FAST-SASA finds. Therefore, as with the conventional simulations, we conclude that FAST-SASA is orders of magnitude more efficient.

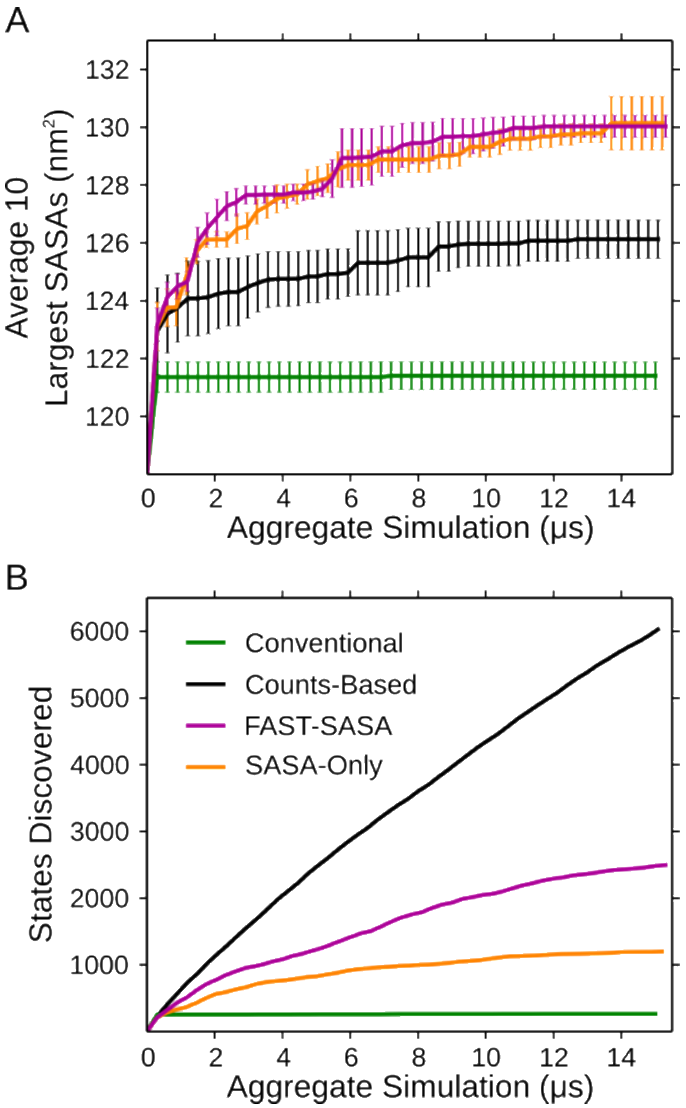

Figure 2.5: Performance of FAST-SASA (magenta) compared to conventional molecular dynamics (green), count-based sampling (black), and SASA-based sampling (orange). (A) The average of the solvent accessible surface areas for the 10 states with the largest surface areas discovered as a function of aggregate simulation time. (B) The number of states discovered as a function of the aggregate simulation time.

2.4.4 FAST-RMSD Efficiently Finds Paths Between Specific Structures

Computer simulations are also frequently employed to discover the transition paths between two distinct structures. As an example of this sort of problem, we sought to discover the preferred paths from the ligand-free crystal structure of β-lactamase discussed in the previous section to a structure with an unexpected allosteric binding pocket (1PZO[65]). To accelerate the discovery of such paths, we used FAST-RMSD to discover structures with low RMSDs to the target structure

and compared the performance of these simulations to conventional molecular dynamics simulations and counts-based adaptive sampling. All the trends are similar to those observed for FAST-SASA in comparison to other sampling methods, as shown in Figure 2.6. Combined with our analysis of synthetic trajectories, as described earlier, we conclude that FAST-RMSD quickly finds target structures and the preferred paths to these structures.

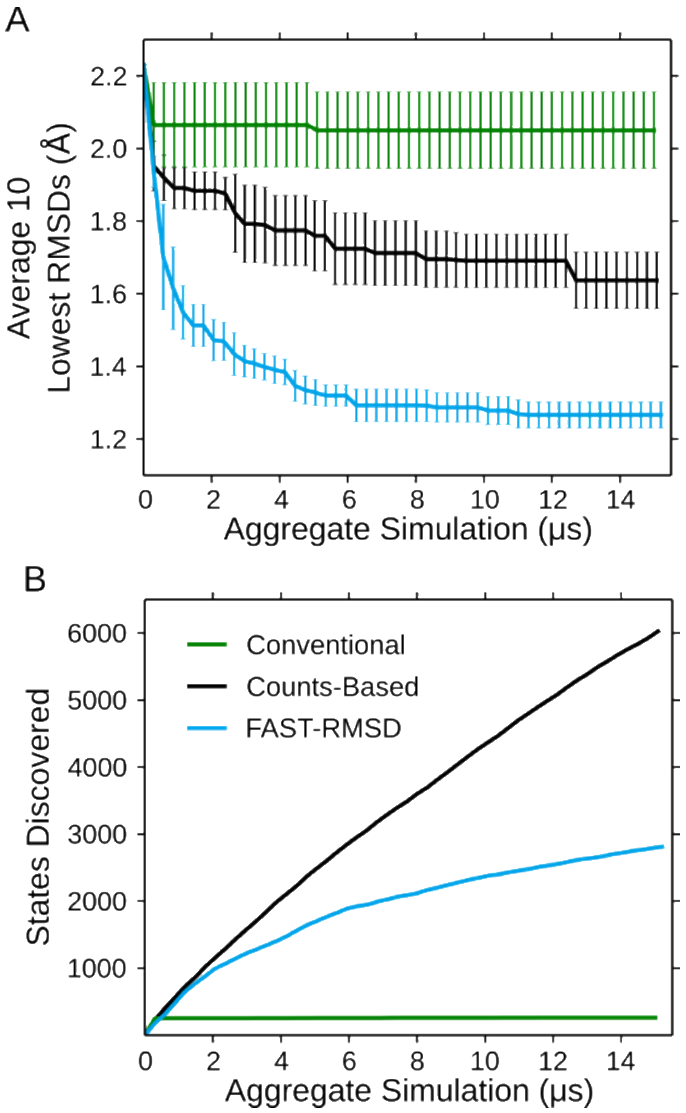

Figure 2.6: Performance of FAST-RMSD (cyan) compared to conventional molecular dynamics (green) and count-based sampling (black). (A) The average of the RMSD to the target structure for the 10 states with the lowest RMSDs discovered as a function of aggregate simulation time. (B) The number of states discovered as a function of the aggregate simulation time.

2.4.5 FAST-Energy Folds Proteins

As a final test of FAST, we applied it to the folding of a variant of the villin headpiece that folds in ~700 ns.[51] Inspired by the idea that proteins fold by following an energy gradient towards their native states, we chose to run FAST-energy to minimize the system's energy. This choice also allows *bona fide* structure predictions, rather than building in the answer with a method like FAST-RMSD. To make a direct comparison with a past study of this protein conducted on the Folding@home distributed computing environment,[52] the same simulation parameters and explicit solvent were used. However, the energies used in FAST's reward function were calculated using implicit solvent because water-water interactions will dominate the energy of any structure with explicit solvent. Implicit solvent, on the other hand, integrates out the water degrees of freedom, allowing FAST-energy to focus on finding preferred protein structures.

FAST-energy simulations fold villin to within 2.5 Å of its crystal structure in just 400 ns of aggregate simulation. Figure 2.7 state A shows the extended starting structure used for these simulations and Figure 2.7 state B shows the predicted structure overlaid on the crystal structure. This result is impressive because there is only an ~60% chance of folding the protein with 700 ns of conventional simulation based on the experimental folding time. Furthermore, the previous Folding@home study that inspired our FAST-energy calculations used 500 μs of conventional simulation[52] and a folding study run on the ANTON supercomputer used 125 μs of simulation.[26]

To understand the structural ensemble explored by FAST-energy, scatter plots of the energies of states from the MSM built from the FAST-energy data vs. their RMSDs to the crystal structure were overlaid with the same information from past Folding@home studies,[56] as shown in Figure 2.7. Overall, the model from FAST-energy covers a similar range of energies and RMSDs to that found by conventional molecular dynamics simulations. However, visual

inspection of the scatter plot suggests that FAST-energy finds more structures with both low energies and low RMSDs. This observation is further supported by the histograms of the energies and RMSDs for the structural states discovered by each method. Taken together, these results demonstrate that FAST successfully discovers the energetically accessible conformations that would eventually be found by conventional simulations but does so with much less simulation.

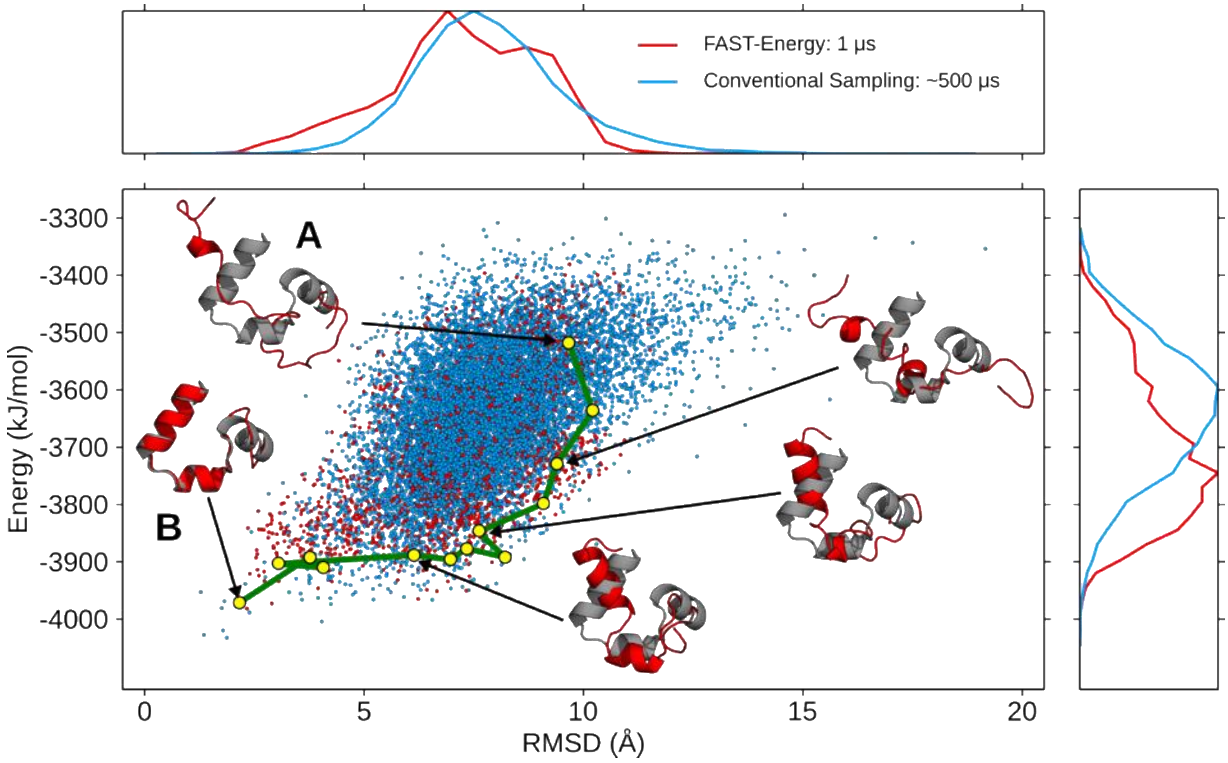

Figure 2.7: The state-space of villin projected onto two order-parameters: potential energy and the RMSD to the native crystal structure. Each point represents a single state discovered within 1 μs of FAST-Energy sampling (red) or 500 μs of unguided sampling from Folding@home (blue). Normalized histograms of the number of states with a given potential energy (right plot) or RMSD (top plot) are shown. The highest flux pathway from the unfolded starting state (state A) to the state with the lowest RMSD (state B) is plotted as a green line, where states along the pathway are identified with yellow points. Five conformations along the FAST-Energy folding pathway (red) are superimposed onto the native crystal structure (grey).

To see if FAST-energy finds similar folding routes to past studies or if the reward function used to choose starting structures for each round of simulation somehow biases the result, the preferred folding pathway from the final MSM was identified. This model ought to

capture the proper thermodynamics and kinetics of the states visited.[37,38] Indeed, the protein first forms some elements of secondary structure and begins to collapse, as observed in previous studies.[66,67] The N-terminal helix is also the last to form, in agreement with previous studies using the same force field.[68] Finally, the slowest implied timescale of the model was calculated as an estimate of the folding time. This calculation yielded a folding time of 830±260 ns, again in reasonable agreement with both experiment and past work using conventional molecular dynamics simulations. Therefore, we conclude that MSMs built from FAST simulations are indeed capable of capturing the proper thermodynamics and kinetics despite the fact that starting conformations are not selected according to a Boltzmann distribution.

2.5 Conclusions

We have introduced a goal-oriented sampling method, called FAST, which rapidly searches through conformational space for structures with desired properties by balancing exploration/exploitation tradeoffs. This algorithm was inspired by the hypothesis that many physical properties have an overall gradient in conformational space, akin to the energetic gradients that are known to guide proteins to their folded states. Indeed, retrospective analysis of existing MSMs supports the idea that structural properties like the RMSD to a target structure, the solvent accessible surface area, and the energy have such gradients. To follow these gradients, we designed FAST to balance between 1) recognizing and amplifying small motions that maximize (or minimize) a selected geometric function and 2) exploring poorly sampled regions of configuration space. This balance is achieved by leveraging ideas from optimization theory regarding exploration/exploitation tradeoffs.

To test FAST, we applied it to a number of common problems and compared its performance to alternative approaches, such as conventional molecular dynamics simulations

and counts-based adaptive sampling. For example, we demonstrated that FAST can find pockets by preferentially sampling structures with large surface areas, it can find paths between specific structures by minimizing the RMSD to a target, and it can fold proteins by minimizing their energies. In each case, FAST outperforms the methods that we compared it to by at least an order of magnitude, and likely considerably more. The success of FAST supports our hypothesis that many physical properties have gradients in conformational space. Moreover, our results demonstrate that FAST is capable of identifying and following these gradients, even overcoming and circumventing barriers that interrupt these trends. In addition to finding structures with a desired property more quickly than other algorithms, FAST also finds a greater diversity of such structures. While the data generated with FAST is not Boltzmann distributed, building an MSM from the data provides the proper thermodynamics and kinetics. The ability to obtain broad sampling while maintaining the proper kinetics is an important advantage over many other sampling algorithms that facilitates a direct connection with kinetic experiments. Therefore, we expect FAST to be of great utility for a wide range of applications. There are also many opportunities for combining FAST with other sampling methods. For example, one could use accelerated molecular dynamics to obtain even broader sampling, though this would sacrifice kinetics. One could also use FAST for state discovery and then refine estimates of the transition probabilities between states with adaptive sampling schemes designed to reduce statistical uncertainty.

Bibliography

(1) Israelewitz, B.; Gao, M.; Schulten, K. Steered Molecular Dynamics and Mechanical Functions of Proteins. *Current Opinion in Structural Biology* **2001**, *11* (2), 224–230.

(2) Laio, A.; Parrinello, M. Escaping Free-Energy Minima. *Proc. Natl. Acad. Sci. U.S.A.* **2002**, *99* (20), 12562–12566.

(3) Huber, T.; Torda, A. E.; van Gunsteren, W. F. Local Elevation: a Method for Improving the Searching Properties of Molecular Dynamics Simulation. *J Computer-Aided Mol Des* **1994**, *8* (6), 695–708.

(4) Maragliano, L.; Fischer, A.; Vanden-Eijnden, E.; Ciccotti, G. String Method in Collective Variables: Minimum Free Energy Paths and Isocommittor Surfaces. *The Journal of Chemical Physics* **2006**, *125* (2), 024106.

(5) Albert C Pan; Deniz Sezer, A.; Benoît Roux. *Finding Transition Pathways Using the String Method with Swarms of Trajectories*; American Chemical Society, 2008; Vol. 112, pp 3432–3440.

(6) MacCallum, J. L.; Perez, A.; Dill, K. A. Determining Protein Structures by Combining Semireliable Data with Atomistic Physical Models by Bayesian Inference. *Proc. Natl. Acad. Sci. U.S.A.* **2015**, *112* (22), 6985–6990.

(7) Lindorff-Larsen, K.; Best, R. B.; DePristo, M. A.; Dobson, C. M.; Vendruscolo, M. Simultaneous Determination of Protein Structure and Dynamics. *Nature 2003 423:6936* **2005**, *433* (7022), 128–132.

(8) Zheng, L.; Chen, M.; Yang, W. Random Walk in Orthogonal Space to Achieve Efficient Free-Energy Simulation of Complex Systems. *Proc. Natl. Acad. Sci. U.S.A.* **2008**, *105* (51), 20227–20232.

(9) Sugita, Y.; Okamoto, Y. Replica-Exchange Molecular Dynamics Method for Protein Folding. *Chemical Physics Letters* **1999**, *314* (1-2), 141–151.

(10) Hamelberg, D.; Mongan, J.; McCammon, J. A. Accelerated Molecular Dynamics: a Promising and Efficient Simulation Method for Biomolecules. *The Journal of Chemical Physics* **2004**, *120* (24), 11919–11929.

(11) Huber, G. A.; Kim, S. Weighted-Ensemble Brownian Dynamics Simulations for Protein Association Reactions. *Biophysical Journal* **1996**, *70* (1), 97–110.

(12) Dickson, A.; Charles L Brooks, I. WExplore: Hierarchical Exploration of High-Dimensional Spaces Using the Weighted Ensemble Algorithm. *J. Phys. Chem. B* **2014**, *118* (13), 3532–3542.

(13) Suárez, E.; Lettieri, S.; Zwier, M. C.; Stringer, C. A.; Subramanian, S. R.; Chong, L. T.; Zuckerman, D. M. Simultaneous Computation of Dynamical and Equilibrium Information Using a Weighted Ensemble of Trajectories. *J. Chem. Theory Comput.* **2014**, *10* (7), 2658–2667.

(14) Chen, Y.; Roux, B. Efficient Hybrid Non-Equilibrium Molecular Dynamics - Monte Carlo Simulations with Symmetric Momentum Reversal. *The Journal of Chemical Physics* **2014**, *141* (11), 114107.

(15) Hinrichs, N. S.; Pande, V. S. Calculation of the Distribution of Eigenvalues and Eigenvectors in Markovian State Models for Molecular Dynamics. *The Journal of Chemical Physics* **2007**, *126* (24), 244101.

(16) Bowman, G. R.; Ensign, D. L.; Pande, V. S. Enhanced Modeling via Network Theory: Adaptive Sampling of Markov State Models. *J. Chem. Theory Comput.* **2010**, *6* (3), 787–794.

(17) Doerr, S.; De Fabritiis, G. On-the-Fly Learning and Sampling of Ligand Binding by High-Throughput Molecular Simulations. *J. Chem. Theory Comput.* **2014**, *10* (5), 2064–2069.

(18) Bacci, M.; Vitalis, A.; Caflisch, A. A Molecular Simulation Protocol to Avoid Sampling Redundancy and Discover New States. *Biochimica et Biophysica Acta (BBA) - General Subjects* **2015**, *1850* (5), 889–902.

(19) Adhikari, A. N.; Freed, K. F.; Sosnick, T. R. Simplified Protein Models: Predicting Folding Pathways and Structure Using Amino Acid Sequences. *Physical Review Letters* **2013**, *111* (2), 028103.

(20) Voelz, V. A.; Elman, B.; Razavi, A. M.; Zhou, G. Surprisal Metrics for Quantifying Perturbed Conformational Dynamics in Markov State Models. *J. Chem. Theory Comput.* **2014**, *10* (12), 5716–5728.

(21) Moyano, G. E.; Collins, M. A. Molecular Potential Energy Surfaces by Interpolation: Strategies for Faster Convergence. *The Journal of Chemical Physics* **2004**, *121* (20), 9769–9775.

(22) Shirts, M. COMPUTING: Screen Savers of the World Unite! *Science* **2000**, *290* (5498), 1903–1904.

(23) Buch, I.; Harvey, M. J.; Giorgino, T.; Anderson, D. P.; De Fabritiis, G. High-Throughput All-Atom Molecular Dynamics Simulations Using Distributed Computing. *J. Chem. Inf. Model.* **2010**, *50* (3), 397–403.

(24) Shaw, D. E.; Bowers, K. J.; Chow, E.; Eastwood, M. P.; Ierardi, D. J.; Klepeis, J. L.; Kuskin, J. S.; Larson, R. H.; Lindorff-Larsen, K.; Maragakis, P.; Moraes, M. A.; Dror, R. O.; Piana, S.; Shan, Y.; Towles, B.; Salmon, J. K.; Grossman, J. P.; Mackenzie, K. M.; Bank, J. A.; Young, C.; Deneroff, M. M.; Batson, B. Millisecond-Scale Molecular Dynamics Simulations on Anton; ACM Press: New York, New York, USA, 2009; p 1.

(25) Voelz, V. A.; Bowman, G. R.; Beauchamp, K.; Pande, V. S. Molecular Simulation of Ab Initio Protein Folding for a Millisecond Folder NTL9(1−39). *Journal of the American Chemical Society* **2010**, *132* (5), 1526–1528.

(26) Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; Shaw, D. E. How Fast-Folding Proteins Fold. *Science* **2011**, *334* (6055), 517–520.

(27) Bowman, G. R.; Geissler, P. L. Equilibrium Fluctuations of a Single Folded Protein Reveal a Multitude of Potential Cryptic Allosteric Sites. *Proc. Natl. Acad. Sci. U.S.A.* **2012**, *109* (29), 11681–11686.

(28) Dror, R. O.; Green, H. F.; Valant, C.; Borhani, D. W.; Valcourt, J. R.; Pan, A. C.; Arlow, D. H.; Canals, M.; Lane, J. R.; Rahmani, R.; Baell, J. B.; Sexton, P. M.; Christopoulos, A.; Shaw, D. E. Structural Basis for Modulation of a G-Protein-Coupled Receptor by Allosteric Drugs. *Nature 2003 423:6936* **2013**, *503* (7475), 295–299.

(29) Kohlhoff, K. J.; Shukla, D.; Lawrenz, M.; Bowman, G. R.; Konerding, D. E.; Belov, D.; Altman, R. B.; Pande, V. S. Cloud-Based Simulations on Google Exacycle Reveal Ligand Modulation of GPCR Activation Pathways. *Nature Chemistry 2017 9:10* **2014**, *6* (1), 15–21.

(30) Onuchic, J. N.; and, Z. L.-S.; Wolynes, P. G. THEORY of PROTEIN FOLDING: the Energy Landscape Perspective. *http://dx.doi.org/10.1146/annurev.physchem.48.1.545* **2003**, *48* (1), 545–600.

(31) Brooks, C. L., III. Simulations of Protein Folding and Unfolding. *Current Opinion in Structural Biology* **1998**, *8* (2), 222–226.

(32) Dill, K. A.; Chan, H. S. From Levinthal to Pathways to Funnels. *Nature Structural & Molecular Biology 1997 4:1* **1997**, *4* (1), 10–19.

(33) Berry, D. A.; Hall, B. F. L. C. A.; 1985. Bandit Problems: Sequential Allocation of Experiments (Monographs on Statistics and Applied Probability). *Springer*.

(34) Poli, R.; Kennedy, J.; Blackwell, T. Particle Swarm Optimization. *Swarm Intell* **2007**, *1* (1), 33–57.

(35) Audibert, J.-Y.; Munos, R.; Szepesvári, C. Exploration–Exploitation Tradeoff Using Variance Estimates in Multi-Armed Bandits. *Theoretical Computer Science* **2009**, *410* (19), 1876–1902.

(36) Auer, P. Using Confidence Bounds for Exploitation-Exploration Trade-Offs. *Journal of Machine Learning Research* **2002**, *3* (Nov), 397–422.

(37) Huang, X.; Bowman, G. R.; Bacallado, S.; Pande, V. S. Rapid Equilibrium Sampling Initiated From Nonequilibrium Data. *Proc. Natl. Acad. Sci. U.S.A.* **2009**, *106* (47), 19765–19769.

(38) Noé, F.; Schütte, C.; Vanden-Eijnden, E.; Reich, L.; Weikl, T. R. Constructing the Equilibrium Ensemble of Folding Pathways From Short Off-Equilibrium Simulations. *Proc. Natl. Acad. Sci. U.S.A.* **2009**, *106* (45), 19011–19016.

(39) *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*; Springer Netherlands: Dordrecht, 2014.

(40) Noé, F. Probability Distributions of Molecular Observables Computed From Markov Models. *The Journal of Chemical Physics* **2008**, *128* (24), 244103.

(41) Bowman, G. R. Improved Coarse-Graining of Markov State Models via Explicit Consideration of Statistical Uncertainty. *The Journal of Chemical Physics* **2012**, *137* (13), 134111.

(42) Weber, J. K.; Pande, V. S. Characterization and Rapid Sampling of Protein Folding Markov State Model Topologies. *J. Chem. Theory Comput.* **2011**, *7* (10), 3405–3411.

(43) Van Der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E.; Berendsen, H. J. C. GROMACS: Fast, Flexible, and Free. *J Comput Chem* **2005**, *26* (16), 1701–1718.

(44) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High Performance Molecular Simulations Through Multi-Level Parallelism From Laptops to Supercomputers. *SoftwareX* **2015**, *1-2*, 19–25.

(45) Kollman, P. A. Advances and Continuing Challenges in Achieving Realistic and Predictive Simulations of the Properties of Organic and Biological Molecules. *Acc. Chem. Res.* **1996**, *29* (10), 461–469.

(46) Onufriev, A.; Bashford, D.; Case, D. A. Exploring Protein Native States and Large-Scale Conformational Changes with a Modified Generalized Born Model. *Proteins: Structure, Function, and Bioinformatics* **2004**, *55* (2), 383–394.

(47) Jelsch, C.; Mourey, L.; Masson, J. M.; Samama, J. P. Crystal Structure of Escherichia Coli TEM1 B-Lactamase at 1.8 Å Resolution. *Proteins: Structure, Function, and Bioinformatics* **1993**, *16* (4), 364–383.

(48) Hess, B. P-LINCS: a Parallel Linear Constraint Solver for Molecular Simulation. *J. Chem. Theory Comput.* **2007**, *4* (1), 116–122.

(49) Feenstra, K. A.; Hess, B.; Berendsen, H. J. C. Improving Efficiency of Large Time-Scale Molecular Dynamics Simulations of Hydrogen-Rich Systems. *J Comput Chem* **1999**, *20* (8), 786–798.

(50) Bussi, G.; Donadio, D.; Parrinello, M. Canonical Sampling Through Velocity Rescaling. *The Journal of Chemical Physics* **2007**, *126* (1), 014101.

(51) Kubelka, J.; Chiu, T. K.; Davies, D. R.; Eaton, W. A.; Hofrichter, J. Sub-Microsecond Protein Folding. *Journal of Molecular Biology* **2006**, *359* (3), 546–553.

(52) Ensign, D. L.; Kasson, P. M.; Pande, V. S. Heterogeneity Even at the Speed Limit of Folding: Large-Scale Molecular Dynamics Study of a Fast-Folding Variant of the Villin Headpiece. *Journal of Molecular Biology* **2007**, *374* (3), 806–816.

(53) DeLano, W. L. PyMOL. **2002**.

(54) Bowman, G. R.; Huang, X.; Pande, V. S. Using Generalized Ensemble Simulations and Markov State Models to Identify Conformational States. *Methods* **2009**, *49* (2), 197–201.

(55) Beauchamp, K. A.; Bowman, G. R.; Lane, T. J.; Maibaum, L.; Haque, I. S.; Pande, V. S. MSMBuilder2: Modeling Conformational Dynamics on the Picosecond to Millisecond Scale. *J. Chem. Theory Comput.* **2011**, *7* (10), 3412–3419.

(56) Bowman, G. R.; Beauchamp, K. A.; Boxer, G.; Pande, V. S. Progress and Challenges in the Automated Construction of Markov State Models for Full Protein Systems. *The Journal of Chemical Physics* **2009**, *131* (12), 124101.

(57) Prinz, J.-H.; Wu, H.; Sarich, M.; Keller, B.; Senne, M.; Held, M.; Chodera, J. D.; Schütte, C.; Noé, F. Markov Models of Molecular Kinetics: Generation and Validation. *The Journal of Chemical Physics* **2011**, *134* (17), 174105.

(58) and, W. C. S.; Pitera, J. W.; Suits, F. *Describing Protein Folding Kinetics by Molecular Dynamics Simulations. 1. Theory†*; American Chemical Society, 2004; Vol. 108, pp 6571–6581.

(59) Hendlich, M.; Rippmann, F.; Barnickel, G. LIGSITE: Automatic and Efficient Detection of Potential Small Molecule-Binding Sites in Proteins. *Journal of Molecular Graphics and Modelling* **1997**, *15* (6), 359–363.

(60) McGibbon, R. T.; Beauchamp, K. A.; Harrigan, M. P.; Klein, C.; Swails, J. M.; Hernández, C. X.; Schwantes, C. R.; Wang, L.-P.; Lane, T. J.; Pande, V. S. MDTraj: a Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophysical Journal* **2015**, *109* (8), 1528–1532.

(61) E, W.; Vanden-Eijnden, E. Towards a Theory of Transition Paths. *J Stat Phys* **2006**, *123* (3), 503–523.

(62) Metzner, P.; Schütte, C.; Vanden-Eijnden, E. Transition Path Theory for Markov Jump Processes. *Multiscale Modeling & Simulation* **2009**, *7* (3), 1192–1219.

(63) Bowman, G. R.; Voelz, V. A.; Pande, V. S. Atomistic Folding Simulations of the Five-Helix Bundle Protein Λ6−85. *Journal of the American Chemical Society* **2010**, *133* (4), 664–667.

(64) Bowman, G. R.; Bolin, E. R.; Hart, K. M.; Maguire, B. C.; Marqusee, S. Discovery of Multiple Hidden Allosteric Sites by Combining Markov State Models and Experiments. *Proc. Natl. Acad. Sci. U.S.A.* **2015**, *112* (9), 2734–2739.

(65) Horn, J. R.; Shoichet, B. K. Allosteric Inhibition Through Core Disruption. *Journal of Molecular Biology* **2004**, *336* (5), 1283–1291.

(66) Duan, Y.; Wang, L.; Kollman, P. A. The Early Stage of Folding of Villin Headpiece Subdomain Observed in a 200-Nanosecond Fully Solvated Molecular Dynamics Simulation. *Proc. Natl. Acad. Sci. U.S.A.* **1998**, *95* (17), 9897–9902.

(67) Bowman, G. R.; Pande, V. S. Protein Folded States Are Kinetic Hubs. *Proc. Natl. Acad. Sci. U.S.A.* **2010**, *107* (24), 10890–10895.

(68) Piana, S.; Lindorff-Larsen, K.; Shaw, D. E. How Robust Are Protein Folding Simulations with Respect to Force Field Parameterization? *Biophysical Journal* **2011**, *100* (9), L47–L49.

Chapter 3

How to Run FAST Simulations

3.1 Preamble

This chapter is adapted from the following article: Zimmerman, M.I. and Bowman, G.R. (2016). "How to Run FAST Simulations", *Methods in Enzymology*, 578, 213-225

3.2 Introduction

One of the largest challenges in using molecular dynamics simulations to study enzymes is achieving adequate sampling to accurately represent its equilibrium structural ensemble and conformational transitions.[1,2] In other words, the conformational space of a protein is extraordinarily large and transitions between two given conformations may be separated by numerous kinetically slow steps that require a great deal of simulation-time to observe. To put this into perspective, many enzymatic reactions/conformational transitions occur on the millisecond-second timescale, but a typical desktop computer may only be able to simulate a few nanoseconds of dynamics per day. Therefore, it could take a desktop computer hundreds to millions of years to simulate a particular event.

One approach to overcoming the limitations of MD simulations is to build specialized supercomputers. For example, the development of powerful purpose-built hardware for MD simulations, such as the ANTON supercomputer, allows for much longer timescale simulations.[3] However, this approach is typically too expensive for the average researcher.

An alternative approach is to run many short timescale simulations on different computers. A single, long simulation will eventually generate multiple independent samples of rare events. Running multiple simulations can capture the same independent events in parallel. Running simulations in parallel maximizes the use of commodity hardware since obtaining larger aggregate simulation times can be easily achieved through the addition of processors rather than increasing a processors speed. For these reasons, massively parallelized distributed computing projects, such as Folding@home, have been very successful in using MD to capture long timescale conformational transitions of proteins such as folding and allostery.[4-10]

Markov state models (MSMs) provide an elegant framework for analyzing protein simulation data, whether it is generated by a single simulation or many of them.[11-13] An MSM is essentially a map of the different conformations a protein adopts. The basic construction of an MSM consists of the following steps: 1) cluster all of the simulation data into discrete "microstates", for example with a k-centers clustering algorithm based on the protein backbone, 2) generate a transition count matrix, an N x N matrix of all the observed transitions from microstate i to j for a specified lag time (i.e. observation interval), 3) and generate a transition probability matrix, an N x N matrix created from the transition count matrix detailing the probability of transitioning from state i to state j. The transition probability matrix contains a wealth of information. For example, the first eigenvector of this matrix specifies the equilibrium probabilities of all the states. Other eigenvalues and eigenvectors specify the rates of transitioning between different sets of states and which states are involved. One powerful attribute of MSMs is that it is equally valid to build them from a single long simulation or a set of simulations, even if these simulations are not initiated from a Boltzmann distribution.

Furthermore, freely available software packages, such as MSMBuilder and pyEMMA, provide a readily accessible means to construct and analyze these models.[14-16]

MSMs' ability to extract the equilibrium thermodynamics and kinetics of a system irrespective of the distribution of starting conformations for a set of simulations opens the doors to interactively sample desired regions of conformational space. For example, adaptive sampling algorithms iteratively run a set of simulations, build an MSM, and then select starting points for new simulations that will help to reduce statistical uncertainty in the model.[17-23] Various adaptive sampling schemes have been developed to enhance and automate the construction of MSMs. For example, Hinrichs and Pande have developed an adaptive sampling scheme that spawns new simulations from the states that contribute most to the statistical uncertainty in an MSMs principle eigenvectors and eigenvalues.[18] Other methods spawn simulations from states based on the number of times they have been observed or the number of neighbors they are connected to in order to discover new states more quickly.[19] These methods will generally explore conformational space more efficiently than brute force simulations. However, they will not necessarily sample specific events of interest to a researcher before thoroughly exploring other, less relevant regions of conformational space.

We have developed a goal-oriented sampling algorithm, called Fluctuation Amplification of Specific Traits (FAST), which draws inspiration from adaptive sampling and the multi-armed bandit problem to efficiently identify structures with a desired physical property.[24] For example, FAST can be used to identify the preferred pathways between active and inactive states of an enzyme or it can be used to identify potentially druggable pockets that are not apparent from existing crystal structures. The FAST algorithm achieves this by balancing between exploiting promising structures (i.e. searching around promising solutions for even better ones) and broad

exploration (i.e. searching unexplored regions of conformational space for entirely new solutions). The following sections of this chapter provide details on the algorithm and the parameters relevant to setting up FAST simulations.

3.3 FAST Algorithm

The FAST algorithm can be used to find structures that optimize any geometric function (ϕ) of protein conformations. At the heart of the FAST-ϕ algorithm is the reward function used to decide which states to simulate for future runs of sampling. The FAST-ϕ reward function is modeled after a simple solution to the multi-armed bandit problem,

$$r_{\phi}(i) = \bar{\phi}(i) + \alpha\bar{\psi}(i)$$

where the reward (r_{ϕ}) for state *i* is the sum of a directed component, $\bar{\phi}(i)$, and an undirected component, $\bar{\psi}(i)$, with scaling parameter α. The set of directed components correspond to a feature-scaled list of traits that one wishes to exploit (such as the RMSD to a target structure) and the set of undirected components correspond to a feature-scaled list of some statistical function that facilitates state-space exploration (such as the number of observations per state). Feature-scaling transforms a list of values to range from 0 to 1. Directed and undirected components to the FAST ranking can be either positively feature-scaled to favor large values

$$\bar{\phi}(i) = \frac{\phi(i) - \phi_{min}}{\phi_{max} - \phi_{min}}$$

or negatively feature-scaled to favor small values

$$\bar{\phi}(i) = \frac{\phi_{max} - \phi(i)}{\phi_{max} - \phi_{min}}$$

The variables ϕ_{min} and ϕ_{max} are the minimum and maximum values of $\phi(i)$ observed in a simulation dataset, respectively.

Although the reward function may change for the specific type of FAST-ϕ sampling, the basic algorithm remains the same:

(1) Start a swarm of *N* simulations from a structure or set of structures, such as one or more known crystal structures,

(2) Cluster all the simulation data collected so far into discrete conformational states. This can be accomplished by using a k-centers algorithm on the RMSD between select protein atoms (such as backbone heavy-atoms), with a specified distance cutoff. The distance cutoff will specify the maximum distance between structures in a cluster to the cluster center and will dictate the total number of clusters generated.

(3) Rank all of the states discovered using the FAST-ϕ reward function.

(4) Start a new swarm of simulations from the top *N* structures that maximize the FAST-ϕ reward function.

(5) Repeat steps 2-4 until some convergence criterion is met or a predetermined amount of simulation has been conducted.

(6) Build an MSM from the final dataset to capture the proper thermodynamics and kinetics, thereby correcting for any bias introduced by selecting starting conformations from each swarm of simulations according to our reward function instead of a Boltzmann distribution.

As mentioned, the directed and undirected components to the ranking can vary depending on the specific problem at hand. Specific traits for the directed component of FAST sampling will be discussed in a later section. In early applications of the FAST-ϕ reward function, the undirected component was chosen to be the negatively feature-scaled number of observations of each state

$$\bar{\psi}(i) = \frac{C_{max} - C(i)}{C_{max} - C_{min}}$$

where C_{min} and C_{max} are the minimum and maximum number of observations of any state, respectively. This version of the undirected component was selected based on a simple Bayesian model that suggests it should maximize the discovery of new states. One could also use alternative statistical measures, such as existing adaptive sampling methods, in place of our counts-based, undirected component.

3.4 FAST Sampling Parameters

The FAST algorithm contains many parameters, in the form of input and output, which can be reduced to those that are relevant for running molecular dynamics simulations, building MSMs, or propagating a run of goal-oriented sampling. With the large number of parameters required for running FAST, it can be a daunting task to set up expensive simulations without a good feel for reasonable values. In this section, we will detail some of the main parameters that are used in FAST simulations, how to determine reasonable values, and how they may interact with one another. For the sake of brevity and clarity, parameters relevant to running individual molecular

dynamics simulations will not be discussed; there are many software packages that can perform these simulations that provide extensive tutorials and user manuals, such as Amber, CHARMM, Gromacs, and NAMD.[25-28]

3.4.1 Number of Runs

Typically one will run FAST sampling until some convergence criterion is achieved. In some circumstances this is very straightforward, although a convergence criteria is not always easy to deduce. Running FAST simulations from a starting structure to a specified target (e.g. using FAST-RMSD between known conformations) will produce a simple convergence criterion since there is a single end state; simulations can be terminated once the end state is discovered. On the other hand, there may not be obvious criteria for terminating a set of FAST simulations for more open-ended problems, such as searching for conformations with large solvent accessible surface areas (SASAs) using FAST-SASA as a heuristic for discovering unknown druggable pockets. In the case of FAST-SASA, one might want to stop simulations when the solvent-accessible surface-area ceases to increase as rounds continue, but we have shown that in this scenario the undirected component to the FAST-SASA reward function will aid in the discovery of multiple pathways to large SASA states, which is desirable because a diversity of potential druggable sites can be discovered. In practice, it is convenient to run simulations for a specified number of runs and continue the runs if sampling is deemed insufficient, since the algorithm is easy to restart from a previous run or preexisting set of data.

3.4.2 The α Scaling Parameter

The scaling parameter, α, is used in the FAST-ϕ reward function to weight the relative importance of exploiting physical traits and increasing state-exploration. Large α values will

increase the exploration of state space by favoring states that have not been observed as frequently, whereas smaller values will place more emphasis on exploiting structures with promising traits. Emphasizing the trait-based component of the reward function will increase the likelihood FAST tries to hop over larger energy barriers rather than try new solutions. Through the analysis of synthetic trajectories generated with existing MSMs, we have seen that sampling results are largely insensitive to values of α between 0.5-1.5. However, it is possible that the energy-landscape of the protein being simulated, as well as the gradient of conformational-space that one is attempting to follow, may change this observation. If traits are very monotonically increasing/decreasing it is expected that smaller values of α will optimize FAST-ϕ's performance, whereas if traits require significant backtracking, larger values of α will optimize FAST-ϕ's performance. For these reasons, unless one has special insight into the nature of a particular protein's energy landscape, an $\alpha = 1$ is a safe choice.

3.4.3 Number of Simulations Per Run

The number of simulations to perform during each run of FAST sampling is an important decision to maximize computational resources, ensure a good swath of conformations, and accelerate the observation of rare-events. A main advantage to running simulations in parallel over generating a single trajectory is that many parallel jobs on multiple processors can be efficiently used to generate sizeable datasets, thus, the biggest factor in selecting the number of simulations per run is attempting to generate the largest aggregate simulation time with the resources available. Additionally, more simulations per run allows for a greater spread of starting states that will identify a diversity of potential pathways to explore, which will better allow for the circumvention of dead-ends. Despite this improvement, the number of simulations should be balanced with the individual simulation lengths so that there is a reasonable amount of aggregate

simulation time per run; having too much aggregate simulation per run means that there will be less total runs and less FAST-enhancement. As an example, if one wishes to observe a process that takes 1 μs of simulation to observe, using 40 simulations per run with 10 ns lengths would generate 400 ns of aggregate simulation per run, meaning that after 3 runs the aggregate simulation is much larger than the expected mean first passage time. Alternatively, 10 simulations per run of 10 ns lengths would take 10 runs to total 1 μs, which will provide more FAST-enhancement by offering extra chances to adaptively explore conformational space.

3.4.4 Simulation Length

Individual simulations must be longer than the Markov time for the final MSM. If simulations are shorter than this timescale, then the final model will violate the Markov assumption and be of little utility. However, one also wants simulations to be as short as possible to maximize the number of different runs that can be performed and to prevent simulations from wandering far from the region of conformational space one hopes to explore. In practice, we have often found that simulation lengths between 10 and 20 ns satisfy these constraints, in large part because models with a Markov time much greater than this would often be insufficient for the applications we have pursued.

3.4.5 Atom Indices Used for Clustering

The atomic indices that are used to cluster simulation data with a specified method into discrete microstates are the core of how states are defined. When clustering simulation data in-between runs of FAST sampling it is important to recognize that the structures within a state are similar only in the atomic indices specified for clustering. Usually it is beneficial to cluster conformations in a holistic fashion, based on the backbone heavy atoms (C_α, C_β, CO, N, and O),

so that different clusters represent global changes in a protein's conformation. However, situations arise where one is interested in an aspect of a protein structure, and that using the entire protein backbone for clustering would include unnecessary detail that drowns out the relevant structural motions. For example, we used FAST-RMSD to study the transition between apo and holo conformations of TEM-1 β-lactamase to understand how a surprising cryptic pocket opens up. These conformations have a global RMSD of ~0.26 Å, so an extremely high-resolution model would be required if the data were clustered based on a global RMSD.[29,30] To avoid an unnecessarily large number of clusters, we chose to instead cluster the data based on just the atoms of the two helices surrounding the pocket we were interested in.

3.4.6 Resolution of Clustering

One must balance between having enough clusters to resolve valuable differences but not so many as to make the statistical component of the reward function ineffective. For example, we have previously used a k-centers clustering algorithm that continues to divide conformational space into smaller groups until the maximum distance from any structure to its cluster center is less than a predetermined distance-cutoff. This distance-cutoff controls the level of structural similarity within and between clusters as well as the total number of clusters created during each run of sampling. Large distance-cutoffs will generate fewer clusters with many structures per cluster, whereas a small distance cutoff will generate numerous clusters with few structures per cluster. A good distance-cutoff value will be small enough that a structure pulled from a cluster will be an accurate representation of other structures in that cluster, but also large enough that the number of observations of each state reflects the sampling for that region in conformational space. If a distance-cutoff is particularly small, there may be many clusters with only a single conformation, which the FAST undirected component will rank extraordinarily highly. This is

not desirable if it is because extremely similar states are falsely considered separate. On the other hand, if the clusters are too coarse, then one may miss a valuable region of conformational space that FAST's reward function would otherwise discover.

3.5 Applications

FAST-ϕ can be tailored to provide pertinent thermodynamic and kinetic information for any region in conformational space that can be identified with a calculable order parameter. The central hypothesis of FAST is that gradients exist in conformational space with respect to specific traits and that they can be exploited through sampling states with large or small values of some trait that one wishes to maximize or minimize respectively. Although there are limitless possibilities for the directed component to the FAST-ϕ ranking, we will discuss a few that have been used to study enzyme function and structural ensembles.

FAST-SASA aims to uncover structural states of enzymes with large-SASA under the assumption that a large-SASA state will likely have large pocket openings that can be used to discover or design novel therapeutics. While enzymatic function is generally critical for cellular and biological processes, enzymatic reactions can be detrimental to human health. As an example, the enzyme TEM-1 β-lactamase is produced in certain bacteria as a means of hydrolyzing β-lactam antibiotics to confer antibiotic resistance.[31] Antibiotic resistant bacteria are swiftly becoming a global health concern due to the overuse of antibiotic treatments, so it is desired to find molecular ways to inhibit the antibiotic resistant nature of β-lactamase.[32] If complete atomistic structures exist where the enzyme has a large pocket opening, computational docking of small molecules to this region can aid in the discovery of ligands that will inhibit its function.[33] While the crystal structure of β-lactamase has a single large pocket (its active site), there is little diversity in locations to dock small molecules against; multiple pocket openings

will increase the likelihood of successful docking. Fortunately, proteins are not static and pockets will emerge during the course of an MD simulation.[9] FAST-SASA will accelerate the observation of large pocket openings by favoring states with an already large SASA, as we have previously shown for the enzyme β-lactamase.[8,24] We foresee that FAST-SASA, or related FAST-ϕ algorithms that more quantitatively detail global or specific pocket volumes, will be an invaluable tool for discovering druggable sites on enzymes that do not display obvious pockets in crystal structures.

FAST-RMSD is intended to reveal the equilibrium conformational transition pathway between two known enzyme structures with accurate thermodynamics and kinetics. Enzymes are dynamic proteins that often transition between many conformations that are relevant to their biological function.[34-36] Knowledge of their transition pathway, along with their kinetic rates and conformational free energies, can provide significant insight into their mechanisms of action and intrinsic regulation. It is often the case that structural studies of enzymes will identify multiple conformational populations, although will be unable to detail the structural intermediates between them. For example, nuclear magnetic resonance spectra may identify conformational heterogeneity through analysis of chemical shifts, although intermediates between populations are too short lived or have too small a population to observe.[37] Additionally, crystallographic studies may detail enzyme structures in various substrate-binding conformations, but they will not reveal the relative populations of these conformations. Given two atomically detailed structures as input, a start and a target, FAST-RMSD can efficiently identify the equilibrium transition pathway between them by biasing the starting structures of simulations originally spawned from the start towards the target. As mentioned earlier, if the conformational change that one is attempting to observe takes place for a portion of the total protein, it is beneficial to

define states based solely on the atom-indices of that region. Additionally, all RMSDs that are used in the reward function should be confined to this region of the protein. By doing this, the global changes to the protein will not wash away the observation of (in terms of RMSD values) relevant conformational transitions.

Clever use of the FAST-ϕ reward function can also provide valuable structural information in cases where experiments suggest a conformational transition but are unable to produce an atomistic description of the relevant structures. For example, a FRET experiment could provide a low-resolution view of a conformational change that occurs in some enzyme. Without an all-atom representation of a target structure, FAST-RMSD would be unable to elucidate the nature of the conformational change. Despite this, directed components to the reward function can be deduced that will explain these data. For example, a FAST-distance algorithm can be devised to favor transitions from some known structure, say where the dyes in the FRET study would be far apart, to new structures where the dyes would be brought together:

$$\bar{\phi}(i) = \frac{d_{max} - d(i)}{d_{max} - d_{min}}$$

where d is the distance between the dyes. The resulting model could then be used to help explain the origins of the experimental observation and to plan new experiments.

Bibliography

(1) Zwier, M. C.; Chong, L. T. Reaching Biological Timescales with All-Atom Molecular Dynamics Simulations. *Curr Opin Pharmacol* **2010**, *10* (6), 745–752.

(2) Dror, R. O.; Dirks, R. M.; Grossman, J. P.; Xu, H.; Shaw, D. E. Biomolecular Simulation: a Computational Microscope for Molecular Biology. *http://dx.doi.org/10.1146/annurev-biophys-042910-155245* **2012**, *41* (1), 429–452.

(3) Shaw, D. E.; Deneroff, M. M.; Dror, R. O.; Kuskin, J. S.; Larson, R. H.; Salmon, J. K.; Young, C.; Batson, B.; Bowers, K. J.; Chao, J. C.; Eastwood, M. P.; Gagliardo, J.; Grossman, J. P.; Ho, C. R.; Ierardi, D. J.; Kolossváry, I.; Klepeis, J. L.; Layman, T.; McLeavey, C.; Moraes, M. A.; Mueller, R.; Priest, E. C.; Shan, Y.; Spengler, J.; Theobald, M.; Towles, B.; Wang, S. C. Anton, a Special-Purpose Machine for Molecular Dynamics Simulation. *Communications of the ACM* **2008**, *51* (7), 91–97.

(4) Shirts, M. COMPUTING: Screen Savers of the World Unite! *Science* **2000**, *290* (5498), 1903–1904.

(5) Pande, V. S.; Baker, I.; Chapman, J.; Elmer, S. P.; Khaliq, S.; Larson, S. M.; Rhee, Y. M.; Shirts, M. R.; Snow, C. D.; Sorin, E. J.; Zagrovic, B. Atomistic Protein Folding Simulations on the Submillisecond Time Scale Using Worldwide Distributed Computing. *Biopolymers* **2003**, *68* (1), 91–109.

(6) Voelz, V. A.; Bowman, G. R.; Beauchamp, K.; Pande, V. S. Molecular Simulation of Ab Initio Protein Folding for a Millisecond Folder NTL9(1−39). *Journal of the American Chemical Society* **2010**, *132* (5), 1526–1528.

(7) Bowman, G. R.; Pande, V. S. Protein Folded States Are Kinetic Hubs. *Proc. Natl. Acad. Sci. U.S.A.* **2010**, *107* (24), 10890–10895.

(8) Bowman, G. R.; Geissler, P. L. Equilibrium Fluctuations of a Single Folded Protein Reveal a Multitude of Potential Cryptic Allosteric Sites. *Proc. Natl. Acad. Sci. U.S.A.* **2012**, *109* (29), 11681–11686.

(9) Bowman, G. R.; Bolin, E. R.; Hart, K. M.; Maguire, B. C.; Marqusee, S. Discovery of Multiple Hidden Allosteric Sites by Combining Markov State Models and Experiments. *Proc. Natl. Acad. Sci. U.S.A.* **2015**, *112* (9), 2734–2739.

(10) Plattner, N.; Noé, F. Protein Conformational Plasticity and Complex Ligand-Binding Kinetics Explored by Atomistic Simulations and Markov Models. *Nature Communications* **2015**, *6* (1), 7653.

(11) *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*; Springer Netherlands: Dordrecht, 2014.

(12) Chodera, J. D.; Noé, F. Markov State Models of Biomolecular Conformational Dynamics. *Current Opinion in Structural Biology* **2014**, *25*, 135–144.

(13) Pande, V. S.; Beauchamp, K.; Bowman, G. R. Everything You Wanted to Know About Markov State Models but Were Afraid to Ask. *Methods* **2010**, *52* (1), 99–105.

(14) Scherer, M. K.; Trendelkamp-Schroer, B.; Paul, F.; Pérez-Hernández, G.; Hoffmann, M.; Plattner, N.; Wehmeyer, C.; Prinz, J.-H.; Noé, F. PyEMMA 2: a Software Package for Estimation, Validation, and Analysis of Markov Models. *J. Chem. Theory Comput.* **2015**, *11* (11), 5525–5542.

(15) Bowman, G. R.; Huang, X.; Pande, V. S. Using Generalized Ensemble Simulations and Markov State Models to Identify Conformational States. *Methods* **2009**, *49* (2), 197–201.

(16) Beauchamp, K. A.; Bowman, G. R.; Lane, T. J.; Maibaum, L.; Haque, I. S.; Pande, V. S. MSMBuilder2: Modeling Conformational Dynamics on the Picosecond to Millisecond Scale. *J. Chem. Theory Comput.* **2011**, *7* (10), 3412–3419.

(17) Bowman, G. R.; Ensign, D. L.; Pande, V. S. Enhanced Modeling via Network Theory: Adaptive Sampling of Markov State Models. *J. Chem. Theory Comput.* **2010**, *6* (3), 787–794.

(18) Hinrichs, N. S.; Pande, V. S. Calculation of the Distribution of Eigenvalues and Eigenvectors in Markovian State Models for Molecular Dynamics. *The Journal of Chemical Physics* **2007**, *126* (24), 244101.

(19) Weber, J. K.; Pande, V. S. Characterization and Rapid Sampling of Protein Folding Markov State Model Topologies. *J. Chem. Theory Comput.* **2011**, *7* (10), 3405–3411.

(20) Doerr, S.; De Fabritiis, G. On-the-Fly Learning and Sampling of Ligand Binding by High-Throughput Molecular Simulations. *J. Chem. Theory Comput.* **2014**, *10* (5), 2064–2069.

(21) Adhikari, A. N.; Freed, K. F.; Sosnick, T. R. Simplified Protein Models: Predicting Folding Pathways and Structure Using Amino Acid Sequences. *Physical Review Letters* **2013**, *111* (2), 028103.

(22) Voelz, V. A.; Elman, B.; Razavi, A. M.; Zhou, G. Surprisal Metrics for Quantifying Perturbed Conformational Dynamics in Markov State Models. *J. Chem. Theory Comput.* **2014**, *10* (12), 5716–5728.

(23) Bacci, M.; Vitalis, A.; Caflisch, A. A Molecular Simulation Protocol to Avoid Sampling Redundancy and Discover New States. *Biochimica et Biophysica Acta (BBA) - General Subjects* **2015**, *1850* (5), 889–902.

(24) Zimmerman, M. I.; Bowman, G. R. FAST Conformational Searches by Balancing Exploration/Exploitation Trade-Offs. *J. Chem. Theory Comput.* **2015**, *11* (12), 5747–5757.

(25) Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. The Amber Biomolecular Simulation Programs. *J Comput Chem* **2005**, *26* (16), 1668–1688.

(26) Brooks, B. R.; Brooks, C. L.; Mackerell, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S.; Caflisch, A.; Caves, L.; Cui, Q.; Dinner, A. R.; Feig, M.; Fischer, S.; Gao, J.; Hodoscek, M.; Im, W.; Kuczera, K.; Lazaridis, T.; Ma, J.; Ovchinnikov, V.; Paci, E.; Pastor, R. W.; Post, C. B.; Pu, J. Z.; Schaefer, M.; Tidor, B.; Venable, R. M.; Woodcock, H. L.; Wu, X.; Yang, W.; York, D. M.; Karplus, M. CHARMM: the Biomolecular Simulation Program. *J Comput Chem* **2009**, *30* (10), 1545–1614.

(27) Van Der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E.; Berendsen, H. J. C. GROMACS: Fast, Flexible, and Free. *J Comput Chem* **2005**, *26* (16), 1701–1718.

(28) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. Scalable Molecular Dynamics with NAMD. *J Comput Chem* **2005**, *26* (16), 1781–1802.

(29) Jelsch, C.; Mourey, L.; Masson, J. M.; Samama, J. P. Crystal Structure of Escherichia Coli TEM1 B-Lactamase at 1.8 Å Resolution. *Proteins: Structure, Function, and Bioinformatics* **1993**, *16* (4), 364–383.

(30) Horn, J. R.; Shoichet, B. K. Allosteric Inhibition Through Core Disruption. *Journal of Molecular Biology* **2004**, *336* (5), 1283–1291.

(31) Proliferation and Significance of Clinically Relevant B-Lactamases. **2013**, *1277* (1), 84–90.

(32) Laxminarayan, R.; Duse, A.; Wattal, C.; Zaidi, A. K. M.; Wertheim, H. F. L.; Sumpradit, N.; Vlieghe, E.; Hara, G. L.; Gould, I. M.; Goossens, H.; Greko, C.; So, A. D.; Bigdeli, M.; Tomson, G.; Woodhouse, W.; Ombaka, E.; Peralta, A. Q.; Qamar, F. N.; Mir, F.; Kariuki, S.; Bhutta, Z. A.; Coates, A.; Bergstrom, R.; Wright, G. D.; Brown, E. D.; Cars, O. Antibiotic Resistance—the Need for Global Solutions. *The Lancet Infectious Diseases* **2013**, *13* (12), 1057–1098.

(33) Jones, G.; Willett, P.; Glen, R. C.; Leach, A. R.; Taylor, R. Development and Validation of a Genetic Algorithm for Flexible Docking. *Journal of Molecular Biology* **1997**, *267* (3), 727–748.

(34) Elber, R.; Karplus, M. Multiple Conformational States of Proteins: a Molecular Dynamics Analysis of Myoglobin. *Science* **1987**, *235* (4786), 318–321.

(35) Benkovic, S. J.; Hammes-Schiffer, S. A Perspective on Enzyme Catalysis. *Science* **2003**, *301* (5637), 1196–1202.

(36) Garcia-Viloca, M.; Gao, J.; Karplus, M.; Truhlar, D. G. How Enzymes Work: Analysis by Modern Rate Theory and Computer Simulations. *Science* **2004**, *303* (5655), 186–195.

(37) Kleckner, I. R.; Foster, M. P. An Introduction to NMR-Based Approaches for Measuring Protein Dynamics. *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics* **2011**, *1814* (8), 942–968.

Chapter 4

Choice of Adaptive Sampling Strategy Impacts State Discovery, Transition Probabilities, and the Apparent Mechanism of Conformational Changes

4.1 Preamble

This chapter is adapted from the following article: Zimmerman, M.I., Porter, J.R., Sun, Xianquiang, S., Silva, R.R., and Bowman, G.R. (2018). "Choice of Adaptive Sampling Strategy Impacts State Discovery, Transition Probabilities, and the Apparent Mechanism of Conformational Change", *Journal of Chemical Theory and Computation*, 14 (11), 5459-5475

4.2 Introduction

The use of all-atom molecular dynamics (MD) simulations for long time-scale phenomena are often thwarted by insufficient computational resources. Many interesting biological processes occur on the millisecond to second timescale, where a single simulation may take longer than a lifetime to gather. Notable attempts to alleviate hardware limitations are the purpose-built ANTON supercomputers.[1,2] These supercomputers are an engineering feat, yet are still limited by sampling and not accessible to many researchers. Since increasing commodity hardware performance by many orders of magnitude is not likely in the immediate future, the observation of interesting biological phenomena requires the use of clever sampling techniques.

A common technique to increase the observation of long time-scale phenomena is to alter the underlying energy landscape. These methods aim to guide a simulation towards some end goal or toward the exploration of a set of order parameters. Some examples include, Gō models,[3,4] replica-exchange,[5-7] steered MD,[8,9] accelerated MD,[10,11] meta-dynamics,[12,13] among others.[14-16] Unfortunately, these methods do not capture proper kinetic information, and can traverse unrealistically high energy barriers. Here, we are particularly interested in sampling methods that access long time-scale phenomena without perturbing the underlying energy landscape, such that both thermodynamic and kinetic properties can be inferred.

As an alternative to a single long simulation, many independent simulations can be run in parallel. Combined, these parallel simulations tractably capture time-scales longer than any single simulation. To illustrate: if we assume that the transition between conformational states A and B follows a Poisson process, the probability of observing a transition to state B is dependent only on the aggregate simulation time from A, not the length of each simulation.[17] Put another way, the probability of traversing a single energy barrier is based on the number of attempts to cross that barrier, regardless of whether they are in parallel or successive. Thus, parallel simulations may offer a significant enhancement in the observation of rare events, since it is usually easier to add more computational resources than to make them faster. This is the strategy of Folding@home, which takes advantage of around 100,000 personal computers, whose resources are donated for massively distributed MD simulations.[18] Additionally, many parallel simulations may provide better estimates of transition rates for this single barrier, since there are more statistics on the outward transitions.

For large sets of independent simulations that are in local equilibrium (i.e. they sample from the underlying energy distribution), we can reconstruct both the proper thermodynamics

and kinetics with the use of Markov state models (MSMs).[19] An MSM is a network model that describes a protein's energy landscape in terms of a set of structural states the protein tends to adopt and the probabilities of transitioning between neighboring states in a fixed time interval. The utility of an MSM depends on accurately estimating the conditional transition probabilities between conformational states, without requiring that any individual simulation achieve global equilibration. As a consequence, the number of times different states are sampled does not need to be Boltzmann distributed for an accurate description of their populations at equilibrium, provided that estimates of transition probabilities are accurate. MSMs have recently succeeded in guiding the design of new proteins[20,21] and allosteric modulators,[22] among many other applications.[19,23-31]

MSMs' ability to integrate information from many parallel simulations whose starting states are not necessarily Boltzmann distributed opens the possibility of performing adaptive sampling. First developed for refining MSMs by identifying conformational states that contribute the most to statistical uncertainty,[32] adaptive sampling schemes typically have the following steps: 1) run simulations, 2) build an MSM from simulations, 3) rank each state by some metric, 4) start new simulations from the highest ranked states, and 5) repeat steps 2-4 for some number of rounds or until a convergence criterion is met. The main difference between adaptive sampling algorithms is in the metric for ranking and selecting states for future sampling.[32-41] Recently, we have developed the goal-oriented sampling algorithm, Fluctuation Amplification of Specific Traits (FAST), that ranks states on some structural metric in addition to a statistical metric.[42,43] We have demonstrated that this method increases the rate of state exploration by at least an order of magnitude, and additionally, can capture thermodynamic and kinetic information that agrees with a multitude of experiments.[21,44]

Each of the equilibrium-based sampling methods mentioned above (long-, parallel-, adaptive-, and FAST-simulations) should converge on identical MSMs, provided with near infinite sampling. Unfortunately, for most systems of interest, simulations are not able to reach global equilibrium, and are usually significantly under-sampled. It should be noted that FAST, and other adaptive sampling algorithms, do not *increase* the amount of sampling, but rather focus sampling efforts to specific regions of conformational space to make the most of limited computational resources. With that, the functional differences between methods are simply the rates at which specific sections of conformational-space are explored. However, it is not completely understood how each of these methods influences the probability of discovering states, nor how this influences the mechanism of conformational changes that is observed, especially when conformational sampling is far from global equilibrium.

In this work, we seek to assess the relative performance of different sampling strategies. We develop an analytical expression for the probability of discovering a conformational state for very simple landscapes. We find that state discovery is dependent on the number and length of simulations, in addition to the shape of the energy landscape. We then examine the performance of the four equilibrium-based sampling methods above in finding the highest-flux pathway between two states, for a variety of energy landscapes. These results are very informative for tuning the many hyperparameters in adaptive sampling, and even identify pitfalls that should be avoided. Lastly, we demonstrate that insights from our simple landscapes are consistent with observations using all-atom MD simulations, by generating folding trajectories of a fast-folding version of the λ-repressor.

4.3 Theory

To understand how the probability of discovering a state on a particular landscape is dependent on sampling, we develop a mathematical formalism for describing the probability that a set of simulations will discover a particular conformational state. First, we consider sampling to occur on a discretized energy landscape with N conformational states, where the state index is represented as n_i, $i = 1, \ldots, N$. Transitions between states are described by the $N \times N$-transition probability matrix, T_{ij}, which is the probability of transitioning from state n_i to n_j at a specified lag-time, τ. A simulation on this landscape of *K*-steps is denoted with the symbol **X**, where the conformation at the *k*-th time step is X_k, $k = 1, \ldots, K$. For a dataset with M simulations, we denote the *m*-th simulation as $\mathbf{X}_m$, $m = 1, \ldots, M$. For multiple simulations of various lengths (different number of time steps), we choose **K** to represent a vector of lengths, where K_m, $m = 1, \ldots, M$, is the length of the m-th simulation.

Towards our goal of describing the probability of discovering a particular conformational state on an energy landscape given sampling parameters, we introduce the $N \times N$-matrix, $D_{ij}^{\mathbf{K},\mathbf{M}}$, which indicates if state n_j is ever discovered within the trajectories $\mathbf{X}_M$, started from state n_i with lengths described by **K**. For example, if state n_j is a state within the trajectories, $D_{ij}^{\mathbf{K},\mathbf{M}}$ is 1, otherwise it is 0. This can be represented with,

$$D_{ij}^{\mathbf{K},\mathbf{M}} = \begin{cases} 1 & if\ n_j \in \mathbf{X}_M \\ 0 & if\ n_j \notin \mathbf{X}_M \end{cases} \qquad [1]$$

While this can be determined for a set of trajectories, we wish to know the probability of having observed a state, *a priori*, or $\mathrm{P}\left(D_{ij}^{\mathbf{K},\mathbf{M}} = 1\right)$. This is the probability of discovering state n_j given

the sampling parameters **K**. For short hand, we call these probabilities the "discover probabilities".

Before providing an expression for $\mathrm{P}\left(D_{ij}^{\mathbf{K},\mathbf{M}}=1\right)$, we must first introduce another $N \times N$-matrix, v_{ij}^{k}, which indicates if the conformation at the k-th step of a single trajectory, **X,** belongs to the state n_j, when started from state n_i.[45]

$$v_{ij}^{k}=\begin{cases}1 & if\ \mathrm{X}_k = n_j \\ 0 & if\ \mathrm{X}_k \neq n_j\end{cases} \qquad [2]$$

Additionally, we are interested in the probability of this event occurring, denoted as $\mathrm{P}\left(v_{ij}^{k}=1\right)$. Since only one conformation at the k-th step can be observed, each row of $\mathrm{P}\left(v_{ij}^{k}=1\right)$ is a normalized probability vector indicating the state index at time k. For the trivial case of the 0$^{\text{th}}$-step (i.e. before a simulation is generated), the probability of being in the starting state is 1, and everywhere else, 0:

$$\mathrm{P}\left(v_{ij}^{k=0}=1\right)=I$$

where I is the identity matrix. Since $\mathrm{P}\left(v_{ij}^{k}=1\right)$ is a list of probability vectors, we can propagate the probabilities a time step (the lag-time, τ) using the transition probability matrix, T.

$$\mathrm{P}\left(v_{ij}^{k}=1\right)=\begin{cases}I & if\ k=0 \\ \mathrm{P}\left(v_{ij}^{k-1}=1\right)T & if\ k>0\end{cases} \qquad [3]$$

This expression is useful for determining $P\left(D_{ij}^{\mathbf{K},\mathbf{M}}=1\right)$, since the probability of ever visiting a state is the complement of not visiting it at each time step. For example, the probability of discovering state n_j after one step is the complement of not discovering it before and after one step:

$$\mathrm{P}\left(D_{ij}^{\mathbf{K}=\{1\},M=1}=1\right)=1-\left(1-\ \mathrm{P}\left(v_{ij}^{0}=1\right)_{ij}\right)*\left(1-\ \mathrm{P}\left(v_{ij}^{1}=1\ \right)_{ij}\right)=1-\left(1-I_{ij}\right)*$$

$$\left(1-T_{ij}\right)=\begin{cases}1 & if\ i=j\\ T_{ij} & if\ i\neq j\end{cases}$$

This reasoning holds true for a single step in a simulation, although does not for more than one step. What is required is an expression for the probability of being in a state at time step, k, conditional on not having discovered state n_j for all of the previous steps. We represent this expression as, $\mathrm{P}\left(v_{i'j'}^{k}=1\mid\left\{v_{ij}^{k'}=0\ \forall\ k'<k\right\}\right)$, which can be evaluated with the following:

$$\mathrm{P}\left(v_{i'j'}^{k}=1\mid\left\{v_{ij}^{k'}=0\ \forall\ k'<k\right\}\right)=\begin{cases}I & if\ k=0\\ \mathrm{P}\left(v_{i'j'}^{k-1}=1\mid\left\{v_{ij}^{k'}=0\ \forall\ k'\leq(k-1)\right\}\right)T & if\ k>0\end{cases}$$

[4]

For each step in the recursive calculation, the j^{th} column of $\mathrm{P}\left(v_{i'j'}^{k-1}=1\right)$ is set to 0, and each row is then normalized to unity. This is described in more detail in the supporting information.

Using this definition, we can extend our expression of the discover probabilities to include an arbitrary number of steps, K. In a single simulation, we can see that the probability of discovering a state is:

$$\mathrm{P}\left(D_{ij}^{\mathbf{K}=\{K\},M=1}=1\right)=1-\prod_{k=0}^{K}\left(1-\mathrm{P}\left(v_{i'j'}^{k}=1\mid\left\{v_{ij}^{k'}=0\;\forall\;k'<k\right\}\right)_{ij}\right) \quad [5]$$

Since the probability of discovering a state within a simulation is independent of the probability in another simulation, the discover probabilities for multiple simulations is the complement of not discovering a state in any of the individual simulations. For example, in the case of two simulations with lengths K_0 and K_1,

$$\mathrm{P}\left(D_{ij}^{\mathbf{K}=\{\mathrm{K}_0,\mathrm{K}_1\},M=2}\right)=1-\left(1-\mathrm{P}\left(D_{ij}^{\mathbf{K}=\{\mathrm{K}_0\},M=1}=1\right)_{ij}\right)*\left(1-\mathrm{P}\left(D_{ij}^{\mathbf{K}=\{\mathrm{K}_1\},M=1}=1\right)_{ij}\right)$$

This can be generalized to an arbitrary number of simulations with arbitrary lengths:

$$\mathrm{P}\left(D_{ij}^{\mathbf{K},\mathbf{M}}=1\right)=1-\prod_{m=1}^{M}\left[\prod_{k=0}^{\mathrm{K}_m}\left(1-\mathrm{P}\left(v_{i'j'}^{k}=1\mid\left\{v_{ij}^{k'}=0\;\forall\;k'<k\right\}\right)_{ij}\right)\right] \quad [6]$$

This gives us our final expression for state discovery as a function of our equilibrium-sampling parameters.

4.4 Results

4.4.1 There Are Different Advantages to Running Many Short or Few Long Simulations

From equation 6, it is clear that the probability of discovering a state is influenced by four parameters: 1) the number of trajectories, 2) the lengths of the trajectories, 3) the starting state, and 4) the shape of the landscape being sampled. Strikingly, this implies that the probability of

discovering a state can be drastically distinct between a single long simulation and many short simulations, though this is only true for finite sampling since $\mathrm{P}\left(D_{ij}^{\mathbf{K},\mathbf{M}} = 1\right) \rightarrow 1$ as either, $\mathbf{M} \rightarrow \infty$, or $\mathrm{K}_m \rightarrow \infty$. It may seem tempting to seek the global optimum sampling parameters, however, sampling is strongly dependent on the specifics of the landscape itself. Additionally, different goals may necessitate different sampling strategies, i.e. is the goal to discover as many states as possible, or to discover a pathway between a particular set of states? From this, our goal is to characterize different sampling strategies for a variety of different landscapes to gain insight into their appropriate uses.

As a first test, we constructed a simple landscape where four states are connected in a linear arrangement (Figure 4.1A). Here, each state can transition to either a neighbor or itself, with differing probabilities. We imagine that these states represent a conformational landscape where each successive state is progress along some order parameter. Starting from state 0, the first state in the chain, we calculate the probability of discovering the other states from either a single trajectory or many parallel trajectories with an equivalent aggregate amount of simulation. Figure 4.1C depicts the probability of discovering states 2 (blue curves) and 3 (red curves) from a single trajectory at various time-steps (solid lines), or some number of parallel trajectories with 4 time-steps each (dashed lines). We see that the long simulations have a higher probability of reaching states 3 and 4 than do parallel simulations. For this shaped landscape, the discrepancy between long and parallel simulations widens with the number of states. This makes intuitive sense from equation 6, because we see that the probability of a simulation making 2 successive transitions is different than one of two simulations making 2 successive transitions.

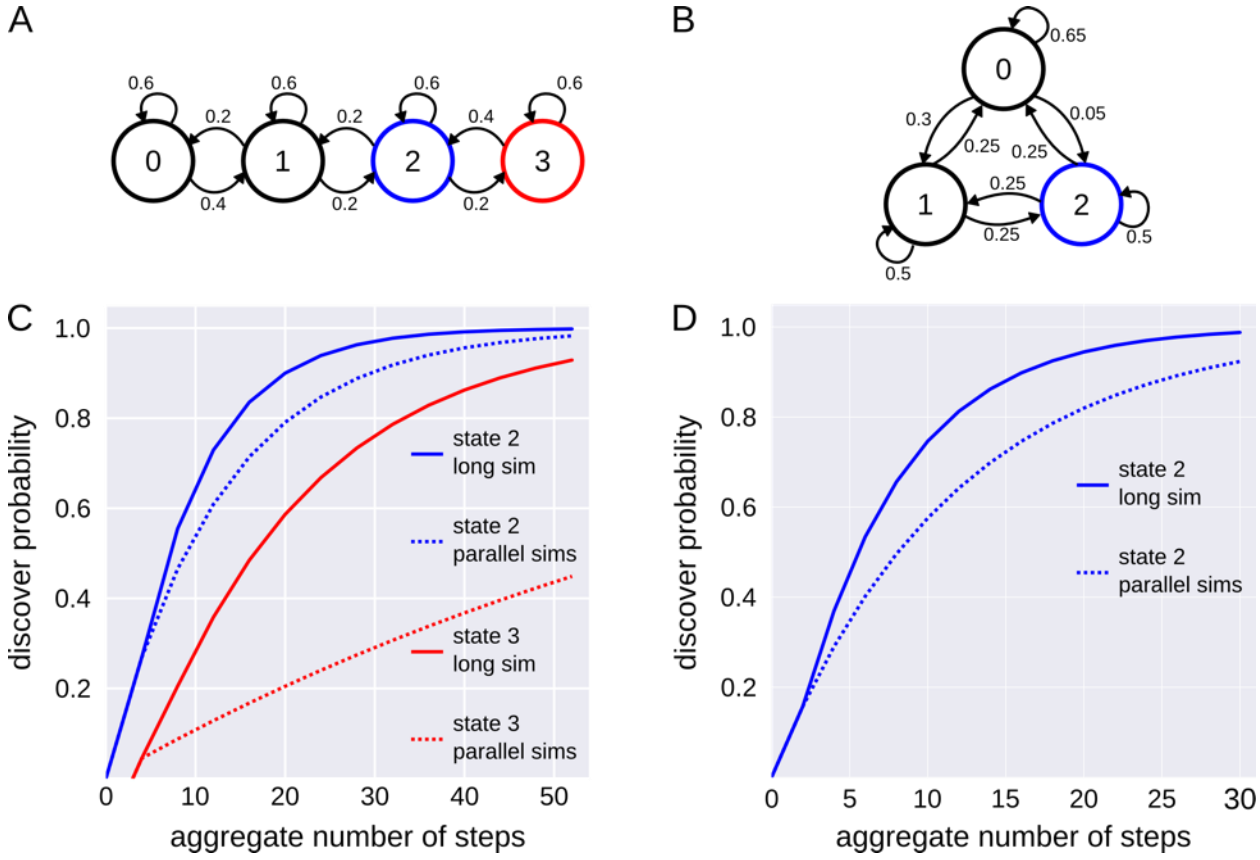

Figure 4.1: The probability of discovering particular states on simplified landscapes as a function of the number and length of simulations from equation 6. (A) Four states, arranged linearly, have transitions to themselves and their direct neighbors to varying degrees. States 2 and 3 are colored blue and red for visual aid. (B) A fully connected 3 state system. The probability of transitioning from state 0 to 2 is very low. (C) The probability of discovering state 2 (blue) or state 3 (red) with either a single long simulation (solid line) or many simulations consisting of 4 steps (dashed line) for the landscape in panel A. (D) The probability of discovering state 2 (blue) with either a single long simulation (solid line) or many simulations consisting of 2 steps (dashed line) for the landscape in panel B.

We should note that the fully connected landscape in Figure 4.1B also displays this property, indicating that it is not an artifact of the way we have drawn the landscape. Here, the probability of transitioning between state 0 to 2 is very low, making the more probable route go through the transition state, 1. This leaves parallel simulations at a disadvantage of having to take the longer route to observe the transition, making this observation less probable. Interestingly, this also indicates that it is possible to consistently stumble upon an incorrect conclusion for the transition pathway; a trivial example being that many 1-step simulations started from state 0 would incorrectly predict the pathway as going directly from state 0 to 2. It is an important point

that this result arises from the probability of discovering certain states, and their transitions, but not from the estimates of each states conditional transition probabilities, which should remain preserved across sampling methods. Therefore, in addition to understanding how sampling affects state discovery, we are interested in how the state discovery influences the predicted mechanism of conformational changes (e.g. the highest probability transition pathways between two sets of states). We investigate this idea in more detail in later sections.

So far, linear and fully connected landscapes might lead one to believe that long simulations are always advantageous in state discovery, but this is not the case when landscapes have entropic barriers. For many realistic systems, it is likely that a particular conformational state has many other states that it can transition to. To capture this transitional entropy, we generated the star-shaped landscape depicted in Figure 4.2A. This landscape has a central state and 5 arms, which is reminiscent of a "kinetic hub" where unfolded/high-energy states typically pass through the folded state to reach other unfolded/high-energy states.[46] Parallel simulations have a significantly higher probability to discover any of the states on this landscape, compared with equal aggregate time of the long simulation. We reason that the long simulations are penalized by having to backtrack to explore each of the arms, whereas the parallel simulations have a high probability of sampling multiple arms simultaneously. This effect becomes more drastic as the dimensionality of the state-space increases. Furthermore, this landscape provides a nice example that the optimal sampling scheme is strongly dependent on the shape of the landscape.

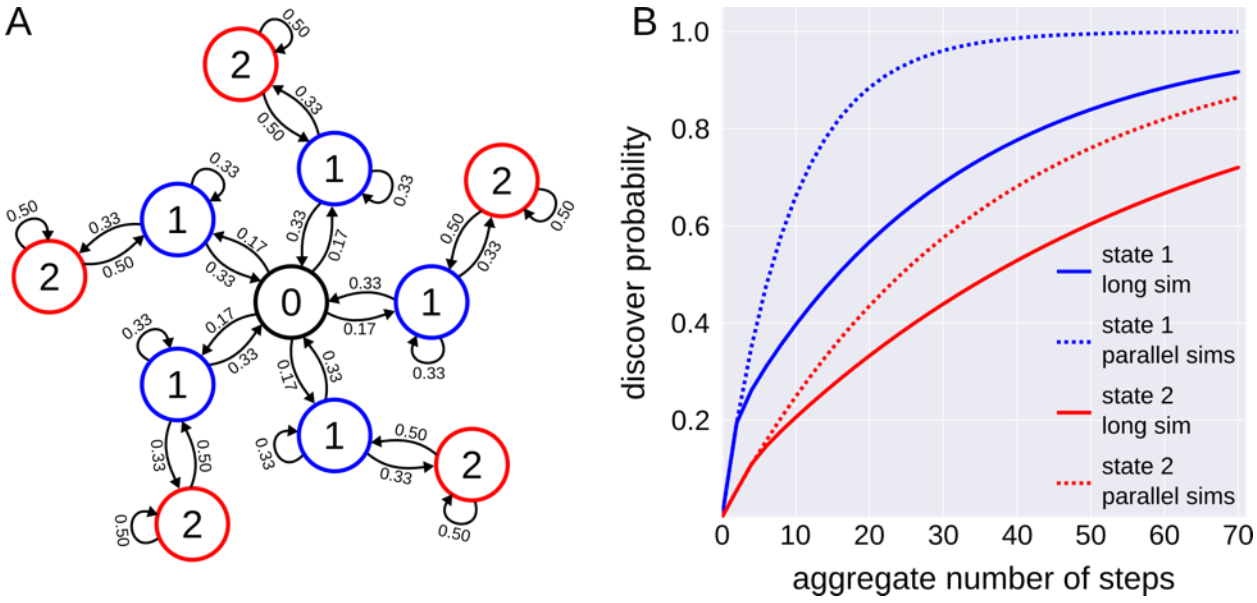

Figure 4.2: The probability of discovering particular states on a star-shaped landscape as a function of simulation length and number of simulations from equation 6. (A) Network representation of the star-shaped landscape. Due to symmetry in the transition probabilities, a simulation started from state 0 has equal probability of reaching any of the states labeled 1, as well as any of the states labeled 2. State 0 also has a self-transition probability of 0.17, but this edge is omitted for visual clarity. (B) The probability of discovering a particular state 1 (blue) or state 2 (red) with either a single long simulation (solid line) or parallel simulations consisting of 2 steps (dashed line).

These simple landscapes provide valuable insight into how long or parallel simulations affect state discovery, setting a baseline for characterizing more complicated sampling schemes, such as adaptive sampling. Towards this goal, we generated a series of larger landscapes, which emulate common challenges in the sampling of proteins. To aid in human intuition, these landscapes are two-dimensional energy surfaces projected onto a grid, where each point on the grid represents a conformational state with a single potential energy. Each state can have up to four connected neighbors, with transitions governed by the Metropolis criterion. In the next few sections, we make use of kinetic Monte Carlo simulations on these landscapes using four different sampling methods: 1) a single long simulation (referred to as "long"), 2) many short simulations (referred to as "parallel"), 3) counts-based adaptive sampling (referred to as "counts"), and 4) our goal-oriented FAST algorithm (referred to as "FAST"). Although there are many adaptive sampling algorithms, we chose to use counts because it has been shown to be the

best at indiscriminately discovering new states.[34,42] The specifics of sampling are described in greater detail in the methods section. Furthermore, we aim to characterize each method based on three different criteria: 1) ability to discover a target state, 2) ability to predict realistic transition pathways, and 3) ability to estimate accurate transition probabilities.

4.4.2 FAST is Most Likely to Discover the Target State

The first landscape that we generated was inspired by the challenge of using MD simulations to find the native state of a cooperatively folding protein. Two common tasks include: 1) to determine the native conformational state given an amino-acid sequence, also known as a structure prediction problem,[47-49] and 2) explore the preferred pathway(s) from an unfolded state to the native state.[50,51] We chose to start with one of the simplest possible models, a minimally frustrated folding-funnel (Figure 4.3).[52,53] Here, there is a reasonably smooth energetic gradient from a high-energy starting-state to the low-energy target state. The solid colored lines represent the three highest-flux pathways from the start to the target.

To characterize state-discovery on this landscape, we performed 5,000 independent trials of each sampling method, with equivalent aggregate simulation times, as is described in the methods. We then calculate the probability of discovering a given state (which we refer to as the discover probabilities) for the four methods, by averaging the results of equation 1 for each trial. We note that we terminate the algorithm after reaching the target state, since we are mainly concerned with the initial pathway to the target; including excessive sampling after reaching the target convolutes the results with what happens afterwards. Additionally, trimming the data after discovering the end state does not affect the discover probabilities of the end state itself.

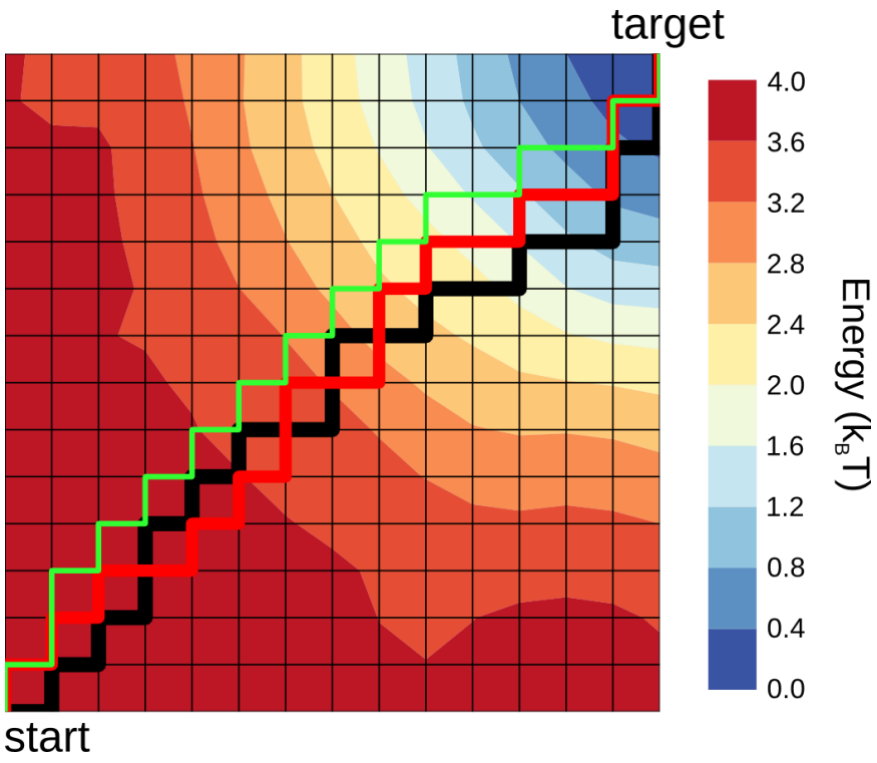

Figure 4.3: An energy landscape inspired by a simple folding funnel. Conformational states are located at the vertices of the grid, where the color at this point represents the energy of that state. States can have up to 4 neighbors to transition with. Solid lines (black, red, and green) indicate the three highest flux pathways from the start to the target state, where line thickness is proportional to the flux along the particular path.

If the goal is to simply reach the end-state, FAST does so with the highest probability. The discover probabilities of the target state are $1.0 \pm 7 \times 10^{-4}$, 0.94, $0.62 \pm 7 \times 10^{-3}$, and 2.2×10^{-5} for FAST, long, counts, and parallel simulations respectively (this value for long and parallel simulations come from equation 6). It is not a surprise that FAST is the best at reaching the end state, since it is the only method tested that uses knowledge of the end state in its sampling and we have previously shown FAST's ability to reach a target state with orders of magnitude less simulation.[42] Of greater interest here is the difference between the observation of states along the way to the target.

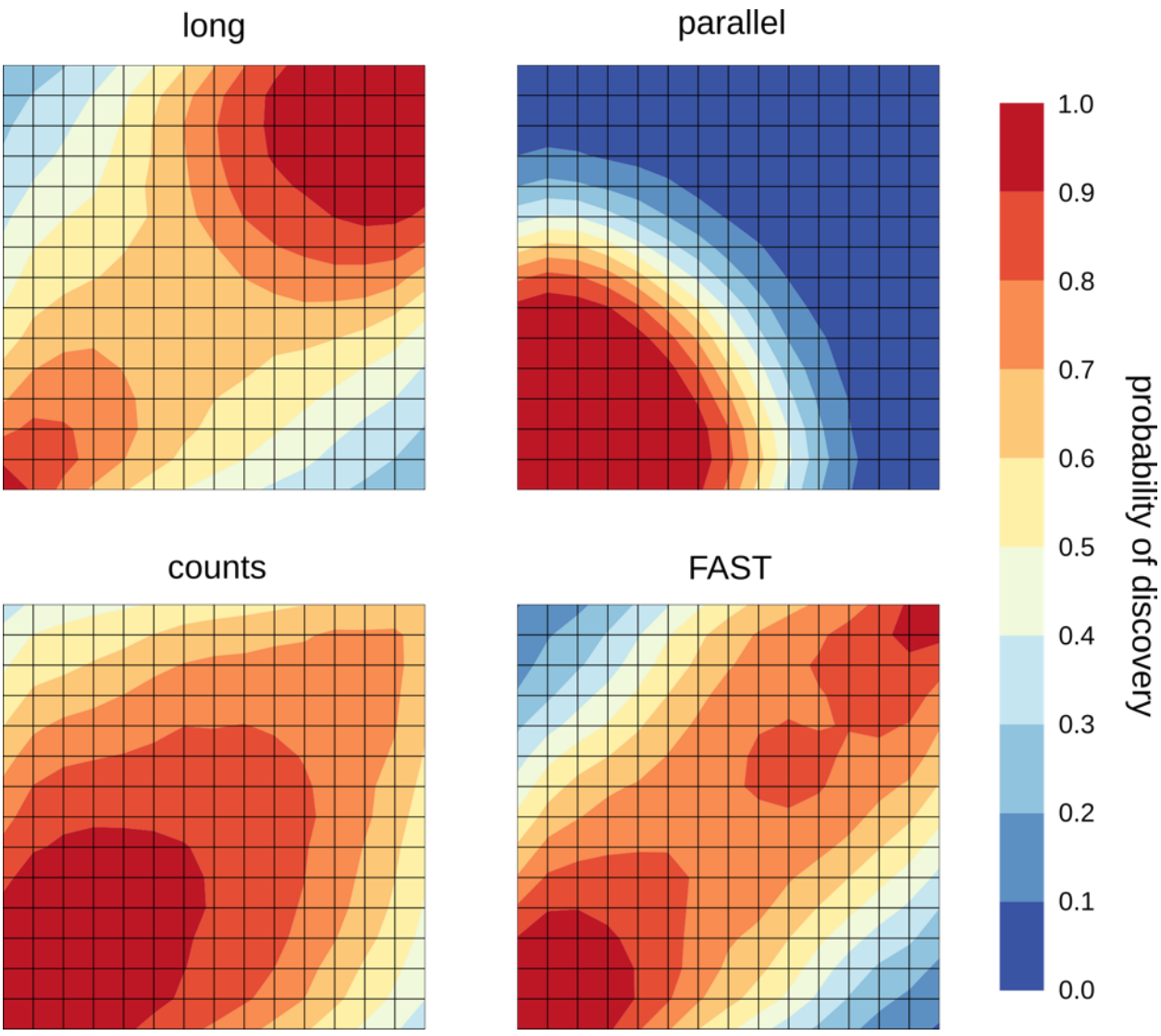

Figure 4.4: The discover probabilities (the probability that a simulation set observes a particular state) on the funneled landscape in Figure 3. Shown are the probabilities for four sampling strategies, a single long simulation, many parallel simulations, counts-based adaptive sampling, and the goal-oriented FAST simulations.

Towards this goal, we plot the discover probabilities for each method in Figure 4.4, which reveals distinct patterns for each sampling method. We find it extremely beneficial to view the discover probabilities for each state in this manner, since it provides intuition for the ways that each method explores the landscape before reaching the target. Analysis of the long simulations indicates that they have a propensity to sample around the native-well before reaching the target state. The 25 states closest to the target have over a 0.9 probability of being

discovered first. Conversely, parallel simulations rarely venture near the target, but thoroughly explore the landscape around the starting state. Strikingly, this suggests that parallel simulations would require orders of magnitude more aggregate simulation time than the long simulation to reliably observe a transition to the target. In fact, this is what we observe for MD simulations of the λ-repressor in a later section.

Unlike the other sampling strategies, counts-based adaptive sampling has an elevated propensity to explore the high-energy edges of the funneled landscape. Compared to the long simulations, counts has almost twice the probability of discovering the states furthest from the start and the target, yet, nearly half the probability of discovering the target itself. This is because counts indiscriminately discovers new states, particularly in high-energy neighborhoods where low count states are prevalent. This property enables counts-based sampling to lead in state discovery, with an average of 183.3 ± 12.3 states discovered, in comparison to 168.5 ± 12.3, 144.2 ± 24.0, and 72.7 ± 10.1 for FAST, long, and parallel simulations respectively. Interestingly, counts-based sampling's propensity to climb energy barriers actually hinders its ability to follow a simple gradient to the global minimum. Therefore, counts-based simulations may actually be a poor choice for many applications, despite its ability to discover many states, because it will dedicate significant computational resources to sampling irrelevant (e.g. high-energy) states. On the other hand, FAST simulations are very directed.

On this funneled landscape, FAST not only has a higher probability of discovering the states along the highest-flux pathways to the global minimum, but also provides the best estimates of their transition probabilities. While there are many ways to estimate transition probabilities to construct an MSM from trajectories, we compare sampling results by row-normalizing transition counts. This is a straightforward method that works well with adaptive

sampling data, as is described in a later section. Using a relative entropy metric to quantify the deviation of MSMs built with each method from the true landscape, as we have done previously,[42,46] we find that FAST and long simulations have the lowest deviations for states in the top three highest-flux pathways. These relative entropies, ascending, are 0.58 ± 0.46, 0.84 ± 0.80, 1.96 ± 0.76, and $2.46 \pm 5 \times 10^{-2}$ for FAST, long, counts, and parallel simulations, respectively. This result suggests that FAST matches long simulations' ability to reach distant conformations, parallel simulations' ability to thoroughly explore particular regions of conformational space, and adaptive sampling's flexibility.

Taken together, the funneled landscape provides a coarse view of each sampling method's behavior. With the perspective that aggregate simulation time is a finite resource, we can imagine the differences between sampling methods being the amount of this resource spent on each region of the conformational landscape. Parallel simulations spend the majority of this resource around the starting state. Counts-based simulations spread it across the landscape. Long simulations distribute it in proportion to neighboring states' energy. FAST spends computing resources on the states that optimize its objective. On the funneled landscape, there are minimal barriers to prevent counts from spreading, and the states that optimize FAST's objective are nearly a straight line from the start to the target. In the following section, we add a layer of complexity to see if adaptive sampling can truly adapt to roadblocks in energy landscapes.

4.4.3 Adaptive Sampling Navigates Obstacles

To mimic the complexities of more realistic landscapes, we generated the rugged landscape in Figure 4.5A. This rugged landscape provides an interesting challenge to not just discover the target state, but also discover the preferred pathways. The three highest-flux pathways between the start and the target state are shown in Figure 4.5A, which each require navigation around

large energy barriers. As an added difficulty, there exist alternative routes around the barriers, with differing fluxes. Although sampling is stochastic, and any individual run has the potential to proceed along an arbitrary path, we expect the distribution of paths to resemble the actual highest-flux paths. Of special interest is how FAST navigates the landscape, since it strongly uses structural information in reseeding simulations. We wish to confirm that it does not cut across high-energy barriers in an effort to maximize its objective function.

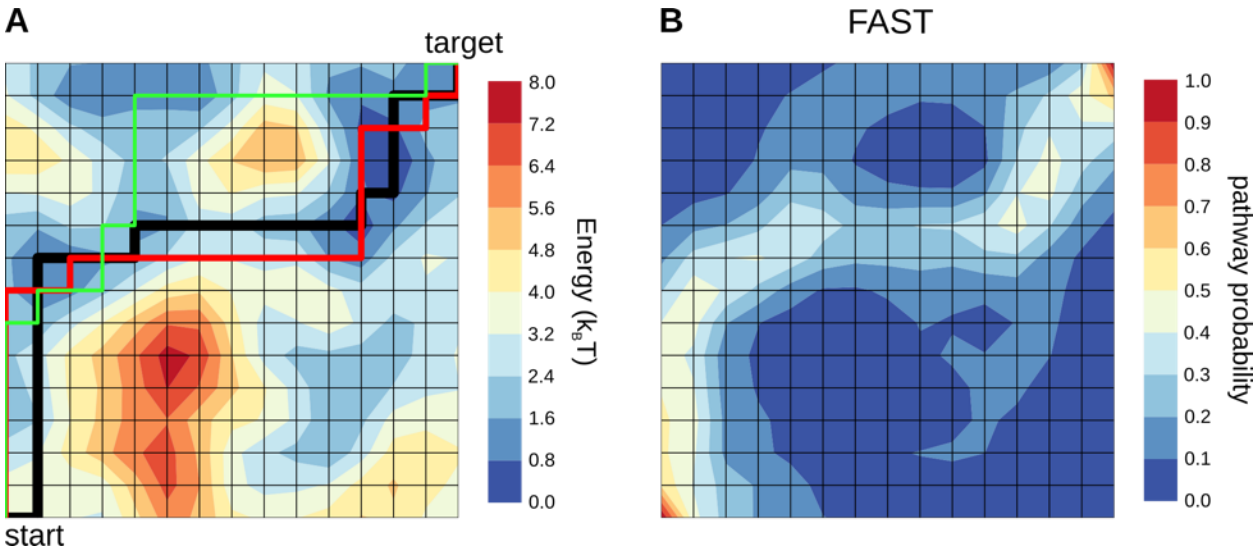

Figure 4.5: The performance of FAST on a rugged landscape. (A) An energy landscape inspired by a folding funnel with random obstacles. Conformational states are located at the intersection of the grid lines, where the color at this point represents the energy of that state. Solid lines (black, red, and green) indicate the three highest flux pathways from the start to the target state, where line thickness is proportional to the flux along the particular path. (B) The probability that a FAST simulation set will predict a state to be in the highest-flux path from the start to the target state.

Similar to the performance on the previous landscape, FAST outperforms the alternative approaches in discovering the target state. This is best seen from each method's discover probabilities on this landscape (Figure A.1.3), where FAST clearly has the highest probability of discovering the target state. In addition, FAST is most likely to discover the states along the actual highest-flux pathways, which suggests that FAST also predicts the correct pathways. To better quantify this, we characterize the probability that a state is predicted to be on pathway

from the start to the target. This is done by calculating the highest-flux pathway for each of our 5,000 trials and determining the number of times a state is observed. Averaging this leaves us with a state value of 1 if it is always observed when transitioning from the start to the target, and 0 if it is never observed.

Inspection of the pathway probabilities for FAST (Figure 4.5B) reveals its ability to navigate around obstacles. The predicted pathways from the start to the target do not pathologically cut across the energy barriers, but mimic the route taken by the three highest-flux pathways that were calculated from the underlying transition probabilities. Furthermore, the predicted pathways of FAST and counts resemble the predicted pathways obtained from the long simulations (Figure A.1.4). This is consistent with the hypothesized benefits of adaptive and goal-oriented sampling: since each simulation is in local equilibrium, the probability of traversing any individual barrier remains unchanged, and thus, transitions will typically occur along realistic pathways.

4.4.4 Pathway Tunneling: Observing an Unfavorable Pathway Due to Sampling Artifacts

The landscapes considered so far have been well suited for use with FAST, largely because the simple geometric function used in our FAST ranking (i.e. distance to the target state) is a reasonable surrogate for kinetic proximity to the target. However, there are many instances where finding a reasonable surrogate may be difficult. For example, there are many systems where transitioning between geometrically similar conformations may require partial unfolding of a protein.[54] In these cases, the optimal transition path would have, at times, unfavorable state rankings.

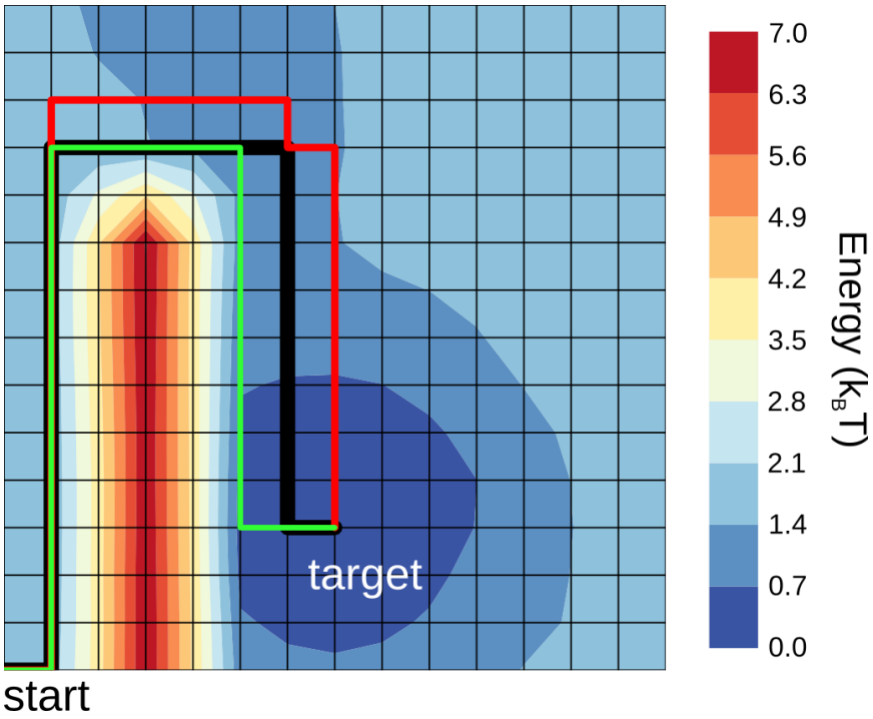

Figure 4.6: An energy landscape where the preferred pathway is not the shortest distance between the start and the target state. Conformational states are located at the intersection of the grid lines, where the color at this point represents the energy of that state. Solid lines (black, red, and green) indicate the three highest flux pathways from the start to the target state, where line thickness is proportional to the flux along the particular path.

To explore the utility of FAST when the preferred pathway is suboptimally described by the geometric ranking function, we modeled a landscape with a large barrier separating the start and target states (Figure 4.6). Here, the three highest-flux pathways all circumnavigate this large barrier rather than taking the geometrically shortest path (across the barrier). Indeed, the long simulations also indicate that the preferred pathway does not cut across the barrier, but follows the longer, low-energy route (Figure 4.7A-B).

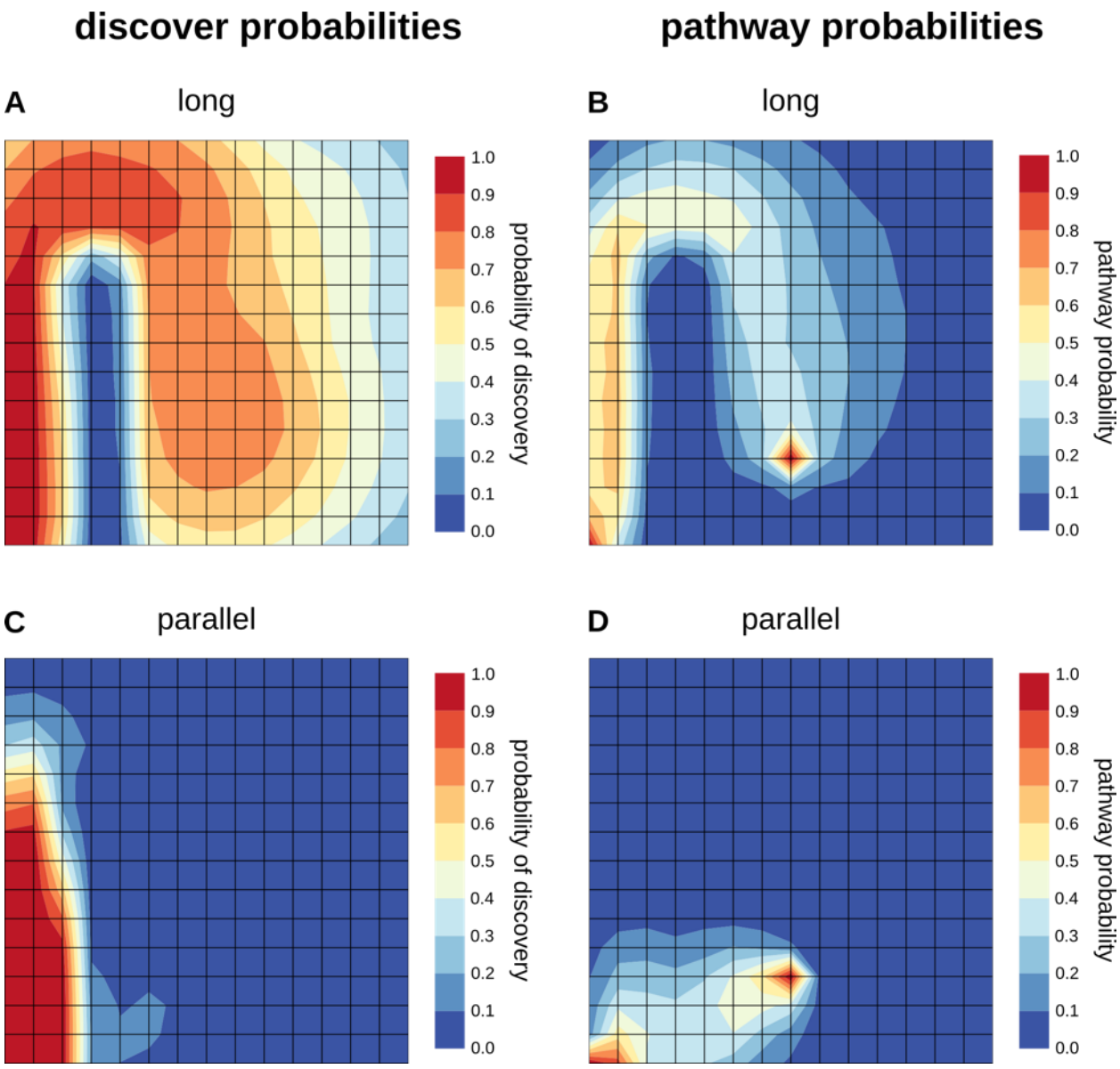

Figure 4.7: The discover probabilities and predicted pathways for long and parallel simulations on the landscape in Figure 6A. (A) The probability that a long simulation discovers a particular state. (B) The probability that a long simulation will predict a state to be in the highest-flux path from the start to the target state. (C) The probability that a set of parallel simulations discovers a particular state. (D) The probability that a set of parallel simulations will predict a state to be in the highest-flux path from the start to the target state.

This landscape highlights a potential pathology of running many short parallel simulations, which consistently predict that the highest-flux pathway cuts across the high-energy barrier. From Figure 4.7C, we observe that the probability that one of the short simulations completes the long path is significantly less than the probability that it hops across the high

energy barrier. This leads to the prediction of a very unrealistic highest-flux pathway, as shown in Figure 4.7D. We name this undesired phenomenon "pathway tunneling", due to its loose similarity to the tunneling through high-energy barriers observed in quantum mechanics. If the length of all the parallel simulations is gradually increased, the probability of pathway tunneling falls monotonically and converges on the correct mechanism. In this example, pathway tunneling is a consequence of not discovering the set of states along the optimal path, although, it can also arise from poor estimates of transition probabilities that result in an overestimate of the probabilities of rare paths (due to insufficient sampling or inaccuracies in MSM construction). We explore the role of MSM estimators on adaptive sampling data in a later section.

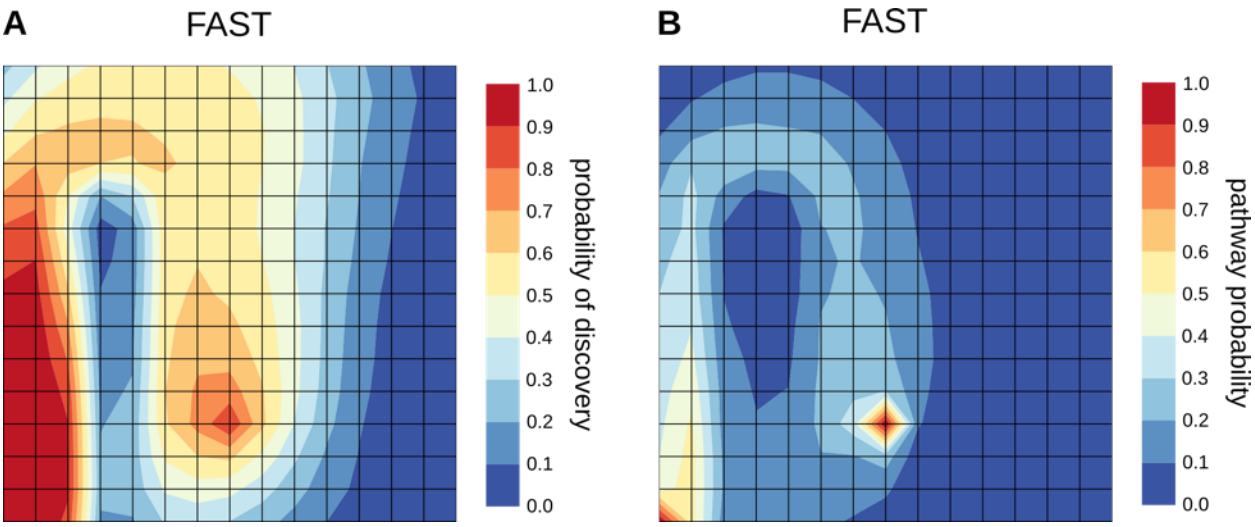

Figure 4.8: FAST simulations navigating a large energy barrier. (A) The probability that a FAST simulation set discovers a particular state. (B) The probability that a FAST simulation set will predict a state to be in the highest-flux path from the start to the target state.

From the discover probabilities in Figure 4.8A, we observe that FAST has a significantly higher probability of discovering the states along the preferred pathway compared to those of the tunneled pathway. It appears that even in the extreme case where the directed component is at times orthogonal to the preferred pathway, FAST's statistical component mitigates pathway tunneling. This is evidenced from counts-based adaptive samplings' ability to discover the

correct pathway (Figure A.1.6-7). However, despite this benefit, compared to the long simulations there is an increased probability of discovering the tunneled states. This isn't an issue if the estimates of the transition probabilities are accurate enough to distinguish the likelihood of each path, although the pathway probabilities in Figure 4.8B show that FAST non-negligibly predicts the tunneled pathway as the preferred pathway. This result suggests that although FAST is able to circumnavigate orthogonal energy barriers, poor selection of a geometric criterion may lead to over estimating the probability of traversing unfavorable paths.

4.4.5 FAST-String Quickly Discriminates between Alternative Pathways

To minimize the probability that FAST falls victim to pathway tunneling, we introduce a new ranking scheme for FAST that refines the transition probabilities along the highest-flux pathways to quantify their relative weights. This method draws inspiration from the string method,[55-57] which refines a proposed transition path by iteratively running short molecular dynamics simulations from regularly spaced conformations along the path and letting them relax towards the true lowest free energy path. Here, we begin FAST-string after first discovering a pathway, or set of pathways, to the target state using the original FAST rankings. Then, we change the ranking function to focus on refining the transition probabilities of the path(s) found. Specifically, we calculate the *n*-highest-flux pathways and rank states found in these paths by some statistical criterion. Thus, our state rankings become:

$$r(i) = \begin{cases} \bar{\psi}(i) & if\ i \in \{w_0, \dots, w_n\} \\ 0 & otherwise \end{cases} \qquad [7]$$

where $r(i)$ is the ranking of state i, $\bar{\psi}(i)$ is the scaled statistical component of the original FAST ranking function, and $\{w_0, \ldots, w_n\}$ represents the states found in the n-highest-flux paths. For our purposes, we use the counts of each state as our statistical component to favor less explored regions of the predicted pathways. We expect that sampling along these states will distinguish favorable paths from unfavorable, if multiple paths are discovered, and help relax the pathway to the preferred path if pathway tunneling has occurred.

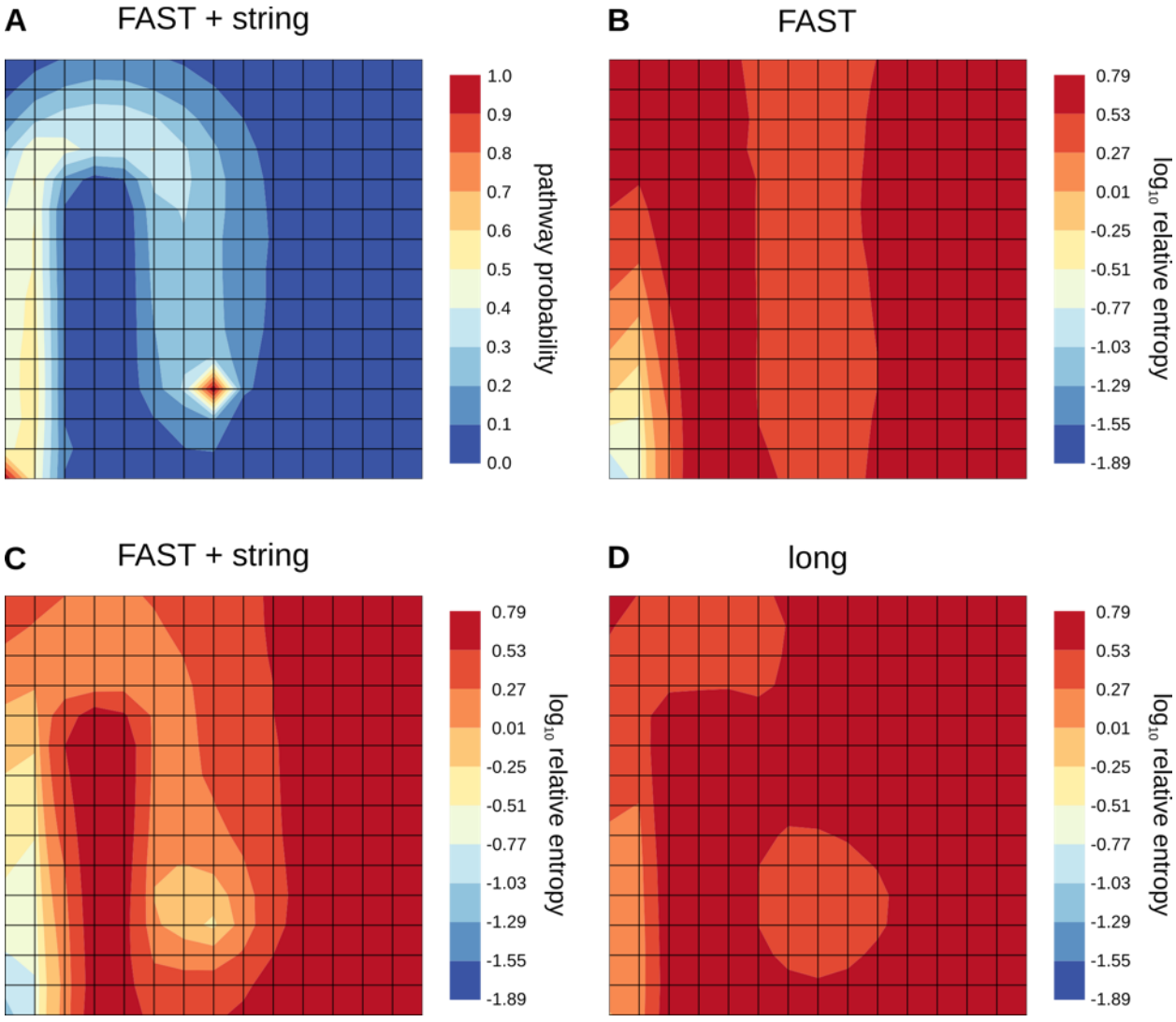

Figure 4.9: A comparison of predicted pathways and estimated transition probabilities between sampling methods on a landscape with a large barrier. (A) The probability that a FAST-string simulation set will predict a state to be in the highest-flux path from the start to the target state. (B-D) The Kullbeck-Liebler divergence of each states conditional transition probabilities to the true transition probabilities. Here, a lower value indicates a lower deviation from the

true underlying landscape. Compared are FAST simulations, FAST simulations followed by FAST-string, and a long simulation. Each of these are produced from equivalent aggregate simulation.

With our FAST-string method, we are able to consistently determine the preferred transition path. Figure 4.9A shows that the tunneled pathway is no longer predicted as the transition pathway. We obtain this result with the same amount of aggregate simulation as the other methods; we run FAST until it discovers the end state, then switch to FAST-string for the remainder of the rounds. Instead of redundantly sampling around the target state once found, FAST-string productively refines estimates of the transition probabilities. From Figure 4.9B-D, we can see that FAST-string has the most accurate estimates of transition probabilities along the highest-flux pathways.

4.4.6 Normalizing Row Counts Provides a Good Balance of Estimating Rates and Equilibrium Populations with Adaptive Sampling Data

In addition to comparing different sampling methods, it is important to ask what the best way of estimating the transition probabilities between states from a given data set is. In other words, what is the best way to use a count-matrix, which counts the observed transitions between every pair of states, to estimate the transition probabilities and equilibrium populations of each state?

The simplest way is to normalize each row in the count-matrix to get an unbiased estimate of each states conditional transition probabilities, where the first eigenvector provides the equilibrium populations.[45] However, this approach does not guarantee microscopic reversibility and can have serious pathologies if the transition probability matrix is not ergodic, especially if transitions are observed from state n_i to n_j but not in the opposite direction. To alleviate this issue, it is customary to assume that, prior to observing any data, each state has equal probability to transition to any other state. This can be represented by adding a pseudo-count, $\tilde{C}$, to each possible transition,

$$T_{ij}^{normalize} = \frac{C_{ij}+\tilde{C}}{\sum_k (C_{ik}+\tilde{C})} \quad [8]$$

where,

$$\tilde{C} = \frac{1}{n} \quad [9]$$

and n is the number of states. An alternative estimator, called the transpose method, enforces detailed balance. At equilibrium, we know that each state transition should be equally populated by the reverse process (running an infinitely long simulation in reverse should not alter the estimates for transition probabilities). Enforcing this is straightforward, by averaging with the transpose of the count-matrix:

$$C_{ij}^{transpose} = \frac{C_{ij}+C_{ji}}{2}$$

and

$$T_{ij}^{transpose} = \frac{C_{ij}^{transpose}}{\sum_k C_{ik}^{transpose}}$$

and the equilibrium populations are calculated as,

$$\pi_i = \frac{\sum_j C_{ij}^{transpose}}{\sum_{k,j} C_{kj}^{transpose}} \quad [10]$$

This has recently been extended for use with simulations at multiple temperatures.[58] More sophisticated methods have also been developed to enforce detailed balance, such as the use of maximum likelihood estimation (MLE),[23,24] and the observable operator model (OOM).[59] In the MLE method, the likelihood of the transition probability matrix given an observed trajectory, **X**, is determined to be,

$$\mathrm{P}(T|\mathbf{X}) \propto \prod_{i,j} T_{ij}^{C_{ij}}$$

Consequently, the most likely transition probability matrix is solved as,

$$T_{ij}^{MLE} = \arg\max_{T_{ij}^*} \mathrm{P}\left(T_{ij}^*|\mathbf{X}\right)$$

A variant of the MLE method, which we will refer to as MLE-CP (MLE- with Constrained Populations), has also been developed to enforce a pre-determined equilibrium probability distribution.[60,61] This is useful with experimental estimates of state populations. Lastly, the OOM was recently developed as a generalization to hidden Markov models,[62] and restructured for use with MSMs.[59,63]

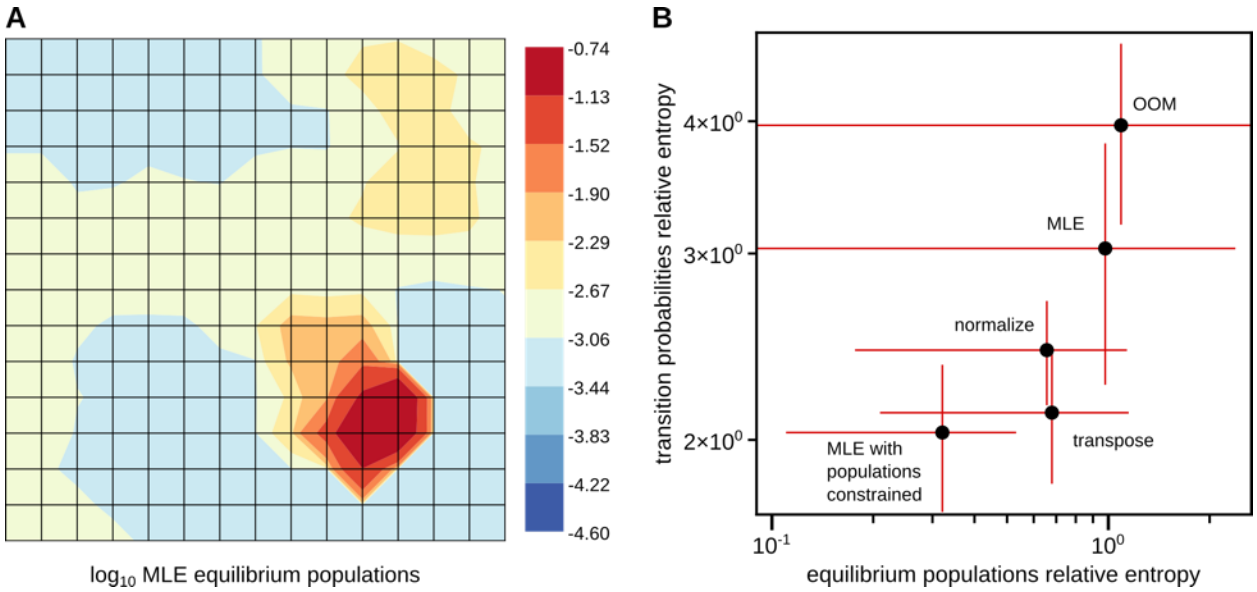

Figure 4.10: An analysis of MSM estimators' performance on the landscape depicted in Figure 5A. (A) The predicted state populations for a single FAST simulation using the MLE method. (B) A comparison of the MLE-CP, transpose, normalize, MLE, and OOM estimators. Solid points are the average relative entropy for transition probabilities and equilibrium populations. Red lines are the standard deviations of these values.

Each of these methods has been studied theoretically, in the limit of infinite data, and on small systems where sampling is not an issue. However, we are interested in the likely scenario where sampling is far from exhaustive. To test MSM construction in this regime, we used the FAST simulation sets on the landscape in Figure 4.5A to generate MSMs using the five methods listed above: 1) normalize, 2) transpose, 3) MLE, 4) OOM, and 5) MLE with constrained populations. We then compared the MSMs predictions of thermodynamics (equilibrium populations) and kinetics (transition probabilities) to the true distributions calculated from the underlying landscape. While the performance of each estimator may depend on the particular landscape being sampled, the case study we present here is representative of our results with other landscapes. Our metrics for performance consists of how well each estimator predicts the kinetics (transition rates) and thermodynamics (equilibrium populations) of the underlying energy landscape, as quantified with a relative entropy metric.

Upon inspection of the predicted equilibrium populations, we find that the MLE and OOM methods have a tendency to significantly overestimate the populations of an arbitrary set of states. Figure 4.10A is an example of this phenomenon for MLE, where four states are predicted to have a total probability of 0.58 even though the probability that they were sampled in the raw data is only 0.016. For reference, the true total probability of these states is 0.029 and the true probability of the most populated state in the underlying landscape is 0.032. In comparison, Figure A.1.9 shows that normalize and transpose give more reasonable predictions. Characterizing this over the entire dataset, we observe that on average, the largest predicted state population for MLE and OOM is 10.5 ± 21.0 and 16.4 ± 53.5 times larger than its true population. Interestingly, the deviation in these predictions are sizable; the most egregious observances of an overinflated state population for MLE and OOM were predictions of a single state containing 0.38 and 0.55 of the total population for each method, respectively. Additionally, MLE and OOM do not regularly overpopulate the same state; the probability that the state with the largest predicted population is truly the most populated state is 0.16 and 0.14 for MLE and OOM, respectively, compared to 0.35 and 0.33 for normalize and transpose, respectively. On the other hand, normalize and transpose have a largest populated state that is only 2.2 ± 2.5 and 2.0 ± 2.7 times its true population. However, while OOM is subject to the same pathology as MLE, severely over estimating the populations for a set of states, it appears to describe most of the landscape quite well (Figure A.1.9). Future developments of OOM could provide an accurate and robust estimator. To further quantify predictions for all states, we compute the relative entropy between each models' prediction of transition probabilities and equilibrium populations to the true distributions.

The MLE-CP is shown to generate an MSM with the most accurate estimates of kinetics and thermodynamics for FAST simulations on this particular landscape. Figure 4.10B shows the average deviations of transition probabilities and equilibrium populations from the true values for the underlying landscape for each MSM method. It is not surprising that constraining the populations to their true values performs well. Also, as has been previously reported, there are significant improvements to estimates of transition probabilities when the equilibrium populations are constrained.[61] However, *a priori* knowledge of the equilibrium distribution is not typically available, so it is not currently possible to adopt this approach as standard practice.

The normalize and transpose methods produce the next most accurate estimates of transition probabilities and equilibrium populations. However, despite transposes' adequate performance on this landscape, it can be shown from equation 10 that the estimated equilibrium populations are directly related to the amount of sampling in each state. This is not thought to be ideal with adaptive sampling, since continually sampling from a state will artificially inflate its estimated equilibrium population. Transpose does well on this particular landscape due to the relatively flat energy surface of the preferred path and would be less favorable with real landscapes. Therefore, we recommend the use of the normalize method with adaptive sampling data for its simplicity and accurate estimates of thermodynamics and kinetics.

4.4.7 Simulations of λ-Repressor Recapitulate the Patterns Observed for Simple Landscapes

Kinetic Monte Carlo simulations on physically inspired landscapes have provided valuable functional insight, but it is important to ensure that our conclusions hold true for the exploration of real protein landscapes. Protein conformational landscapes are hyper-dimensional and likely have many barriers, both enthalpic and entropic. Thus, we turn to using all-atom MD simulations

for three sampling methods: 1) long simulations, 2) massively parallel simulations, and 3) FAST-contacts (which ranks states by the fraction of native contacts that are present). Each method uses the same unfolded starting structure and simulation parameters, where extended details are described in the Methods. As for a model system, we chose to simulate a fast-folding variant of the λ-repressor.[64] Due to its speed of folding and size, the kinetics of this protein have been extensively studied, both experimentally and computationally, making it ideal for use when comparing sampling strategies.

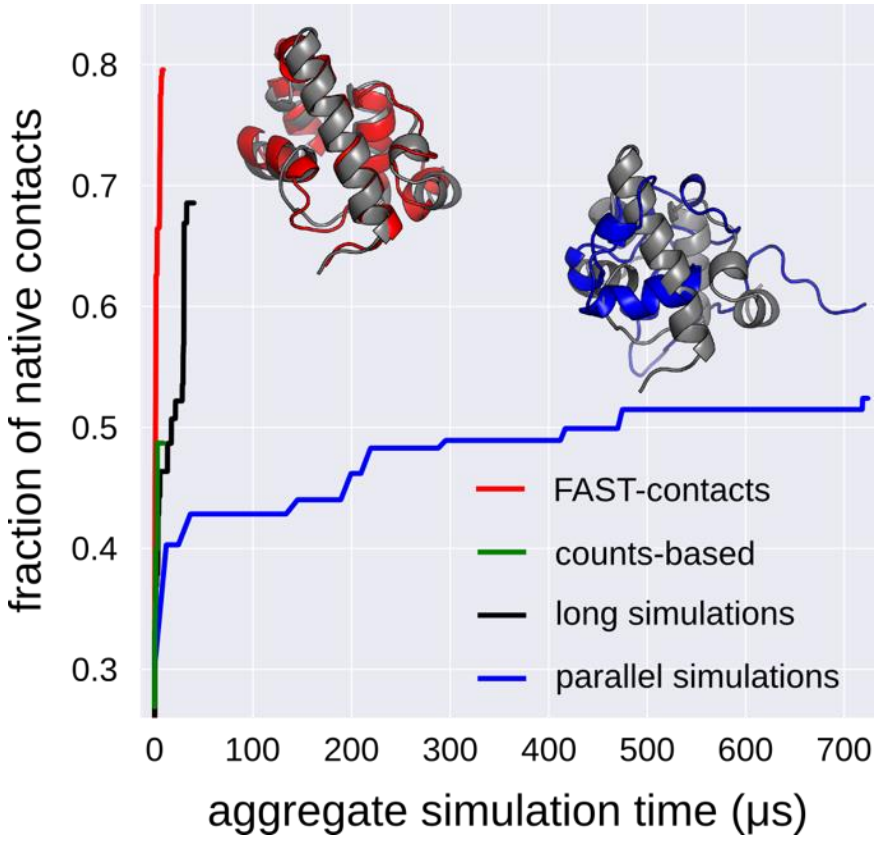

Figure 4.11: The largest observed fraction of native contacts as a function of aggregate simulation time for three equilibrium-based sampling methods. Simulation sets were generated from the same initial structure, which had a fraction of native contacts of 0.17 formed. Structures indicate the largest fraction of native contacts observed in a single run of FAST (red) or parallel simulations (blue) in contrast with the crystal structure (gray) (PDBID: 1LMB).

Unlike the simple landscapes in previous sections, all-atom MD simulations are computationally expensive and sample along vast conformational landscapes. As a consequence,

we cannot run thousands of iterations to robustly characterize the probability of discovering a particular state. Instead, we can compare the performance of each method by focusing on a more coarse-grained metric of interest, such as the computational time required to reach the folded state, as measured by the fraction of native contacts present.

Analysis of the three sampling methods reveals that adaptive sampling yields similar benefits to those found on our simple landscapes. Figure 4.11 shows the highest fraction of native contacts observed for each sampling method as a function of the aggregate simulation time. Remarkably, FAST-contacts folds the λ-repressor with ~4 μs of aggregate simulation, which is faster than its experimental folding time. By comparison, it takes nearly 40 μs of long simulations to achieve a similar level of foldedness. Furthermore, the massively parallelized simulations, with over 700 μs of aggregate simulation time, and counts-based adaptive sampling do not discover the folded state. Due to the high dimensionality of the λ-repressor, compared with our generated landscapes, counts-based adaptive sampling appears to hinder discovery of the folded state; low count states are continually discovered and selected in orthogonal directions to the fraction of native contacts. These results are in strong agreement with the discovery predictions from the landscapes in Figures 4.3 and 4.5.

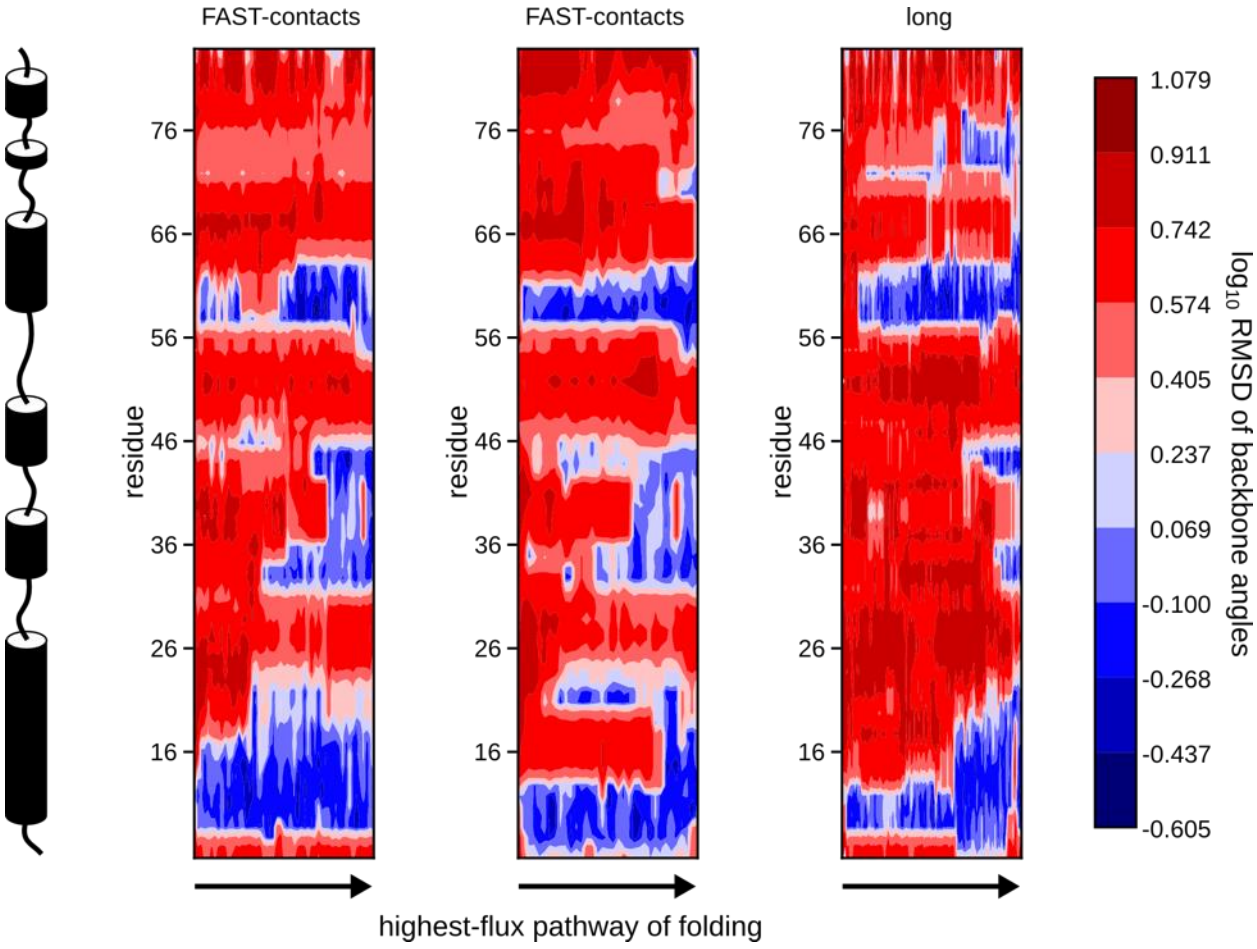

Figure 4.12: Analysis of predicted folding pathways for λ-repressor using the RMSD of each residues' backbone ϕ and ψ angles to the crystal structure (PDBID: 1LMB). Folding pathways are defined as an MSMs' highest-flux path from the starting state to the state with the largest fraction of native contacts. The time evolution of each residue's backbone RMSDs are shown along the x-axis for the predicted folding pathway from two separate runs of FAST-contacts and a single set of long simulations.

In addition to understanding the probability of observing a folded state, we are interested in the predicted folding pathways. However, the idea of characterizing a pathway for all-atom MD simulations is more complicated than on the theoretical landscapes; state-space is significantly larger, computational limitations prevent multiple trials to assess the stochasticity, and the optimal (human-intuitive) parameters to define a pathway are not straightforward. The long and parallel simulations require too much computational resources to gather statistics on, although we were able to generate five independent trials of FAST in a reasonable timeframe.

For the purposes of defining a pathway, others have successfully taken the approach of characterizing folding by the rate of formation of secondary structural elements.[65-67] Thus, we also aim to characterize the rate of secondary structure formation by determining each residues' root-mean-square deviation (RMSD) of backbone dihedrals from the crystal structure, for states along the predicted highest-flux pathway. We plot these deviations for two representative runs of FAST-contacts and the long simulations in Figure 4.12.

The predicted pathways for each of these methods are reasonably consistent with one another. FAST-contacts predominantly predicts the folding of helices 1 and 4 before helices 2 and 3. This is consistent with our prediction using the single set of long simulations. Additionally, this is what has been seen with previous simulation reports,[68,69] and hydrogen exchange experiments.[70] Interestingly, this is counter to the results from a Gō model, which has been previously used and describes helices 1-4 folding cooperatively.[71] This difference suggests that FAST-contacts is not simply an expensive Gō model.

4.5 Conclusions

We have presented a systematic comparison of different sampling strategies on a variety of representative energy landscapes. We first developed an analytic expression for the probability of discovering states on a landscape that depends on the number, length, and starting state of simulations. From this we find that long simulations have a higher probability of discovering states on landscapes with reduced dimensionality, though parallel simulations have a higher probability of discovering states as the dimensionality increases. To build upon this, we used kinetic Monte Carlo simulations on more complex landscapes to compare four sampling strategies (long simulations, parallel simulations, counts adaptive sampling, and FAST), which each reveal a unique state discovery signature. Understanding the differences in how these

sampling strategies discover states has provided insight into their advantages and disadvantages. Specifically, long simulations provide an unbiased estimates of transition paths, although requires significant computational resources compared to adaptive sampling or FAST and produces less accurate MSMs. Parallel simulations thoroughly explore around the starting state and provide excellent estimates of transition probabilities (for the states discovered) but are unlikely to explore distant regions of conformational space and may provide erroneous transition paths. Counts-based adaptive sampling discovers the most states along a variety of paths, although these states are likely to be unproductive for a given goal, especially on landscapes with large dimensionality.

Throughout our analysis, we have taken special interest in the performance of our recently developed goal-oriented sampling algorithm, FAST. On our simple landscapes, we find that FAST consistently has the highest probability of discovering a target state, predicts reasonable pathways, and provides the best estimates of transition probabilities for an entire MSM as well as of the true highest-flux pathway (Table S1). Furthermore, we demonstrate the utility of FAST using all-atom MD simulations of the λ-repressor. FAST produces an accurate folding pathway with an order of magnitude less aggregate simulation than long simulations, and orders of magnitude less than parallel simulations.

4.6 Methods

4.6.1 Generation and Simulation of Simple Landscapes

The three physically inspired potential energy landscapes were generated by selectively adding Gaussian potentials to an otherwise flat surface. These potential energy landscapes were then converted to a transition probability matrix using the following relations:

$$\zeta_{ij} = \begin{cases} e^{\varepsilon_i - \varepsilon_j} & if\ \varepsilon_i < \varepsilon_j \\ 1 & if\ \varepsilon_i \geq \varepsilon_j \end{cases}$$

for all j that are neighbors of i , and where ε_i is the potential energy of state n_i in units of $k_B T$. This can then be row-normalized to obtain,

$$T_{ij} = \frac{\zeta_{ij}}{\sum_j \zeta_{ij}}$$

Kinetic Monte Carlo simulations were then performed with this transition probability matrix for four sampling schemes: 1) long simulations, 2) parallel simulations, 3) counts-based adaptive sampling, and 4) FAST simulations. For each of the sampling schemes, 5,000 independent sets of simulations were generated, each with a total of 1,000 time-steps. For the long simulations, this consisted of 5,000 single trajectories, of 1,000 steps. Each set for the parallel simulations consisted of 25 trajectories with 40 steps.

Counts-based adaptive sampling and FAST both followed the same basic protocol: 1) generate 5 trajectories of 20 steps each from the initial state, 2) build an MSM, 3) rank states, 4) generate 5 more trajectories of 20 steps each from the top 5 states with the highest ranking, 5) repeat steps 2-4 for a total of 10 rounds. The difference between counts-based adaptive sampling and FAST is in the manner of ranking states between each round. For counts adaptive sampling, states were ranked by their observed counts in the MSM, with lower counts being more favorable. For FAST, we used the following ranking,

$$r_\phi(i) = \bar{\phi}(i) + \alpha\bar{\psi}(i) + \beta\chi(i) \qquad [11]$$

where $\bar{\phi}$ is the feature-scaled directed component (Euclidean distance to the target state), $\bar{\psi}$ is the feature scaled undirected component, χ is a similarity penalty, and α and βcontrol the weights of $\bar{\psi}$ and χ, respectively, as has been published previously.[21] Here, $\bar{\psi}(i)$ is taken to be the state counts and a value of 1 was used for both α and β. The directed component for each state on the landscapes was the grid distance to the target state. The similarity penalty for each state selected is defined with,

$$\chi(i) = \begin{cases} 0 & if\ N = 0 \\ \frac{1}{N}\sum_{j=1}^{N}\left(1 - e^{\frac{-d_{ij}^2}{2w^2}}\right) & if\ N > 0 \end{cases} \quad [12]$$

which is the average of the Gaussian weighted grid distance, d, from state $n_i$to the *N* states that have been selected for reseeding so far, where w is the Gaussian width (set to the clustering radius). Thus, selecting states proceeds as follows: 1) rank all states by the FAST ranking and select the top state, 2) add the similarity penalty and select the top-ranking state as the next state, 3) repeat step 2 until the desired number of states have been selected.

After generating the state trajectories on the landscapes from the sampling methods, state discover probabilities, pathway probabilities, and relative entropies were calculated. The discover probabilities were calculated by first using equation 1 to indicate if a state was discovered for each simulation set. These values for $D_{ij}^{\mathbf{K},\mathbf{M}}$ were then averaged over the 5,000 trials to determine the probability of discovering a state in the simulation set, $\mathrm{P}\left(D_{ij}^{\mathbf{K},\mathbf{M}} = 1\right)$. Similar to the discover probabilities, the pathway probabilities were calculated by averaging the

output of a selector function, over the simulation sets, that indicated if a state was present in the predicted highest-flux pathway. The highest-flux pathway for each simulation set was calculated using MSMBuilder.[72] The relative entropies of each state were calculated as the Kullback-Leibler divergence between the estimated conditional transition probabilities from that state and those of the underlying energy distribution:

$$D_{KL}^{i}(P_i||Q_i) = -\sum_i P_i \log\left(\frac{Q_i}{P_i}\right)$$

where D_{KL}^{i} is the relative entropy for state i, P_i is the i-th row of the true transition probability matrix, and Q_i is the i-th row of the transition probability matrix reconstructed from synthetic trajectories. The relative entropy of the entire MSM is a population weighted average of these values, as is described previously.[42,46] MSMs were constructed with either the MSMBuilder or PyEMMA software packages.[72-74]

4.6.2 Molecular Dynamics Simulations

Four sets of all-atom molecular dynamics simulations for the λ-repressor were generated: 1) 7,005 parallel simulations (103.4 ± 82.0 ns each), 2) 16 long simulations (2.5 μs each), 3) FAST-contacts simulations (30 rounds of 10 simulations per round, with 30 ns per simulation), and 4) counts-based adaptive sampling (30 rounds of 10 simulations per round, with 30 ns per simulation). Each of these simulations were run with Gromacs 5.1.1[75] using the AMBER03 force field with explicit TIP3P solvent.[76,77]

Each of these sets of simulations began from the same starting structure, which was prepared as follows. First, a linear structure of the λD14A mutant[64] was generated using the

VMD software package.[78] The linear structure was equilibrated for 1 ns at 420 K with OBC GBSA implicit solvent.[79] The final conformation was then placed in a dodecahedron box that extended 1.0 nm beyond the protein in any dimension, with a total of 46,450 atoms in the system. This system was then energy minimized with the steepest descent algorithm until the maximum force fell below 100 kJ/mol/nm using a step size of 0.01 nm and a cutoff distance of 1.2 nm for the neighbor list, Coulomb interactions, and van der Waals interactions.

For production runs, all bonds were constrained with the LINCS algorithm and virtual sites were used to allow a 4 fs time step. Cutoffs of 1.0 nm were used for the neighbor list, Coloumb interactions, and van der Waals interactions. The Verlet cutoff scheme was used for the neighbor list. The stochastic velocity rescaling (*v*-rescale) thermostat was used to hold the temperature at 360 K and conformations were stored every 50 ps.[80]

4.6.3 FAST Simulations

Five sets of FAST-contacts simulations were generated that each observed an independent folding trajectory for the λ-repressor. Each set of FAST-contacts consisted of 9 μs of aggregate simulation time: 30 rounds, of 10 simulations per round, where each simulation was 30 ns. Between each round, discrete states were generated by clustering atomic coordinates of backbone atoms using a *k*-centers algorithm based on RMSD between conformations until every cluster center had a radius less than 3.0 Å. States were selected for reseeding based on the ranking function and selection criterion described with equations 11 and 12. The similarity penalty used was RMSD between cluster centers, where the Gaussian width, w, was set to the clustering radius of 3.0 Å. The directed component to the FAST ranking was the feature scaled values of the fraction of native contacts, described elsewhere.[81]

4.6.4 MSM Construction and Analysis

MSMs were built of each simulation set using MSMBuilder.[72,73] The construction of each MSM followed the same basic protocol: 1) cluster conformations into discrete states, 2) count transitions between these states at a specified lag-time, and 3) generate each states' conditional transition probabilities. For the first step, atomic coordinates of backbone heavy atoms (CO, C_α, O, N) and C_β atoms were clustered with a *k*-centers clustering algorithm until every cluster center had a radius of less than 3.0 Å. A lag-time of 5 ns was used for counting transitions between states. Each states' conditional transition probabilities were computed using the normalize method with a prior-counts, as described with equations 8 and 9. Structural analysis was aided with the use of MDTraj.[82]

Bibliography

(1) Shaw, D. E.; Deneroff, M. M.; Dror, R. O.; Kuskin, J. S.; Larson, R. H.; Salmon, J. K.; Young, C.; Batson, B.; Bowers, K. J.; Chao, J. C.; Eastwood, M. P.; Gagliardo, J.; Grossman, J. P.; Ho, C. R.; Ierardi, D. J.; Kolossváry, I.; Klepeis, J. L.; Layman, T.; McLeavey, C.; Moraes, M. A.; Mueller, R.; Priest, E. C.; Shan, Y.; Spengler, J.; Theobald, M.; Towles, B.; Wang, S. C. Anton, a Special-Purpose Machine for Molecular Dynamics Simulation. *Communications of the ACM* **2008**, *51* (7), 91–97.

(2) Shaw, D. E.; Grossman, J. P.; Bank, J. A.; the, B. B. P. O.; 2014. Anton 2: Raising the Bar for Performance and Programmability in a Special-Purpose Molecular Dynamics Supercomputer. *dl.acm.org*.

(3) Takada, S. Go-Ing for the Prediction of Protein Folding Mechanisms. *Proc. Natl. Acad. Sci. U.S.A.* **1999**, *96* (21), 11698–11700.

(4) Go, N. Theoretical Studies of Protein Folding. *Annu. Rev. Biophys. Bioeng.* **1983**, *12* (1), 183–210.

(5) Sugita, Y.; Okamoto, Y. Replica-Exchange Molecular Dynamics Method for Protein Folding. *Chemical Physics Letters* **1999**, *314* (1-2), 141–151.

(6) Fukunishi, H.; Watanabe, O.; Takada, S. On the Hamiltonian Replica Exchange Method for Efficient Sampling of Biomolecular Systems: Application to Protein Structure Prediction. *The Journal of Chemical Physics* **2002**, *116* (20), 9058–9067.

(7) Faraldo-Gómez, J. D.; Roux, B. Characterization of Conformational Equilibria Through Hamiltonian and Temperature Replica-Exchange Simulations: Assessing Entropic and Environmental Effects. *J Comput Chem* **2007**, *28* (10), 1634–1647.

(8) Izrailev, S.; Stepaniants, S.; Isralewitz, B.; Kosztin, D.; Lu, H.; Molnar, F.; Wriggers, W.; Schulten, K. Steered Molecular Dynamics. In *Computational Molecular Dynamics: Challenges, Methods, Ideas*; Lecture Notes in Computational Science and Engineering; Springer Berlin Heidelberg: Berlin, Heidelberg, 1999; Vol. 4, pp 39–65.

(9) Isralewitz, B.; Gao, M.; Schulten, K. Steered Molecular Dynamics and Mechanical Functions of Proteins. *Current Opinion in Structural Biology* **2001**, *11* (2), 224–230.

(10) Voter, A. F. Hyperdynamics: Accelerated Molecular Dynamics of Infrequent Events. *Physical Review Letters* **1997**, *78* (20), 3908–3911.

(11) Hamelberg, D.; Mongan, J.; McCammon, J. A. Accelerated Molecular Dynamics: a Promising and Efficient Simulation Method for Biomolecules. *The Journal of Chemical Physics* **2004**, *120* (24), 11919–11929.

(12) Laio, A.; Parrinello, M. Escaping Free-Energy Minima. *Proc. Natl. Acad. Sci. U.S.A.* **2002**, *99* (20), 12562–12566.

(13) Laio, A.; Gervasio, F. L. Metadynamics: a Method to Simulate Rare Events and Reconstruct the Free Energy in Biophysics, Chemistry and Material Science. *Reports on Progress in Physics* **2008**, *71* (12), 126601.

(14) Perez, A.; MacCallum, J.; Dill, K. A. Meld: Modeling Peptide-Protein Interactions. *Biophysical Journal* **2013**, *104* (2), 399a.

(15) Perez, A.; MacCallum, J. L.; Dill, K. A. Accelerating Molecular Simulations of Proteins Using Bayesian Inference on Weak Information. *Proc. Natl. Acad. Sci. U.S.A.* **2015**, *112* (38), 11846–11851.

(16) Zheng, L.; Chen, M.; Yang, W. Random Walk in Orthogonal Space to Achieve Efficient Free-Energy Simulation of Complex Systems. *Proc. Natl. Acad. Sci. U.S.A.* **2008**, *105* (51), 20227–20232.

(17) Shirts, M. R.; Pande, V. S. Mathematical Analysis of Coupled Parallel Simulations. *Physical Review Letters* **2001**, *86* (22), 4983–4987.

(18) Shirts, M. COMPUTING: Screen Savers of the World Unite! *Science* **2000**, *290* (5498), 1903–1904.

(19) Bowman, G. R.; Pande, V. S.; Noé, F. Introduction and Overview of This Book. In *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*; Advances in Experimental Medicine and Biology; Springer Netherlands: Dordrecht, 2014; Vol. 797, pp 1–6.

(20) Hart, K. M.; Ho, C. M. W.; Dutta, S.; Gross, M. L.; Bowman, G. R. Modelling Proteins' Hidden Conformations to Predict Antibiotic Resistance. *Nature Communications* **2016**, *7*, 12965.

(21) Zimmerman, M. I.; Hart, K. M.; Sibbald, C. A.; Frederick, T. E.; Jimah, J. R.; Knoverek, C. R.; Tolia, N. H.; Bowman, G. R. Prediction of New Stabilizing Mutations Based on Mechanistic Insights From Markov State Models. *ACS Cent. Sci.* **2017**, *3* (12), 1311–1321.

(22) Hart, K. M.; Moeder, K. E.; Ho, C. M. W.; Zimmerman, M. I.; Frederick, T. E.; Bowman, G. R. Designing Small Molecules to Target Cryptic Pockets Yields Both Positive and Negative Allosteric Modulators. *PLOS ONE* **2017**, *12* (6), e0178678.

(23) Bowman, G. R.; Beauchamp, K. A.; Boxer, G.; Pande, V. S. Progress and Challenges in the Automated Construction of Markov State Models for Full Protein Systems. *The Journal of Chemical Physics* **2009**, *131* (12), 124101.

(24) Metzner, P.; Noé, F.; Schütte, C. Estimating the Sampling Error: Distribution of Transition Matrices and Functions of Transition Matrices for Given Trajectory Data. *Physical Review E* **2009**, *80* (2), 021106.

(25) Pande, V. S.; Beauchamp, K.; Bowman, G. R. Everything You Wanted to Know About Markov State Models but Were Afraid to Ask. *Methods* **2010**, *52* (1), 99–105.

(26) Chodera, J. D.; Noé, F. Markov State Models of Biomolecular Conformational Dynamics. *Current Opinion in Structural Biology* **2014**, *25*, 135–144.

(27) Mukherjee, S.; Pantelopulos, G. A.; Voelz, V. A. Markov Models of the <I>Apo</I>-MDM2 Lid Region Reveal Diffuse Yet Two-State Binding Dynamics and Receptor Poses for Computational Docking. *Scientific Reports 2016 6* **2016**, *6* (1), 31631.

(28) Zhou, G.; Pantelopulos, G. A.; Mukherjee, S.; Voelz, V. A. Bridging Microscopic and Macroscopic Mechanisms of P53-MDM2 Binding with Kinetic Network Models. *Biophysical Journal* **2017**, *113* (4), 785–793.

(29) Plattner, N.; Doerr, S.; De Fabritiis, G.; Noé, F. Complete Protein–Protein Association Kinetics in Atomic Detail Revealed by Molecular Dynamics Simulations and Markov Modelling. *Nature Chemistry 2017 9:10* **2017**, *9* (10), 1005–1011.

(30) Wang, W.; Cao, S.; Zhu, L.; Huang, X. Constructing Markov State Models to Elucidate the Functional Conformational Changes of Complex Biomolecules. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2018**, *8* (1), e1343.

(31) Husic, B. E.; Pande, V. S. Markov State Models: From an Art to a Science. *Journal of the American Chemical Society* **2018**, *140* (7), 2386–2396.

(32) Hinrichs, N. S.; Pande, V. S. Calculation of the Distribution of Eigenvalues and Eigenvectors in Markovian State Models for Molecular Dynamics. *The Journal of Chemical Physics* **2007**, *126* (24), 244101.

(33) Bowman, G. R.; Ensign, D. L.; Pande, V. S. Enhanced Modeling via Network Theory: Adaptive Sampling of Markov State Models. *J. Chem. Theory Comput.* **2010**, *6* (3), 787–794.

(34) Weber, J. K.; Pande, V. S. Characterization and Rapid Sampling of Protein Folding Markov State Model Topologies. *J. Chem. Theory Comput.* **2011**, *7* (10), 3405–3411.

(35) Doerr, S.; De Fabritiis, G. On-the-Fly Learning and Sampling of Ligand Binding by High-Throughput Molecular Simulations. *J. Chem. Theory Comput.* **2014**, *10* (5), 2064–2069.

(36) Voelz, V. A.; Elman, B.; Razavi, A. M.; Zhou, G. Surprisal Metrics for Quantifying Perturbed Conformational Dynamics in Markov State Models. *J. Chem. Theory Comput.* **2014**, *10* (12), 5716–5728.

(37) Bacci, M.; Vitalis, A.; Caflisch, A. A Molecular Simulation Protocol to Avoid Sampling Redundancy and Discover New States. *Biochimica et Biophysica Acta (BBA) - General Subjects* **2015**, *1850* (5), 889–902.

(38) Kukharenko, O.; Sawade, K.; Steuer, J.; Peter, C. Using Dimensionality Reduction to Systematically Expand Conformational Sampling of Intrinsically Disordered Peptides. *J. Chem. Theory Comput.* **2016**, *12* (10), 4726–4734.

(39) Doerr, S.; Harvey, M. J.; Noé, F.; De Fabritiis, G. HTMD: High-Throughput Molecular Dynamics for Molecular Discovery. *J. Chem. Theory Comput.* **2016**, *12* (4), 1845–1852.

(40) Sultan, M. M.; Pande, V. S. Decision Functions From Supervised Machine Learning Algorithms as Collective Variables for Accelerating Molecular Simulations. February 28, 2018.

(41) Noé, F.; Banisch, R.; Clementi, C. Commute Maps: Separating Slowly Mixing Molecular Configurations for Kinetic Modeling. *ACS Publications* **2016**, *12* (11), 5620–5630.

(42) Zimmerman, M. I.; Bowman, G. R. FAST Conformational Searches by Balancing Exploration/Exploitation Trade-Offs. *J. Chem. Theory Comput.* **2015**, *11* (12), 5747–5757.

(43) Zimmerman, M. I.; Bowman, G. R. How to Run FAST Simulations. *Methods in Enzymology* **2016**, *578*, 213–225.

(44) Bowman, G. R.; Geissler, P. L. Equilibrium Fluctuations of a Single Folded Protein Reveal a Multitude of Potential Cryptic Allosteric Sites. *Proc. Natl. Acad. Sci. U.S.A.* **2012**, *109* (29), 11681–11686.

(45) Grinstead, C. M.; Snell, J. L. *Introduction to Probability*; 2012.

(46) Bowman, G. R.; Pande, V. S. Protein Folded States Are Kinetic Hubs. *Proc. Natl. Acad. Sci. U.S.A.* **2010**, *107* (24), 10890–10895.

(47) Wang, Y.; Lv, J.; Zhu, L.; Ma, Y. Crystal Structure Prediction via Particle-Swarm Optimization. *Phys. Rev. B* **2010**, *82* (9), 094116.

(48) Moult, J. A Decade of CASP: Progress, Bottlenecks and Prognosis in Protein Structure Prediction. *Current Opinion in Structural Biology* **2005**, *15* (3), 285–289.

(49) Floudas, C. A.; Fung, H. K.; McAllister, S. R.; Mönnigmann, M.; Rajgaria, R. Advances in Protein Structure Prediction and De Novo Protein Design: a Review. *Chemical Engineering Science* **2006**, *61* (3), 966–988.

(50) Voelz, V. A.; Bowman, G. R.; Beauchamp, K.; Pande, V. S. Molecular Simulation of Ab Initio Protein Folding for a Millisecond Folder NTL9(1−39). *Journal of the American Chemical Society* **2010**, *132* (5), 1526–1528.

(51) Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; Shaw, D. E. How Fast-Folding Proteins Fold. *Science* **2011**, *334* (6055), 517–520.

(52) Leopold, P. E.; Montal, M.; Onuchic, J. N. Protein Folding Funnels: a Kinetic Approach to the Sequence-Structure Relationship. *Proc. Natl. Acad. Sci. U.S.A.* **1992**, *89* (18), 8721–8725.

(53) Dill, K. A.; Chan, H. S. From Levinthal to Pathways to Funnels. *Nature Structural & Molecular Biology 1997 4:1* **1997**, *4* (1), 10–19.

(54) Whitford, P. C.; Onuchic, J. N. What Protein Folding Teaches Us About Biological Function and Molecular Machines. *Current Opinion in Structural Biology* **2015**, *30*, 57–62.

(55) E, W.; Ren, W.; Vanden-Eijnden, E. String Method for the Study of Rare Events. *Phys. Rev. B* **2002**, *66* (5), 052301.

(56) Maragliano, L.; Fischer, A.; Vanden-Eijnden, E.; Ciccotti, G. String Method in Collective Variables: Minimum Free Energy Paths and Isocommittor Surfaces. *The Journal of Chemical Physics* **2006**, *125* (2), 024106.

(57) Albert C Pan; Deniz Sezer, A.; Benoît Roux. *Finding Transition Pathways Using the String Method with Swarms of Trajectories*; American Chemical Society, 2008; Vol. 112, pp 3432–3440.

(58) Leahy, C. T.; Kells, A.; Hummer, G.; Buchete, N.-V.; Rosta, E. Peptide Dimerization-Dissociation Rates From Replica Exchange Molecular Dynamics. *The Journal of Chemical Physics* **2017**, *147* (15), 152725.

(59) Nüske, F.; Wu, H.; Prinz, J.-H.; Wehmeyer, C.; Clementi, C.; Noé, F. Markov State Models From Short Non-Equilibrium Simulations—Analysis and Correction of Estimation Bias. *The Journal of Chemical Physics* **2017**, *146* (9), 094104.

(60) Trendelkamp-Schroer, B.; Noé, F. Efficient Bayesian Estimation of Markov Model Transition Matrices with Given Stationary Distribution. *The Journal of Chemical Physics* **2013**, *138* (16), 164113.

(61) Trendelkamp-Schroer, B.; Noé, F. Efficient Estimation of Rare-Event Kinetics. *Phys. Rev. X* **2016**, *6* (1), 011009.

(62) Jaeger, H. Observable Operator Models for Discrete Stochastic Time Series. *http://dx.doi.org/10.1162/089976600300015411* **2006**, *12* (6), 1371–1398.

(63) Wu, H.; Prinz, J.-H.; Noé, F. Projected Metastable Markov Processes and Their Estimation with Observable Operator Models. *The Journal of Chemical Physics* **2015**, *143* (14), 144101.

(64) Yang, W. Y.; Gruebele, M. Folding Λ-Repressor at Its Speed Limit. *Biophysical Journal* **2004**, *87* (1), 596–608.

(65) Hu, W.; Walters, B. T.; Kan, Z.-Y.; Mayne, L.; Rosen, L. E.; Marqusee, S.; Englander, S. W. Stepwise Protein Folding at Near Amino Acid Resolution by Hydrogen Exchange and Mass Spectrometry. *Proc. Natl. Acad. Sci. U.S.A.* **2013**, *110* (19), 7684–7689.

(66) Piana, S.; Lindorff-Larsen, K.; Shaw, D. E. How Robust Are Protein Folding Simulations with Respect to Force Field Parameterization? *Biophysical Journal* **2011**, *100* (9), L47–L49.

(67) Henry, E. R.; Best, R. B.; Eaton, W. A. Comparing a Simple Theoretical Model for Protein Folding with All-Atom Molecular Dynamics Simulations. *Proc. Natl. Acad. Sci. U.S.A.* **2013**, *110* (44), 17880–17885.

(68) Liu, Y.; Strümpfer, J.; Freddolino, P. L.; Gruebele, M.; Schulten, K. Structural Characterization of Λ-Repressor Folding From All-Atom Molecular Dynamics Simulations. *ACS Publications* **2012**, *3* (9), 1117–1123.

(69) Bowman, G. R.; Voelz, V. A.; Pande, V. S. Atomistic Folding Simulations of the Five-Helix Bundle Protein Λ6−85. *Journal of the American Chemical Society* **2010**, *133* (4), 664–667.

(70) Yu, W.; Baxa, M. C.; Gagnon, I.; Freed, K. F.; Sosnick, T. R. Cooperative Folding Near the Downhill Limit Determined with Amino Acid Resolution by Hydrogen Exchange. *Proc. Natl. Acad. Sci. U.S.A.* **2016**, *113* (17), 4747–4752.

(71) Shoemaker, B. A.; Wang, J.; Wolynes, P. G. Exploring Structures in Protein Folding Funnels with Free Energy Functionals: the Transition State Ensemble. *Journal of Molecular Biology* **1999**, *287* (3), 675–694.

(72) Harrigan, M. P.; Sultan, M. M.; Hernández, C. X.; Husic, B. E.; Eastman, P.; Schwantes, C. R.; Beauchamp, K. A.; McGibbon, R. T.; Pande, V. S. MSMBuilder: Statistical Models for Biomolecular Dynamics. *Biophysical Journal* **2017**, *112* (1), 10–15.

(73) Beauchamp, K. A.; Bowman, G. R.; Lane, T. J.; Maibaum, L.; Haque, I. S.; Pande, V. S. MSMBuilder2: Modeling Conformational Dynamics on the Picosecond to Millisecond Scale. *J. Chem. Theory Comput.* **2011**, *7* (10), 3412–3419.

(74) Scherer, M. K.; Trendelkamp-Schroer, B.; Paul, F.; Pérez-Hernández, G.; Hoffmann, M.; Plattner, N.; Wehmeyer, C.; Prinz, J.-H.; Noé, F. PyEMMA 2: a Software Package for Estimation, Validation, and Analysis of Markov Models. *J. Chem. Theory Comput.* **2015**, *11* (11), 5525–5542.

(75) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High Performance Molecular Simulations Through Multi-Level Parallelism From Laptops to Supercomputers. *SoftwareX* **2015**, *1-2*, 19–25.

(76) Duan, Y.; Wu, C.; Chowdhury, S.; Lee, M. C.; Xiong, G.; Zhang, W.; Yang, R.; Cieplak, P.; Luo, R.; Lee, T.; Caldwell, J.; Wang, J.; Kollman, P. A Point-Charge Force Field for Molecular Mechanics Simulations of Proteins Based on Condensed-Phase Quantum Mechanical Calculations. *J Comput Chem* **2003**, *24* (16), 1999–2012.

(77) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. Comparison of Simple Potential Functions for Simulating Liquid Water. *The Journal of Chemical Physics* **1998**, *79* (2), 926–935.

(78) VMD: Visual Molecular Dynamics. *Journal of Molecular Graphics* **1996**, *14* (1), 33–38.

(79) Onufriev, A.; Bashford, D.; Case, D. A. Exploring Protein Native States and Large-Scale Conformational Changes with a Modified Generalized Born Model. *Proteins: Structure, Function, and Bioinformatics* **2004**, *55* (2), 383–394.

(80) Bussi, G.; Donadio, D.; Parrinello, M. Canonical Sampling Through Velocity Rescaling. *The Journal of Chemical Physics* **2007**, *126* (1), 014101.

(81) Best, R. B.; Hummer, G.; Eaton, W. A. Native Contacts Determine Protein Folding Mechanisms in Atomistic Simulations. *Proc. Natl. Acad. Sci. U.S.A.* **2013**, *110* (44), 17874–17879.

(82) McGibbon, R. T.; Beauchamp, K. A.; Harrigan, M. P.; Klein, C.; Swails, J. M.; Hernández, C. X.; Schwantes, C. R.; Wang, L.-P.; Lane, T. J.; Pande, V. S. MDTraj: a Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophysical Journal* **2015**, *109* (8), 1528–1532.

Chapter 5

Prediction of New Stabilizing Mutations Based on Mechanistic Insights from Markov State Models

5.1 Preamble

This chapter is adapted from the following article: Zimmerman, M.I., Hart., K.M., Sibbald, C.A., Frederick, T.E., Jimah, J.R., Knoverek, C.R., Tolia, N.H., and Bowman, G.R. (2017). "Prediction of New Stabilizing Mutations Based on Mechanistic Insights from Markov State Models", *American Chemical Society Central Science*, 3 (12), 1311-1321

5.2 Introduction

Studying the evolution of antibiotic resistance has provided many insights into how proteins acquire new functions, but the mechanistic basis for how mutations alter a protein's activity and stability often remains unclear. For example, studying how bacteria evolve variants of TEM β-lactamase that confer resistance to new antibiotics by degrading these drugs has revealed that many of the mutations that give rise to new functions are destabilizing. Therefore, it is common for proteins to acquire one or more mutations that alter their function and then to acquire additional mutations that restore stability.[1] M182T is one such stabilizing mutation in TEM, and it has appeared in numerous clinical isolates and directed evolution experiments.[2-4]

This substitution occurs far from the active site (Figure 5.1A) and, on its own, has little effect on TEM's activity. It is often called a global suppressor because of its ability to counterbalance the

destabilizing effects of a wide variety of other substitutions that do alter TEM's activity.[3] Despite over two decades of work on this variant, the mechanism of stabilization by M182T is not understood well enough to predict new stabilizing mutations. Elucidating the mechanism underlying this stabilization would provide a basis for predicting other global suppressors and eventually developing quantitative design principles.

A mechanistic understanding of how M182T stabilizes TEM remains elusive because of a lack of methods that provide both a detailed structural model of the relevant species and their relative populations. Spectroscopic studies have revealed that TEM-1, which we will refer to as wild-type TEM, populates at least three states at equilibrium: a native state (N), an intermediate (I), and an unfolded state (U).[5] Introducing the M182T substitution appears to reduce the number of equilibrium states to two.[4] However, there is debate over whether this results from M182T stabilizing the native state or destabilizing the intermediate.[6] Moreover, these spectroscopic experiments do not directly provide a structural model for how M182T shifts the relative populations of these states. Two competing structural models based on crystallographic data have been proposed to explain M182T's ability to stabilize the enzyme. In the first crystal structure, Thr182 is poised to form a hydrogen bond between TEM's two structural domains, interacting with the backbone carbonyls of Glu63 and Glu64 in an adjacent loop[7] (Figure 5.1B). Therefore, it was proposed that M182T stabilizes TEM by strengthening the interface between the α-helix and β-sheet domains. However, in a later structure, Thr182 is oriented to form hydrogen bonds with the backbone amide of Ala185 (Figure 5.1C).[1] Based on this model, it was proposed that M182T stabilizes the protein by forming a hydrogen bond between its sidechain and an unfulfilled backbone donor at the end of helix 9 in a classic N-capping interaction. In all likelihood, both of these structures are present at thermal equilibrium, but it is impossible to

conclude which, if either of these interactions, plays a dominant role in stabilizing TEM from the crystallographic data.

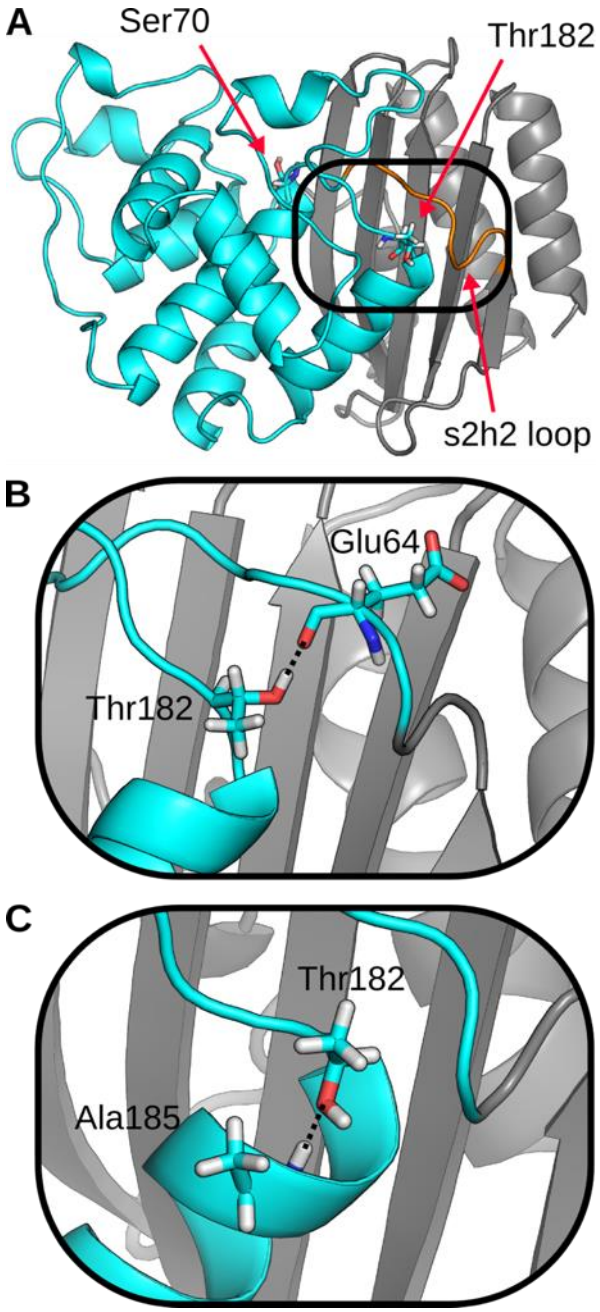

Figure 5.1: Representative structures of TEM that highlight two potential mechanisms for stabilization by Thr182. (A) Crystal structure of TEM with mutation M182T (PDB 1JWP). The backbone of the α-helix domain (cyan), β-sheet domain (gray), and s2h2 loop (orange) are represented as a cartoon. Active site residue, Ser70, and the stabilizing mutation, Thr182, are shown in sticks. (B) A structure where Thr182 hydrogen bonds to the s2h2 loop. (C) A second structure where Thr182 caps helix 9.

Here, we employ Markov state models (MSMs)[8-10] to understand how M182T shifts the distribution of different structures that TEM adopts. These models provide a quantitative description of a protein's thermodynamics and kinetics by defining its structural states and the rates of transitioning between them. We have previously compared MSMs of variants that alter TEM's specificity to understand how they change the proteins function.[11] In this study, we compare MSMs of the wild-type and M182T variants to infer how M182T stabilizes TEM. We then predict the effects of other mutations, including new global suppressor mutations, and experimentally test our predictions using a combination of spectroscopic measurements of protein stability, nuclear magnetic resonance (NMR) measurements of chemical shifts, a crystal structure, and *in vivo* measurements of the fitness of bacteria expressing our newly designed TEM variants.

5.3 Results

5.3.1 M182T Stabilizes the Native State

Uncertainty over whether M182T stabilizes the native state or destabilizes the intermediate stems from the limited ability of any one spectroscopic observable to clearly distinguish all three thermodynamic states. For example, circular dichroism (CD) fails to adequately capture M182T's intermediate state. By CD, there are three distinguishable states for wild-type[5] but only two for M182T[4] (Figure 5.2A); however, the dependence of M182T's native-state stability on denaturant, as reflected in its *m*-value, is shallower than expected for a protein of its size.[12] This indicates that like wild-type, M182T likely populates more than two states at equilibrium,[13] rendering a two-state model insufficient. Fluorescence also fails to capture all three thermodynamic states for both wild-type and M182T (Figure 5.2B). Previous studies of β-

lactamases have established that the intermediate state has the same fluorescence as the unfolded state,[14] so fluorescence captures only the transition between the native and intermediate states.

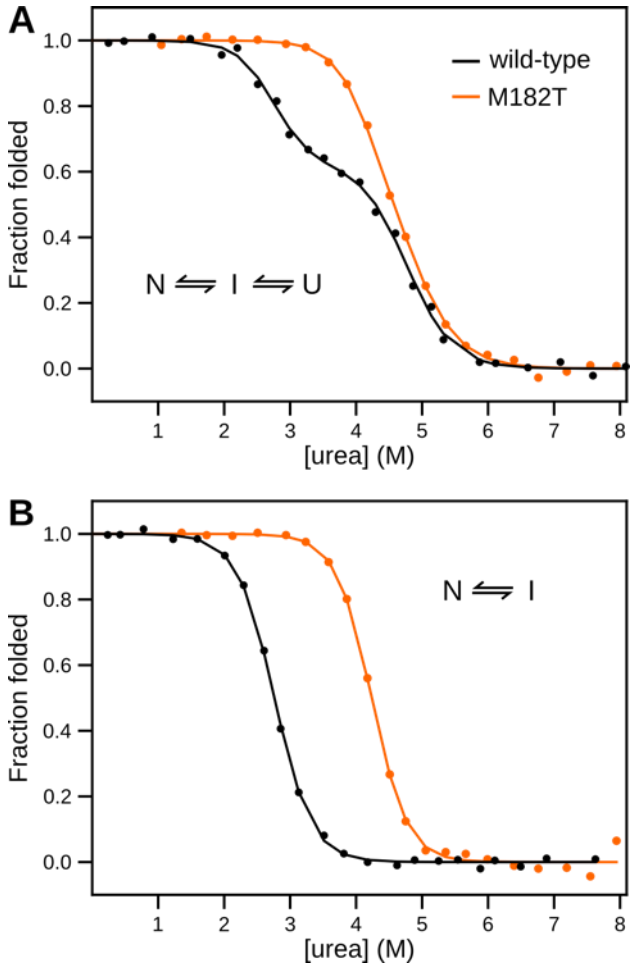

Figure 5.2: Chemical melts of TEM. Shown are the fractions of folded protein for wild-type TEM (black) and TEM M182T (orange) as a function of [urea]. (A) Monitoring circular dichroism signal. (B) Monitoring intrinsic fluorescence at 340 nm.

To overcome the limitations of a single spectroscopic observable, we performed global fits to the fluorescence and CD data for each variant, assuming that the first transition observed by CD is the same as that observed by fluorescence. Doing so allows us to disambiguate the two transitions captured by CD by leveraging the single transition captured by fluorescence. Our global fits reveal that M182T stabilizes the native state without destabilizing the intermediate. The free energy difference between the native and intermediate states of M182T is 3.3 kcal/mol

greater than that for wild-type (Table 5.1). In contrast, the free energy differences between the intermediate and unfolded states are the same, within error, for both variants.

Table 5.1. Stabilities of TEM β-lactamase variants*

	ΔG_{un} (*kcal mol^{-1}*)	m_{un} (*kcal mol^{-1} M^{-1}*)	$\Delta G_{in}^{\dagger}$ (*kcal mol^{-1}*)	$m_{in}^{\dagger}$ (*kcal mol^{-1} M^{-1}*)	$\Delta G_{ui}^{\dagger}$ (*kcal mol^{-1}*)	$m_{ui}^{\ddagger}$ (*kcal mol^{-1} M^{-1}*)
wild-type	14.3 ± 0.3	3.8 ± 0.2	6.0 ± 0.1	2.1 ± 0.1	8.3 ± 0.1	1.7 ± 0.1
M182T	17.7 ± 0.4	4.1 ± 0.2	10.0 ± 0.6	2.4 ± 0.2	7.8 ± 0.2	1.7, fixed
M182S	18.5 ± 0.5	4.4 ± 0.1	10.6 ± 0.5	2.7 ± 0.1	7.9 ± 0.4	1.7, fixed
M182V	13.5 ± 0.3	3.8 ± 0.1	5.4 ± 0.2	2.1 ± 0.1	8.2 ± 0.1	1.7, fixed
M182N	13.7 ± 0.5	3.8 ± 0.1	5.9 ± 0.4	2.1 ± 0.1	7.8 ± 0.4	1.7, fixed

*All measurements were repeated three times. Errors are standard deviations.
†Determined using a global fit of fluorescence data to a two-state (I-N) model and CD data to a three-state (U-I-N) model using the linear extrapolation method (see *Methods*).
‡The value for m_{ui} was fixed to the average value determined for wild-type. The addition of m_{ui} as a parameter did not significantly improve the quality of the fit, as determined by F-tests (values in the range of $1x10^{-10}$ – $1x10^{-7}$, see *Appendix 2*).

5.3.2 M182T Stabilizes Helix 9

Given our assumption that M182T does not affect the unfolded ensemble, and thus, primarily stabilizes the native state, we reason that it should be possible to infer the mechanism of stabilization from analysis of native-state ensembles. To accomplish this, we use MSMs to provide an atomically-detailed representation of conformational heterogeneity in the native state that is currently unavailable to many experimental techniques. Doing so enables us to quantify the probabilities of various interactions in a manner that is not possible with the static structures from techniques like crystallography. Furthermore, by identifying interactions that are formed in M182T's native-state ensemble but not that of wild-type TEM we can narrow down the secondary effects of this mutation.

To efficiently identify the interactions that Thr182 forms, we employed our FAST simulation method[15,16] to build MSMs of the wild-type and M182T variants of TEM. FAST is a goal-oriented adaptive sampling method in which we 1) run a batch of simulations, 2) build an MSM from all the simulation data collected so far, 3) rank each state with a function that favors

states that optimize some geometric criteria, as well as a statistical criterion that favors poorly sampled states, 4) run a new batch of simulations from the highest ranked states, 5) repeat steps 2-4 for some number of iterations, and 6) build a final MSM from all the simulation data. For this study, we sought to maximize the RMSD from the starting structure to maximize the number of different structures identified by the final model. We have previously established that FAST captures rare events with one or two orders of magnitude less simulation data than conventional molecular dynamics simulations.[15] Therefore, the 6.5 microseconds of simulation data we collected for each variant should be sufficient to construct a quantitatively predictive map of the native-state ensemble.[17]

Analysis of our FAST simulations reveals that M182T prefers to N-cap helix 9. This conclusion comes from quantifying the probabilities of all the different contacts Thr182's sidechain can form. Doing so reveals that Thr182 predominantly caps helix 9 by forming a hydrogen bond with Ala185 with a probability of 0.72±0.023. Thr182 also forms a hydrogen bond with the backbone carbonyl of Glu64 with a probability of 0.12±0.017. Thus, we observe both conformations captured in the two competing crystal structures. The probabilities of other contacts, such as the hydrogen bond with the backbone carbonyl of Glu63, are negligible.

While it is tempting to conclude that capping is sufficient for global stabilization, we instead propose that the stability of helix 9 is a better predictor of TEM's stability. Our model's distinction between capping and helix stability was motivated by the observation that other residues capable of N-capping have not been observed at position 182 either in clinical isolates or in directed evolution studies.[2] It might seem intuitive that capping would stabilize helix 9, but, in the next section, we defy this intuition by identifying a residue that caps without conferring global stabilization. Previous work on the folding of β-lactamases provides a foundation for our

model by suggesting that the α-helix domain is largely folded in the intermediate state but the β-sheet domain is unstructured.[18] Taking inspiration from this model, we propose that helix 9 is unstructured in the intermediate state. In our model, M182T stabilizes helix 9's native conformation and reduces its conformational heterogeneity. Because this helix is an important part of the interface between the α-helix and β-sheet domains, we propose that stabilizing the helix stabilizes the entire interface between the two domains, thereby stabilizing TEM's native conformation. Helix 9 being unstructured in the intermediate state in our model is consistent with the fact that the free energy difference between the unfolded and intermediate states is unaffected by M182T (Table 5.1).

As a proxy for assessing the stability of helix 9, we quantify the distribution of distances between its backbone hydrogen bonding partners. Our simulations capture transitions between weak and moderate hydrogen bonds. Following past work,[19,20] we define a moderate hydrogen bond as having a hydrogen bond acceptor to hydrogen distance less than 2.2 Å, where a weak hydrogen bond has a distance between 2.2-2.5 Å. Assuming that weak hydrogen bonds are more likely to break on longer timescales, we can infer M182T's effect on helix stability by comparing the local fluctuations of its hydrogen bonds to that of wild-type.

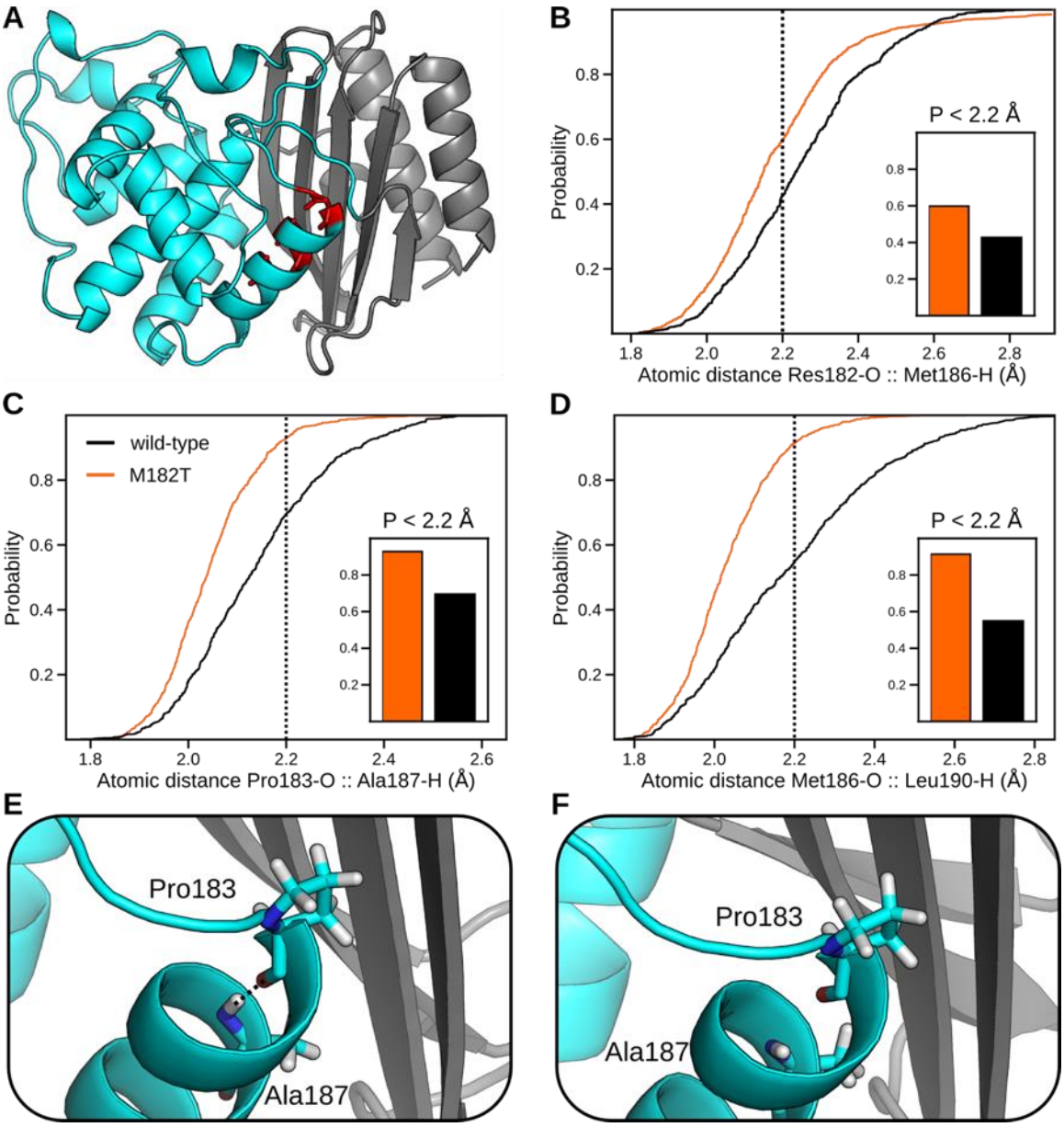

Figure 5.3: Effect of M182T on the stability of helix 9, as judged by the distributions of distances between hydrogen-bonding partners. (A) Structure highlighting hydrogen-bonding partners residue 182 and Met186, Pro183 and Ala187, and Met186 and Leu190, which are colored red. (B-D) Cumulative distribution functions of the hydrogen-bonding partners listed in (A) for wild-type (black) and M182T (orange). These plots indicate the probability of observing an atomic distance less than the specified value. Our cutoff distance for moderate hydrogen bonds, 2.2 Å, is shown as a dotted line. Probabilities of moderate hydrogen bonds for each pair are shown in the inset. (E, F) Representative structures, from our MSMs, of helix 9 with a moderate hydrogen bond (observed in M182T) and a broken hydrogen bond (observed in wild-type). The backbone of the α-helix domain (cyan) and β-sheet domain (gray) are represented as a cartoon.

Quantifying the distance distributions of hydrogen bonds reveals that M182T stabilizes helix 9. M182T increases the probability of moderate strength hydrogen bonds between three

pairs of residues: 182-186, 183-187, and 186-190 (Figure 5.3). As stated above, since moderate strength hydrogen bonds are less likely to break we conclude that they are stabilizing. The distributions for other hydrogen bonds are not altered significantly by the M182T substitution. Interestingly, all the residues with increased hydrogen bonding strength reside on the face of the helix that points into the core of the protein, along the interface between the two domains (Figure 5.3A).

5.3.3 Helix Capping Alone is Not Sufficient to Stabilize the Native State

Mutagenesis at position 182 presents a valuable opportunity to test our model and probe why other mutations may or may not stabilize helix 9. In particular, studying other capping residues could reveal that capping is sufficient for stabilization, or alternatively, lead to the identification of other stabilizing factors. To discover these factors, we modeled mutations at position 182, predicted their stability relative to wild-type, and performed experimental tests.

We selected three alternative substitutions at position 182 to study. First, we selected M182N because asparagine is the most frequently observed N-capping residue in proteins with known structures[21] and the most stabilizing N-cap,[22] so one might expect it to be even more stabilizing than threonine. Second, we chose M182S because serine has a hydroxyl group that is analogous to threonine's, so it might form a similar capping interaction and have a comparable effect on stability. Third, we modeled M182V because valine mimics threonine sterically but lacks the ability to cap since it has a methyl group instead of a hydroxyl group. Therefore, comparing M182V with the other substitutions could help elucidate the relative importance of capping and sterics.

Consistent with our expectations, MSMs show that M182S and M182N cap helix 9 (Figure A.2.1). The probabilities that Ser182 and Asn182 cap by hydrogen bonding with Ala185

are 0.61±0.02 and 0.79±0.02, respectively. Each residue can also hydrogen bond with Glu64 in the s2h2 loop. M182S forms this interaction with a probability of 0.22±0.02 and M182N forms this interaction with a probability of 0.59±0.03. Notably, M182N has the ability to simultaneously cap helix 9 and interact with the s2h2 loop. Therefore, if capping were sufficient to predict helix stability we would expect that M182S and M182N would be stabilizing mutations, while M182V would not.

Quantifying the degree that each of these substitutions stabilizes helix 9 suggests that capping is not sufficient to stabilize TEM. Comparing the probabilities of moderate hydrogen bonds along the length of helix 9 reveals that M182S is stabilizing, whereas M182V and M182N are not (Figure A.2.2). The fact that M182N is not stabilizing is particularly surprising given that it caps as frequently as M182T and can simultaneously hydrogen bond with Glu64. If true, this would highlight the predictive power of our model, since it defies biochemical intuition. To test these predictions, we experimentally measured the stability of each TEM variant.

Free energy differences of each variant, derived from chemical melts, and a crystal structure are consistent with our model for global stability. As predicted, M182S stabilizes TEM to a similar extent to M182T (Table 5.1, Figure A.2.3). Furthermore, M182N and M182V are not stabilizing. To provide additional evidence that M182N caps helix 9 without conferring stability, we solved a crystal structure of this variant to 2.0 Å resolution (Figure A.2.4, Table A.2.5). This structure further supports our prediction that Asn182 caps, since the x-ray density around position 182 is best fit with a rotamer that caps helix 9 by hydrogen bonding with Ala185 (Figure A.2.4).

Understanding why M182N does not stabilize TEM despite its strong propensity for capping helix 9 presents a valuable opportunity for dissecting the mechanisms of stabilization by

M182T and M182S. Given that capping is generally stabilizing, we reasoned that Asn182 must form other interactions that counterbalance this effect. If this is true, we would expect the stability of helix 9 in isolation from the rest of the protein to correlate with the propensity of residue 182 to cap the helix. To test this prediction, we simulated helix 9 (residues 181-197) with each of the following residues at position 182: threonine, serine, asparagine, valine, and methionine.

Probing the helical propensity of each variant suggests that capping is sufficient to stabilize helix 9 in isolation. We quantify helical propensity by measuring the probability that at least 80% of the residues adopt a conformation in the α-helical region of the Ramachandran plot. We find that each of the helix 9 variants with an N-terminal capping residue (Thr, Ser, or Asn) at position 182 have a similar helical propensity of ~45% (Figure A.2.5). Furthermore, variants that lack a capping residue have much lower helical propensity (12-23% for Val and Met). These trends remain the same if the cutoff for considering a structure helical is changed. Therefore, it appears that any capping interaction will stabilize helix 9 in isolation, consistent with our hypothesis that Asn182 must be forming other destabilizing interactions in the context of the full-length protein. To determine the reason that M182N does not stabilize helix 9 in the context of the full sequence, we next examine the differences in Asn182's conformations between the full-length sequence and in the isolated helix.

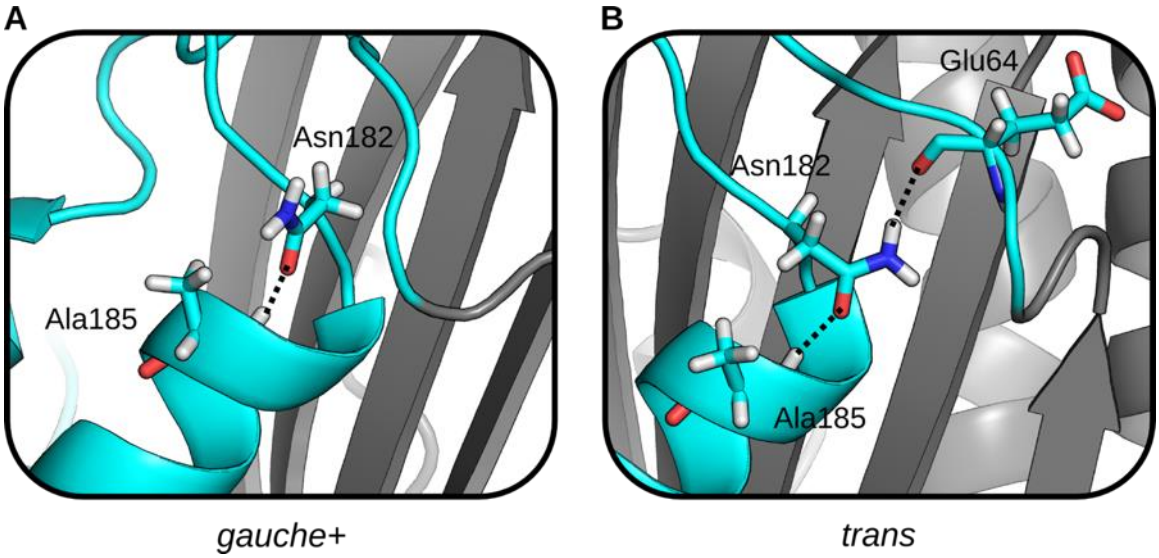

Figure 5.4: Two commonly observed sidechain conformations of Asn182 in MSMs, which can be characterized by their χ_1 angle. (A) A representative structure with Asn182 in the gauche+ conformation. The sidechain amine points out into solution. (B) A representative structure with Asn182 in the trans conformation. The sidechain amine hydrogen-bonds with Glu64 in the s2h2 loop. In both conformations, the sidechain hydrogen-bonds with Ala185. The backbone of the α-helix domain (cyan) and β-sheet domain (gray) are represented as a cartoon.

In both sets of simulation for M182N, the isolated helix and the full-length sequence, Asn182 largely populates only two conformations. These conformations differ in whether the χ_1-angle is in the *gauche+* (χ_1 : 0° → 120°) or *trans* (χ_1 : 120° → 240°) rotamer. Both conformations are capable of capping helix 9 but only the *trans* rotamer hydrogen bonds with Glu64 (Figure 5.4A and 5.4B). In the isolated helix, Asn182 adopts the *trans* rotamer with a probability of 0.75±0.01, while the probability of this conformation is only 0.58±0.03 in the context of the full-length protein (Figure 5.5). In contrast, Thr182 and Ser182 overwhelmingly adopt the *gauche+* rotamer (Figure A.2.6).

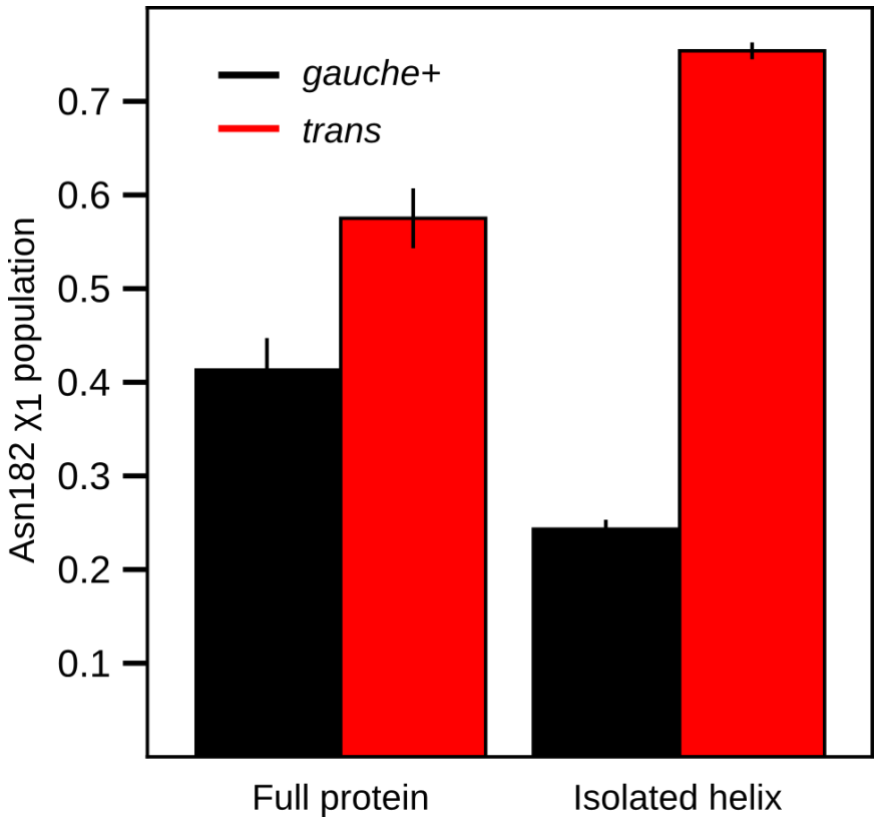

Figure **5.5:** Asn182 rotamer populations for the full protein and isolated helix. Shown are the gauche+ (black) and trans (red) rotamer populations from MSMs of the full protein and isolated helix.

Asn182's rotamer populations suggest that the *trans* rotamer stabilizes helix 9 but that competing interactions in the context of the full-length protein mitigate these stabilizing effects by favoring the *gauche*+ conformation. As a test of this hypothesis, we calculated two sets of distance distributions for hydrogen bonds along helix 9: one for the set of conformations when Asn182 is in the *trans* rotamer, and one for the *gauche*+ rotamer (Figure A.2.7). Comparing these distributions confirms that the *trans* rotamer stabilizes helix 9 by increasing the probability of moderate strength hydrogen bonds, while the *gauche*+ rotamer behaves more like wild-type. We next examined structures from each rotameric state of Asn182 to understand why *gauche*+ appears so frequently given that it doesn't stabilize helix 9.

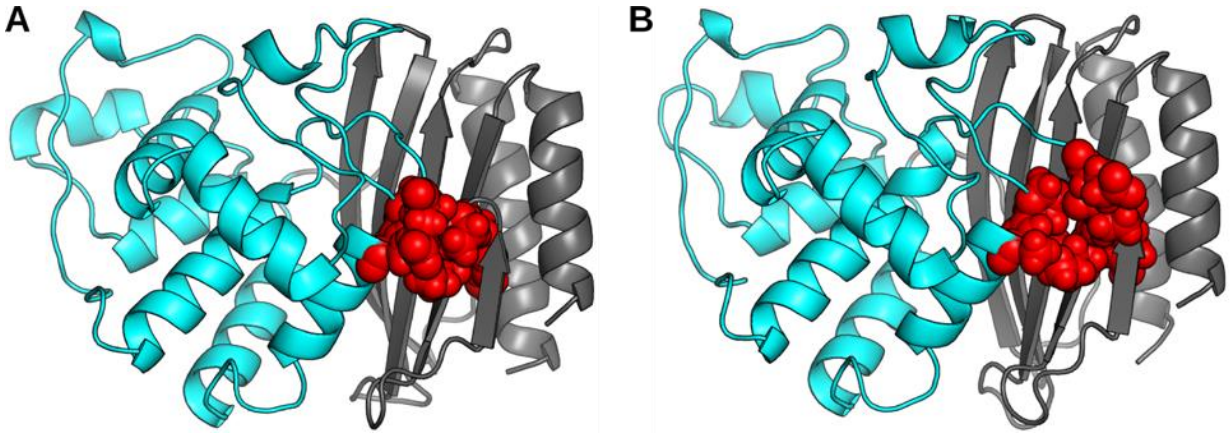

Figure 5.6: Representative structures that highlight the effects of different Asn182 rotamers on packing at the interface of the s2h2 loop and α-helix/β-sheet domains. (A) A representative structure of the gauche+ rotamer. (B) A representative structure of the trans rotamer. Residues whose packing is affected by Asn182's rotamer (Tyr46, Ile47, Pro62, Glu63, Pro182, and Ala184) are shown as red spheres. The backbone of the α-helix domain (cyan) and β-sheet domain (gray) are represented as a cartoon.

We find the *trans* and *gauche+* conformations of Asn182 to have distinct effects on packing at the interface between the α-helix and β-sheet domains. In the *gauche+* state, which does not stabilize helix 9, the domain interface is well-packed (Figure 5.6A). In contrast, when Asn182 adopts the *trans* conformation, it appears to disrupt the packing of this interface and increase the exposure of a number of hydrophobic moieties to solvent (Figure 5.6B). Specifically, a pocket forms between Tyr46 and Ile47 from the β-sheet domain, Pro62 and Glu63 of the s2h2 loop, and Pro183 and Ala184 of the α-helix domain. To quantify this effect, we calculated the average solvent accessible surface area of these residues for the ensembles of structures where Asn182 adopts either the *trans* or *gauche+* rotamer. Doing so reveals that when Asn182 adopts the *trans* state, this surface area increases by ~20% compared to when Asn182 is in the *gauche+* state (Figure A.2.8). Furthermore, much of the increased surface area is contributed by hydrophobic portions of these residues. Since exposure of buried hydrophobic groups is thermodynamically destabilizing, we propose that opening of this pocket counterbalances the stabilizing effects of capping. Therefore, M182N fails to stabilize helix 9

and ultimately the entire protein. This result is also consistent with the observation of the *gauche+* rotamer in the crystal structure of M182N, since each rotamer has roughly equal population and crystal packing forces will favor the more compact structure. Finally, our results for Asn182 are consistent with our proposal that the domain interface is a crucial determinant of the stability of TEM's native state.

5.3.4 Stabilizing Mutations Stabilize the Domain Interface

As a further test of our model, and the importance of helix 9 to the domain interface, we turned to NMR spectroscopy. We use NMR because it can provide site specific details on protein structure and dynamics. Here, we performed ^{1}H-^{15}N heteronuclear single quantum coherence (HSQC) experiments for each variant and calculated chemical shift perturbations (CSPs) relative to wild-type. Since each chemical shift reports on a nuclei's unique local magnetic environment, a CSP indicates a change in the structure and dynamics at this site. Thus, the CSPs for each variant will identify all regions affected by the mutation, regardless of their proximity to the mutation.

Consistent with our proposed mechanism for stabilization, most of the statistically significant CSPs for M182T are found in helix 9 and the adjacent β-sheets (Figure 5.7). Significant CSPs are observed on the first two turns of helix 9, as is expected from our prediction that M182T increases the propensity of moderate hydrogen bonds. We also observe significant CSPs on the β-sheet domain, not only in residues that interact directly with helix 9 (i.e. Ile47, Leu49, and Val262), but also in more distant residues (i.e. Val44, Phe60, and Thr265). Together, these results demonstrate that M182T alters the structure and dynamics of helix 9 and that these effects are propagated to distant residues along the domain interface. This is consistent with our model that M182T stabilizes helix 9, which in turn stabilizes the interface between the β-sheet

and α-helix domains. To explore this idea further we next examined the CSPs of the other variants.

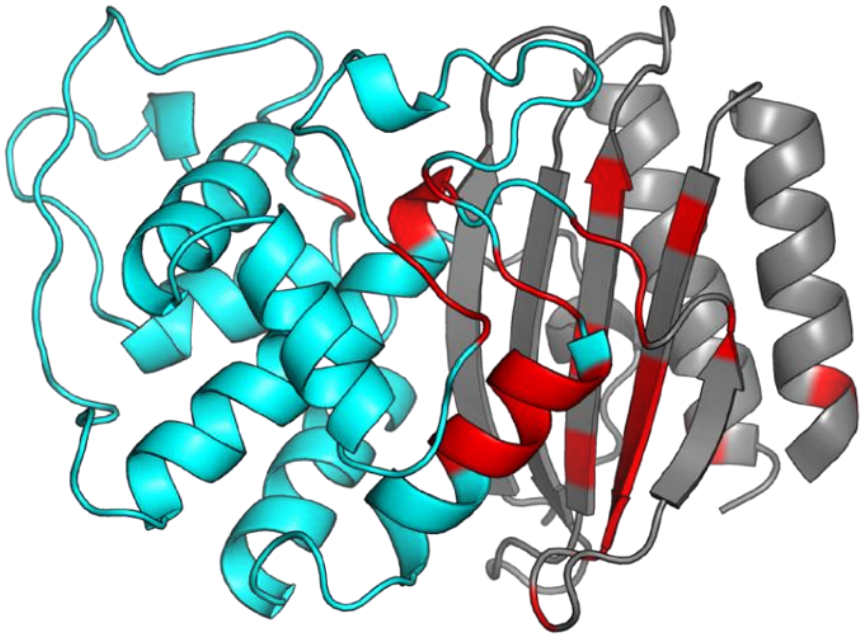

Figure 5.7: Backbone amide chemical shift perturbations of TEM M182T. The backbone of the α-helix domain (cyan) and β-sheet domain (gray) are represented as a cartoon. Residues with statistically significant chemical shift perturbations are colored red.

Comparing the magnitude and direction of CSPs on the β-sheet between each variant suggests that stabilizing mutations stabilize the domain interface. Similar to M182T, each variant predominately displays CSPs on helix 9 and the interface of the α-helix and β-sheet domains (Figure A.2.9). This indicates that each of our substitutions at position 182 alters the structure and dynamics of the domain interface. Although one might conclude from the common locations of CSPs between the variants that each mutation perturbs TEM in a similar manner, we find that the magnitude of CSPs on the β-sheet differs between variants (Figure 5.8). Additionally, these CSPs are not randomly scattered. Instead, there is a clear trend from the least stable to the most stable variant. Taking all of our observations together, we propose that CSPs closer to wild-type represent a more loosely packed, weaker interface, whereas those closer to M182T/M182S

represent a more tightly packed, stronger interface. Therefore, we conclude that global stability is not only achieved through helix 9 stabilization, but also through stabilization of the domain interface.

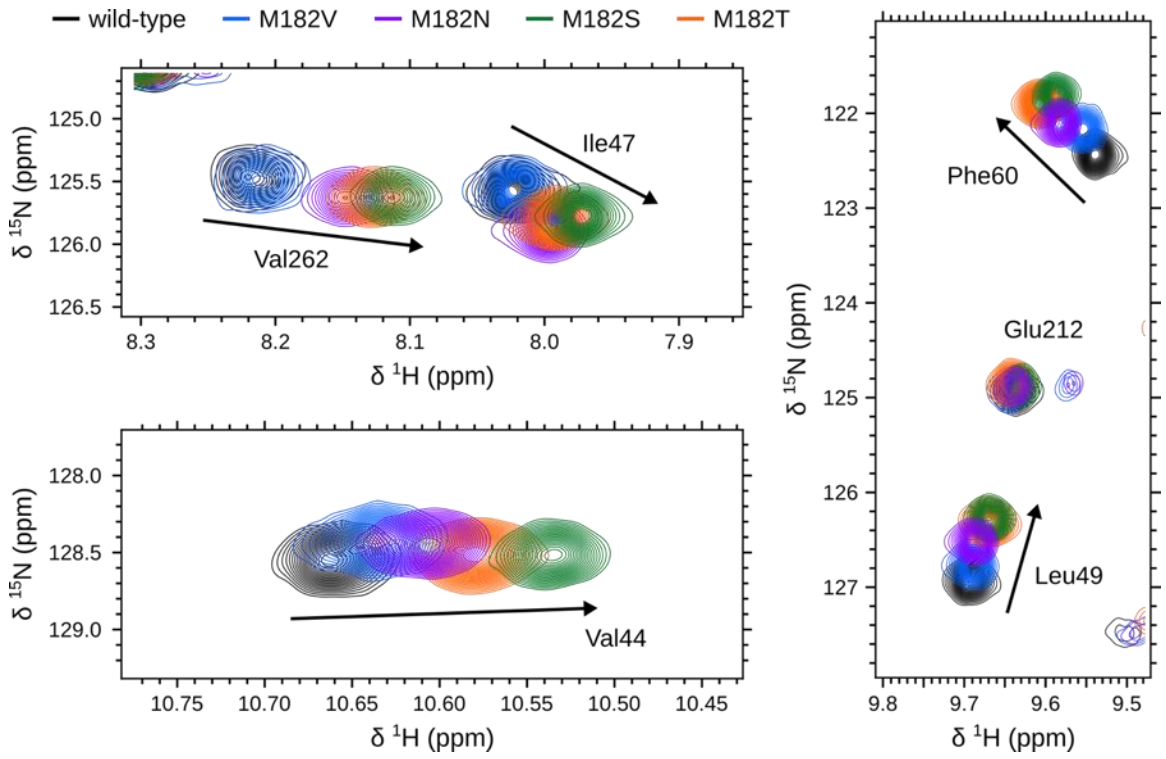

Figure 5.8: Representative backbone amide chemical shifts, located on TEM's β-sheet, for 5 sequence variants. Shown are the chemical shifts for wild-type (black), M182V (blue), M182N (purple), M182S (green), and M182T (orange) for residues located on the β-sheet: Val44, Ile47, Leu49, Phe60, and Val262. For reference, Glu212 is not located on the β-sheet and does not display significant perturbations upon mutation.

5.3.5 Stabilizing Mutations are Global Suppressors

If stabilization by M182T is the biophysical mechanism for its ability to suppress the impact of other deleterious mutations, then we would expect the stabilities of the three new variants we selected to correlate with their ability to act as global suppressors. To test this hypothesis, we introduced our three substitutions into a background that also contains the substitution G238S. G238S is known to confer TEM with cefotaxime resistance at the expense of protein stability[1,5]. Furthermore, a variant with both G238S and M182T is more resistant to cefotaxime than a

variant with just one of these substitutions. Therefore, we expect M182S/G238S to have similar levels of cefotaxime resistance to M182T/G238S while we expect M182N/G238S and M182V/G238S to have similar levels of cefotaxime resistance to G238S alone.

Table 5.2. MICs for *E. coli* strains expressing TEM β–lactamase variants*

		cefotaxime (μM)	
		Single mutant	**Double mutant (+G238S)**
Wild-type TEM		<0.035	0.141
Suppressor/stabilizing			
	M182T	0.070	72.000
	M182S	0.070	36.000
Wild-type-like/neutral			
	M182V	<0.035	0.141
	M182N	<0.035	0.141
	M182C	<0.035	0.281
	M182A	<0.035	0.281
Deleterious			
	M182G	ND†	<0.035
	M182P	ND†	<0.035
	M182I	ND†	<0.035
	M182L	ND†	<0.035
	M182F	ND†	<0.035
	M182W	ND†	<0.035
	M182Y	ND†	<0.035
	M182R	ND†	<0.035
	M182H	ND†	<0.035
	M182K	ND†	0.070
	M182D	ND†	<0.035
	M182E	ND†	<0.035
	M182Q	ND†	<0.035
	M182G	ND†	<0.035

*MIC determination was performed in triplicate. Values are most commonly observed concentration with an error of +/- one well, which differ by 2-fold in concentration.
†Not determined.

Minimal inhibitory concentrations (MICs) of bacteria expressing our TEM variants in the background of G238S in the presence of varying levels of cefotaxime reveals that global stabilization of the domain interface leads to global suppression. As predicted, M182S/G238S resembles M182T/G238S while the other variants are more similar to G238S alone (Table 5.2).

The observation that M182S is a global suppressor mutation that has not been reported previously lead us to question if other global suppressors may exist.

MICs for every other possible variant at position 182, in combination with G238S, reveal that there are no other possible global suppressor mutations at this position (Table 5.2). Substituting Met182 with valine, asparagine, cysteine, or alanine in a G238S background is neutral. All other double mutants have lower MICs than G238S alone, suggesting that they are deleterious. Therefore, M182T and M182S are the only global suppressor mutations at this residue. Together with the previous sections, these results are consistent with our hypothesis that stabilization of helix 9 and the domain interface are responsible for M182T's ability to stabilize TEM and suppress the effects of other destabilizing substitutions.

5.4 Conclusions

Our MSMs have provided a new mechanistic understanding of the stabilizing effects of M182T, which we successfully use to predict the effects of new mutations at position 182. Previous crystallographic studies have proposed that M182T's stabilizing effect is a result of Thr182 either N-capping or forming a hydrogen bond between the α-helix and β-sheet domain interface. Since MSMs are able to capture conformational heterogeneity in a way that cannot be inferred from static structures, we are able to propose that M182T stabilizes helix 9, which in turn stabilizes the interface between the α-helix and β-sheet domains. In support of the validity of our model, it has superior predictive power compared to previous models: we correctly predict that M182S is stabilizing but not M182V and M182N, whereas the hydrogen bonding model incorrectly predicts M182N to be stabilizing. Furthermore, NMR chemical shift perturbations support our dynamical predictions. The fact that our MSMs make successful predictions that

defy biochemical intuition is a strong testament to the accuracy and value of these atomically-detailed models.

The ability to predict new stabilizing mutations is an important step towards designing proteins with new or improved functions. The fact that M182S hasn't been observed suggests that nature hasn't exhaustively identified all possible stabilizing mutations. Our work raises interesting questions, such as why hasn't M182S been observed in nature. Furthermore, combining our ability to predict new stabilizing mutations with our previous work on predicting how mutations impact activity could enable the design of proteins with new or improved function.

5.5 Methods

5.5.1 MD Simulations

All simulations were run with Gromacs 5.1.1.[23] β-Lactamase simulations were run at 300 K using the AMBER03 force field with explicit TIP3P solvent.[24,25] We have previously shown that the AMBER03 forcefield is sufficient to capture the relevant conformational states of TEM β-lactamases for a range of problems.[11,26-28] The single starting structure for TEM-1 β-Lactamase simulations was generated from the crystallographic structure (PDB ID: 1JWP)[1]. The starting structures for each TEM variant was generated by mutating the side chain at position 182 to the respective amino acid using PDBFixer, followed by an energy minimization for 1,000 steps using the AMBER03 force field with the OBC GBSA implicit solvent model.[24,29,30] Starting structures for the individual helix simulations were taken as residues 181-197 from the starting structures of the full sequence. For each full-length sequence, 2.5 μs of conventional sampling and 4 μs of FAST-RMSD adaptive sampling (described below) was performed. For the individual helix simulations, 4 μs of each sequence was performed: 20 simulations of 200 ns.

Simulations were prepared by placing the starting structure for each sequence in a dodecahedron box that extended 1.0 Å beyond the protein in any dimension. Each system was then energy minimized with the steepest descent algorithm until the maximum force fell below 100 kJ/mol/nm using a step size of 0.01 nm and a cutoff distance of 1.2 nm for the neighbor list, Coulomb interactions, and van der Waals interactions. For production runs, all bonds were constrained with the LINCS algorithm and virtual sites were used to allow a 4 fs time step.[31,32] Cut-offs of 1.0 nm were used for the neighbor list, Coulomb interactions, and van der Waal interactions. The Verlet cutoff scheme was used for the neighbor list. The stochastic velocity rescaling (v-rescale) thermostat was used to hold the temperature at 300 K.[33] Conformations were stored every 20 ps.

5.5.2 Adaptive Sampling

The FAST algorithm was used to generate simulation data.[15] FAST-RMSD was run for each sequence for 10 rounds, of 10 simulations per round, where each simulation was 40 ns in length; a total of 4 μs per sequence. The FAST-ranking favored states that maximized the RMSD to the starting structure. RMSD calculations were performed between all heavy atoms in residues within 1.0 nm of position 182 in the crystallographic starting structure. To enhance the conformational diversity of states that are chosen for reseeding simulations, the FAST-ranking function was modified with a term that penalizes states conformationally similar to others selected. This ensures that each round of sampling contains a good spread of conformations. Procedurally, states are selected one at a time, where the modified term is recomputed and added to the original ranking for each selection. The modified ranking takes the form,

$$r_{\phi}(i) = \bar{\phi}(i) + \alpha\bar{\psi}(i) + \beta\chi(i)$$

where $\bar{\phi}$ is the directed component, $\bar{\psi}$ is the undirected component, and α and β control the weights of $\bar{\psi}$ and χ respectively. Here, $\bar{\psi}(i)$ is taken to be the state counts and a value of 1 was used for both α and β. The additional term,

$$\chi(i) = \begin{cases} 0 & if\ N = 0 \\ \frac{1}{N}\sum_{j=1}^{N}\left(1 - e^{\frac{-RMSD_{ij}^2}{2w^2}}\right) & if\ N > 0 \end{cases}$$

is calculated as the average of Gaussian weighted RMSDs from state *i* to the *N* states that have been selected for reseeding so far, where *w* is the Gaussian width (set to the clustering radius). Thus, the procedure for selecting states to reseed simulations from each round is as follows: 1) rank all states by the FAST-ranking and select the top state as the first state to reseed, 2) add the similarity penalization term to the FAST-ranking and select the top state as another state to reseed, 3) update the penalization term and repeat step 2 until the desired number of states for reseeding have been selected.

5.5.3 MSM Construction and Analysis

All MSMs were built using MSMBuilder.[34,35] An MSM is a network representation of an energy landscape, where nodes are discrete conformational states and directed edges are conditional transition probabilities. MSMs provide a statistically rigorous way of mapping of protein dynamics, even from parallel simulations with starting structures that are not Boltzmann distributed. Using an MSM, we can quantify thermodynamic and kinetic changes that aid in understanding molecular motions.

Simulation datasets for each TEM variant were combined and clustered into a single shared state-space. Each dataset consisted of 4 μs FAST-RMSD and 2.5 μs conventional simulations. With 5 sequences, this gives a total of 32.5 μs of total simulation. The shared state-space was defined using all heavy atoms on residues within 1.0 Å of position 182 in the crystallographic structure of TEM β-lactamase (PDB ID: 1JWP). The sidechain atoms of position 182 were not included, since they vary between sequences. These atomic coordinates were then clustered with a k-centers algorithm based on RMSD between conformations until every cluster center had a radius less than 1.0 Å. Then, 10 sweeps of a k-medoids update step was used to center the clusters on the densest regions of conformational space. Following clustering, the cluster assignments were split and a unique MSM was constructed for each TEM sequence with a lagtime of 2 ns. To obey microscopic reversibility, transition count matrices were symmetrized. Representative cluster centers were saved for each state in each sequence for analysis.

Geometric analysis of representative cluster centers was performed using MDTraj;[36] in particular, RMSDs, solvent-accessible surface areas, and atomic distances. Ensemble average values within MSMs were calculated as the expectation value for a particular observable. i.e. the expectation of observable z is calculated as:

$$E(z) = \sum_i P(i) * z(i)$$

where *P(i)* is the population of state *i* and *z*(*i*) is the value of state *i*.

5.5.4 Protein Expression and Purification

TEM-1 was subcloned using NdeI and XhoI restriction sites into the multiple cloning site of a pET24 vector (Life Technologies), and its native export signal sequence was replaced by the OmpA signal sequence to maximize export efficiency. Site-specific variants were constructed via site-directed mutagenesis and verified by DNA sequencing. Plasmids were then transformed into BL21(DE3) Gold cells (Agilent Technologies) for expression under T7 promoter control.

Cells were induced with 1 mM IPTG at OD = 0.6 and grown at 18 °C for 15 h before harvesting. TEM β-lactamases were isolated from the periplasmic fraction using osmotic shock lysis: Cells were resuspended in 30 mM Tris pH 8, 20% sucrose and stirred for 10 min at room temperature. After centrifugation, the pellet was re-suspended in ice-cold 5 mM $MgSO_4$ and stirred for 10 min at 4 °C. After centrifugation, the supernatant was dialyzed against 20 mM sodium acetate, pH 5.5 and purified using cation exchange chromatography (BioRad UNOsphere Rapid S column) followed size exclusion chromatography (BioRad ENrich SEC 70 column) into storage buffer (20 mM Tris, pH 8.0).

5.5.5 Protein Stability Measurements

Fluorescence data were collected using a Photon Technology International QuantaMaster 800 Rapid Excitation Spectrofluorometer with Quantum Northwest Inc. TC-125 Peltier-controlled cuvette holder. Melts were performed by monitoring intrinsic protein fluorescence, exciting at 280 nm and detecting emission intensity at 340 nm. Melts were carried out in a 1-cm pathlength cuvette (50 μg/mL protein, 20 mM Tris pH 7). Samples with varying concentrations of urea were prepared individually, equilibrated overnight and allowed to stir in the instrument for 2 minutes before data collection.

Circular dichroism data were collected using an Applied Photophysics Chirascan with a Quantum Northwest Inc. TC-125 Peltier-controlled cuvette holder. Melts were performed by monitoring CD signal at 222 nm and were carried out in a 1-cm pathlength cuvette (50 μg/mL protein, 20 mM Tris pH 7). For urea melts, samples with varying concentrations of urea were prepared individually, equilibrated overnight and allowed to stir in the instrument for 2 minutes before data collection, which was averaged over 60 seconds.

Urea melt data for each variant were globally fit. Fluorescence data were fit by a two-state model (I-to-N), and CD data simultaneously were fit by a three-state model (U-to-I-to-N) using a linear extrapolation method:[37]

$$Fluorescence\ (F) = \frac{F_i + F_n e^{-(\Delta G_{in} + m_{in}[urea])/RT}}{1 + e^{-(\Delta G_{in} + m_{in}[urea])/RT}} \quad \text{Equation 1}$$

$$CD\ (\Theta) = \frac{\Theta_u + \Theta_i e^{-(\Delta G_{in} + m_{in}[urea])/RT} + \Theta_n e^{-(\Delta G_{in} + m_{in}[urea])/RT} e^{-(\Delta G_{ui} + m_{ui}[urea])/RT}}{1 + e^{-(\Delta G_{in} + m_{in}[urea])/RT} + e^{-(\Delta G_{in} + m_{in}[urea])/RT} e^{-(\Delta G_{ui} + m_{ui}[urea])/RT}} \quad \text{Equation 2}$$

where F_i and F_n are the fluorescence signals for the intermediate and native states, fit as lines, and Θ_u, Θ_i and Θ_n are the CD signals for the unfolded, intermediate and native states, fit as lines. ΔG_{in} is the extrapolated free energy of folding relative to the intermediate in the absence of denaturant, and m_{in} is a proportionality constant related to the steepness of the I-to-N transition. ΔG_{ui} and m_{ui} are the free energy and m-value describing the U-to-I transition.

The m_{ui}-value was fixed to 1.7 kcal/mol*M, the average derived for wild-type TEM, because we hypothesize the intermediate species is the same between variants. *m*-values correlate with the change in solvent-exposed surface area upon folding[12] and are characteristic of

a particular folded or partially folded state. For comparison, all data were also fit using a floating m_{ui}-value, and F-tests were performed with the null hypothesis that any improvement to the fit due to the additional parameter occurs by chance. The F-values obtained were all in the range of $1x10^{-10}$—$1x10^{-7}$ (much lower than ~4.2, the critical F-value for $p<0.05$), and thus the F-tests strongly support our hypothesis that holding the m_{ui}-value fixed is reasonable.

5.5.6 Minimal Inhibitory Concentration (MIC) Measurements

Levels of antibiotic resistance of BL21(DE3) cells containing TEM expression plasmids were determined by measuring their minimum inhibitory concentrations (MIC90's) using the broth microdilution method according to the Clinical and Laboratory Standards Institute (CLSI, formerly the NCCLS) guidelines. [38] Strains were grown to saturation overnight in Luria Miller broth with kanamycin and 1 mM IPTG. Each well of a 96-well microtiter plate was filled with 50 μL of sterile Mueller Hinton II (MHII) media broth (Sigma). Antibiotic was dissolved in water making a 20 mM solution, then diluted with sterile MHII media broth to 288 μM cefotaxime (CFX). Exactly 50 μL of the compound solution was added to the first well of the microtiter plate, and 2-fold serial dilutions were made down each row of the plate. Exactly 50 μL of bacterial inoculum (diluted to 5×10^5 CFU mL^{-1} from the overnight cultures) was then added to each well giving a total volume of 100 μL $well^{-1}$ and compound concentration gradients of 72 μM–0.04 μM CFX. The plate was incubated at 37 °C for 17 h, and then each well was examined for bacterial growth. The MIC90 was recorded as the lowest compound concentration required to inhibit 90% of bacterial growth as judged by turbidity of the culture media relative to a row of wells filled with a water standard. Gentamicin was included in a control row at a concentration gradient of 174 μM–0.09 μM.

5.5.7 Nuclear Magnetic Spectroscopy

Uniform ^{15}N labeled TEM-1 was expressed in M9 minimal media containing $^{15}NH_4Cl$ (1 g/L), D-glucose (4 g/L), and 2.5 mM betaine. The cells were incubated at 37 °C and 240 rpm until OD_{600} » 0.6, then an additional 30 minutes at 18 °C and 225 rpm. Cells were induced with IPTG and incubated approximately 36 hours prior to harvesting. Protein was purified from both the periplasm and the media; the media was concentrated to approximately 100 mL using an Amicon stirred cell (EMD Millipore) and dialyzed overnight into TEM-1 S loading buffer. Purification followed the periplasmic prep.

$^{15}N/^{1}H$ HSQC spectra were recorded at 303 K on a 600 MHz (1H) Bruker Avance III spectrometer. TEM-1 samples were concentrated to 100 μM in 25 mM sodium phosphate, 4 mM imidazole pH 6.6 and 10% D_2O. Wild type TEM-1 assignments were previously reported (BMRB entry 16392).[39]

5.5.8 X-ray Crystallography

Screening for crystal growth conditions was performed with Mosquito® (TTP LabTech Limited) using 25 mg/mL protein. Optimized crystals were grown via hanging drop vapor diffusion at 18°C by mixing 1 μl of protein at 25 mg/ml with 1μl of reservoir containing 0.1 M sodium phosphate dibasic/citric acid pH 4.2, 0.1 M lithium sulfate, and 20% PEG 1000. Crystals were cryoprotected in oil (Hampton Research Parabar 10312 HR2-862) before flash-freezing in liquid nitrogen. X-ray diffraction data was collected at beamline 4.2.2 of the Advanced Light Source in Berkeley, CA and processed with XDS.[40] Phase determination was by molecular replacement using PHENIX[41] with the coordinates from PDB 1JWP used as a search model. Iterative model building in COOT[42] and refinement with PHENIX[41] accounting for crystal twinning led to the current model of M182N with R_{work}/R_{free} of 22.46%/28.26%. The final refined model had a

Ramachandran plot with 96.54% of residues in the favored region and none in the disallowed region (MolProbity[43]). A summary of the data collection and refinement statistics is shown in Table S1. Structure factors and coordinates are deposited in the RSCB Protein Structure Database under PDB ID 6B2N.

Bibliography

(1) Wang, X.; Minasov, G.; Shoichet, B. K. Evolution of an Antibiotic Resistance Enzyme Constrained by Stability and Activity Trade-Offs. *Journal of Molecular Biology* **2002**, *320* (1), 85–95.

(2) Salverda, M. L. M.; De Visser, J. A. G. M.; Barlow, M. Natural Evolution of TEM-1 B-Lactamase: Experimental Reconstruction and Clinical Relevance. *FEMS Microbiol. Rev.* **2010**, *34* (6), 1015–1036.

(3) Huang, W.; Palzkill, T. A Natural Polymorphism in B-Lactamase Is a Global Suppressor. *Proc. Natl. Acad. Sci. U.S.A.* **1997**, *94* (16), 8801–8806.

(4) Kather, I.; Jakob, R. P.; Dobbek, H.; Schmid, F. X. Increased Folding Stability of TEM-1 B-Lactamase by in Vitro Selection. *Journal of Molecular Biology* **2008**, *383* (1), 238–251.

(5) Raquet, X.; Vanhove, M.; Brasseur, J. L.; Goussard, S.; Courvalin, P.; Frère, J. M. Stability of TEM B-Lactamase Mutants Hydrolyzing Third Generation Cephalosporins. *Proteins: Structure, Function, and Bioinformatics* **1995**, *23* (1), 63–72.

(6) Sideraki, V.; Huang, W.; Palzkill, T.; Gilbert, H. F. A Secondary Drug Resistance Mutation of TEM-1 B-Lactamase That Suppresses Misfolding and Aggregation. *Proc. Natl. Acad. Sci. U.S.A.* **2001**, *98* (1), 283–288.

(7) Orencia, M. C.; Yoon, J. S.; Ness, J. E.; Willem P. C. Stemmer; Stevens, R. C. Predicting the Emergence of Antibiotic Resistance by Directed Evolution and Structural Analysis. *Nature Structural & Molecular Biology 1997 4:1* **2001**, *8* (3), 238–242.

(8) Bowman, G. R.; Pande, V. S.; Noé, F. Introduction and Overview of This Book. In *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*; Advances in Experimental Medicine and Biology; Springer Netherlands: Dordrecht, 2014; Vol. 797, pp 1–6.

(9) Chodera, J. D.; Noé, F. Markov State Models of Biomolecular Conformational Dynamics. *Current Opinion in Structural Biology* **2014**, *25*, 135–144.

(10) Schütte, C.; Sarich, M. *Metastability and Markov State Models in Molecular Dynamics*; 2013.

(11) Hart, K. M.; Ho, C. M. W.; Dutta, S.; Gross, M. L.; Bowman, G. R. Modelling Proteins' Hidden Conformations to Predict Antibiotic Resistance. *Nature Communications* **2016**, *7*, 12965.

(12) Myers, J. K.; Pace, C. N.; Scholtz, J. M. Denaturant M Values and Heat Capacity Changes: Relation to Changes in Accessible Surface Areas of Protein Unfolding. *Protein Science* **1995**, *4* (10), 2138–2148.

(13) Spudich, S.; Marqusee, S. *A Change in the Apparent M Value Reveals a Populated Intermediate Under Equilibrium Conditions in Escherichia Coli Ribonuclease HI†*; American Chemical Society, 2000; Vol. 39, pp 11677–11683.

(14) Annabelle Lejeune; Roger H Pain; Paulette Charlier; Jean-Marie Frère, A.; André Matagne. TEM-1 B-Lactamase Folds in a Nonhierarchical Manner with Transient Non-Native Interactions Involving the C-Terminal Region†. *Biochemistry* **2008**, *47* (4), 1186–1193.

(15) Zimmerman, M. I.; Bowman, G. R. FAST Conformational Searches by Balancing Exploration/Exploitation Trade-Offs. *J. Chem. Theory Comput.* **2015**, *11* (12), 5747–5757.

(16) Zimmerman, M. I.; Bowman, G. R. How to Run FAST Simulations. *Methods in Enzymology* **2016**, *578*, 213–225.

(17) Bowman, G. R. Accurately Modeling Nanosecond Protein Dynamics Requires at Least Microseconds of Simulation. *J Comput Chem* **2016**, *37* (6), 558–566.

(18) Vanhove, M.; Lejeune, A.; Pain, R. H. B-Lactamases as Models for Protein-Folding Studies. *CMLS, Cell. Mol. Life Sci.* **2014**, *54* (4), 372–377.

(19) Jeffrey, G. A.; Saenger, W. *Hydrogen Bonding in Biological Structures*; 2012.

(20) Baker, E. N.; Hubbard, R. E. Hydrogen Bonding in Globular Proteins. *Progress in Biophysics and Molecular Biology* **1984**, *44* (2), 97–179.

(21) Richardson, J. S.; Richardson, D. C. Amino Acid Preferences for Specific Locations at the Ends of Alpha Helices. *Science* **1988**, *240* (4859), 1648–1652.

(22) Doig, A. J.; Baldwin, R. L. N- and C-Capping Preferences for All 20 Amino Acids in A-Helical Peptides. *Protein Science* **1995**, *4* (7), 1325–1336.

(23) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High Performance Molecular Simulations Through Multi-Level Parallelism From Laptops to Supercomputers. *SoftwareX* **2015**, *1-2*, 19–25.

(24) Duan, Y.; Wu, C.; Chowdhury, S.; Lee, M. C.; Xiong, G.; Zhang, W.; Yang, R.; Cieplak, P.; Luo, R.; Lee, T.; Caldwell, J.; Wang, J.; Kollman, P. A Point-Charge Force Field for Molecular Mechanics Simulations of Proteins Based on Condensed-

Phase Quantum Mechanical Calculations. *J Comput Chem* **2003**, *24* (16), 1999–2012.

(25) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. Comparison of Simple Potential Functions for Simulating Liquid Water. *The Journal of Chemical Physics* **1998**, *79* (2), 926–935.

(26) Bowman, G. R.; Geissler, P. L. Equilibrium Fluctuations of a Single Folded Protein Reveal a Multitude of Potential Cryptic Allosteric Sites. *Proc. Natl. Acad. Sci. U.S.A.* **2012**, *109* (29), 11681–11686.

(27) Bowman, G. R.; Bolin, E. R.; Hart, K. M.; Maguire, B. C.; Marqusee, S. Discovery of Multiple Hidden Allosteric Sites by Combining Markov State Models and Experiments. *Proc. Natl. Acad. Sci. U.S.A.* **2015**, *112* (9), 2734–2739.

(28) Hart, K. M.; Moeder, K. E.; Ho, C. M. W.; Zimmerman, M. I.; Frederick, T. E.; Bowman, G. R. Designing Small Molecules to Target Cryptic Pockets Yields Both Positive and Negative Allosteric Modulators. *PLOS ONE* **2017**, *12* (6), e0178678.

(29) Eastman, P.; Friedrichs, M. S.; Chodera, J. D.; Radmer, R. J.; Bruns, C. M.; Ku, J. P.; Beauchamp, K. A.; Lane, T. J.; Wang, L.-P.; Shukla, D.; Tye, T.; Houston, M.; Stich, T.; Klein, C.; Shirts, M. R.; Pande, V. S. OpenMM 4: a Reusable, Extensible, Hardware Independent Library for High Performance Molecular Simulation. *J. Chem. Theory Comput.* **2012**, *9* (1), 461–469.

(30) Onufriev, A.; Bashford, D.; Case, D. A. Exploring Protein Native States and Large-Scale Conformational Changes with a Modified Generalized Born Model. *Proteins: Structure, Function, and Bioinformatics* **2004**, *55* (2), 383–394.

(31) Feenstra, K. A.; Hess, B.; Berendsen, H. J. C. Improving Efficiency of Large Time-Scale Molecular Dynamics Simulations of Hydrogen-Rich Systems. *J Comput Chem* **1999**, *20* (8), 786–798.

(32) Hess, B. P-LINCS: a Parallel Linear Constraint Solver for Molecular Simulation. *J. Chem. Theory Comput.* **2007**, *4* (1), 116–122.

(33) Bussi, G.; Donadio, D.; Parrinello, M. Canonical Sampling Through Velocity Rescaling. *The Journal of Chemical Physics* **2007**, *126* (1), 014101.

(34) Beauchamp, K. A.; Bowman, G. R.; Lane, T. J.; Maibaum, L.; Haque, I. S.; Pande, V. S. MSMBuilder2: Modeling Conformational Dynamics on the Picosecond to Millisecond Scale. *J. Chem. Theory Comput.* **2011**, *7* (10), 3412–3419.

(35) Harrigan, M. P.; Sultan, M. M.; Hernández, C. X.; Husic, B. E.; Eastman, P.; Schwantes, C. R.; Beauchamp, K. A.; McGibbon, R. T.; Pande, V. S. MSMBuilder:

Statistical Models for Biomolecular Dynamics. *Biophysical Journal* **2017**, *112* (1), 10–15.

(36) McGibbon, R. T.; Beauchamp, K. A.; Harrigan, M. P.; Klein, C.; Swails, J. M.; Hernández, C. X.; Schwantes, C. R.; Wang, L.-P.; Lane, T. J.; Pande, V. S. MDTraj: a Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophysical Journal* **2015**, *109* (8), 1528–1532.

(37) Santoro, M. M.; Bolen, D. W. Unfolding Free Energy Changes Determined by the Linear Extrapolation Method. 1. Unfolding of Phenylmethanesulfonyl .Alpha.-Chymotrypsin Using Different Denaturants. *Biochemistry* **2002**, *27* (21), 8063–8068.

(38) Balouiri, M.; Sadiki, M.; Ibnsouda, S. K. Methods for in Vitro Evaluating Antimicrobial Activity: a Review. *Journal of Pharmaceutical Analysis* **2016**, *6* (2), 71–79.

(39) and, P.-Y. S.; Gagné, S. M. *Backbone Dynamics of TEM-1 Determined by NMR: Evidence for a Highly Ordered Protein†*; American Chemical Society, 2006; Vol. 45, pp 11414–11424.

(40) Kabsch, W.; IUCr. Xds. *Acta Crystallogr Sect D Biol Crystallogr* **2010**, *66* (2), 125–132.

(41) Adams, P. D.; Afonine, P. V.; Bunkóczi, G.; Chen, V. B.; Davis, I. W.; Echols, N.; Headd, J. J.; Hung, L. W.; Kapral, G. J.; Grosse-Kunstleve, R. W.; McCoy, A. J.; Moriarty, N. W.; Oeffner, R.; Read, R. J.; Richardson, D. C.; Richardson, J. S.; Terwilliger, T. C.; Zwart, P. H.; IUCr. PHENIX: a Comprehensive Python-Based System for Macromolecular Structure Solution. *Acta Crystallogr Sect D Biol Crystallogr* **2010**, *66* (2), 213–221.

(42) Emsley, P.; Cowtan, K.; IUCr. Coot: Model-Building Tools for Molecular Graphics. *Acta Crystallogr Sect D Biol Crystallogr* **2004**, *60* (12), 2126–2132.

(43) Chen, V. B.; Arendall, W. B.; Headd, J. J.; Keedy, D. A.; Immormino, R. M.; Kapral, G. J.; Murray, L. W.; Richardson, J. S.; Richardson, D. C.; IUCr. MolProbity: All-Atom Structure Validation for Macromolecular Crystallography. *Acta Crystallogr Sect D Biol Crystallogr* **2010**, *66* (1), 12–21.

Chapter 6

Enspara: Modeling Molecular Ensembles with Scalable Data Structures and Parallel Computing

6.1 Preamble

This chapter is adapted from the following article: Porter, J.R., Zimmerman, M.I., and Bowman, G.R. (2019). "Enspara: Modeling Molecular Ensembles with Scalable Data Structures and Parallel Computing", *Journal of Chemical Physics*, 150, 044108

6.2 Introduction

Markov state models (MSMs)[1-4] are a powerful tool for representing the complexity of dynamics in protein conformational space. They have proven useful both as quantitative models of protein behavior[5-8] and for producing insights about the mechanism of protein conformational transitions.[9-12] And, with the rise of special-purpose supercomputers,[13,14] distributed-computing platforms,[15] and the dramatic increases in the power of consumer-grade processors (especially GPUs), the size of molecular dynamics (MD) data sets that MSMs are built on have grown in size commensurately.

With the increasing size of MD datasets, there is ongoing and substantial interest in making more tractable models by distilling protein landscapes into a small number of essential states. Typically, this is achieved by making assumptions about the relevant features. In

particular, existing MSM libraries PyEMMA2[16] and MSMBuilder3[17-19] over state-of-the-art, modular components for the newest theoretical developments from the MSM community. These libraries emphasize early conversion to coarse-grained models, particularly through the use of time-lagged independent components analysis (tICA),[20-22] but also through deep learning[23,24] or explicit state-merging.[25-28] All these approaches merge states that are kinetically close to one another to build a more interpretable model.

Kinetic coarse-graining is effective when the most interesting process is also the slowest, for example, when studying folding. However, physiologically-relevant conformational changes also can occur quickly. For example, the opening of druggable cryptic allosteric sites can occur many orders of magnitude faster than the global unfolding process.[29,30] Thus, for biological questions where the underlying physical chemistry is irreducibly high-dimensional or the features in which it is low-dimensional are not known, building models with a large number of states is an effective strategy for ensuring that important states are not overlooked. An alternative approach to extracting insight from large MD datasets is to retain the size and high dimensionality, and to manually learn which features are relevant to the biological question. For example, one approach to understanding sequence-function relationships is to compare simulations of different sequences to form hypotheses about which features are important, which can then be used to propose experiments. This approach has been successfully leveraged to, for example, understand the determinants of protein stability,[8] enzyme catalysis,[6] and biochemical properties.[29] The downside of this approach is that it is substantially more computationally demanding, due to the much larger size of both the input features and the resulting model.

In this paper, we present enspara, which implements methods that improve the scalability of the MSM methods. We implement a “ragged array" data structure that enables memory-

efficient in-memory handling of data with heterogeneous lengths, and develop tools which use sparse matrices, vastly reducing memory usage of the models themselves while speeding up certain calculations on them. We further introduce clustering methods that can be parallelized across multiple nodes in a supercomputing cluster using MPI, a user-friendly command-line interface (CLI) for large clustering tasks, thread-parallelized routines for information-theoretic calculations, and a new framework for rapid experimentation with methods for estimating MSMs.

6.3 Results and Discussion

6.3.1 Ragged Arrays

The most computation-intensive step in any molecular dynamics-based approach is actually generating the simulation data. One approach to mustering the computation necessary to solve this problem is to harness the power of distributed computing to generate many parallel simulations on many computers. Indeed, one of the points where MSMs excel is in unifying such parallel simulations into a single model. An example of this is the distributed computing project Folding@home.[15] However, in these scenarios, individual trajectories often substantially differ in their lengths. In Folding@home, the trajectory length distribution shows strong positive skew, with a few trajectories one or more orders of magnitude longer than the median trajectory. Historically, atomic coordinates, as well as features computed on trajectories, have been represented as `square' arrays of $n_{\text{trajectories}} \times n_{\text{timepoints}} \times n_{\text{features}}$ (or $n_{\text{atoms}} \times 3$), which assumes uniform trajectory length.[16,31]

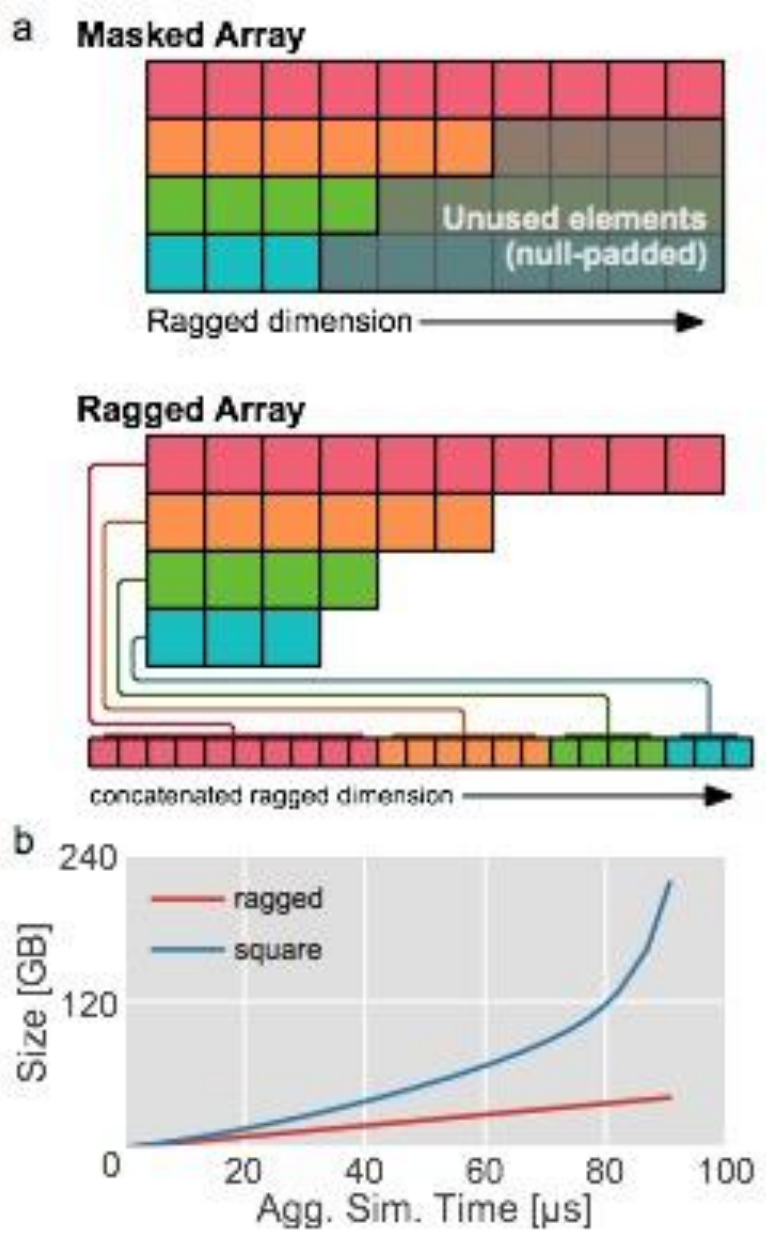

Figure 6.1: Ragged arrays compactly store non-uniform length data in memory. (a) A schematic comparison between the memory footprint of a masked, uniform array, and our implementation of the ragged array interface. In the masked array, rows of length lower than the longest row are padded with additional, null valued elements to preserve the uniformity of the array. In the ragged array, however, rows are stored concatenated and memory is not expended. (b) A plot of memory used by traditional and ragged arrays as a function of aggregate simulation time as trajectories of increasing length are added from a previously published Folding@home dataset.[11]

To represent non-uniform trajectory lengths, a number of approaches exist. One approach, found in MSMBuilder2,[17] is to use a two-dimensional square array with the 'overhanging' timepoints filled with a null value. This is also the solution provided by numpy,[32]

with its masked array object. While this approach maintains the in-memory arrangement that makes array slicing and indexing fast, it can dramatically inflate the memory footprint of datasets with highly non-uniform length distributions. The other approach, used by the latest version of MSMBuilder3[19] sacrifices speed for memory by building a python list of numpy arrays. While this is more memory-efficient, it cannot easily be sliced, cannot easily take advantage of numpy's vectorized array computations, and can be very slow to read and write from disk via python's general-purpose pickle library.

In enspara, we introduce an implementation of the ragged array, a data structure that relaxes the constraint that the rows in a two-dimensional array be the same length (Figure 6.1a). The ragged array maintains an end-to-end concatenated array of rows in memory. When the user requests access to particular elements using a slice or array indices, the object translates these array slices or element coordinates appropriately to the concatenated array, uses these translated coordinates to index into the concatenated array, and then reshapes the data appropriately and returns it to the user. On trajectory the length distributions described, the ragged array scales much better than the padded square array (Figure 6.1b), such as the square array used in MSMBuilder2 while retaining the useful properties of an array which are lost in a list-of-arrays representation.

6.3.2 SIMD Clustering Using MPI

Among the more expensive and worst-scaling steps in the Markov state model construction processes is clustering, and substantial effort has been spent on improving the speed of these calculations.[33,34] The most popular clustering algorithms for use in the MSM community are k-means[35] (generally composed of k-means++ initialization and Lloyd's algorithm[36] for refinement) for featurized data, and k-hybrid[17] (composed of k-centers[37] initialization and k-

medoids[38] refinement) for raw atomic coordinates. Both of these algorithms scale roughly with $O(nkdi)$, where n is the number of observations, d is the number of features per observation, k is the number of desired cluster centers, and i is the number of iterations required to converge. Unfortunately, with the possible exception of i, these numbers are all generally very large. As discussed below (Section II D), the number of clusters k must be large for some problems, proteins are intrinsically high-dimensional objects (i.e. high d), and the increasing speed of simulation calculations[39] has increased the number of timepoints that must be clustered, n, into the millions.

To address the poor scaling of clustering, the MSM community has developed a number of approaches to managing this problem. One approach is to reduce the number of observations by subsampling data[31] so that only every nth frame is used. Another approach is to reduce the number of features by including only certain atoms (as in Refs[8,40,41]), using a dimensionality reduction algorithm like principal components analysis (PCA),[42,43] or creating a hand-tuned set of order parameters (e.g. specific, relevant pairwise atomic distances). Yet a third approach is to use tICA as a dimensionality reduction, which has the benefit of reducing both the number of features and the number of clusters needed to satisfy the Markov assumption, but has the disadvantage that it may obscure important fast motions and can be sensitive to parameter choices (in particular the lag time).

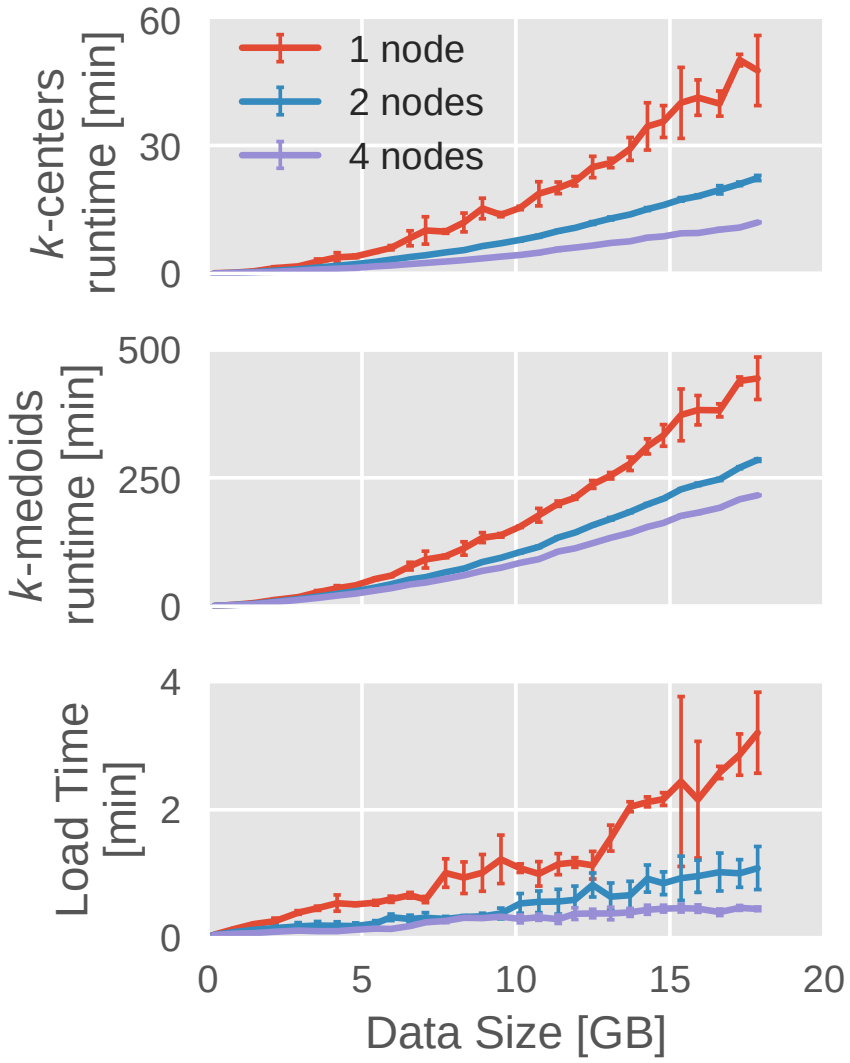

Figure 6.2: SIMD reformulation of clustering algorithms allows greater scaling. (a) The runtime of the parallelized k-centers code as a function of data input size. (b) The runtime of the parallelized k-medoids code as a function of data input size. (c) The load time of the parallel code as a function of input data size. Points represent the average and error bars the standard deviation across three trials.

An alternative or complimentary approach to preprocessing data to reduce input size is to parallelize the clustering algorithms themselves so that many hundreds, rather than many tens, of cores can be simultaneously utilized. Message Passing Interface (MPI)[44] is a parallel computing framework that enables communication between computers that are connected by low-latency, high-reliability computer networks, like those commonly encountered in academic cluster computing environments. This approach to interprocess communication has enabled numerous

successful parallel applications including molecular dynamics codes like GROMACS[45,46] (among many others). This approach to interprocess communication allows information to be shared easily across a network between an arbitrary number of distinct computers. Thus, for a successfully MPI-parallelized program, the amount of main memory and number of cores available is increased from what can be fit into one computer to what can be fit into one supercomputing cluster—a difference of one or two dozens of processors to hundreds of processors. However, because interprocess communication is potentially many orders of magnitude slower than, for example, in thread-parallelization, single-core algorithms must generally be adjusted to scale well under these constraints.

In this work, we present low-communication, same-instruction-multiple-data (SIMD) variants of clustering algorithms that are popular in the MSM community, *k*-centers, *k*-medoids, and *k*-hybrid. Specifically, data—atomic coordinates/features and distances between coordinates and medoids—are distributed between parallel processes which can reside on separate computers, allowing more data to be held in main memory, and allowing more processors in toto to be brought to bear on the data.

The k-centers initialization algorithm repeatedly computes the distance of all points to a particular point, and then identifies the maximum distance amongst all distances computed this way. This introduces the need for communication to (1) distribute the point to which distances will be computed and (2) collectively identify which distance is largest. (1) is solved trivially by the MPI scatter directive and (2) is solved by computing local maxima and then distributing these maxima with MPI allgather. Implementation details of k-medoids are somewhat more complex but follow a similar pattern. The full code is available on our GitHub repository. In brief, during each iteration, (1) all nodes must collaborate to choose a new random centroid for

each existing center—achieved by choosing a random number on the highest-ranked node and MPI scattering it to all other nodes—before (2) recomputing the assignment of each frame that could possibly have changed its state assignments. This step is potentially embarrassingly parallel in the number of frames assigned to the cluster. Finally, (3) the costs (usually mean-squared distances from each point to its cluster center) are computed and compared between the new and old assignments, and the cheaper assignment is accepted.

The performance characteristics of this implementation as a function of data input size is plotted in Figure 6.2a and b, which show marked decreases in runtime as additional computers are added to the computation. In both the k-centers and the k-medoid case, growth of runtime as a function of data input size is roughly quadratic. While this is expected for k-medoids, it may be surprising that k-centers also grows quadratically (see, for example, Ref. [34]). This is because we have chosen a fixed cluster radius for k-centers (rather than a fixed number of cluster centers). As new data (molecular dynamics trajectories with different initial velocities) are added, both the number of cluster centers and the number of data points to which each center must be compared increase, apparently roughly proportionally, leading to roughly quadratic scaling.

A further advantage of a parallelized algorithm is that, if configured correctly, it can also decrease load times. In the traditional high-performance computing (HPC) environment used in many academic settings, data typically resides on a single central, “head" node and it is distributed to \worker" nodes via a network file system (NFS). The NFS can transfer data to any particular worker node only as quickly as the network allows, which is generally orders of magnitude slower than the rate at which it can be loaded from disk into memory. However, if network topology allows nodes to independently communicate with the head node (and hence filesystem), the network bottleneck is reduced or removed, and load times can be substantially

decreased, as shown in Figure 6.2c. While load times do not dominate the overall runtime of the algorithms we discuss here, low load times are desirable since many forms of misconfiguration can only be detected after data has been loaded.

6.3.3 Flexible, Well-Scaling Clustering CLI

In this section, we illustrate how enspara can be used to analyze an MD dataset using our clustering command-line interface (CLI), and use the flexibility enspara offers to compare the usefulness of different ways of clustering the same MD trajectories.

Clustering, or assigning frames of the trajectory to discrete states, is the first step in analyzing most MD datasets using MSM technology. In enspara, we focus on offering mechanisms for clustering large datasets into many states, since other libraries already offer excellent mechanisms for reducing data size using various preprocessing strategies like tICA. For this purpose, enspara provides a command-line application, in addition to a clustering API, which handles some common tasks (Figure 6.3a-c). This clustering application can take trajectories in formats accepted by MDTraj (Figure 6.3a) or numpy arrays of numerical features (Figure 6.3c), supports several different distance metrics, provides easy support for clustering different topologies into shared state spaces (Figure 6.3b), and supports execution under MPI.

```
                --input   --configuration   --output
a)  enspara cluster
      --trajectories tem_0.xtc tem_1.xtc tem_2.xtc
      --topology  tem_wt.pdb
      --cluster-radius 0.3
      --cluster-iterations 20
      --algorithm khybrid
      --atoms 'name N or name CA or name C'
      --distances tem-distances.h5
      --assignments  tem-assignments.h5
      --center-features  tem-ctr-structs.h5
b)  enspara cluster
      --trajectories tem_wt_*.xtc
      --topology  tem_wt.pdb
      --trajectories tem_g238s_m182s_*.xtc
      --topology  tem_g238s_m182s.pdb
      --cluster-number 1000
      --algorithm kcenters
      --atoms '(not type H) and (residue 69 to 103)'
      --center-indices tem-combo-ctr-inds.npy
      --distances tem-combo-distances.h5
      --assignments  tem-combo-assignments.h5
      --center-features  tem-combo-ctr-structs.pkl
c)  enspara cluster
      --features tem1-sasas-*.npy
      --cluster-distance euclidean
      --cluster-number 100
      --subsample 10
      --algorithm khybrid
      --center-indices tem1-ctr-inds.npy
      --distances tem1-distances.h5
      --assignments tem1-assignments.h5
      --center-features tem1-center-structs.npy
```

d)

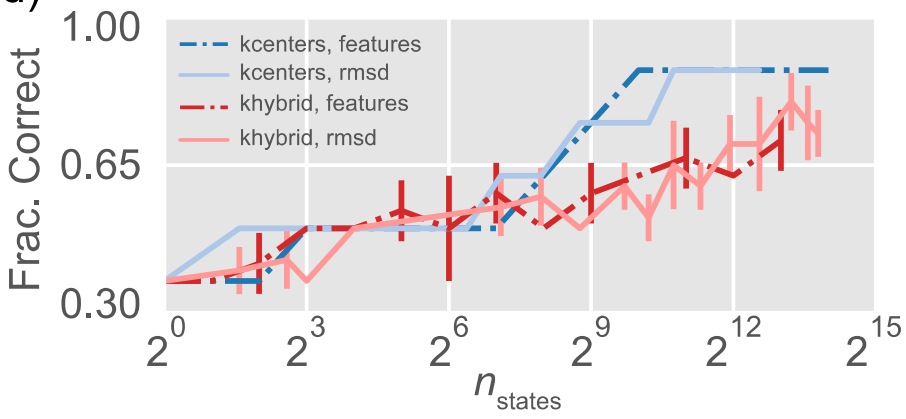

Figure 6.3: enspara offers a flexible, well-scaling, and multipurpose clustering CLI. (a) A CLI invocation clustering trajectories with a shared topology with the k-hybrid algorithm using backbone RMSD, stopping k-centers at 3 Å, and 20 rounds of k-medoids refinement. (b) A CLI invocation clustering trajectories with differing topologies by a small subset of shared atoms using the k-centers algorithm to discover 1000 states. (c) A CLI invocation clustering Euclidean distances between feature vectors representing frames stored in a group of numpy NPY-format files using

k-hybrid. (d) An MSM's ability to predict the results of an experimental measurement of solvent exposure as a function of number of clusters. Dashed lines are models constructed using Euclidean distances between vectors of residue sidechain solvent accessible surface areas, whereas solid lines use backbone RMSD. Blue traces used k-centers, and red traces used k-hybrid. The experimental measurement is a previously published[29] biochemical labeling assay that classifies a residue as exposed, buried, or transiently exposing. Residues exposure class was predicted as "buried" if no state exists where the residue was exposed, "exposed" if the residue is never buried, and "transient" if the residue populates both exposed and buried states in the MSM. The y-axis represents the fraction of these residues that were classified correctly. Error bars represent the standard deviation of three trials (k-centers are deterministic and have no error bars).

In enspara, we have implemented many of these options because different choices for cluster size/number, clustering algorithm, and cluster distance metric can dramatically impact an MSM's predictive power. As an example, in Figure 6.3d, we investigate the effect of clustering algorithm (k-centers vs. k-hybrid) and cluster number on the ability of an MSM to retrodict a previously-described biochemical thiol labeling assay.[29,30] In this case, the MSM's ability to sufficiently represent the protein's state space is positively related to the number of clusters used to represent the state space. Interestingly, k-centers appears to perform better than k-hybrid in this case. This may be related to the fact that these exposed states are high energy and hence rare, giving rise to a tendency in k-medoids to lump these rare states in with more populous adjacent states.

Because of this potential need for very large state spaces, it is often necessary to handle a large amount of data. In part, this challenge is a computer scientific one, which can be addressed by new parallel algorithms, such as that described above (Sec. 2). In addition to efficient algorithms, however, there are also software engineering concerns like effective memory management. Our CLI places an emphasis on these large clustering tasks and large state spaces, and hence scales better than existing codes that place an emphasis on smaller state spaces (Figure 6.4). For purposes of reference, clustering of the TEM-1 data set used all 2026 protein heavy atoms across 90.5 μs total simulation time saved every 100 ps and the G_q dataset used all 2655 protein heavy atoms across 20.5 μs saved every 10 ps. All these values trade off against one

another, however, meaning that if every 10th frame were used to cluster the G_q dataset, 205 μs of data could be clustered on a single node (and up to 1.03 ms on 5 nodes using MPI).

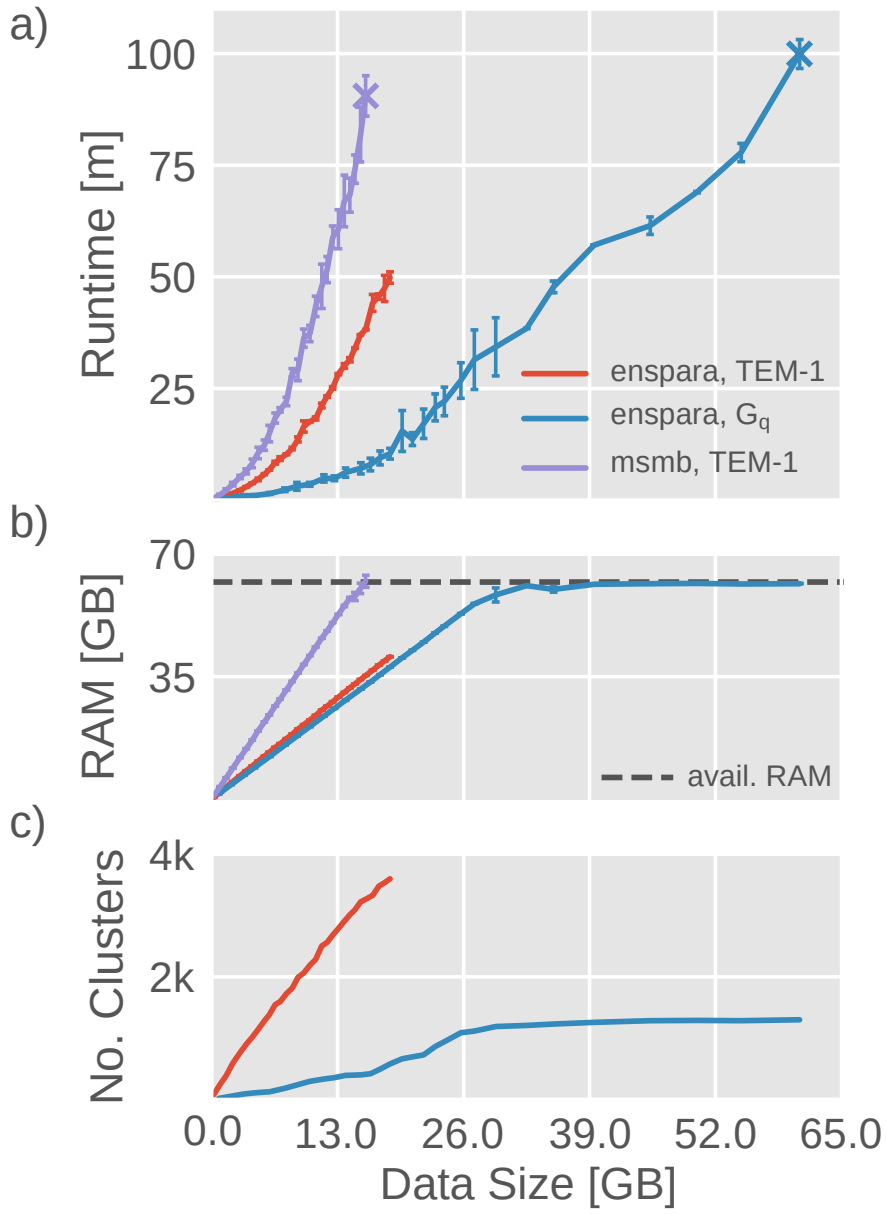

Figure 6.4: The CLI provided by enspara has favorable memory and performance characteristics. (a) Runtime as a function of data input size for the enspara cluster CLI on the TEM-1 and G_q datasets, and the MSMBuilder CLI on the TEM-1 dataset. For TEM-1/MSMBuilder and G_q/enspara, the final point represents the largest data size that can be run without exceeding available memory. (b) Process-allocated memory usage as a function of data input size for the enspara cluster CLI on the TEM-1 and G_q datasets, and the MSMBuilder CLI on the TEM-1 dataset. Apparent memory use by enspara appears to stop growing after 32 GB because, on the computer system tested (see section 6.5), the operating system allocates double the necessary RAM to enspara. Where MSMBuilder runs out of RAM loading ~16 GB, enspara is capable of using almost all of the available 64 GB RAM. (c) Number of clusters as a function of data input size for TEM-1 and G_q datasets. The change in runtime growth of the G_q dataset around 26

GB of data loaded is a consequence of the slowdown in state discovery as the new data are added. For (a) and (b), error bars represent the standard deviation of three trials.

6.3.4 Sparse Matrix Integration

Building a Markov state model with tens of thousands of states presents some methodological challenges. One of these is the representation of the transition counts and transition probability matrices. Most straightforwardly, this is achieved using dense arrays, such as the array or matrix classes available in numpy, and this is the strategy employed by extant MSM softwares, MSMBuilder3and PyEMMA. The problem with this representation is that the memory usage of these matrices grows with the square of the number of states in the model. To make matters worse, the computational cost of the eigendecomposition that is typically required to calculate a model's stationary distribution (equilibrium probabilities) and principal relaxation modes grows with the cube of the number of elements in the matrix. [47]

To address the computational challenges posed by traditional arrays, enspara has been engineered to support sparse arrays wherever possible. Sparse arrays have been supported by MSMBuilder in the past but were dropped with version 3. PyEMMA also makes heavy use of dense arrays, although there is some support for sparse arrays. Sparse arrays, rather than growing strictly with the square of the number of states, grow linearly in the number of nonzero elements in the array. In the worst case, where every element of the transition counts matrix is non-zero (i.e. every possible transition between pairs of states is observed) this becomes the dense case. However, this is very unusual: the number of observed transitions is generally several orders of magnitude smaller than the number of possible transitions (Figure 6.5a). By implementing routines that support scipy's sparse matrices, it becomes possible to keep much larger Markov state models in memory (Figure 6.5b) and analyze those models much more quickly (Figure 6.5c).

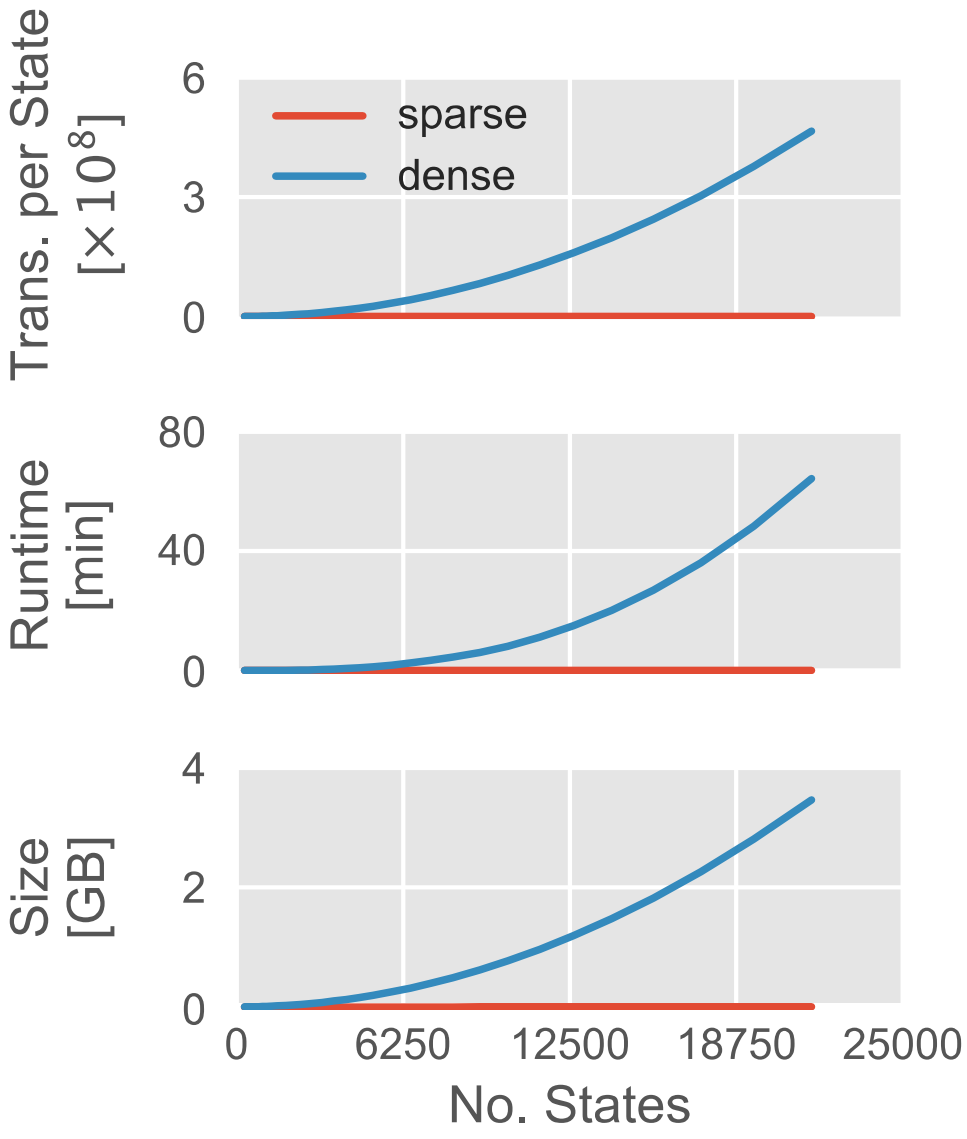

Figure 6.5: The performance characteristics of sparse and dense metrics representing the same MSM. (a) The mean number of transitions per state in a transition counts matrix as a function of the number of states in the model. Any pair of states with an observed transition between them has a nonzero entry in the transition counts matrix and consumes memory in both sparse and dense cases. In contrast, a sparse matrix does not require memory for the zero elements of the transition counts matrix. (b) The runtime of an eigendecomposition as a function of the number of states in a model. (c) The memory footprint of the transition probability matrix as a function of the number of states in a model.

6.3.5 Fast and MSM-Ready Information Theory Routines

Recent work[48-50] has demonstrated the usefulness of information theory, and mutual information (MI) in particular, for identifying and understanding the salient features of conformational

ensembles. MI is a nonlinear measurement of the statistical non-independence of two random variables. MI is given by

$$\mathrm{MI}(\mathrm{X},\mathrm{Y}) = \sum_{\mathrm{y}\in\mathrm{Y}} \sum_{\mathrm{x}\in\mathrm{X}} \mathrm{P}(\mathrm{x},\mathrm{y}) \log\left(\frac{\mathrm{P}(\mathrm{x},\mathrm{y})}{\mathrm{P}(\mathrm{x})\mathrm{P}(\mathrm{y})}\right) \qquad (1)$$

where $\mathrm{P}(\mathrm{x})$ is the probability that random variable X takes on value x, $\mathrm{P}(\mathrm{y})$ is the probability that random variable Y takes on value y, and $\mathrm{P}(\mathrm{x},\mathrm{y})$ is the joint probability that random variable X takes on value x and that random variable Y takes on value y.

Historically, the joint distribution $\mathrm{P}(\mathrm{x},\mathrm{y})$ is estimated by counting the number of times that combination of features appeared in each frame. This computation can become a bottleneck when it must be computed over hundreds or thousands of different features and for datasets with hundreds of thousands or millions of observations. This is because it is highly iterative—which is notoriously slow in many higher-level programming languages like python or Matlab—and because the number of joint distributions that must be calculated grows with the square of the number of features to be tracked. Consequently, in the worst case, this involves examining every frame of a trajectory n^2 times, where n is the number of random variables of interest.

In enspara, we take two overlapping approaches to address the problem of the poor scalability of pairwise MI calculations. The first approach is to use the joint distribution implied by the equilibrium probabilities of a Markov state model, rather than by counting co-occurrences from full trajectories. Specifically, the joint probability $\mathrm{P}(\mathrm{x},\mathrm{y})$ is estimated by $\sum_{s\in S} \pi(s)$, where $\pi(s)$ is the equilibrium probability of states from the MSM and S is the set of states where x = X and y = Y . This works by reducing the number of individual observations, usually by orders of

magnitude. Existing codes[49,51] either do not provide the option to compute MI with weighted observations or require a specific object-based framework to do so.[52]

Our second approach is to implement a fast joint counts calculation routine. This routine is both thread-parallelized and much faster than the equivalent numpy routine even on a single core. This approach is needed because, in some cases, information from a Markov state model cannot be trivially substituted for frame-by-frame calculations. To address this case, we also implement a function using cython[53] and OpenMP[54] that takes a trajectory of n features and returns a four-dimensional joint counts array with dimension $n \times n \times s_n \times s_n$, where s_n is the number of values each feature n can take on. The value of returning this four-dimensional joint counts matrix is that it renders the problem embarrassingly parallel in the number of trajectories: this function can be run on each trajectory totally independently, and the resulting joint counts matrices can be summed before being normalized to compute joint probabilities. We recommend combining this with a pipelining software like Jug.[55]

Additionally, in this package, we include a reference implementation of Correlation of All Rotameric and Dynamical States framework (CARDS).[49] In brief, this method takes a series of molecular dynamics trajectories and computes the mutual information (MI) between all pairs of dihedral angle rotameric states, and between all pairs of dihedral angle order/disorder states. A dihedral angle is considered disordered if it frequently hops between rotameric states. This implementation parallelizes across cores on a single machine using the thread-parallelization described in Section 6.3.5.

6.3.6 Flexible and Interoperable Model Fitting and Analysis

With enspara, a major goal is maximal flexibility. This means loosely-coupled, function-based components and the use of widely-accepted datatypes for input and output of these functions.

This helps us maximize interoperability with existing MSM softwares, other python libraries, and prototypes of novel analysis strategies in the future. One important way we achieve flexibility in enspara is by constructing an API that accepts widely-used datatypes, rather than datatypes that are unique to enspara. This is most important for our analysis functions, which accept parameters of MSMs rather than MSM objects themselves. For example, mutual information calculations (Section 6.3.5) that use equilibrium probabilities from an MSM accept a vector of probabilities rather than an MSM object. (Note also that any function that accepts a RaggedArray will also accept a numpy array.) A crucial consequence of this API pattern is that enspara's MSM analysis routines are interoperable with both PyEMMA's and MSMBuilder's MSM objects. It also allows integration with simple, hand-crafted models, as it was used to do in Zimmerman *et al.*[56]

Another way we achieve flexibility is to preference function-based semantics over object-based semantics. A successful and prominent API pattern for machine learning tasks was promulgated by scikit-learn, which represents various machine learning tasks (clustering, featurization, etc.) as objects. While this nicely contains the logic and complexities of each algorithm inside a fairly uniform API, it also makes the behavior of these algorithms difficult to modify with novel approaches, since new ideas must either be integrated into the existing object completely or the object must be entirely duplicated. An object can also obscure state from the user, hindering comprehension, modification, or reuse of code. To address this in enspara, wherever we have created object interfaces exist, they are thin wrappers for chains of function calls. Consequently, an interested user can then easily intercept control ow to inject new behavior.

A noteworthy example of this in enspara is our semantic for estimating transition probability matrices. Estimating a transition probability matrix from observed state transitions is a crucial step in building an MSM, yet there is not a uniform procedure for accomplishing this that works in all cases. Many different estimators exist, and more are in active development.[31,56-64] Perhaps the simplest procedure to estimate the transition probability matrix, T, is to row-normalize the transition count matrix, C,

$$T_{ij}^{normalize} = \frac{C_{ij}}{\sum_k C_{ik}} \quad (2)$$

where T_{ij} is the probability of observing a transition from state i to j and C_{ij} is the number of times such a transition was observed. While this method is simple, it can and often does generate a non-ergodic state space. In an effort to address this difficulty and to condition the MSM to be well-behaved, one can include an additional pseudocount, $\tilde{C}$, before estimation,

$$T_{ij}^{pseduo} = \frac{C_{ij}+\tilde{C}}{\sum_k C_{ik}+\tilde{C}} \quad (3)$$

which ensures ergodicity. A more dramatic conditioning comes when forcing the counts matrix to obey detailed balance by averaging forward and reverse transitions:

$$C_{ij}^{transpose} = \frac{(C_{ij}+C_{ji})}{2} \quad (4)$$

$$T_{ij}^{transpose} = \frac{C_{ij}^{transpose}}{\sum_k C_{ik}} \quad (5)$$

Yet a third proposed way of estimating an MSM is to find the maximum likelihood estimate for T subject to the constraint that it satisfies detailed balance.[4,31] Framed as a Bayesian inference, the transition probabilities are solved as the most likely given a transition counts matrix, such that,

$$T_{ij}^{MLE} = \arg\max P\left(T_{ij}^{*} \middle| C_{ij}\right) \quad (6)$$

Additionally, there exist more sophisticated schemes of estimation, such as those that draw on inspiration from observable operator models,[57] and projected MSMs.[65] While it is beyond the scope of this article to review this area of study in exhaustive detail, we hope these few examples demonstrate the variety and importance of estimators. This poses a major challenge to writing a framework that can readily estimate a transition probability matrix; estimators are an active area of research, and a flexible framework that allows users to quickly modify an existing estimator or try a new one would be of great utility.

To address this difficulty, we treat fitting methods as simple functions, which we call builders, that take a transition counts matrix and return transition and equilibrium probabilities. These built-in functions, along with our MSM object can be used to quickly fit an MSM using commonly-used approaches (Figure 6.6a). Alternatively, for users who wish to slightly modify existing MSM estimation methods, the function-level interface provides fine-grained control over the steps in fitting an MSM (Figure 6.6b). Finally, for users who wish to prototype entirely new MSM estimation methods, any function or callable object is accepted as a builder, as long as

it accepts a transition counts matrix C as input and returns a 2-tuple of transition probabilities and equilibrium probabilities.

a)
```
from enspara import msm

m = msm.MSM(lag_time=10,
            method=msm.builders.transpose)
m.fit(assignments)
```

b)
```
from enspara.msm import builders

C = msm.assigns_to_counts(assignments,
                          lag_time=10)
T, pi = msm.builders.transpose(C)
```

c)
```
def custom_builder(C, alpha, *args, **kwargs):
    """A custom builder that creates a convex
    combination of the transpose and normalize
    builders.
    """
    T1, pi1 = msm.builders.transpose(
         C, *args, **kwargs)
    T2, pi2 = msm.builders.normalize(
         C, *args, **kwargs)

    T = alpha*T1 + (1-alpha)*T2
    pi = alpha*pi1 + (1-alpha)*pi2

    return T, pi
```

Figure 6.6: (a) An example usage of the high-level, object-based API to fit a Markov state model. (b) An example usage of enspara's low-level, function-based API to fit a Markov state model. (c) A custom method that fits a Markov state model and is interoperable with enspara's existing API.

6.4 Conclusions

In this work, we have presented enspara, a library for building Markov state models at scale. We introduced an implementation of the ragged array, which dramatically improved the memory footprint of MSM-associated data. We developed a low-communication, parallelized version of the classic *k*-centers and *k*-medoids clustering algorithms, which simultaneously reduce runtime

and load time while vastly increasing the ceiling on memory use for those algorithms by allowing execution on multiple computers simultaneously. Enspara also has turn-key sparse matrix usage. Finally, we implement a function-based API for MSM estimators that greatly increases the flexibility of MSM estimation to enable rapid experimentation with different methods of fitting. Taken together, these features make enspara the ideal choice of MSM library for many-state, large-data MSM construction and analysis.

6.5 Methods

6.5.1 Source Code and Documentation

The source code to enspara is available on GitHub at https://github.com/bowman-lab/enspara, where installation instructions can also be found. In brief, it can be downloaded from GitHub and installed using setup.py.

Documentation takes two forms, docstrings and a documentation website. Individual functions and objects are documented as docstrings, which indicate parameters and return values, and briefly describe each functions role. The library as a whole is documented at https://enspara.readthedocs.io, which gives a high-level description of the library's functionality, as well as providing worked-through examples of enspara's use.

Finally, at https://enspara.readthedocs.io/tutorial, we give an in-depth tutorial example analyzing data from a public dataset.

6.5.2 Libraries and Hardware

Eigenvector/eigenvalue decomposition experiments were performed on a Ubuntu 16.04.5 (xenial) workstation with an Intel i7-5820K CPU @ 3.30GHz (12 cores) with 32GB of RAM using SciPy version 1.1.0 and numpy 1.13.3. Probabilities were represented as 8-byte floating

point numbers.

Thread parallelization experiments were performed on the same hardware using OpenMP 4.0 (2013.07) with gcc 5.4.0 (2016.06.09) and cython 0.26 in Python 3.6.0, distributed by Continuum Analytics in conda 4.5.11.

Clustering scaling experiments were performed on identical computers running CentOS Linux release 7.3.1611 (Core) with Intel Xeon E5-2697 v2 CPUs @ 2.70GHz and 64 GB of RAM linked to a head node with two Intel 10-Gigabit X540-AT2 ethernet adapters and nfs-utils 1.3.0. We used the mpi4py[66-68] and Python 3.6.0 with Open MPI 2.0.2. Clustering used as a distance metric the RMSD function provided in the MDTraj 1.9.1.[33]

6.5.3 Simulation Data

For example, simulation data, we used a previously-published 90.5 μs TEM-1 β-lactamase dataset[9] and a 122.6 μs G_q dataset.[69] As described previously, simulations were run at 300 K with the GROMACS software package[45] using the Amber03 force field[70] and TIP3P34 explicit solvent. Data was generated using the Folding@home distributed computing platform.[71]

6.5.4 Residue Labeling Analysis

Residue labeling behavior for residues A150, L190, S203, A232, A249, I260, and L286 was measured in Bowman *et al.*[29] and for S243 in Porter *et al.*[30]. "Exposed" residues label almost immediately, "pocket" or "transiently-labeling" residues label on the order of 10^{-3} or 10^{-4} s^{-1}, and buried residues label on the order over days.

Residue labeling behavior was predicted according to the procedure described in Ref. 50. In brief, sidechain atoms' solvent exposure to a 2.8 Å probe was calculated (using the Shrake-

Rupley[72] algorithm implemented by MDTraj[33]) for the representative structure for each MSM state, and the residue was called as exposed if its exposed area exceeded 2 $Å^2$.

Bibliography

(1) Bowman, G. R.; Pande, V. S.; Noé, F. Introduction and Overview of This Book. In *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*; Advances in Experimental Medicine and Biology; Springer Netherlands: Dordrecht, 2014; Vol. 797, pp 1–6.

(2) Husic, B. E.; Pande, V. S. Markov State Models: From an Art to a Science. *Journal of the American Chemical Society* **2018**, *140* (7), 2386–2396.

(3) Pande, V. S.; Beauchamp, K.; Bowman, G. R. Everything You Wanted to Know About Markov State Models but Were Afraid to Ask. *Methods* **2010**, *52* (1), 99–105.

(4) Prinz, J.-H.; Wu, H.; Sarich, M.; Keller, B.; Senne, M.; Held, M.; Chodera, J. D.; Schütte, C.; Noé, F. Markov Models of Molecular Kinetics: Generation and Validation. *The Journal of Chemical Physics* **2011**, *134* (17), 174105.

(5) Chodera, J. D.; Swope, W. C.; Pitera, J. W.; Dill, K. A. Long-Time Protein Folding Dynamics From Short-Time Molecular Dynamics Simulations. *Multiscale Modeling & Simulation* **2006**, *5* (4), 1214–1226.

(6) Hart, K. M.; Ho, C. M. W.; Dutta, S.; Gross, M. L.; Bowman, G. R. Modelling Proteins' Hidden Conformations to Predict Antibiotic Resistance. *Nature Communications* **2016**, *7*, 12965.

(7) Noé, F.; Doose, S.; Daidone, I.; Löllmann, M.; Sauer, M.; Chodera, J. D.; Smith, J. C. Dynamical Fingerprints for Probing Individual Relaxation Processes in Biomolecular Dynamics with Simulations and Kinetic Experiments. *Proc. Natl. Acad. Sci. U.S.A.* **2011**, *108* (12), 4822–4827.

(8) Zimmerman, M. I.; Hart, K. M.; Sibbald, C. A.; Frederick, T. E.; Jimah, J. R.; Knoverek, C. R.; Tolia, N. H.; Bowman, G. R. Prediction of New Stabilizing Mutations Based on Mechanistic Insights From Markov State Models. *ACS Cent. Sci.* **2017**, *3* (12), 1311–1321.

(9) Bowman, G. R.; Geissler, P. L. Equilibrium Fluctuations of a Single Folded Protein Reveal a Multitude of Potential Cryptic Allosteric Sites. *Proc. Natl. Acad. Sci. U.S.A.* **2012**, *109* (29), 11681–11686.

(10) Bowman, G. R.; Pande, V. S. Protein Folded States Are Kinetic Hubs. *Proc. Natl. Acad. Sci. U.S.A.* **2010**, *107* (24), 10890–10895.

(11) Buch, I.; Giorgino, T.; De Fabritiis, G. Complete Reconstruction of an Enzyme-Inhibitor Binding Process by Molecular Dynamics Simulations. *Proc. Natl. Acad. Sci. U.S.A.* **2011**, *108* (25), 10184–10189.

(12) Collins, A. P.; Anderson, P. C. Complete Coupled Binding–Folding Pathway of the Intrinsically Disordered Transcription Factor Protein Brinker Revealed by Molecular Dynamics Simulations and Markov State Modeling. *Biochemistry* **2018**, *57* (30), 4404–4420.

(13) Shaw, D. E.; Deneroff, M. M.; Dror, R. O.; Kuskin, J. S.; Larson, R. H.; Salmon, J. K.; Young, C.; Batson, B.; Bowers, K. J.; Chao, J. C.; Eastwood, M. P.; Gagliardo, J.; Grossman, J. P.; Ho, C. R.; Ierardi, D. J.; Kolossváry, I.; Klepeis, J. L.; Layman, T.; McLeavey, C.; Moraes, M. A.; Mueller, R.; Priest, E. C.; Shan, Y.; Spengler, J.; Theobald, M.; Towles, B.; Wang, S. C. Anton, a Special-Purpose Machine for Molecular Dynamics Simulation. *Communications of the ACM* **2008**, *51* (7), 91–97.

(14) Shaw, D. E.; Maragakis, P.; Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; Eastwood, M. P.; Bank, J. A.; Jumper, J. M.; Salmon, J. K.; Shan, Y.; Wriggers, W. Atomic-Level Characterization of the Structural Dynamics of Proteins. *Science* **2010**, *330* (6002), 341–346.

(15) Shirts, M. COMPUTING: Screen Savers of the World Unite! *Science* **2000**, *290* (5498), 1903–1904.

(16) Scherer, M. K.; Trendelkamp-Schroer, B.; Paul, F.; Pérez-Hernández, G.; Hoffmann, M.; Plattner, N.; Wehmeyer, C.; Prinz, J.-H.; Noé, F. PyEMMA 2: a Software Package for Estimation, Validation, and Analysis of Markov Models. *J. Chem. Theory Comput.* **2015**, *11* (11), 5525–5542.

(17) Beauchamp, K. A.; Bowman, G. R.; Lane, T. J.; Maibaum, L.; Haque, I. S.; Pande, V. S. MSMBuilder2: Modeling Conformational Dynamics on the Picosecond to Millisecond Scale. *J. Chem. Theory Comput.* **2011**, *7* (10), 3412–3419.

(18) Bowman, G. R.; Huang, X.; Pande, V. S. Using Generalized Ensemble Simulations and Markov State Models to Identify Conformational States. *Methods* **2009**, *49* (2), 197–201.

(19) Harrigan, M. P.; Sultan, M. M.; Hernández, C. X.; Husic, B. E.; Eastman, P.; Schwantes, C. R.; Beauchamp, K. A.; McGibbon, R. T.; Pande, V. S. MSMBuilder: Statistical Models for Biomolecular Dynamics. *Biophysical Journal* **2017**, *112* (1), 10–15.

(20) Naritomi, Y.; Fuchigami, S. Slow Dynamics in Protein Fluctuations Revealed by Time-Structure Based Independent Component Analysis: the Case of Domain Motions. *The Journal of Chemical Physics* **2011**, *134* (6), 065101.

(21) Pérez-Hernández, G.; Paul, F.; Giorgino, T.; De Fabritiis, G.; Noé, F. Identification of Slow Molecular Order Parameters for Markov Model Construction. *The Journal of Chemical Physics* **2013**, *139* (1), 015102.

(22) Schwantes, C. R.; Pande, V. S. Modeling Molecular Kinetics with tICA and the Kernel Trick. *J. Chem. Theory Comput.* **2015**, *11* (2), 600–608.

(23) Mardt, A.; Pasquali, L.; Wu, H.; Noé, F. VAMPnets for Deep Learning of Molecular Kinetics. *Nature Communications* **2018**, *9* (1), 5.

(24) Paul, F.; Wu, H.; Vossel, M.; de Groot, B. L.; Noé, F. Identification of Kinetic Order Parameters for Non-Equilibrium Dynamics. *The Journal of Chemical Physics* **2019**, *150* (16), 164120.

(25) Deuflhard, P.; Dellnitz, M.; Junge, O.; Schütte, C. Computation of Essential Molecular Dynamics by Subdivision Techniques. In *Computational Molecular Dynamics: Challenges, Methods, Ideas*; Lecture Notes in Computational Science and Engineering; Springer, Berlin, Heidelberg: Berlin, Heidelberg, 1999; Vol. 4, pp 98–115.

(26) Deuflhard, P.; Weber, M. Robust Perron Cluster Analysis in Conformation Dynamics. *Linear Algebra and its Applications* **2005**, *398*, 161–184.

(27) Sheong, F. K.; Silva, D.-A.; Meng, L.; Zhao, Y.; Huang, X. Automatic State Partitioning for Multibody Systems (APM): an Efficient Algorithm for Constructing Markov State Models to Elucidate Conformational Dynamics of Multibody Systems. *J. Chem. Theory Comput.* **2014**, *11* (1), 17–27.

(28) Wang, W.; Liang, T.; Sheong, F. K.; Fan, X.; Huang, X. An Efficient Bayesian Kinetic Lumping Algorithm to Identify Metastable Conformational States via Gibbs Sampling. *The Journal of Chemical Physics* **2018**, *149* (7), 072337.

(29) Bowman, G. R.; Bolin, E. R.; Hart, K. M.; Maguire, B. C.; Marqusee, S. Discovery of Multiple Hidden Allosteric Sites by Combining Markov State Models and Experiments. *Proc. Natl. Acad. Sci. U.S.A.* **2015**, *112* (9), 2734–2739.

(30) Porter, J. R.; Moeder, K. E.; Sibbald, C. A.; Zimmerman, M. I.; Hart, K. M.; Greenberg, M. J.; Bowman, G. R. Cooperative Changes in Solvent Exposure Identify Cryptic Pockets, Switches, and Allosteric Coupling. *Biophysical Journal* **2019**, *116* (5), 818–830.

(31) Bowman, G. R.; Beauchamp, K. A.; Boxer, G.; Pande, V. S. Progress and Challenges in the Automated Construction of Markov State Models for Full Protein Systems. *The Journal of Chemical Physics* **2009**, *131* (12), 124101.

(32) van der Walt, S.; Colbert, S. C.; Varoquaux, G. The NumPy Array: a Structure for Efficient Numerical Computation. *Computing in Science & Engineering* **2011**, *13* (2), 22–30.

(33) McGibbon, R. T.; Beauchamp, K. A.; Harrigan, M. P.; Klein, C.; Swails, J. M.; Hernández, C. X.; Schwantes, C. R.; Wang, L.-P.; Lane, T. J.; Pande, V. S. MDTraj: a Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophysical Journal* **2015**, *109* (8), 1528–1532.

(34) Zhao, Y.; Sheong, F. K.; Sun, J.; Sander, P.; Huang, X. A Fast Parallel Clustering Algorithm for Molecular Simulation Trajectories. *J Comput Chem* **2013**, *34* (2), 95–104.

(35) Hartigan, J. A.; Wong, M. A. Algorithm as 136: a K-Means Clustering Algorithm. *Applied Statistics* **1979**, *28* (1), 100.

(36) theory, S. L. I. T. O. I.; 1982. Least Squares Quantization in PCM. *sites.cs.ucsb.edu*.

(37) Gonzalez, T. F. Clustering to Minimize the Maximum Intercluster Distance. *Theoretical Computer Science* **1985**, *38*, 293–306.

(38) Dasgupta, S. Performance Guarantees for Hierarchical Clustering. In *Computational Learning Theory*; Lecture Notes in Computer Science; Springer, Berlin, Heidelberg: Berlin, Heidelberg, 2002; Vol. 2375, pp 351–363.

(39) Lane, T. J.; Shukla, D.; Beauchamp, K. A.; Pande, V. S. To Milliseconds and Beyond: Challenges in the Simulation of Protein Folding. *Current Opinion in Structural Biology* **2013**, *23* (1), 58–65.

(40) Bowman, G. R.; Geissler, P. L. Extensive Conformational Heterogeneity Within Protein Cores. *J. Phys. Chem. B* **2014**, *118* (24), 6417–6423.

(41) Schwantes, C. R.; Pande, V. S. Improvements in Markov State Model Construction Reveal Many Non-Native Interactions in the Folding of NTL9. *J. Chem. Theory Comput.* **2013**, *9* (4), 2000–2009.

(42) Jain, A.; Hegger, R.; Stock, G. Hidden Complexity of Protein Free-Energy Landscapes Revealed by Principal Component Analysis by Parts. *J. Phys. Chem. Lett.* **2010**, *1* (19), 2769–2773.

(43) Shlens, J. A Tutorial on Principal Component Analysis. April 3, 2014.

(44) Clarke, L.; Glendinning, I.; Hempel, R. The MPI Message Passing Interface Standard. In *Programming Environments for Massively Parallel Distributed Systems*; Birkhäuser, Basel: Basel, 1994; pp 213–218.

(45) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High Performance Molecular Simulations Through Multi-Level Parallelism From Laptops to Supercomputers. *SoftwareX* **2015**, *1-2*, 19–25.

(46) Van Der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E.; Berendsen, H. J. C. GROMACS: Fast, Flexible, and Free. *J Comput Chem* **2005**, *26* (16), 1701–1718.

(47) Pan, V. Y.; Chen, Z. Q. *The Complexity of the Matrix Eigenproblem*; ACM: New York, New York, USA, 1999; pp 507–516.

(48) McClendon, C. L.; Friedland, G.; Mobley, D. L.; Amirkhani, H.; Jacobson, M. P. Quantifying Correlations Between Allosteric Sites in Thermodynamic Ensembles. *J. Chem. Theory Comput.* **2009**, *5* (9), 2486–2502.

(49) Singh, S.; Bowman, G. R. Quantifying Allosteric Communication via Both Concerted Structural Changes and Conformational Disorder with CARDS. *J. Chem. Theory Comput.* **2017**, *13* (4), 1509–1517.

(50) Wayment-Steele, H. K.; Hernández, C. X.; Pande, V. S. Modelling Intrinsically Disordered Protein Dynamics as Networks of Transient Secondary Structure. *bioRxiv* **2018**, *140*, 377564.

(51) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, É. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12* (Oct), 2825–2830.

(52) Hernández, C. X.; Software, V. P. J. O. S.; 2017. MDEntropy: Information-Theoretic Analyses for Molecular Dynamics. *theoj.org*.

(53) Behnel, S.; Bradshaw, R.; Citro, C.; Dalcin, L.; Seljebotn, D. S.; Smith, K. Cython: the Best of Both Worlds. *Computing in Science & Engineering* **2011**, *13* (2), 31–39.

(54) Dagum, L.; Engineering, R. M. C. I. S.; 1998. OpenMP: an Industry-Standard API for Shared-Memory Programming. *cs.swarthmore.edu*.

(55) Coelho, L. P. Jug: Software for Parallel Reproducible Computation in Python. *Journal of Open Research Software* **2017**, *5* (1), 022109.

(56) Zimmerman, M. I.; Porter, J. R.; Sun, X.; Silva, R. R.; Bowman, G. R. Choice of Adaptive Sampling Strategy Impacts State Discovery, Transition Probabilities, and the Apparent Mechanism of Conformational Changes. May 11, 2018.

(57) Wu, H.; Prinz, J.-H.; Noé, F. Projected Metastable Markov Processes and Their Estimation with Observable Operator Models. *The Journal of Chemical Physics* **2015**, *143* (14), 144101.

(58) Trendelkamp-Schroer, B.; Noé, F. Efficient Estimation of Rare-Event Kinetics. *Phys. Rev. X* **2016**, *6* (1), 011009.

(59) Trendelkamp-Schroer, B.; Noé, F. Efficient Bayesian Estimation of Markov Model Transition Matrices with Given Stationary Distribution. *The Journal of Chemical Physics* **2013**, *138* (16), 164113.

(60) Stelzl, L. S.; Hummer, G. Kinetics From Replica Exchange Molecular Dynamics Simulations. *J. Chem. Theory Comput.* **2017**, *13* (8), 3927–3935.

(61) Metzner, P.; Noé, F.; Schütte, C. Estimating the Sampling Error: Distribution of Transition Matrices and Functions of Transition Matrices for Given Trajectory Data. *Physical Review E* **2009**, *80* (2), 021106.

(62) Leahy, C. T.; Kells, A.; Hummer, G.; Buchete, N.-V.; Rosta, E. Peptide Dimerization-Dissociation Rates From Replica Exchange Molecular Dynamics. *The Journal of Chemical Physics* **2017**, *147* (15), 152725.

(63) Grinstead, C. M.; Snell, J. L. *Introduction to Probability*; 2012.

(64) and, N.-V. B.; Hummer, G. *Coarse Master Equations for Peptide Folding Dynamics†*; American Chemical Society, 2008; Vol. 112, pp 6057–6069.

(65) Noé, F.; Wu, H.; Prinz, J.-H.; Plattner, N. Projected and Hidden Markov Models for Calculating Kinetics and Metastable States of Complex Molecules. *The Journal of Chemical Physics* **2013**, *139* (18), 184114.

(66) Dalcin, L.; Paz, R.; Storti, M. MPI for Python. *Journal of Parallel and Distributed Computing* **2005**, *65* (9), 1108–1115.

(67) Dalcin, L.; Paz, R.; Storti, M.; D'Elía, J. MPI for Python: Performance Improvements and MPI-2 Extensions. *Journal of Parallel and Distributed Computing* **2008**, *68* (5), 655–662.

(68) Dalcin, L. D.; Paz, R. R.; Kler, P. A.; Cosimo, A. Parallel Distributed Computing Using Python. *Advances in Water Resources* **2011**, *34* (9), 1124–1139.

(69) Sun, X.; Singh, S.; Blumer, K. J.; eLife, G. B.; 2018. Simulation of Spontaneous G Protein Activation Reveals a New Intermediate Driving GDP Unbinding. *cdn.elifesciences.org*.

(70) Duan, Y.; Wu, C.; Chowdhury, S.; Lee, M. C.; Xiong, G.; Zhang, W.; Yang, R.; Cieplak, P.; Luo, R.; Lee, T.; Caldwell, J.; Wang, J.; Kollman, P. A Point-Charge Force Field for Molecular Mechanics Simulations of Proteins Based on Condensed-Phase Quantum Mechanical Calculations. *J Comput Chem* **2003**, *24* (16), 1999–2012.

(71) Shirts, M.; Pande, V. S. Screen Savers of the World Unite! *Science* **2000**, *290* (5498), 1903–1904.

(72) Shrake, A.; Rupley, J. A. Environment and Exposure to Solvent of Protein Atoms. Lysozyme and Insulin. *Journal of Molecular Biology* **1973**, *79* (2), 351–371.

Chapter 7

Conclusions

7.1 Main Findings

As summarized in chapter 1, accessing a proteins' conformational ensemble using MD simulations has been a tantalizing goal. Before the work in this thesis, obtaining meaningful conformational dynamics of even small protein systems has only been possible with special purpose or massively parallelized computing platforms. While there is no free lunch when it comes to sampling, this thesis has shown that the developed FAST algorithm can make more efficient use of computational hardware to explore protein conformational landscapes. In this section, I review the main findings and results in the development, analysis, and application of the FAST algorithm.

The FAST algorithm is first introduced in chapters 2-3, where the algorithmic details are given and potential applications are explored. A fundamental principle of FAST is that there exist gradients in conformational space that can be followed—i.e. if we want to find states with large solvent accessible surface areas, sampling from the states with the largest surface area will be more likely to discover even larger surface areas than states with very low surface areas. Evidence for this is provided from analyzing the highest-flux pathways to large surface area states in a pre-generated MSM of β-lactamase, which shows a nearly monotonically increasing SASA. I then show that following these gradients can be a general-purpose strategy for exploring conformational space by tackling three difficult challenges in sampling: 1) identifying cryptic

pockets on proteins, 2) observing transitions between known conformations, and 3) folding proteins. For each of these challenges, the FAST simulations acquire a solution with orders of magnitude less aggregate simulation time. Furthermore, there is evidence that the MSMs produce reasonable statistics, since we are able to retrodict the correct folding time of the villain headpiece.

Chapter 4 takes a more in-depth look at how equilibrium-based sampling influences exploration on conformational landscapes. Particularly, if FAST is able to generate MSMs that are reasonable for biological inference. The perspective taken is one that focuses on state discovery, since this is the only factor that differs between sampling methods. To investigate the relationship between state discovery and state-space exploration, a relationship between state discovery and the length, number, and starting state of simulations is derived. From this, we can see that each of these parameters could have a drastic impact on exploration, although general purpose conclusions are difficult to obtain because results can vary depending on the landscape sampled. Assessing sampling on a variety of physically inspired landscapes, it can be seen that the FAST algorithm has the ability to avoid obstacles and provide realistic pathways. However, many parallel simulations could have the pathology that they can traverse unrealistically high energy barriers with an inflated probability compared to long simulations, which is termed pathway tunneling. In the case of a very poorly selected geometric component, FAST simulations have a finite probability of pathway tunneling. To alleviate the chances of tunneling, the FAST-string method is developed, which resamples along the highest-flux pathways and corrects for any tunneling artifacts. It is then shown that use of FAST followed by FAST-string, even with an extremely poor selection of geometric component, will generate correct transition pathways in addition to thermodynamic and kinetic predictions. These results are even more

accurate than those of a single long simulation with equivalent aggregate simulation time. This suggests that FAST simulations are not only easier to generate but also create better MSMs than traditional sampling. This is further evidenced with all-atom MD simulations of the λ-repressor.

From chapters 2-4, FAST simulations are shown to be great at exploring the conformational landscapes defined by a force-field, however, require further evidence that the set of states discovered can be biologically insightful. While very rare-event states can be discovered with many orders of magnitude less aggregate simulation time, a real application of FAST is necessary to demonstrate that it is complementary to experiments. Towards this goal, chapter 5 applies FAST simulations to understand the mechanistic determinates of stabilization in the TEM-1 β-lactamase M182T variant. Previous studies of crystal structures have provided two competing models, N-terminal helix capping or loop stabilization. FAST simulations are more consistent with the helix capping model over loop stabilization (the Thr182 sidechain spends more time in this conformation in MSMs) but suggest a different model altogether. Comparing the MSMs of the two variants, a set of distances down helix-9 are observed to be stabilized in the presence of the stabilizing mutation. The developed computational model is therefore that mutations that stabilize helix-9, and the interdomain interface where it docks, are the key determinants of stabilization. This model is the only proposed model that is able to predict the stabilities of three other point mutations. Of particular interest is the M182N variant that we designed, that does not provide significant stabilization despite crystallographic evidence of helix-capping. In a triumph of computational predictions, the FAST simulations are able to provide a reasonable justification for this: the asparagine sidechain is caught between either stabilizing helix-9 *or* the domain interface, not both. In all, the significant agreement between the

experiments and computational predictions highlight the power of using FAST simulations as a biophysical tool to assess conformational landscapes.

Lastly, in chapter 6, tools for efficiently building and analyzing MSMs are presented. The software enspara is developed to efficiently handle large amounts of aggregate simulation data. Enspara is a fast and flexible framework for building and analyzing trajectory data, with some key developments being the implementation of a "ragged array", sparse-matrix implementations of key MSM algorithms, and an interoperable framework for MSM construction. These tools have been pivotal in the development of FAST and will undoubtedly be necessary as simulation sets get larger as we push the limits of MD.

Combined, the data presented in this thesis is expected to significantly aid in the ability to characterize a proteins' conformational ensemble. Conformational space is extraordinarily vast (an understatement) and brute force sampling is doomed to fail for larger systems. The only way that many relevant conformational ensembles can be documented is with the aid of goal-oriented sampling. Whether FAST be the end all, or a stepping stone for future development, we are approaching the age where knowledge of a proteins' conformational ensemble is becoming a reality.

7.2 Future Directions

Leonardo Da Vinci once said, "art is never finished, only abandoned", and the same can be said of methods development. The FAST algorithm has proven incredibly useful, however, there is still much room for improvement. Additionally, the algorithm has opened the doors for many future applications. In this final section, the possibilities of a few new directions and applications will be explored.

It is surprising that the incredibly simple counts-based adaptive sampling has worked so well at discovering new states. This strategy has even outperformed more sophisticated methods.[1] Despite this positive performance when discovering new states, it can lead to pathologies as the statistical component in the FAST ranking.

Oftentimes, FAST simulations travel down a deep energy well and hit a dead end with no prospect of discovering states that further optimize a particular order parameter. Most of the time this is not an issue; as the dead-end states are sampled more, their counts go up, their rankings go down, and FAST chooses states that will circumvent the trap. Unfortunately, if the dead-end pathway has a large entropic component, new states can be discovered very quickly within this energy minima. These states will be subtly different from one another, though sufficient enough to classify as geometrically distinct. Since these states will be new, they will have a small number of counts and will be mistakenly favored in the FAST ranking. An example of this might arise during the task of folding a protein: if half of the protein misfolds, though partially optimizes the trait-based objective, and has a disordered tail with significant conformational heterogeneity, geometric clustering will produce many new states from this trap with low counts and hinder backtracking. One might argue for using a kinetic clustering to lump all of these states together, although kinetic clustering obscures a significant portion of conformational heterogeneity and will perform poorly with large entropic barriers. A better solution would be to keep the benefits of geometric clustering, however, devise a statistical ranking that considers each states' position in the context of the known conformational landscape—i.e. its neighbors sampling quality, their neighbors sampling quality, *etc*.—instead of considering the state in isolation. Such a holistic view of conformational space would better identify sampling quality in

particular regions and recognize when an extraordinary amount of sampling is being spent in a dead-end.

One such scheme to better identify sampling quality comes from the algorithm initially used to power Google's search engine. The PageRank algorithm ranks a website based on its connectivity within the world-wide web. Specifically, the ranking of website i is calculated as a sum of the rankings of each website, j, that links to i. Each of the rankings in this sum are normalized by the number of connections in j; websites give a share of their ranking to connected pages. This type of ranking has a strong analog to Markov state models and could be used as the statistical component within the FAST ranking to improve sampling. With this framework, states would be ranked based on their number of observations *in addition* to the number of observations in their connected neighbors. That means that if a large number of low count states were generated in a dead end of conformational space, the ranking would identify this as a densely connected region and conclude that it is well sampled.

Another region that could benefit from improvement comes from the trait-based portion to the FAST ranking. The goal of FAST is to automate the process of exploring conformational space, although choosing an order parameter that is amenable to efficient exploration is often very challenging. Selection is still an art, rather than a science, and may require insider knowledge of the particular system being studied. Automated selection of trait-based components to the FAST ranking would greatly simplify its use and expand adoption by non-experts. Much inspiration can come from the field of machine learning. There have been a number of recent attempts to use machine learning on proteins for dimensionality reduction,[2-5] or identification of simple order parameters.[6-9] While the specific path is unclear, it seems possible to retrain a model, within each round of FAST, to identify the most productive order parameter.

This could adaptively identify energetic barriers and reframe the problem to be tackled with reinforcement learning.[10]

Finally, with FAST as a tool to quickly explore conformational landscapes, the natural question arises: what can we do with an increased understanding of conformational space? The most logical next steps are to simulate proteins with mutational differences that influence disease severity, as we are pursuing with clinical mutations found on a protein related to Alzheimer's disease, Apolipoprotein E.[11] We can also use the exploration of conformational space to identify cryptic pockets to develop drugs for otherwise undruggable proteins, such as the Ebola virus protein, VP35.[12,13] Additionally, mechanistic pathways between known conformational states can be incredibly valuable in understanding how biological systems operate, and as such, an ambitious project would be to use FAST to obtain a complete biological pathway, such as the myosin cycle.[14] It will be exciting to see what the future holds for FAST and its abilities to tackle larger systems.

Bibliography

(1) Weber, J. K.; Pande, V. S. Characterization and Rapid Sampling of Protein Folding Markov State Model Topologies. *J. Chem. Theory Comput.* **2011**, *7* (10), 3405–3411.

(2) Liu, Y.; Amzel, L. M. Conformation Clustering of Long MD Protein Dynamics with an Adversarial Autoencoder. May 31, 2018.

(3) Lemke, T.; Peter, C. EncoderMap: Dimensionality Reduction and Generation of Molecule Conformations. *J. Chem. Theory Comput.* **2019**, *15* (2), 1209–1215.

(4) Wehmeyer, C.; Noé, F. Time-Lagged Autoencoders: Deep Learning of Slow Collective Variables for Molecular Kinetics. *The Journal of Chemical Physics* **2018**, *148* (24), 241703.

(5) Mardt, A.; Pasquali, L.; Wu, H.; Noé, F. VAMPnets for Deep Learning of Molecular Kinetics. *Nature Communications* **2018**, *9* (1), 5.

(6) McGibbon, R. T.; Husic, B. E.; Pande, V. S. Identification of Simple Reaction Coordinates From Complex Dynamics. *The Journal of Chemical Physics* **2017**, *146* (4), 044109.

(7) Sultan, M. M.; Wayment-Steele, H. K.; Pande, V. S. Transferable Neural Networks for Enhanced Sampling of Protein Dynamics. *J. Chem. Theory Comput.* **2018**, *14* (4), 1887–1894.

(8) Sittel, F.; Stock, G. Perspective: Identification of Collective Variables and Metastable States of Protein Dynamics. *The Journal of Chemical Physics* **2018**, *149* (15), 150901.

(9) Sultan, M. M.; Pande, V. S. Automated Design of Collective Variables Using Supervised Machine Learning. *The Journal of Chemical Physics* **2018**, *149* (9), 094106.

(10) Shamsi, Z.; Cheng, K. J.; Shukla, D. Reinforcement Learning Based Adaptive Sampling: REAPing Rewards by Exploring Protein Conformational Landscapes. *J. Phys. Chem. B* **2018**, *122* (35), 8386–8395.

(11) Corder, E. H.; Saunders, A. M.; Strittmatter, W. J.; Schmechel, D. E.; Gaskell, P. C.; Small, G. W.; Roses, A. D.; Haines, J. L.; Pericak-Vance, M. A. Gene Dose of Apolipoprotein E Type 4 Allele and the Risk of Alzheimer's Disease in Late Onset Families. *Science* **1993**, *261* (5123), 921–923.

(12) Basler, C. F.; Wang, X.; Mühlberger, E.; Volchkov, V.; Paragas, J.; Klenk, H.-D.; García-Sastre, A.; Palese, P. The Ebola Virus VP35 Protein Functions as a Type I IFN Antagonist. *Proc. Natl. Acad. Sci. U.S.A.* **2000**, *97* (22), 12289–12294.

(13) Basler, C. F.; Mikulasova, A.; Martinez-Sobrido, L.; Paragas, J.; Mühlberger, E.; Bray, M.; Klenk, H.-D.; Palese, P.; García-Sastre, A. The Ebola Virus VP35 Protein Inhibits Activation of Interferon Regulatory Factor 3. *Journal of Virology* **2003**, *77* (14), 7945–7956.

(14) Preller, M.; Manstein, D. J. Myosin Structure, Allostery, and Mechano-Chemistry. *Structure* **2013**, *21* (11), 1911–1922.

Appendices

A.1 Appendix to Chapter 4

A.1.1 Calculation of Discover Probabilities

Given the landscape depicted in Figure 4.1B, with the transition probability matrix,

$$T_{ij} = \begin{bmatrix} 0.65 & 0.3 & 0.05 \\ 0.25 & 0.5 & 0.25 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$

we can calculate the probability of discovering state n_j from simulations starting from state n_i given 3 simulations of length 2, $\mathrm{P}\left(D_{ij}^{\mathbf{K}=\{\mathbf{2},\mathbf{2},\mathbf{2}\},\mathbf{M}=\mathbf{3}} = 1\right)$, by first calculating the probability that a single simulation of length 2 discovers state n_j, $\mathrm{P}\left(D_{ij}^{\mathbf{K}=\{2\},M=1} = 1\right)$. To do this, we use equation 4 to determine the probability of being in any of the three states at each timestep, conditional on not having discovered state n_j yet. Before simulations ($k = 0$), the probability of discovering state n_j is 1 if starting from state n_j, and 0 otherwise, which is simply the identity matrix,

$$\mathrm{P}\left(v_{ij}^{k=0} = 1\right) = I$$

To determine the probability of being in any state after the first timestep, we propagate the probabilities with the transition probability matrix,

$$\mathrm{P}\left(v_{ij}^{k=1}=1\right)=\mathrm{P}\left(v_{ij}^{k=0}=1\right)\mathrm{T}=T_{ij}$$

For the second step, we propagate the probabilities conditional to not having discovered state n_j yet

$$\mathrm{P}\left(v_{i'j'}^{k=2}=1\mid\left\{v_{ij}^{k'}=0\;\forall\;k'<2\right\}\right)=\mathrm{P}\left(v_{i'j'}^{k=1}=1\mid v_{ij}^{k=1}=0\right)\mathrm{T}$$

where, for $j=2$, we have,

$$\mathrm{P}\left(v_{i'j'}^{k=1}=1\mid v_{i2}^{k=1}=0\right)=\begin{bmatrix}0.68 & 0.32 & 0\\ 0.33 & 0.67 & 0\\ 0.5 & 0.5 & 0\end{bmatrix}$$

which are the renormalized rows of $\mathrm{P}\left(v_{ij}^{k=1}=1\right)$ after setting column 2 to 0. Propagating this by the transition probability matrix, we obtain the probability of being in state 2 given that it was not discovered previously,

$$\mathrm{P}\left(v_{i'j'}^{k=2}=1\mid\left\{v_{i2}^{k'}=0\;\forall\;k'<2\right\}\right)=\begin{bmatrix}0.524 & 0.363 & 0.113\\ 0.383 & 0.433 & 0.183\\ 0.450 & 0.400 & 0.150\end{bmatrix}$$

Combining these probabilities of being in state 2 at various time-steps, we calculate the probability of discovering state 2, $\mathrm{P}\left(D_{i2}^{\mathbf{K}=\{2\},M=1}=1\right)$, as,

$$\mathrm{P}\left(D_{i2}^{\mathbf{K}=\{2\},M=1}=1\right)=1-\left(1-\mathrm{P}\left(v_{ij}^{k=0}=1\right)\right)_{i2}*\left(1-\ \mathrm{P}\left(v_{ij}^{k=1}=1\right)\right)_{i2}*\Big(1-$$

$$\mathrm{P}\Big(v_{i'j'}^{k=2}=1\mid\left\{v_{i2}^{k'}=0\ \forall\ k'<2\right\}\Big)\Big)_{i2}=1-\begin{bmatrix}1-0\\1-0\\1-1\end{bmatrix}*\begin{bmatrix}1-0.05\\1-0.25\\1-0.5\end{bmatrix}*\begin{bmatrix}1-0.113\\1-0.183\\1-0.150\end{bmatrix}=\begin{bmatrix}0.16\\0.39\\1.0\end{bmatrix}$$

Calculating the columns for $j=\{0,1\}$, we get the full discover probabilities between any n_i and n_j as,

$$\mathrm{P}\left(D_{ij}^{\mathbf{K}=\{2\},M=1}=1\right)=\begin{bmatrix}1.0&0.51&0.16\\0.44&1.0&0.39\\0.44&0.45&1.0\end{bmatrix}$$

Next, this is used to calculate the discover probabilities of 3 independent simulations of length 2 using the following,

$$\mathrm{P}\left(D_{ij}^{\mathbf{K},\mathbf{M}}=1\right)=1-\left[1-\mathrm{P}\left(D_{ij}^{\mathbf{K}=\{2\},M=1}=1\right)\right]^3=1-\left[1-\begin{bmatrix}1.0&0.51&0.16\\0.44&1.0&0.39\\0.44&0.45&1.0\end{bmatrix}\right]^3$$

$$=\begin{bmatrix}1.0&0.88&0.41\\0.82&1.0&0.77\\0.82&0.83&1.0\end{bmatrix}$$

A.1.2 Supporting Figures

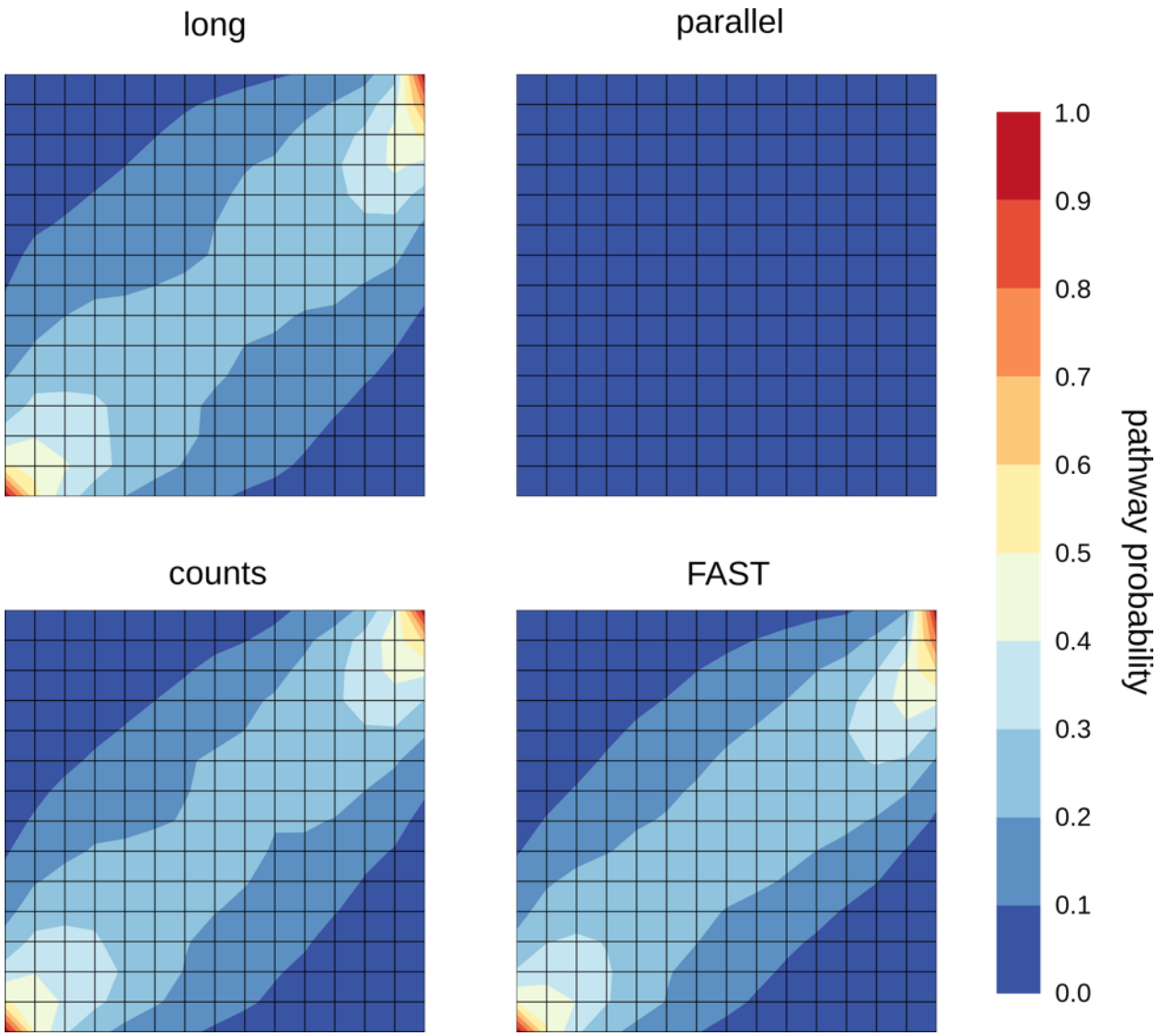

Figure *A.1.1*: The pathway probabilities (the probability that a state is predicted to be in the highest-flux pathway from the start to the target) for the funneled landscape in Figure 3. Shown are the probabilities for four sampling strategies, a single long simulation, many parallel simulations, counts-based adaptive sampling, and the goal-oriented FAST simulations. The parallel simulations did not observe a transition, and thus, do not have a pathway.

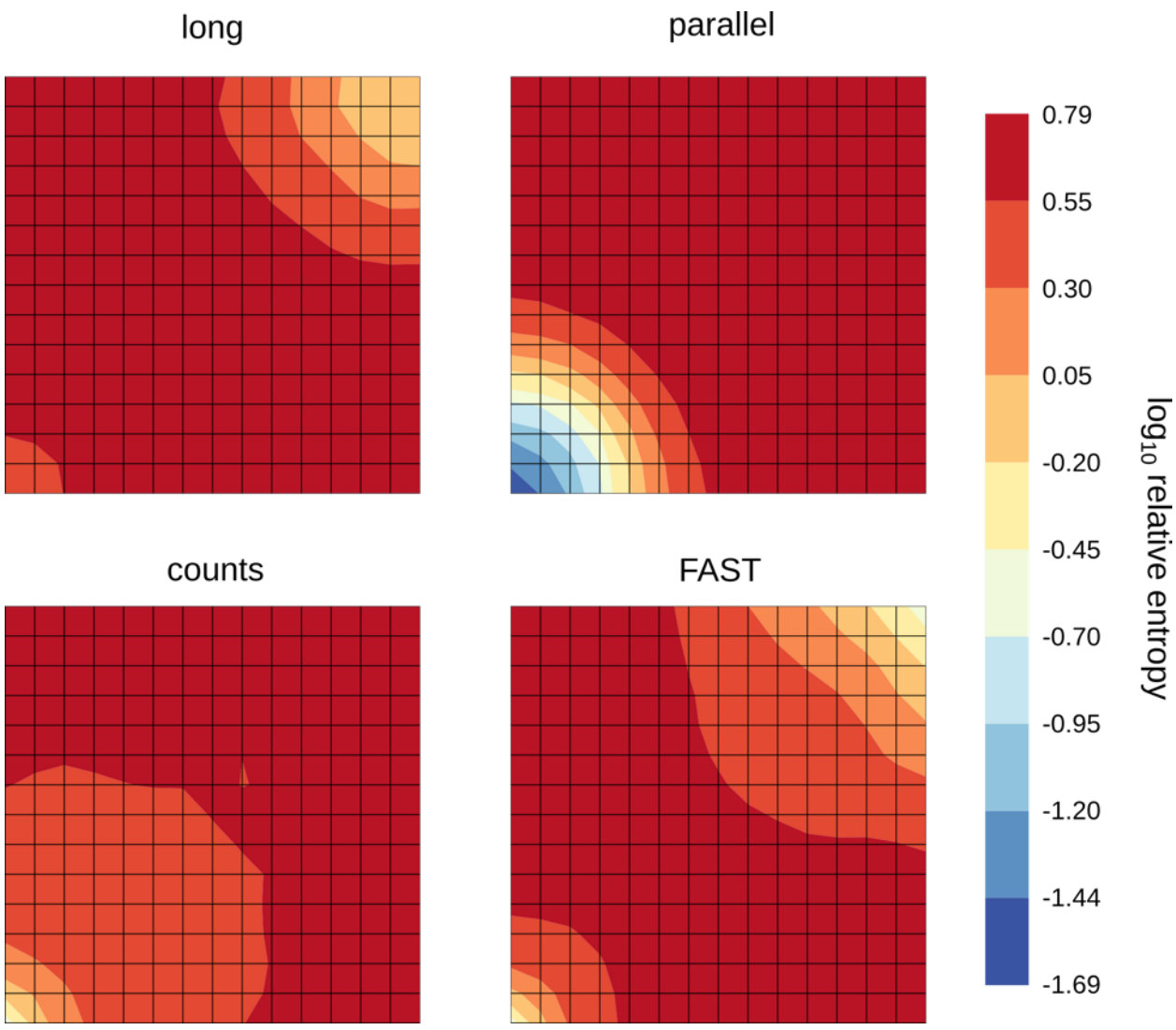

Figure *A.1.2*: The average Kullbeck-Liebler divergence of each states conditional transition probabilities to the true transition probabilities for the funneled landscape in Figure 3. Shown are the average divergences of each state for four sampling strategies, a single long simulation, many parallel simulations, counts-based adaptive sampling, and the goal-oriented FAST simulations.

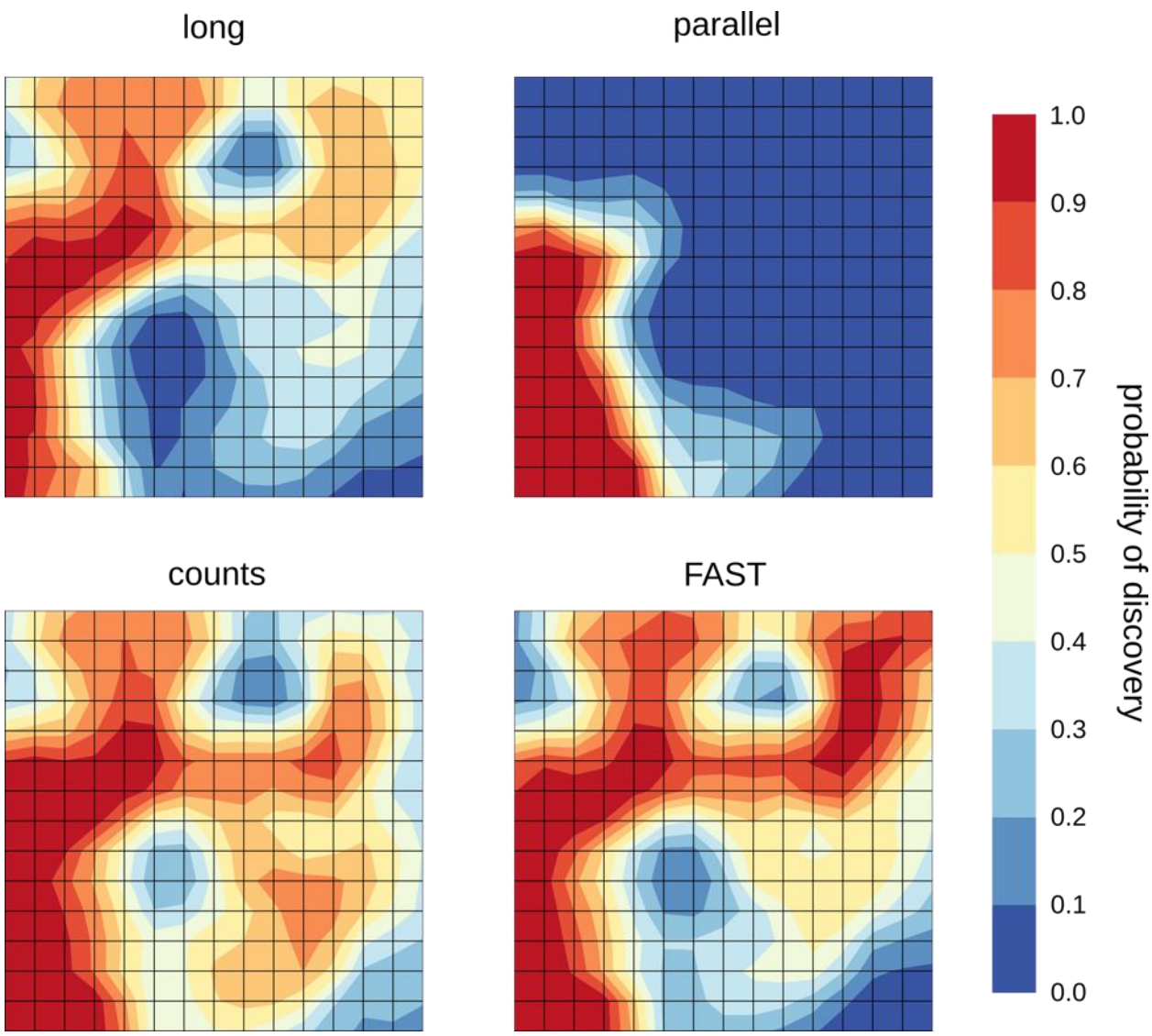

Figure A.1.3: The discover probabilities (the probability that a simulation set observes a particular state) on the random barriered landscape in Figure 5A. Shown are the probabilities for four sampling strategies, a single long simulation, many parallel simulations, counts-based adaptive sampling, and the goal-oriented FAST simulations.

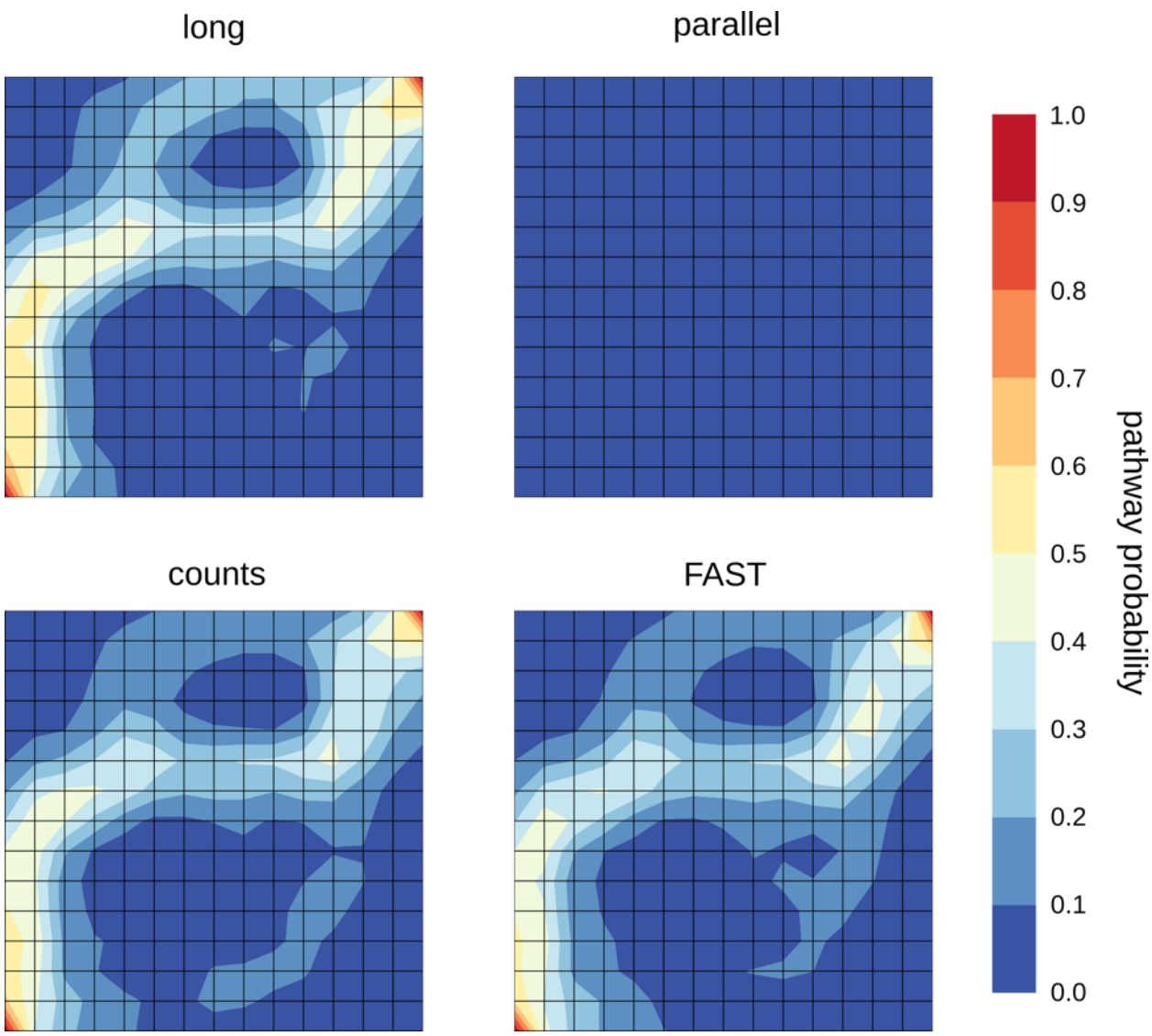

Figure A.1.4: The pathway probabilities (the probability that a state is predicted to be in the highest-flux pathway from the start to the target) for the random barriered landscape in Figure 5A. Shown are the probabilities for four sampling strategies, a single long simulation, many parallel simulations, counts-based adaptive sampling, and the goal-oriented FAST simulations. The parallel simulations did not observe a transition, and thus, do not have a pathway.

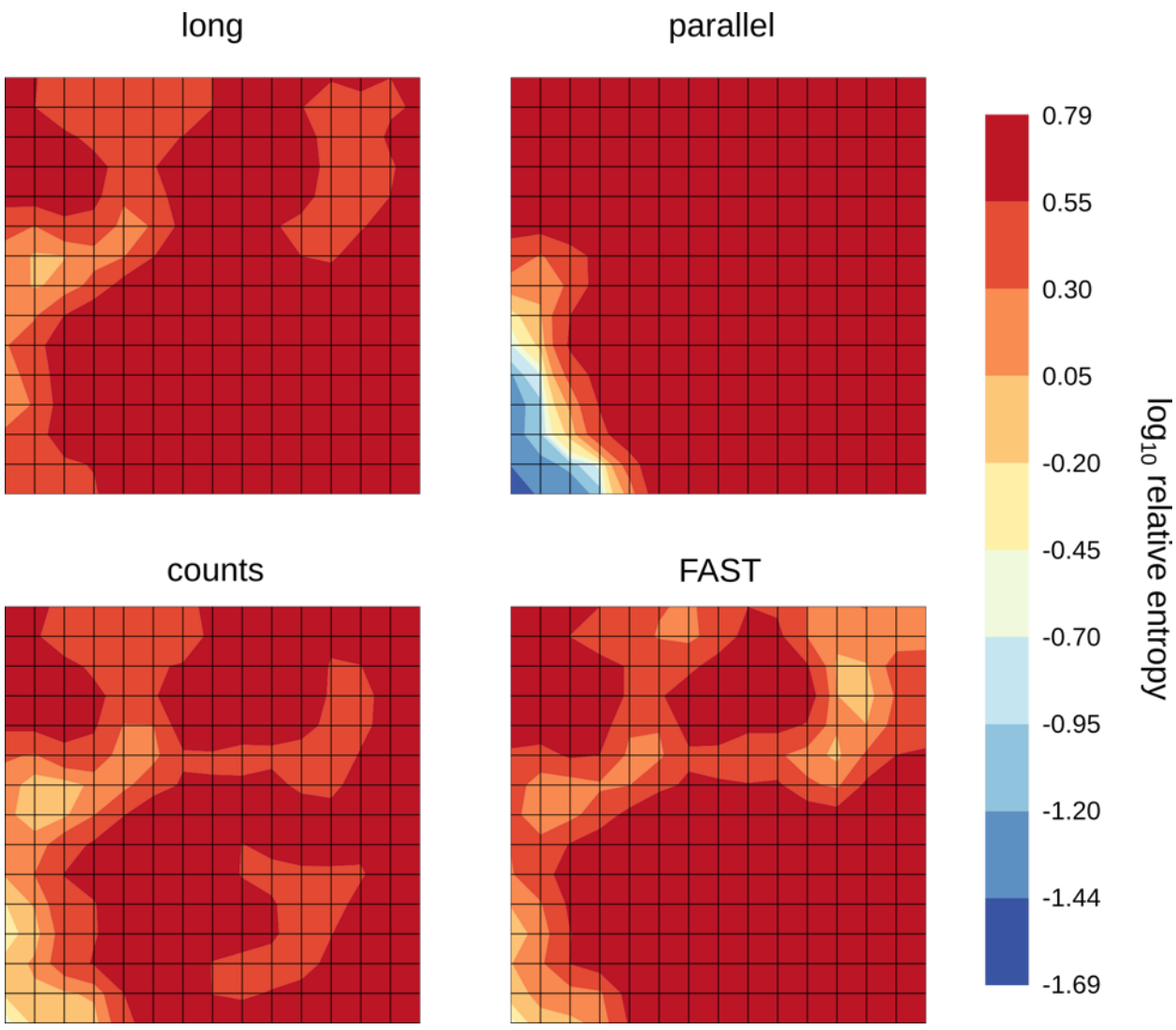

Figure A.1.5: The average Kullbeck-Liebler divergence of each states conditional transition probabilities to the true transition probabilities for the random barriered landscape in Figure 5A. Shown are the average divergences of each state for four sampling strategies, a single long simulation, many parallel simulations, counts-based adaptive sampling, and the goal-oriented FAST simulations.

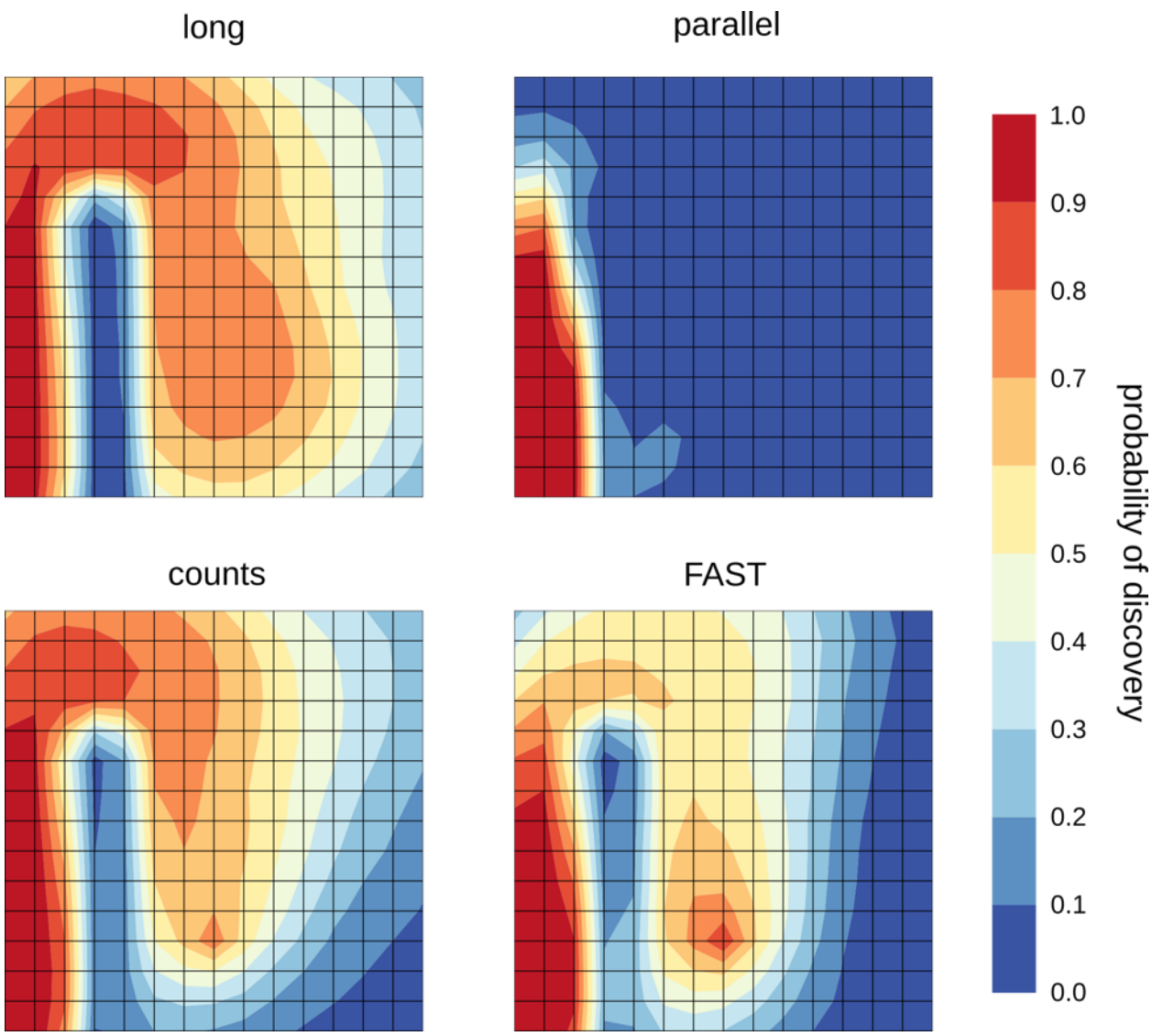

Figure A.1.6: The discover probabilities (the probability that a simulation set observes a particular state) on the large barriered landscape in Figure 6. Shown are the probabilities for four sampling strategies, a single long simulation, many parallel simulations, counts-based adaptive sampling, and the goal-oriented FAST simulations.

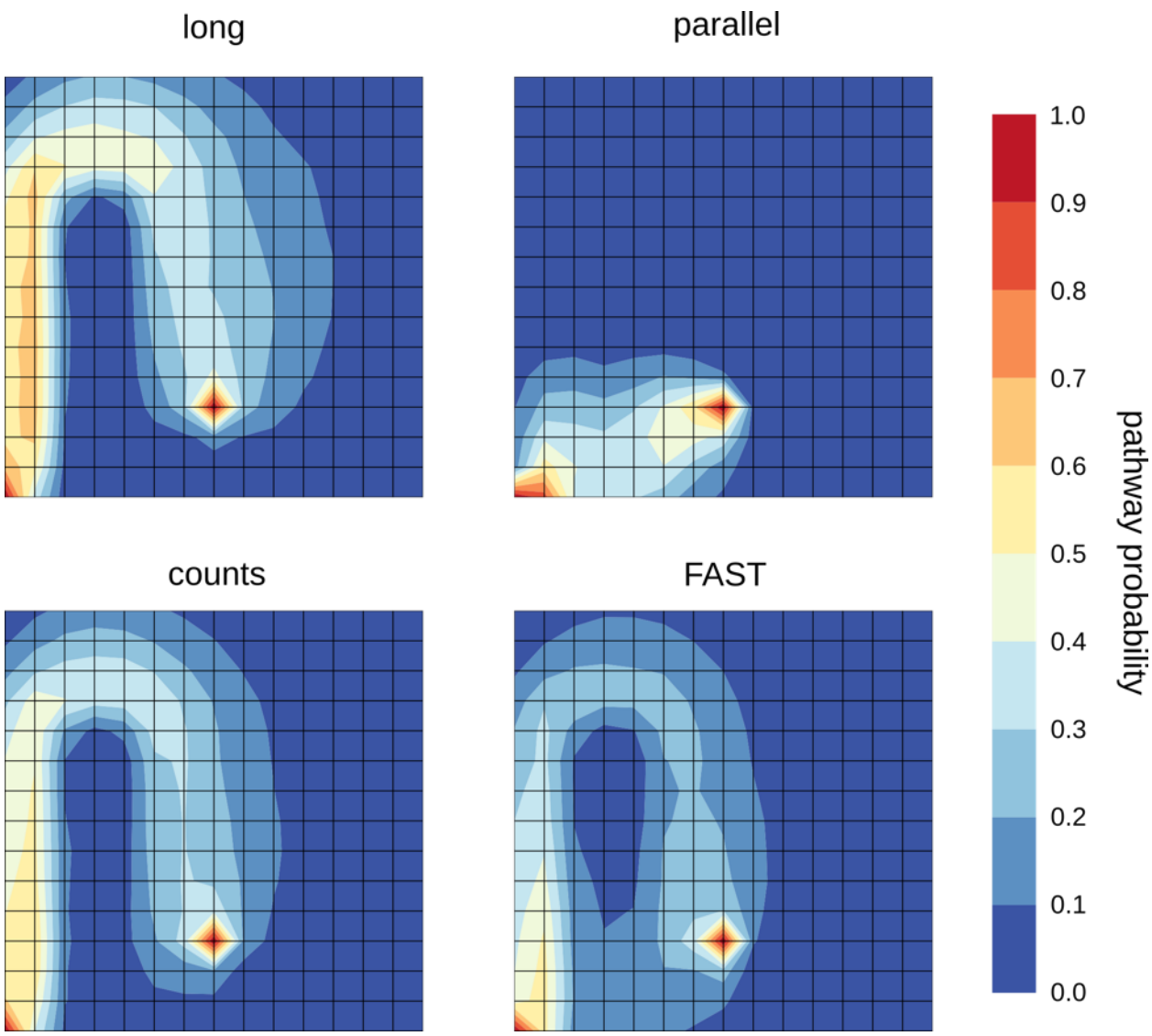

Figure A.1.7: The pathway probabilities (the probability that a state is predicted to be in the highest-flux pathway from the start to the target) for the large barriered landscape in Figure 6. Shown are the probabilities for four sampling strategies, a single long simulation, many parallel simulations, counts-based adaptive sampling, and the goal-oriented FAST simulations.

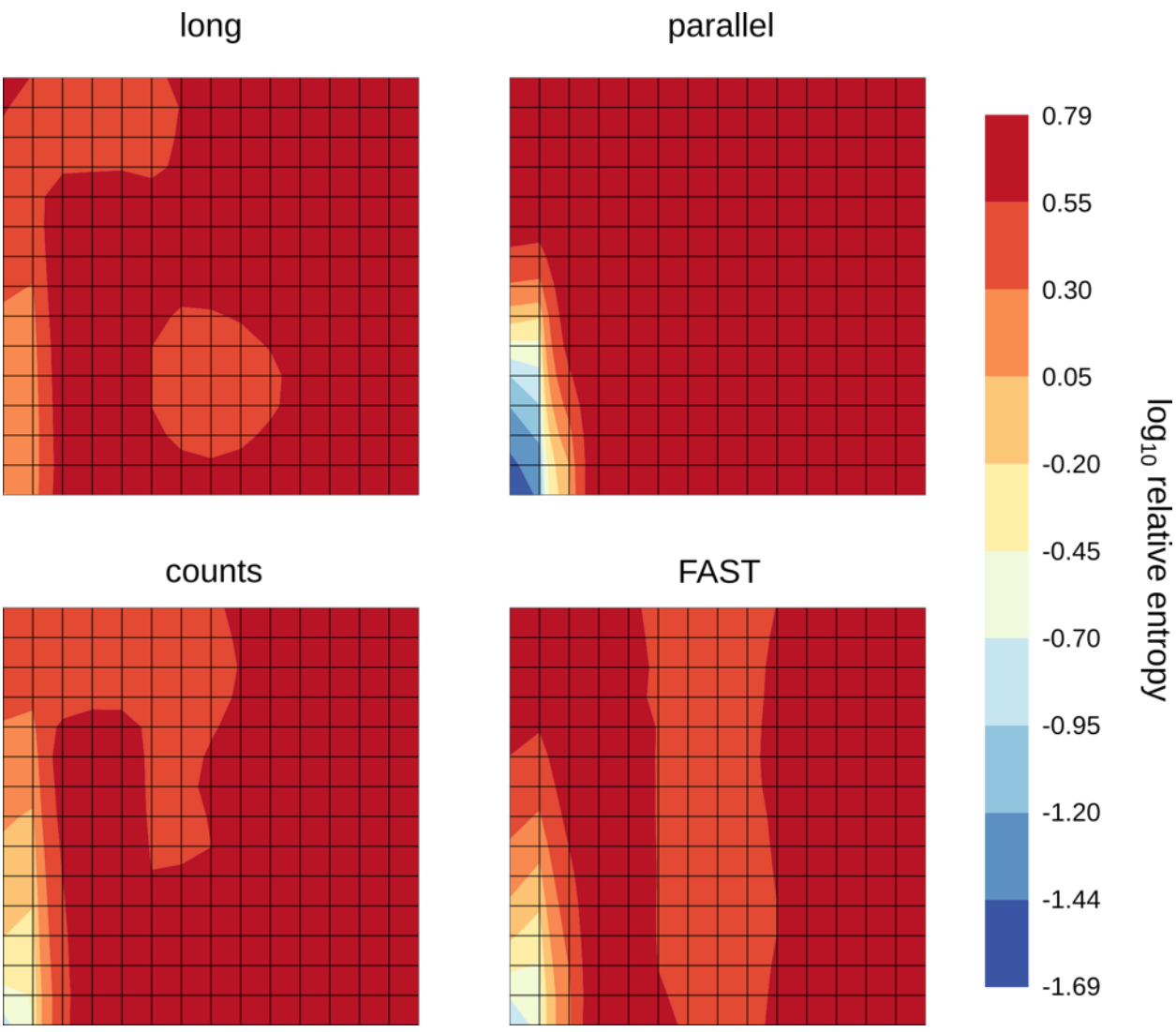

Figure A.1.8: The average Kullbeck-Liebler divergence of each states conditional transition probabilities to the true transition probabilities for the large barriered landscape in Figure 6. Shown are the average divergences of each state for four sampling strategies, a single long simulation, many parallel simulations, counts-based adaptive sampling, and the goal-oriented FAST simulations.

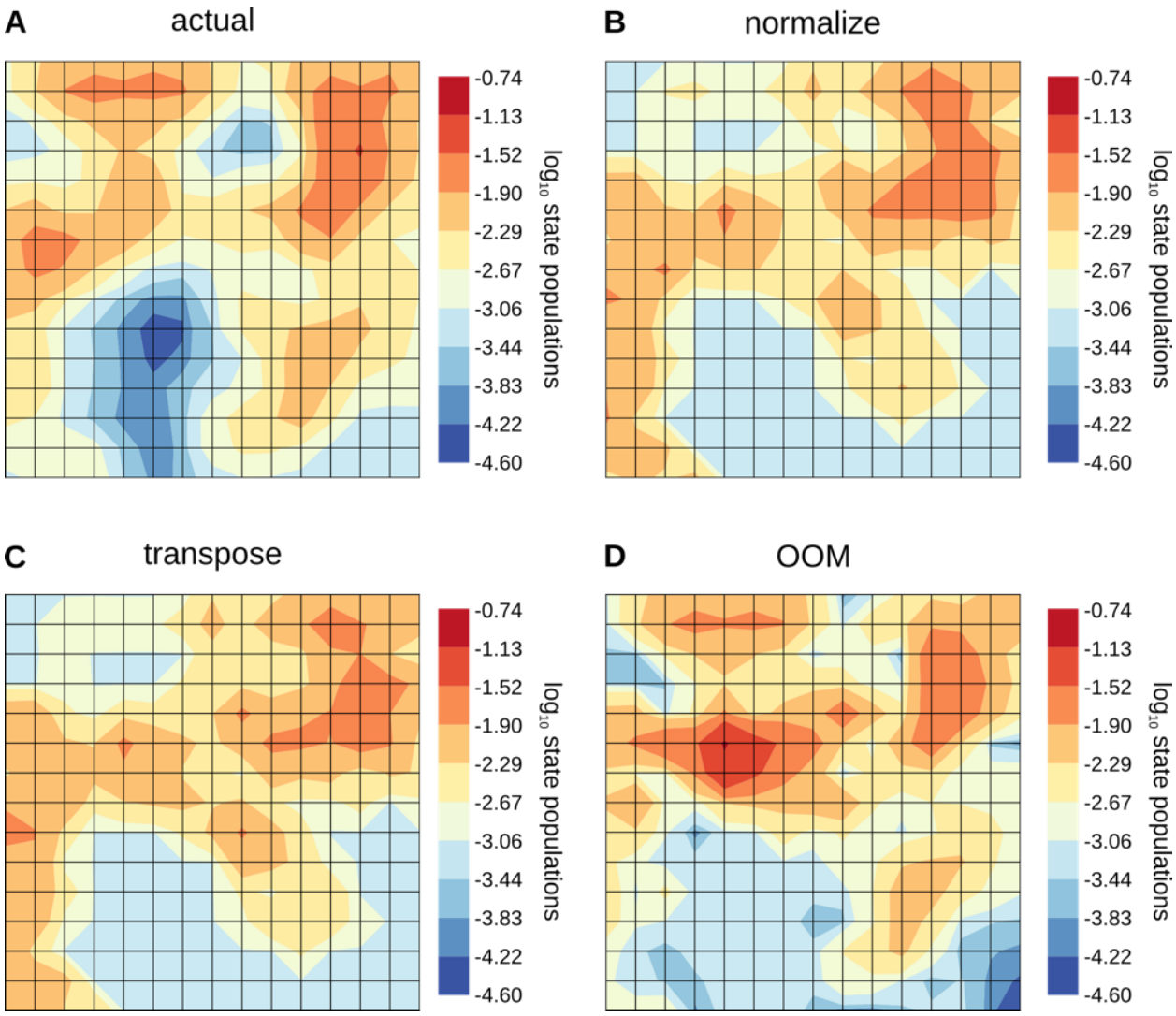

Figure A.1.9: Comparison of MSM estimators' prediction of state populations for a single FAST simulation set. The data set used is the same as is shown in Figure 10A. Shown are (A) the true populations of each state at equilibrium, (B) the predictions from the normalize method, (C) the predictions from the transpose method, and (D) the predictions from the OOM method.

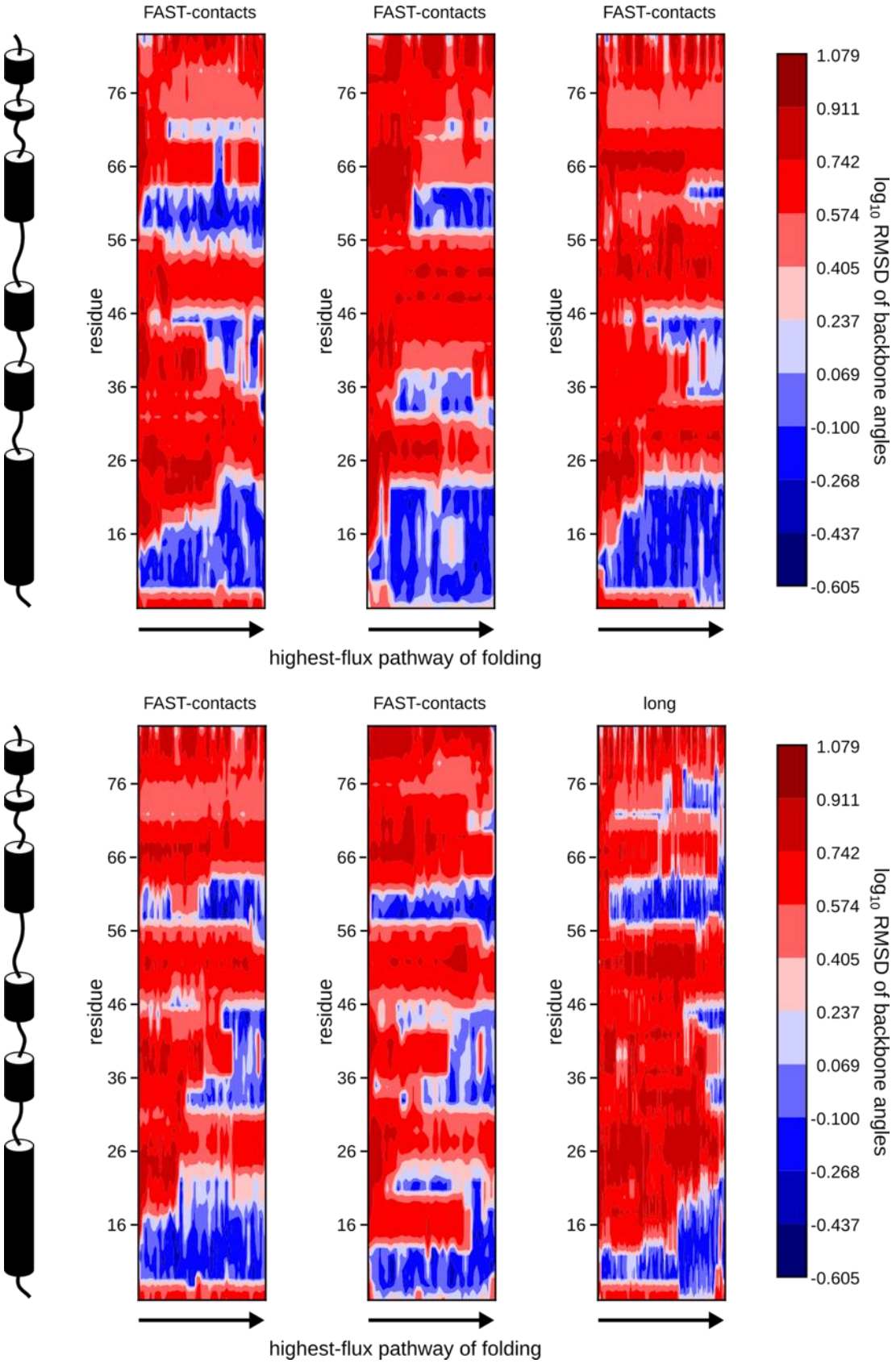
FAST-contacts
FAST-contacts
FAST-contacts
residue
76
66
56
46
36
26
16
highest-flux pathway of folding
1.079
0.911
0.742
0.574
0.405
0.237
0.069
-0.100
-0.268
-0.437
-0.605
log10 RMSD of backbone angles
FAST-contacts
FAST-contacts
long
residue
highest-flux pathway of folding
log10 RMSD of backbone angles

Figure A.1.10: Analysis of λ-repressor predicted folding pathways using the RMSD of each residues' backbone ϕ and ψ angles to the crystal structure (PDBID: 1LMB). Folding pathways are determined as an MSMs highest-flux path from the starting state to the state with the largest fraction of native contacts. From left to right on each plot are the residue backbone RMSDs for each state in the predicted folding pathway from five separate runs of FAST-contacts and a single set of long simulations.

Table A.1.1: Probabilities of discovering the target state, average number of states discovered, and relative entropies of transition probabilities for long, parallel, counts, and FAST simulations. Results are shown for 3 landscapes in the main text: 1) funneled landscape depicted in Figure 3, 2) the random barriered landscape depicted in Figure 5A, and 3) the large barrier depicted in Figure 6. Standard deviations of the discover probabilities come from bootstrapping the kinetic Monte Carlo simulations. The discover probabilities for parallel simulations on the funneled and random barriered landscape come from Equation 6 and do not have a calculated standard deviation, since none of these simulations observed a transition to the target state. The optimal value for a given parameter and landscape is bolded.

Landscape/method	Probability of discovering the target state	Number of states discovered	Relative entropy	Relative entropy of highest-flux paths
Funneled				
Long	0.94 ± 3.2E-3	144.2 ± 24.0	2.53 ± 1.82	0.84 ± 0.80
Parallel	2.2E-5	72.7 ± 10.1	5.38 ± 0.19	2.46 ± 0.05
Counts	0.62 ± 6.9E-3	**183.3 ± 12.3**	4.18 ± 1.74	1.96 ± 0.76
FAST	**1.0 ± 7.4E-4**	168.5 ± 12.3	**2.02 ± 1.19**	**0.58 ± 0.46**
Random barriers				
Long	0.50 ± 7.0E-3	108.9 ± 24.7	3.15 ± 2.17	1.46 ± 1.29
Parallel	8.5E-7	50.2 ± 8.6	5.03 ± 0.36	2.64 ± 0.22
Counts	0.34 ± 6.6E-3	141.7 ± 18.6	3.60 ± 1.69	1.89 ± 0.92
FAST	**0.91 ± 4.0E-3**	**143.5 ± 15.7**	**2.61 ± 1.62**	**0.89 ± 0.89**
Large barrier				
Long	0.74 ± 5.9E-3	129.7 ± 34.1	3.69 ± 2.04	0.67 ± 0.36
Parallel	0.059 ± 3.3E-3	33.0 ± 7.0	5.30 ± 0.20	0.83 ± 0.03
Counts	0.78 ± 5.6E-3	146.1 ± 18.7	3.97 ± 1.64	0.63 ± 0.26
FAST	**0.90 ± 4.3E-3**	124.8 ± 24.2	3.60 ± 1.69	0.63 ± 0.29
FAST + string	**0.90 ± 4.3E-3**	**160.6 ± 25.1**	**2.67 ± 1.83**	**0.46 ± 0.32**

A.2 Appendix to Chapter 5

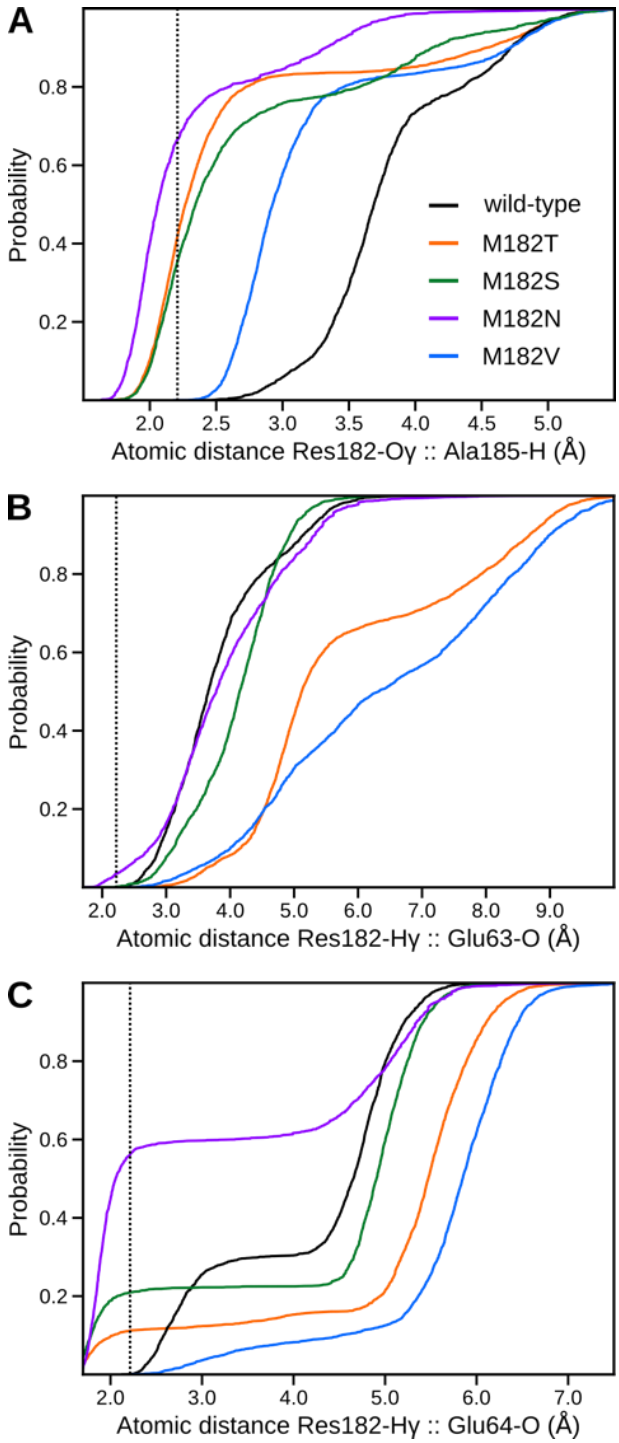

Figure A.2.1: Analysis of N-terminal capping probabilities for each TEM variant. (A-C) Cumulative distributions functions of three distances: Res182-Oγ (or equivalent) to Ala185-H, Res182-Hγ (or equivalent) to Glu63-O, and Res182-Hγ (or equivalent) to Glu64-O, for five TEM variants: wild-type (black), M182T (red), M182S (green), M182N (purple), and M182V (blue). This indicates the probability of observing an atomic distance less than the

specified value. The dotted line indicates the distance of transition from moderate to weak hydrogen bond strength (2.2 Å).

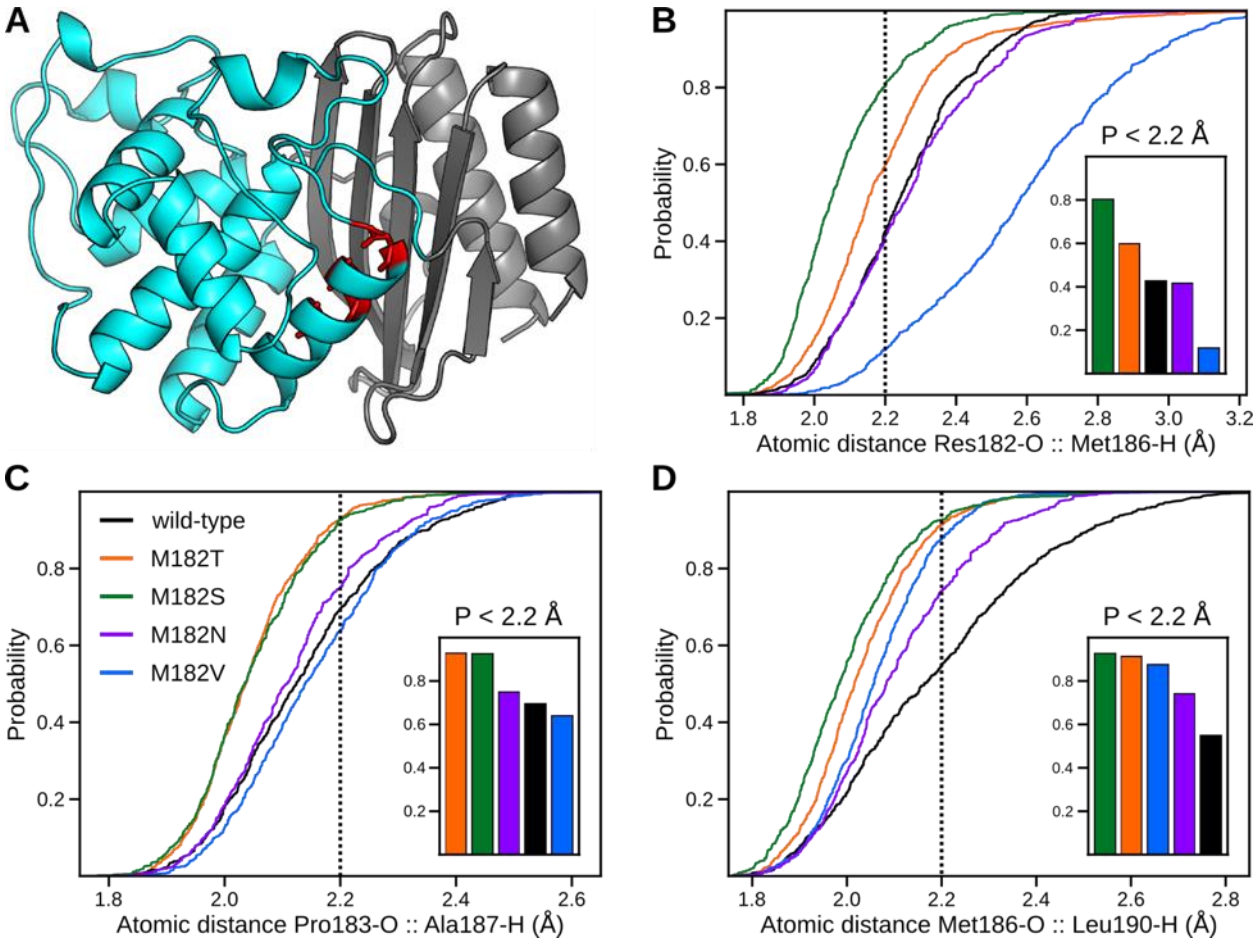

Figure A.2.2*:* Effect of M182T on the stability of helix 9, as judged by the distributions of distances between hydrogen-bonding partners. (A) Structure highlighting hydrogen-bonding partners Residue 182 and Met186, Pro183 and Ala187, and Met186 and Leu190, which are colored red. (B-D) Cumulative distribution functions of the hydrogen bonding partners listed above for wild-type (black) and M182T (orange), M182S (green), M182N (purple), and M182V (blue). These plots indicate the probability of observing an atomic distance less than the specified value. Our cutoff distance for moderate hydrogen bonds, 2.2 Å, is shown as a dotted line. Probabilities of moderate hydrogen bonds for each pair are shown in the inset.

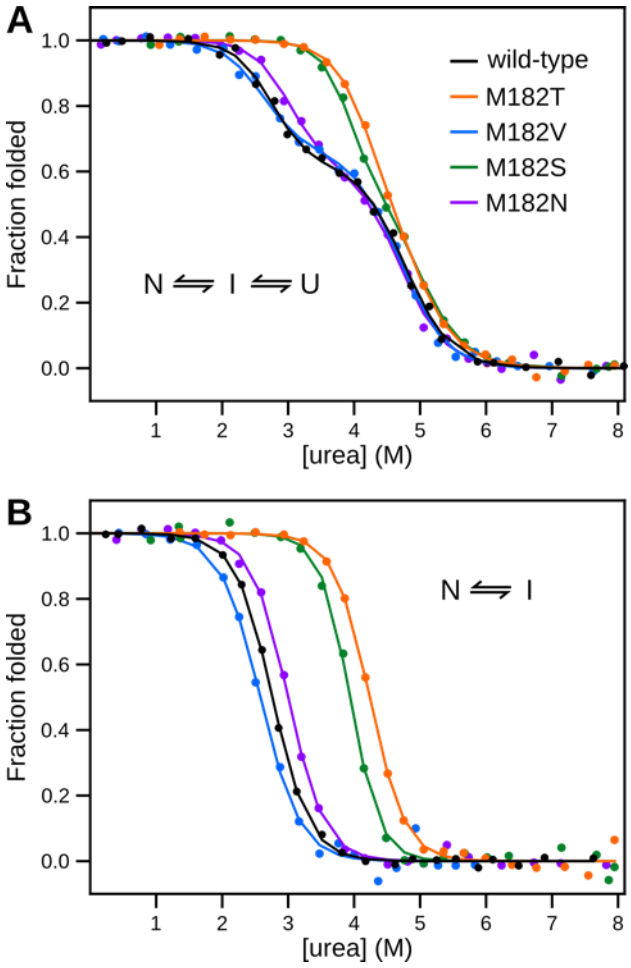

Figure A.2.3: Chemical melts of TEM variants. Shown are the fractions of folded protein for wild-type TEM (black) and TEM M182T (red), M182V (blue), M182S (green), and M182N (purple) as a function of urea. (A) Monitoring signal from circular dichroism. (B) Monitoring signal from fluorescence.

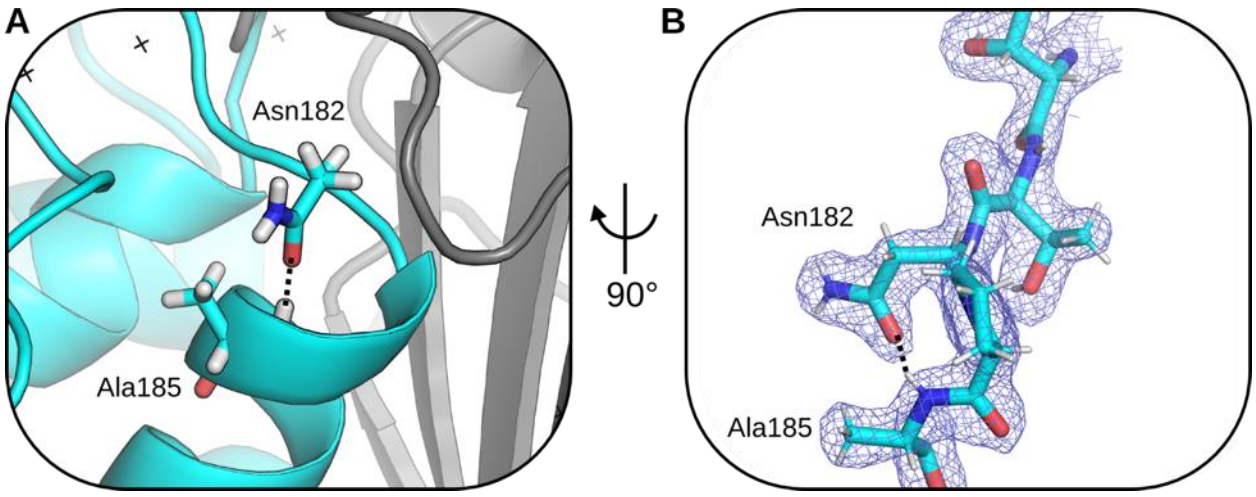

Figure A.2.4: The best fit rotamer of Asn182 from the crystal structure of M182N. Shown is a representative TEM structure from the crystal lattice, solved to 2.0 Å. Asn182 is observed to form a hydrogen bond with Ala185.

Additionally, the sidechain amine has no hydrogen bonding partner and points outward to solvent. (A) Asn182 and Ala185 are represented as sticks, with the backbone of the α-helix domain (cyan) and β-sheet domain (gray) represented as a cartoon. (B) Electron density around Asn182.

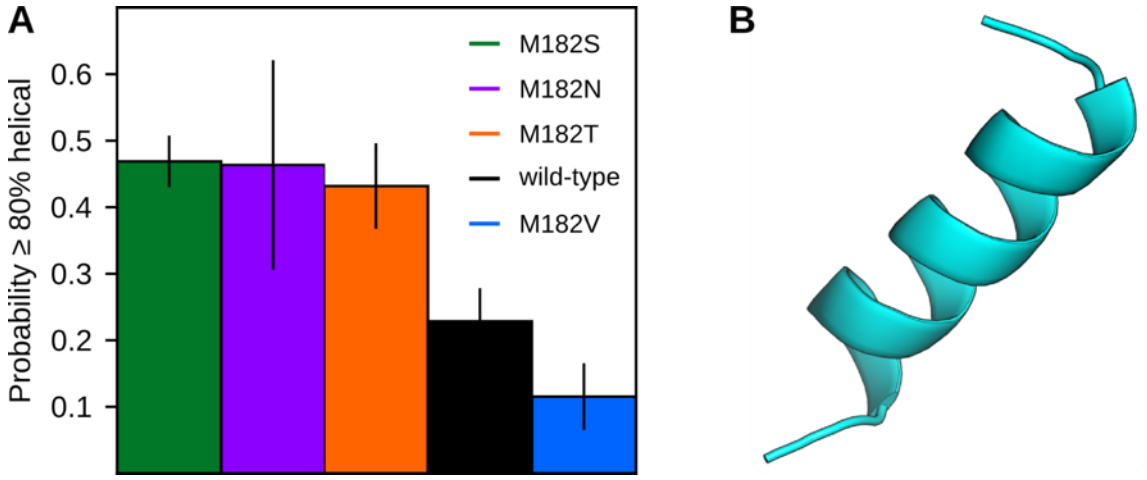

Figure A.2.5: Investigation of helix 9 stability in isolation between TEM variants. (A) Probability of each variant's helix 9 having greater than or equal to 80% of its native helicity. Probabilities come from the MSMs of the isolated helix 9 for wild-type (black), M182T (orange), M182S (green), M182V (blue), and M182N (purple). (B) Helix 9 in isolation (residues 181-197), and the starting structure for simulations.

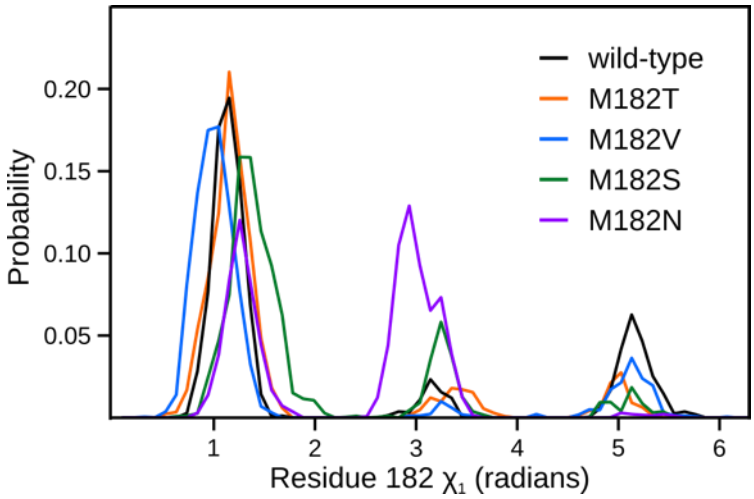

Figure A.2.6: Residue 182 χ1 probabilities. Shown are the χ1 probabilities of each TEM sequence: wild-type (black), M182T (red), M182V (blue), M182S (green), and M182N (purple). These probabilities come from MSMs of the full protein.

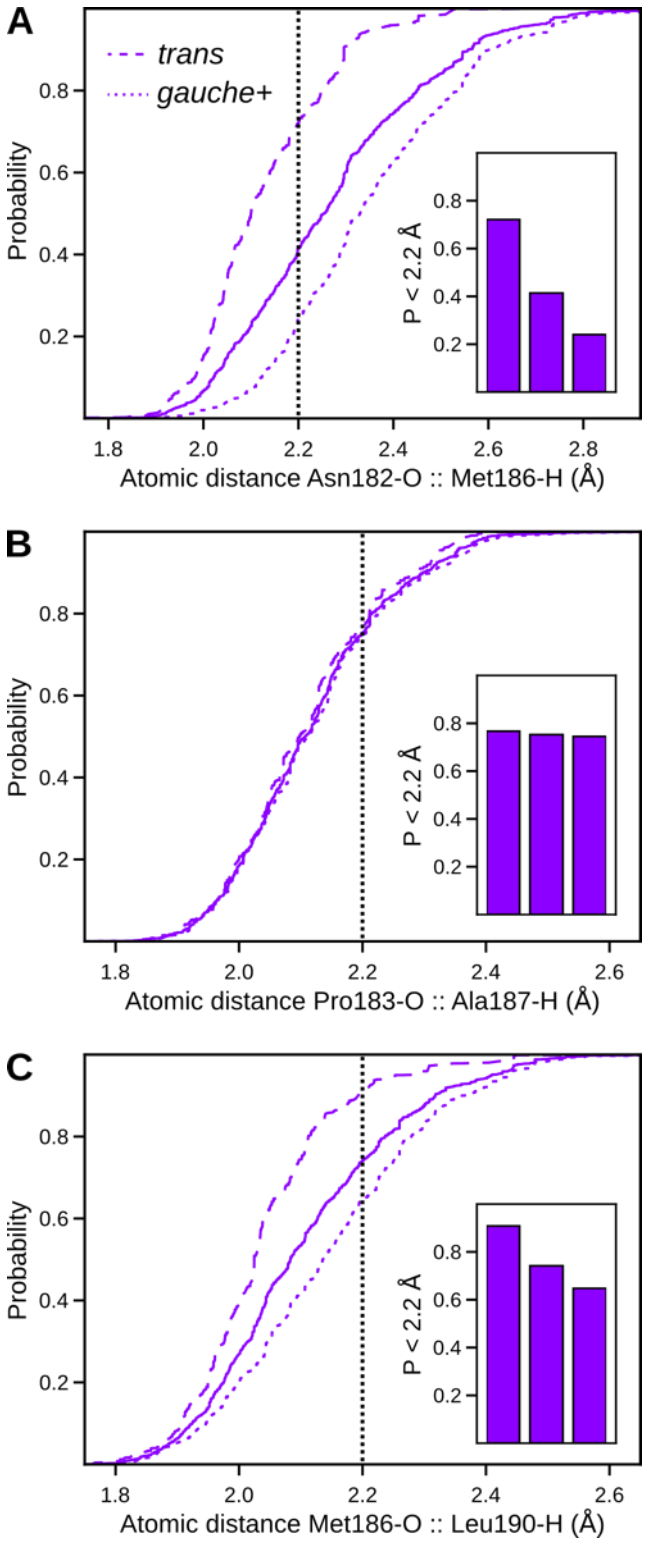

Figure A.2.7: M182N distance distributions, conditional on Asn182's rotamer conformation, for three helix 9 backbone hydrogen bonding partners. (A-C) Cumulative distribution plots, conditional to M182N's rotamer, of the three distances represented in Fig S2. Shown are the distributions for Asn182 adopting the trans rotamer (dashed lines), the gauche+ rotamer (dotted lines), and all rotamers (solid lines). These plots indicate the probability of

observing an atomic distance less than the specified value. Our cutoff distance for moderate hydrogen bonds, 2.2 Å, is shown as a dotted line. Probabilities of moderate hydrogen bonds for each pair are shown in the inset, which show a significant difference between the trans and gauche+ rotamer for two out of three distances.

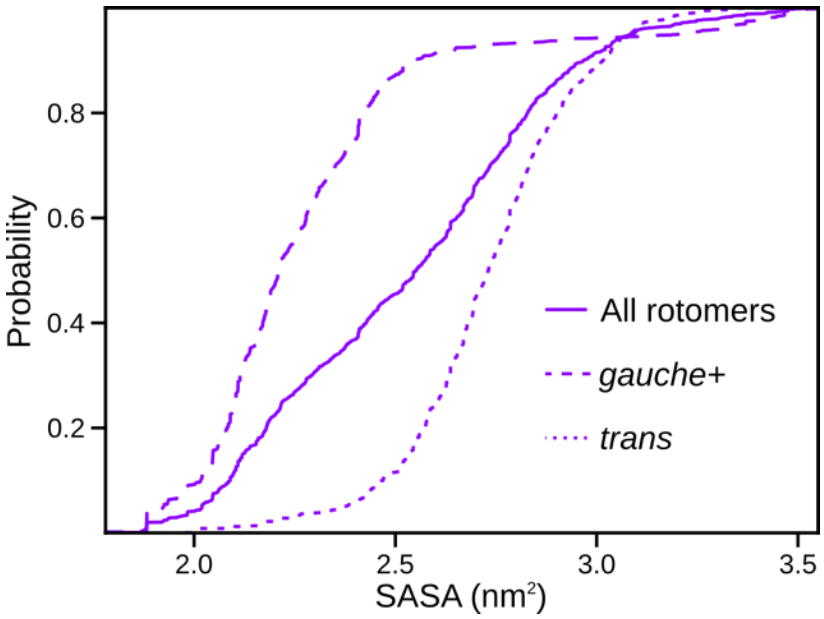

Figure A.2.8: Solvent accessibility distributions at the domain interface, conditional on Asn182's rotamer conformation. Shown are the cumulative distribution functions for the solvent accessible surface area of six residues: Tyr46, Ile47, Pro62, Glu63, Pro183, and Ala184, illustrated in Figure 5.6. These residues are located at the interface of the s2h2 loop, helix 9, and the β-sheet domain. Shown are the distributions for Asn182 adopting the trans rotamer (dashed line), the gauche+ rotomer (dotted line), and all rotamers (solid line). These plots indicate the probability of observing solvent-accessible surface area less than the specified value.

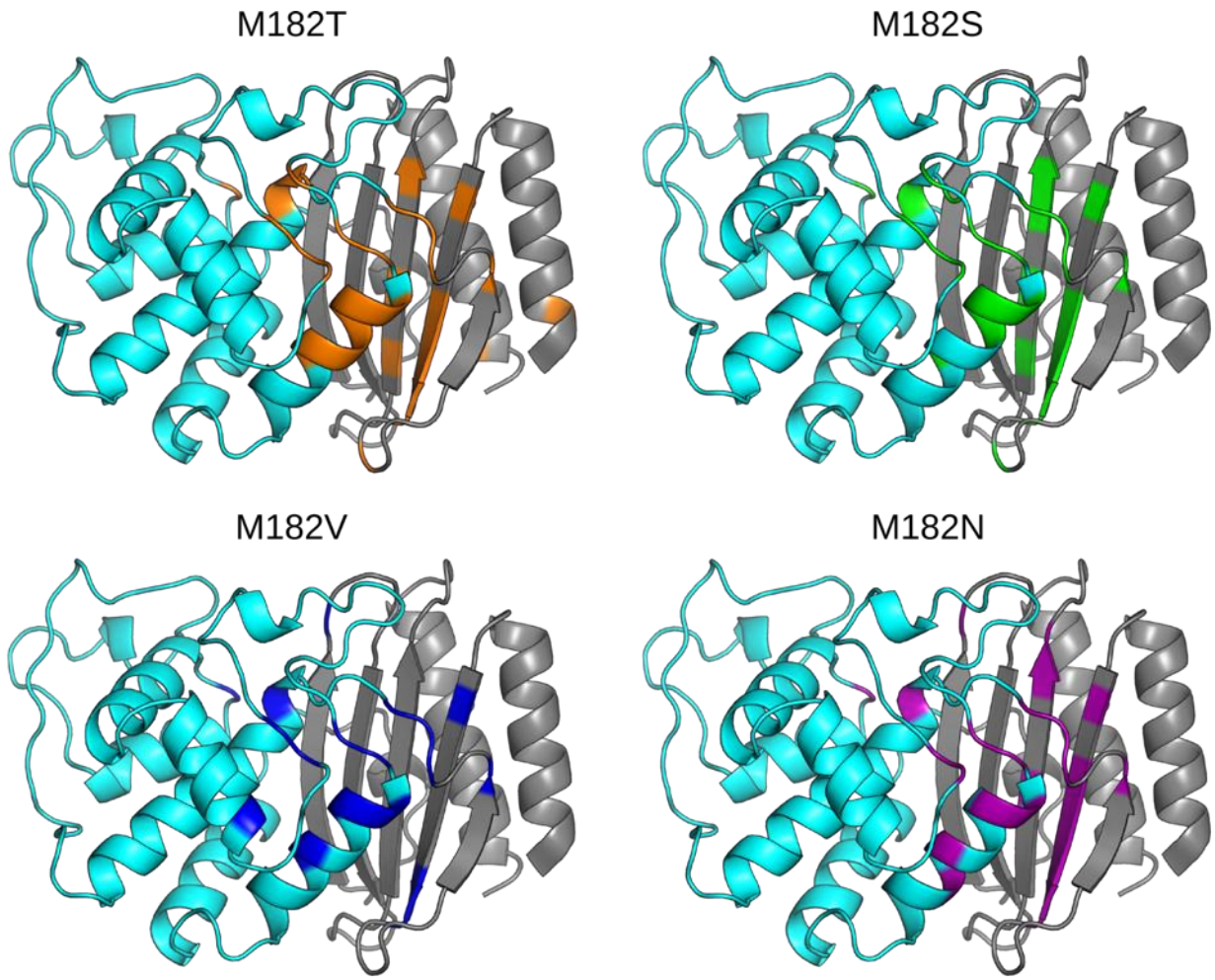

Figure A.2.9: Chemical shift perturbations for each TEM variant. The backbone of the α-helix domain (cyan) and β-sheet domain (gray) are represented as a cartoon. Highlighted residues indicate the locations of backbone amide chemical shifts that are perturbed significantly relative to wild-type. Chemical shift perturbations are shown for each TEM variant: M182T (orange), M182S (green), M182V (blue) and M182N (purple).

Curriculum Vitae

Maxwell I. Zimmerman
Email: mizimmer@wustl.edu
Phone: (954) 593-2930

SUMMARY:
Over the course of my graduate career, my goal has been to computationally design proteins that fold and function as efficiently as their natural counterparts. To accomplish this, major advances need to take place for understanding how a protein's sequence determines its structural ensemble. Towards this goal, my graduate work has primarily focused on developing and applying computational algorithms to characterize the diverse set of structures a protein adopts. These algorithms make use of molecular dynamics simulations and the construction of Markov State models and have been fruitful in discovering unexpected druggable pockets on proteins, detailing the thermodynamics and kinetics of conformational transitions in proteins responsible for antibiotic resistance, and the *de novo* prediction of protein folding pathways. Although I have made great strides in developing tools for understanding the ensembles of naturally occurring proteins, there is more that needs to be done to facilitate robust computational design. My recent endeavors consist of using deep learning tools for predicting the effect of mutations on conformational landscapes.

EDUCATION:

Ph.D. in Computational and Molecular Biophysics, 2019
Washington University School of Medicine in St. Louis,
Washington University in St. Louis

M.S. in Chemical Engineering, 2014
National High Magnetic Field Laboratory, Florida State University

B.S. in Chemical Engineering, 2012
National High Magnetic Field Laboratory, Florida State University

HONORS:

Monsanto/Bayer Graduate Research Fellowship 2016
WUSTL Center for Biological and Systems Engineering Scholar 2016
NSF-Graduate Research Fellowship Program (Honorable Mention) 2014

RESEARCH EXPERIENCE:

Graduate Research Assistant 2014-Present
Washington University School of Medicine, *Washington University in St. Louis*,

Advisor: Dr. Gregory R. Bowman

Graduate Research Assistant 2012-2014
National High Magnetic Field Laboratory, *Florida State University*,
Advisor: Dr. Anant K. Paravastu

Undergraduate Research Assistant 2010-2012
National High Magnetic Field Laboratory, *Florida State University*,
Advisor: Dr. Anant K. Paravastu

Undergraduate Research Assistant 2008-2010
Institute of Molecular Biophysics, *Florida State University,*
Advisor: Dr. Kenneth A. Taylor

TEACHING EXPERIENCE:
***Washington University in St. Louis, Division of Biology and Biomedical Sciences*:**
Macromolecular Interactions Spring 2016

***Florida State University, Department of Chemical and Biomedical Engineering*:**
Chemical Engineering Senior Design II Spring 2014
Chemical Engineering Computations Fall 2012

PUBLICATIONS:
1) Behring, J.B., van der Post, S., Mooradian, A.D., Egan, M.J., Zimmerman, M.I., Clements, J.L., Bowman, G.R., Held., J.M., "Spatial and temporal alterations in Protein Structure by EGF regulate Cryptic Cysteine Oxidation", bioRxiv **2019**
2) Porter, J.R., Moeder, K.E., Sibbald, C.A., Zimmerman, M.I., Hart, K.M., Greenberg, M.J., Bowman, G.R., "Cooperative Changes in Solvent Exposure Identify Cryptic Pockets, Switches, and Allosteric Coupling", *Biophys. J.* **2019**
3) Porter, J.R., Zimmerman, M.I., Bowman, G.R., "ENSPARA: Modeling Molecular Ensembles with Scalable Data Structures and Parallel Computing", *J. Chem. Phys.*, **2019**, 150 (4), 044108
4) Zimmerman, M.I., Porter, J.R., Sun, X., Silva, R.R., Bowman, G.R., "Choice of Adaptive Sampling Strategy Impacts State Discovery, Transition Probabilities, and the Apparent Mechanism of Conformational Changes", *J. Chem. Theory Comput.*, **2018**, 14, 5459-5475
5) Zimmerman, M.I., Hart, K.M., Sibbald, C.A., Frederick, T.E., Jimah, J.R., Knoverek, C.R., Tolia, N.H., Bowman, G.R., "Prediction of New Stabilizing Mutations Based on Mechanistic Insights from Markov State Models", *ACS Cent. Sci.*, **2017**, 3 (12), 1311-1321.
6) Hart, K.M., Moeder, K.E., Ho, C.M.W., Zimmerman, M.I., Frederick, T.E., Bowman, G.R., "Designing Small Molecules to Target Cryptic Pockets Yields Both Positive and Negative Allosteric Modulators", *PLoS one*, **2017**, 12 (6), e0178678
7) Zimmerman, M.I., Bowman, G.R. "How to run FAST simulations." *Methods in Enzymology*, **2016**, 578, 213-225
8) Zimmerman, M.I., Bowman, G.R., "FAST Conformational Searches by Balancing Exploration/Exploitation Trade-Offs." *J. Chem. Theory Comput.* **2015**, 11, 5747-5757

9) Huang, D., Zimmerman, M.I., Martin, P.K., Nix, A.J., Rosenberry, T.L., Paravastu, A.K., "Antiparallel β-Sheet Structure within the C-Terminal Region of 42-Residue Alzheimer's Amyloid-β Peptides When They Form 150-kDa Oligomers." *J. Mol. Biol.*, **2015**, 427 (13), 2319-2328
10) Cormier, A.R., Pang, X., Zimmerman, M.I., Zhou, H.X., Paravastu, A.K., "Molecular Structure of RADA16-I Designer Self-Assembling Peptide Nanofibers." *ACS Nano,* **2013**, 7 (9), 7562-7572
11) Leonard, S.R., Cormier, A.R., Pang, X., Zimmerman, M.I., Zhou, H.X., Paravastu, A.K., "Solid-State NMR Evidence for β-hairpin structure within MAX8 designer peptide nanofibers." *Biophys. J.* **2013**, 105(1), 222-230

ORAL PRESENTATIONS:

1) **Bayer Graduate Fellows Symposium**.
Prediction of New Stabilizing Mutations Based on Mechanistic Insights from Markov State Models.
St. Louis, MO, November 2018.
2) **Canadian Chemistry Conference and Exhibition, 101st Meeting.**
Identifying Cryptic Pockets Using FAST Conformational Searches.
Edmonton, AB, Canada, May 2018.
3) **Workshop on Free Energy Methods, Kinetics, and Markov State Models in Drug Design.**
Choice of Adaptive Sampling Strategy Impacts State Discovery, Transition Probabilities, and the Apparent Mechanism of Conformational Changes.
Boston, MA, May 2018.
4) **Monsanto Graduate Fellows Symposium.**
FAST Forward Protein Folding and Design.
St. Louis, MO, November 2017.
5) **Protein Folding Consortium.**
FAST Conformational Searches by Balancing Exploration/Exploitation Tradeoffs.
San Francisco, CA, June 2017
6) **Biophysical Society 61st Annual Meeting.**
FAST Forward Protein Folding.
New Orleans, LA, February 2017.
7) **Center for Biological and Systems Engineering Seminar Series.**
FAST Methods for Protein Folding and Design.
St. Louis, MO, November 2016.
8) **Gibbs Conference on Biomolecular Thermodynamics.**
FAST Conformational Searches by Balancing Exploration/Exploitation Trade-Offs.
Carbondale, IL, October 2015.
9) **Florida State University, Master's Thesis Defense.**
Peptide Nanostructure Formation Through Self-Assembly: Computations Guide Experimental Characterization and Amino Acid Sequence Design.
Tallahassee, FL, April 2014.