On the challenges and rewards of analyzing molecular dynamics at the terabyte and millisecond scale

Justin Roy Porter
Washington University in St. Louis

Follow this and additional works at: https://openscholarship.wustl.edu/art_sci_etds

Part of the Bioinformatics Commons, and the Biophysics Commons

Recommended Citation

This Dissertation is brought to you for free and open access by the Arts & Sciences at Washington University Open Scholarship. It has been accepted for inclusion in Arts & Sciences Electronic Theses and Dissertations by an authorized administrator of Washington University Open Scholarship. For more information, please contact digital@wumail.wustl.edu.
On the Challenges and Rewards of Analyzing Molecular Dynamics at the Terabyte and Millisecond Scale

by

Justin Roy Porter

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

May 2022
St. Louis, Missouri
Contents

List of Figures viii

List of Tables ix

Acknowledgments x

Abstract xiv

1 Introduction 1

1.1 Protein dynamics are crucial for function. 1

1.2 Molecular dynamics (MD) are structurally-detailed a model for protein dynamics. 3

1.3 Interregnum: The laws of Moore and Amdahl 5

1.4 Markov state models (MSMs) accurately and concisely summarize protein dynamics and thermodynamics. 8

1.5 Allostery is crucial for physiology. 10

1.6 Cryptic pockets present new drug opportunities 12

1.7 Myosins are technically challenging group of proteins with special importance to both physics and health. 15

1.8 Thesis aims 17
2 Modeling molecular ensembles at scale

2.1 Abstract ......................................................... 20

2.2 Introduction .................................................. 20

2.3 Results & Discussion ......................................... 22

  2.3.1 Ragged arrays .............................................. 22
  2.3.2 SIMD clustering using MPI ................................. 25
  2.3.3 Flexible, well-scaling clustering CLI ....................... 29
  2.3.4 Sparse matrix integration ................................. 31
  2.3.5 Fast and MSM-ready information theory routines .............. 33
  2.3.6 Flexible and interoperable model fitting and analysis ........ 36

2.4 Methods ....................................................... 40

  2.4.1 Source code and documentation ........................... 40
  2.4.2 Libraries and Hardware .................................. 40
  2.4.3 Simulation data ............................................ 41
  2.4.4 Residue labeling analysis ................................ 41

2.5 Conclusion ..................................................... 42

3 Cooperative changes in solvent exposure identify functional motions 43

3.1 Abstract ....................................................... 43

3.2 Introduction .................................................. 45

3.3 Methods ....................................................... 48

  3.3.1 Simulations ................................................ 48
  3.3.2 Solvent exposure featurization ............................. 48
  3.3.3 Markov state models ...................................... 49
  3.3.4 Exposon calculation ...................................... 49
  3.3.5 Labeling rate predictions ................................ 52
4.3.5 Markov state models ............................................. 87
4.3.6 Construction of the P-loop free energy surface ................. 89
4.3.7 Selection of myosin motor domain PDB crystal structures ....... 90
4.3.8 Hierarchical clustering of the P-loop .......................... 90
4.3.9 Assignment of new conformations to P-loop states ............... 91
4.3.10 Estimation of equilibrium probability of P-loop states ........... 91
4.3.11 Duty Ratios of Crystal Structures ............................. 91
4.3.12 Visualization .................................................. 92
4.3.13 Code and model availability .................................... 92

4.4 Results & Discussion ............................................. 93
4.4.1 In simulation, the P-loop adopts conformational states that are rare
in crystal structures. .................................................. 93
4.4.2 Simulations suggest that the nucleotide-free motor explores distinct
nucleotide-favorable and nucleotide-unfavorable states. .......... 94
4.4.3 The balance between nucleotide-favorable and nucleotide-unfavorable
P-loop states predicts duty ratio. ................................. 98
4.4.4 Simulations predict ADP release rates better than loop 1 length does
by capturing sequence-specific effects. ........................... 101

4.5 Conclusion ....................................................... 107

5 Main findings and future directions .............................. 111

5.1 Main Findings ................................................... 111
5.1.1 Summary ....................................................... 111
5.1.2 Key Conclusions ............................................... 113

5.2 Future Directions ............................................... 114

A Appendix A ........................................................ 117
A.1  Implied timescales ............................................... 117
A.2  Affinity clustering is stable to damping parameter changes ........... 119
A.3  Time-resolved DTNB thiol labeling by stopped-flow ..................... 121
A.4  Global unfolding rates of TEM-1 S243C and CTX-M-9 ................. 123
A.5  Protein stability measurements ......................................... 124
A.6  Activity of labeled enzyme ............................................. 125
A.7  Synthetic labeling of TEM-1 residues .................................... 126
A.8  Estimation of druggability of S243 pocket .............................. 128
A.9  $k_{\text{obs}}$ is bounded above by $k_{\text{obs}}^{(\text{EX1})}$ and $k_{\text{obs}}^{(\text{EX2})}$ .............................................................. 129

References  ................................................................. 131
# List of Figures

1.1 Amdahl’s law ......................................................... 7

2.1 Ragged arrays ....................................................... 24

2.2 SIMD formulation of \( k \)-centers and -medoids .................. 28

2.3 enspara CLI ......................................................... 30

2.4 enspara memory performance ..................................... 32

2.5 enspara time performance ........................................ 34

2.6 enspara library (API) use ......................................... 39

3.1 Exposon pipeline schematic ....................................... 58

3.2 TEM-1 exposons ..................................................... 62

3.3 eNP exposons ....................................................... 66

3.4 CAP exposons ....................................................... 69

3.5 Exposon centrality .................................................. 71

3.6 Exposons identify novel cryptic pockets ......................... 73

4.1 Myosin is a conserved fold ....................................... 84

4.2 P-loop conformation is broader than crystal structure suggest. 95

4.3 Excited P-loop states are less compatible with nucleotide. .... 99

4.4 P-loop free energy landscape .................................... 102
List of Tables

4.1 Summary of myosin simulations ............................................. 86
4.2 Parameters for whole-motor myosin MSMs ............................. 88
A.1 Linderstrøm-Lang fits of DTNB labeling .............................. 121
A.2 TEM-1 and CTX-M thermodynamic parameters ..................... 124
A.3 TEM-1 and CTX-M Michaels-Menten fits .............................. 125
I would like to thank my advisor, Greg Bowman. I feel extremely fortunate to have landed—a bit by accident, as I suspect it always is—in your lab. You were outstanding, and in ways that it’s only possible for me to recognize in hindsight. Thank you for always thinking of me, for pushing me in the right direction when I was astray, and for letting me muddle along to the answer when I wasn’t. It was profoundly generous, it’s made me a better scientist. Through you I learned to trust my own ideas, to make really make the argument when I think it could be right. This goes not just for in the lab, but in the rest of my life as well. I will not be just a better scientist, but a better person for having had you as a mentor.

Michael Greenberg, thank you for taking me on almost as one of your own students. For reading my specific aims pages when you had grants due and trolling the conference floor for famous people to come to my poster and for laughing when I complain about how your field names things and for explaining, probably a dozen times, about how rates work. I think I finally get it.

I am also grateful to the additional members of my thesis committee—Gautam Dantas,
Jay Ponder, and Andrea Soranno—for their mentorship and guidance over the years. I am similarly grateful to the faculty that guided me through rotations, Fred Prior, Jim Havranek, and Josh Swamidass. And to Eric Galburt, who always listens to me as though I have something to say.

I am grateful to the outstanding competence and leadership of the WashU MSTP—Brian, Liz, Linda, and Christy, who have gently guided all of us through what would otherwise have been a bewildering process and Wayne, who always helped me remember what was important and what was not. And thank you of course to the NHLBI and NIGMS for paying for the show.

I also want to thank to each one of the many members of the Bowman lab. In particular Maxwell Zimmerman, for each of many, many conversations. You always tell me what you think, and never let me sneak anything by you—even when I seem like I hate it, I’m thankful for it. And, for the rest of my days, whenever I use einsum, I will whisper a tiny prayer to the numpy gods and wish you were still sitting behind me, ready to help. Thank you also to my co-author Katie Moeder, for her experiments and for believing that I could be right—I can’t wait to be colleagues again someday. Gracias a Matt y Mick por hacer que parezca posible a aprender un idioma otra vez. And thank you to Neha and Upasana for teaching me that what I had cooked was actually daal, and for recognizing my first (and only) pun in Hindi. And of course thank you to Catie, Katie, Carrie, Catherine, Matt, Sukrit, Tom, Emily, and Shreya. Each of you enriched my time in the Bowman lab, and I am glad to have worked by your side.

Thank you to my other friends in science, especially Brian Weitzner helped me start more different types of journeys than anyone. Someday I hope to stand as tall, speak as clearly, and believe as completely as you do. I’m working on it.
Thank you to Oliver Lange, for every conversation about science and each walk to the bakery, and to Oliver and Mel together for everything else. Most of all for trusting me to listen when you explained things. Don’t think I’ll think about much of anything the same way ever again.

And to Jeff Gray who was there before I knew anything. Without Jeff there might have been no Germany, no Rosetta, and maybe no science at all. Thank you to Bill Schief, for introducing me to proteins at the edge of medicine, and for encouraging me to pursue this crazy and wonderful path I’m on.

Thank you to my friends who continue to make Seattle feel like home. Drew, Pat, and Dylan, who hold down the fort while I’m away; and also Larry and Will who have found themselves far away but who I probably see there more than anywhere. Tamir and especially Michael, for helping me through the hard times and making the good times better.

Thank you to my friends at Washington University, who supported, encouraged, and entertained me through this long, short four years. Thank you to Nakul and Minerva, for trusting me to guide you in the right direction. Thanks to the cool kids—you know who you are—for teaching me I could be cool too, if I wanted. Thank you to Dillan, for your goodheartedness and honesty. To Liz, for pushing me to push myself, and to forgive myself when I fall. To Alaric, for earnestly asking me hard questions as though I have something worth saying, even when I don’t have any reason to have anything worth saying. To Artur, for the faith you put in me and my ideas. Thank you to Rachel, for being kinder to me that I am able to be to myself. And thank you to Lucas, for pushing me to take bets when I think I’m right, and for always taking my calls.


Ich darf auch nicht die St.-Louis-Deutsche vergessen. Eike und Sascha, aber auch Nate, Brian, Gwendolyn und die anderen die es machbar gemacht haben, dass ich nicht alles verlernt und dadurch verloren habe.

And, of course, thank you to my parents, Kathy and Trevor. From you both I learned my love of learning, and my curiosity about how things work. Thank you, Mom, for reminding me to keep trying even when it isn’t easy. And thank you, Dad, for telling me that you’re proud when I do. Someday—when this is all over and the world hasn’t ended—we can have a party.

Finally, thank you to Elizabeth. It is difficult to express without sliding hopelessly into cliché, but it is also hardly sufficient to simply say thank you. You took my work seriously even when yours is serious as it comes. Thank you for being by my side when I needed you, and for being with me for and sharing with me in this mysterious, bizarre, and sometimes-disorienting but mostly-beautiful journey. Ich bin dein Flugzeug.

Justin R. Porter
Washington University in St. Louis
May 2022
ABSTRACT OF THE DISSERTATION

On the Challenges and Rewards of Analyzing Molecular Dynamics at the Terabyte and Millisecond Scale

by

Justin Roy Porter

Doctor of Philosophy in Computational and Systems Biology

Washington University in St. Louis, 2022

Professor Gregory R. Bowman, Chair

Molecular Dynamics (MD) and Markov state models (MSMs) are powerful tools for estimating and concisely representing the conformational ensemble accessible to biological macromolecules, particularly proteins. Conformational ensembles are of special importance to biological function, both in health and disease, because biology derives from molecules’ entire conformational distribution rather than any single structure. Consequently, MD is poised to become a powerful tool for personalized medicine and for the study of molecular sequence-function relationships generally. However, because of their hyperdimensionality and size, just generating MD datasets and Markov state models (MSMs) that represent biologically relevant molecules is a substantive technical challenge. Then, even once these models are generated, it is not immediately obvious how the conformational ensemble represented by an MSM encodes function. In this thesis, I first present ensparsa, a python library that makes it possible to build and analyze MSMs at an unprecedented scale. Then, I present “exposons,” an unsupervised machine learning method for discovering substructure these colossal datasets by searching for cooperative changes in a protein’s surface. This method is applied to several small systems of biological interest. Finally, I demonstrate the power these technologies to analyze the kinetic diversity of motor protein myosin, among the longest-studied proteins in all biochemistry, and in so doing address a longstanding mystery in the field of myosin
biochemistry. The applicability of these technologies is almost certainly not limited to the handful of systems I study here. Therefore, this work likely has broad implications for the future of biochemistry, personalized medicine, and the study of biology.
Chapter 1

Introduction

He did not want to compose another Quixote—which is easy—but the Quixote. Needless to say, he never contemplated a mechanical transcription of the original; he did not propose to copy it. His admirable intention was to produce a few pages which would coincide—word for word and line for line—with those of Miguel de Cervantes.

Pierre Menard, Author of the Quixote, J.L. Borges

1.1 Protein dynamics are crucial for function.

Proteins can and do adopt many difference configurations. A small protein of 101 amino acids in length has 200 rotatable bonds along its peptide backbone. In turn, they have, assuming only three possible states for each bond, $3^{100} = 5 \times 10^{47}$ possible configurations.
Thus, even if the protein is able to sample conformations at a rate of $10^6 \text{s}^{-1}$, it would take about $10^{37}$ times longer than the universe has existed to exhaust all possibilities [3]. This is called “Levinthal’s Paradox” [4]. Although not all of the combinations are possible (some, for example, are prohibited by excluded volume effects), there is clearly nonetheless substantial freedom in the way each polypeptide chain is arranged.

Many experimental techniques bear out this conclusion regarding the importance of dynamics to molecules’ behavior at room temperature. For instance, hydrogen-deuterium exchange (HDX) relies on the property that duterons replace amide protons in polypeptides’ backbone more slowly when the proton participates in a hydrogen bond. The complex kinetics in even early HDX experiments suggested that proteins are able to adopt diverse conformations [5]. Similarly, Nuclear Magnetic Resonance (NMR) relies on certain nuclei’s sensitivity to both local chemical environment and a large external magnetic field. Because it is not restricted to amide protons and because of the complex physics of nuclear spin, the NMR signal can contain information about many different features of the molecular ensemble. For instance, early NMR work showed that the sidechain benzyl ring of phenylalanine can rotate in the folded conformation, despite the apparently-tight packing in static crystal structures [6].

However, interpreting results from techniques like HDX and NMR often require—and are almost always at least improved by—access to detailed structural information. (Although NMR can also be used for structure determination, X-ray crystallography is an older and more popular technique [7]. *) In crystallography, a concentrated protein solution is slowly dried so that crystals form (similar to the production of sea salt from sea water). These crystals are then illuminated by coherent X-ray beams which, because of the regular pattern of the crystal phase, creates a diffraction pattern. Then, the crystallographer uses the diffraction pattern estimate the (usually single) structure that is most likely to have given

*Some of my own past work, in particular Zhang et al. [8], tackled the challenge of predicting structure from only NM resonances and a strong biophysical prior (our knowledge of protein physical chemistry).
rise to that diffraction pattern. These conformations encode a wealth of information, but they are fundamentally incomplete. The first crystal structure ever solved [9] serves as an illustrative case: myoglobin is known to bind molecular oxygen, but the structure provides no obvious explanation for how the oxygen molecule penetrates the core to interact with the heme group at its center. Another example that will be relevant to this thesis is the motor protein myosin, which has at least six different conformational states for different stages of its motor cycle [10]. Because of its size, there are no high-quality NMR structures that include the entire myosin motor domain.

Thus, there is increasing agreement that dynamics is crucial for function but is difficult to interpret structurally. In contrast, structures lack critical dynamical and distributional information [3]. The grand challenge, and one we have made progress on in this thesis, is to combine these two sources of information into a single model of a proteins’ behavior.

1.2 Molecular dynamics (MD) are structurally-detailed a model for protein dynamics.

Molecular dynamics (MD) is a method that models the movement of atoms in a protein in a physically-realistic and atomically-detailed way [11], thus providing a framework for investigating the conformational fluctuations that make protein function possible [3]. In MD, atoms are represented by charged, sticky spheres and are connected by springs. At each point in time, each atom has a position, a velocity, and is acted on by forces from each other atom. At the beginning of a simulation, atoms are in the position suggested by a homology model or an experiment, such as by X-ray crystallography. Because molecular motion is caused by random thermal fluctuations, atoms’ initial velocities are drawn randomly from
a Boltzmann distribution. Then, for each subsequent time, Newton’s laws of motion are integrated to produce a trajectory: at each time step, velocities are used to update positions, new positions imply new forces, which then update velocities, which in turn update positions, and so on *ad infinitum*.

As a model for protein physical chemistry, MD has proven successful across a wide range of applications. This include providing detailed models for how conformational transitions might occur, for the pathways by which proteins fold, and for how small changes to protein sequence function to create large differences in phenotype. The literature on this subject is vast, and is reviewed excellently by Dror et al. [12]. To give a specific illustrative example, by identifying a previously-unobserved state in the $\beta_2$-adrenergic receptor ($\beta_2$AR), Dror et al. [13] were able to resolve an apparent paradox in the interpretation of certain experimental data involving the so-called ionic lock which, despite biochemical evidence that it stabilizes the inactive state of the $\beta_2$AR, was disrupted in the inactive crystal structure. Indeed, MD simulations showed that, although the wild-type protein samples both formed and broken states of the ionic lock, the modified version for the crystal structure sampled only the broken state, thus explaining the crystal conformation [14]. Similarly, studies of protein folding using molecular dynamics have proven useful for tackling longstanding questions in protein folding, such as whether or not proteins fold along a single dominant pathway [15].

Along similar lines, Zimmerman et al. [16] used long simulations of the protein $\beta$-lactamase to explain the atomic basis for the differences in thermodynamic stability of several mutations associated with antibiotic resistance. Ultimately, they found that each mutation contributed to differing degrees to the stability of a helix at an important interdomain interface.

Historically, MD as a methodology has been limited by two primary factors: force field accuracy and sampling. Traditional MD force fields, like those used in this work, are imperfect and are an incomplete description of reality [12]. While improving these force fields is an
important and active area of research, in this thesis they will mostly be taken for granted. In this thesis, we will be primarily concerned with the sampling challenge, which is simply the task of enumerating or observing a statistically-relevant fraction of the possible and plausible conformations of a molecule. This has been particularly challenging because the molecular motions we are most interested in, such as folding, ligand binding, and major conformational changes, occur at the speed of microseconds to milliseconds [17]. In contrast, because of the vibrational speeds of interatomic bonds, the computer must calculate new positions every 2 fs (or, with some simplifications, occasionally 4 fs) [11]. Thus, until very recently, simulations were limited to the hundreds of nanoseconds. Fortunately, advances in compute hardware (and GPUs in particular), the development of special-purpose hardware [18], the assembly of distributed computing projects [19], the development of enhanced sampling strategies [20, 21] and (of particular importance to this work) the development of Markov state models (MSMs) [22, 23] have all contributed to the capacity to surmount this challenge, in at least a subset of important cases.

1.3 Interregnum: The laws of Moore and Amdahl

In 1965, Gordon Moore claimed that that the number of transistors on a chip would, for the foreseeable future, double every 2 years without a commensurate increase in power consumption [24]. The implication—if not the literal statement—was that single-core computer speeds would continue to rise exponentially with time. And, indeed, during the subsequent decades in which it held true both in spirit and in fact, it became known as “Moore’s law.” However, in the modern day, although transistor counts do continue grow more or less at this rate [25], the spirit of the law has been violated for some time—in 2017, for example, single-core performance improved by only 3% [26].
This is not to say that computer hardware technology has stagnated, however. Quite the opposite. Enormous strides have been made in areas such as security (notwithstanding mistakes such as Spectre [27] and Meltdown [28]), domain specific devices (field-programmable logic arrays and Anton [18], for instance) and, most importantly for this thesis, parallelization. In particular, the number of physical cores in a CPU has exploded, as has the speed of GPUs (for many reasons including core number) [25]. So too has the availability of cheap, consumer-grade processors such as those used in smartphones and embedded systems like “smart home” devices.

Despite these advances, it is simply not the case that a doubling of the number of parallel-processing units is equivalent to the doubling of a single-core speed. A similar mathematics is at play when we observe that two women cannot be teamed up to give birth to a single child in 18 weeks. Economists call this property ‘inseparability.’ In computer science, because inseparability is the rule, any fully separable calculation is termed ‘embarrassingly parallel.’

The diminishing returns of increasing parallelization for incompletely separable calculations is described by Amdahl’s law [29]:

\[
S = \frac{1}{(1 - p) + p/s}
\] (1.1)

where \(p\) is the parallelizable part of the program (the part that would benefit from increasing the number of threads, for example), \(s\) is parallelization factor (the increase in number of threads, for instance), and \(S\) is the observed speedup of the program.

The key consequence of Amdahl’s law is that, under modest parallelization, even small amounts of inseparable code embedded in otherwise parallel programs quickly dominate the runtime. That is, even for values of \(p\) close to 1 (highly parallel), increases in \(s\) show
Figure 1.1: Amdahl’s law implies that, with an increasing degree of parallelization, the serial part of a code increasingly dominates computation time. Left, for fixed $p$, $S$ rapidly diverges from $s$ if $p < 1$. Right, for fixed $s$, $S$ drops more sharply away from $s$ as $p$ declines from 1.

rapidly diminishing returns. This effect is illustrated in Fig. 1.1. These plots show, left, the diminishing returns of high $s$ when $p < 1$ and, right, the rapid decline in $S$ with low but non-zero values of $(1 - p)$ (right).

The relevance to molecular simulation is that it is much cheaper to produce many parallel trajectories on parallel hardware than it is to produce the equivalent amount of simulation as a single long trajectory. Consequently, technologies have been built that use various types of assemblages of commodity hardware to generate enormous parallel datasets. Of particular note is high-performance cluster technologies (HPC), new models of parallel computing (e.g. MapReduce [30]), the application of GPUs to folding [31], and distributed computing projects (e.g. Folding@home [19]). It is with the large cost difference between producing many parallel simulations and one serial simulation that we begin our discussion of the Markov state model.
1.4 Markov state models (MSMs) accurately and concisely summarize protein dynamics and thermodynamics.

In the absence of special hardware like Anton [18], the sampling problem takes on special importance. In this setting, even with expensive commodity hardware, the length of any single trajectory is limited to (at the time of this writing) on the order of hundreds of nanoseconds per day for most single-chain soluble proteins (ca. hundreds of amino acids in size). This means that, while the number of computers (and hence the number of trajectories) can be increased at linear financial cost, the length of these individual trajectories can only be increased by better hardware and, generally, a faster than linear increase in cost.

One solution to this problem is to modify the simulation somehow to increase the rate at which a desired event occurs. Some methods rely on the introduction of exogenous forces (for instance, metadynamics or steered MD [32]). This has the disadvantage, however, that these exogenous forces make it difficult to interpret the rate that events occur. Others methods modify temperature or other variables to enhance molecular exploration, such as replica exchange and simulated annealing [33]. These methods have disadvantages as well, such as temperature replica exchange’s worsened performance in surmounting entropic barriers. Still more strategies, like FAST [34], use criteria that adaptively restart simulations from the most interesting regions of conformational space.

Another important tool, and a complimentary one to adaptive resampling strategies like FAST, is the Markov state model (MSM). The key advantage of these models is that they allow the integration of many parallel trajectories into a single, unifying description of a molecule’s conformational landscape. Specifically, MSMs are network models of protein
kinetics and thermodynamics [23] wherein each node represents a conformational state, and each connection is weighted to represent the probability that the molecule jumps from some state to some other state. Formally, an MSM is represented by an $n \times n$ row-stochastic matrix, $T$, often referred to as the transition probability. Each element $T_{i,j}$ represents the probability that a molecule in state $i$ jumps to state $j$ in some time $\tau$, which is called the lag time. Although exactly how $T$ is estimated from trajectory data is an active area of research (see Sec. 2.3.6 for a somewhat more detailed discussion, as well as [21, 35–43]), broadly these transition probabilities are estimated by examining the number of times that the molecule was found in state $i$ and later found in state $j$ after $\tau$ time.

A particularly notable advantage of this representation is that the stationary distribution of $T$ (usually inexpensively† computable through an eigenvector/eigenvalue decomposition of $T$), yields the equilibrium probability distribution across the states in the MSM. This is important because the fraction of the time that a simulation or set of simulations spends in state $i$ is not equivalent to the probability of observing state $i$ in a simulation fully at equilibrium ($P(i)$). In any simulation of finite length, states more kinetically accessible to the initial configuration will be disproportionately represented. The MSM elegantly sidesteps this issue by considering only conditional probabilities. The matrix $T$ models the conditional probability $P(j@t+\tau \mid i@t)$—the chance that the system is observed in state $j$ at time $t+\tau$ after having been found in state $i$ at time $t$. This is advantageous because molecules, while not in rapid exchange with their entire landscape, are in rapid exchange with adjacent conformations. (And, indeed, the lag time $\tau$ must be chosen to be the time past which exchange between neighboring states can be said to be fast.) This property is particularly important for data sources like Folding@home [19], which often produce many thousands of short, independent trajectories.

† for larger models this does eventually become problematic, see Sec. 2.3.4 for strategies we developed to circumnavigate those problems.
Since their first application to proteins almost two decades ago [44], MSMs have proven a remarkably effective way of describing proteins’ thermodynamics and kinetics. They have been used to describe the folding landscape of fast-folding proteins [45], decipher the pH dependence of protein-protein interaction [46], understand thermodynamic stability [16], and predict catalytic efficiency [47], among many other tasks. Among the reasons for their success is that all states in the conformational ensemble are first-class citizens. There is no privileged status for “the structure,” but rather many conformations are explicitly considered, each with a weight corresponding to the energy of that conformation.

MSMs are limited, of course, in that they are discrete models of protein space, creating the need for a timescale faster than which the contents of each discrete state are in “fast” (meaning approximately instantaneous) exchange. It also requires us to make choices in how to actually draw these states, which is the source of much active work in the field but, in all cases, necessarily throws out information. Finally, MSMs also require us to sample the transition between states repeatedly to produce accurate estimates of the relative energies of states, which is a major challenge when sampling is already the “limiting reagent.”

1.5 Allostery is crucial for physiology.

Proteins’ tendency to fluctuate between many structures can create allostery, which is coupling between the conformation in distant parts of the protein. Allostery was first described as a property of the homotetrameric supramolecular assembly hemoglobin, which was shown by crystallography to move between a “tense” and “relaxed” state upon binding oxygen [48].

‡ Other important examples include the transmission of signals about binding events across...
membranes [49] and the modulation of catalytic activity, for example, in metabolic negative feedback loops [50].

Historically, allostery has been thought of as coordinated structural changes, typically from one to another of a small number of well-defined structural states of an entire polypeptide chain. In the classic two-state model of the homotetramer hemoglobin [48], each monomer is in an equilibrium between the relaxed (R) state, which binds molecular oxygen, and tense (T) state, which does not. Although monomers are identical, each monomer’s preference for tense is increased if its neighbors take on that state. Consequently, as oxygen is added to the tetramer, the tetramer’s affinity for the “next” oxygen declines. The result is a sharp sensitivity to ambient oxygen levels, which is utilized by biology to create efficient oxygen transport [51].

The relationship between neighboring T and R conformations can be characterized a change to the probability that monomer $i$ ($M_i$) takes on the tense conformation, $P(M_i = T)$ such that it is conditional on its neighbors $j$ and $k$:

$$P(M_i = T) \neq P(M_i = T \mid M_j = T, M_k = T)$$
$$\neq P(M_i = T \mid M_j = R, M_k = R)$$

This formulation makes it clear that allostery is simply a probabilistic, conditional dependence of one part of a molecule or assembly on another. Consequently, there ought not be any no requirement that all of the monomers be identical.

The value of this statistical view of allostery is that it allows the importation of numerous undergraduate biochemistry courses and medical schools.
tools from other areas of mathematics, statistics, and physics. Of particular interest in this thesis is mutual information (MI), which is a nonlinear measure, in terms of bits, the correlation between two random variables (such as a particular monomer’s conformation, for instance). MI has been used to quantify allostery with the MutInf method [52], and MI has been further evaluating correlations between order and disordered states [53], and even building networks of allostery throughout a protein [54]. Importantly, these analysis and others have revealed that this type of coupling exists in protein that were previously considered non-allosteric [55].

1.6 Cryptic pockets present new drug opportunities

One potential application of allostery, and the capacity of MD to model conformational ensembles, is to identify conformational rearrangements that are coupled with function and that are druggable. The addition of a drug to stabilize (or destabilize) some functional conformation could be one powerful way of modulating molecular function. An especially powerful feature of this approach is that it could enable rational drug design for proteins that are normally considered undruggable—for instance because their crystal structures have no obvious pocket—and it can also be used to activate desirable activities, whereas most directly-acting compounds are generally inhibitory because they act by sterically hindering natural function.

The value of these sites, however, has been incompletely realized in part because cryptic pockets are hard to identify. These issues are discussed in more detail in Chapter 3. As mentioned above, crystallography has a tendency to identify only one or a few conformations that are low-energy. Likewise, NMR methods that report on structure generally require that those conformations have a high population (low energy) before signal is detectable. Other
reasons that targeting cryptic pockets has been challenging is that designing drugs to target a particular site is still an unsolved problem.

Simulations, however, present an exciting method for identifying rare conformations that, if stabilized by a drug, could modulate function. Because of their high spatiotemporal resolution, they are ideally suited to identify rare conformations, and generate hypotheses about what sorts of compounds might bind to those conformations. And, because of the high level of detail offered by these simulations, they can be checked against and help interpret experimental data that does not otherwise immediately lend itself to a structural interpretation. This includes such applications as designing drugs against an entire ensemble (with techniques such as Boltzmann docking [56]) or comparison with kinetic data [57].

However, as datasets grow in size, actually examining large datasets to identify pockets has grown difficult. Although many years of work have gone into identifying pockets [58–61], it remains a matter of some dispute what exactly constitutes a pocket. That is, the very notion of a pocket may not be a fully coherent concept. Even a seemingly simple formulation like “a pocket is a concavity on a protein’s surface” gives rise to many formalized definitions. One formalization is the “alpha shape,” the largest sphere that can be drawn between atoms without encompassing any atoms. (A favorite analogy is that an alpha shape is the largest scoop that can be taken out of chocolate chip ice cream without containing a chocolate chip.) Another, used by LIGSITE [59], says that a pocket is a set of points such that, when rays are cast out many directions, they intersect protein before reaching the outside. Yet a third conceptualizes pockets as a set of points deep to a plane that is laid flat against a protein’s surface [58].

Despite the intense work on this subject, most practitioners will concede that each of these definitions frequently produces objects that do not match their intuitive sense of what a
pocket should be. This, however, is a as much a problem of philosophy as technology. For instance, consider the following question: are two large volumes separated by a small channel one pocket or two pockets? By the alpha shapes definition, they do not. By the LIGSITE definition, however, they do. These sorts of difficult decisions have given rise to entire toolkits for finding, describing, and refining pocket-finding strategies [61].

Another problem presented by structure-based pocket definitions is that the number of pockets discovered for a particular dataset grows more or less linearly with the number of structures in that dataset (i.e. the number of frames). For the data sets under consideration in this thesis, this numbers in the many millions of frames. Of course, methods exist for reducing this number (clustering, in Euclidean space, of all pocket volumes, for example [62]), but these methods are still laborious and require the choice of many hyperparameters.

In this thesis, I describe a method for segmenting a protein into regions that display cooperative solvent exposure. Pockets are one class of object that display this behavior, but as we will see, there are others. Crucially, unlike structure-based pocket detection strategies, where the number of pockets that must be analyzed grows with the data set size, exposons are a function of the entire ensemble, and simply become more better-estimated as increasing amounts of data are collected.
1.7 Myosins are technically challenging group of proteins with special importance to both physics and health.

The motor protein myosin is among the best- and longest-studied proteins in biology [63]. They are a diverse and ubiquitous class of molecular motors and, like all molecular motors, myosins catalyze the conversion of chemical potential energy into mechanical work. There are 38 myosin isoforms encoded by the human genome, and members of this group are responsible for much of the macroscopic force generated by the human body. The human genome encodes 38 different isoforms of myosin [10], and members of this group act as force sensors or generators for a diverse set of processes throughout the body [64–68], including their best-recognized role as the engine that contracts muscle [63]. Their diversity is encoded both in the head of the molecule (the “motor domain”) as well as in their variable tails [10].

As we might expect in a group with so many roles, the kinetic properties of individual myosin isoforms vary greatly. A straightforward example is speed: superfast extraocular myosin-II and β-cardiac myosin have steady-state ATPase rates that differ by at least two orders of magnitude.[67] Another often-discussed property is the ‘duty ratio.’ This is the fraction of time spent attached to the actin filament, and myosins span nearly the entire range: fast skeletal myosin-II has a duty ratio of 4% [69] and myosin-VI has a duty ratio upwards of 90% [66] (in the absence of load). There many more kinetic parameters, and this diversity is seen across almost all of them. Using a variety of kinetic parameters, it has been proposed that there are at least five different biophysical classes of myosin: fast, slow, calcium-sensitive force sensor, calcium-insensitive force sensor, and processive/gate [70, 71].

Unfortunately, it has remained impossible to predict even basic properties of myosin motors,
such as duty ratio or ADP release rate. The ADP release rate is of special importance because it is the rate-limiting step for actin detachment \[72\] (at saturating ATP concentrations).

Phylogeny does not reliably predict myosin properties: although more related myosin motor domains are perhaps more likely to have similar duty ratios, this is likely an accident of history rather than general principle since several myosin classes contain both high and low duty ratio motors \[70\]. Similarly, crystallography has not provided answers to these questions. Despite more than 100 available crystal models \[10\], there is no known general property of these structures that distinguishes high and low duty ratio motors. Finally, these proteins’ size (600-800 residues for just the motor domain) and the timescales of their motions (usually seconds) has made simulations to predict these properties prohibitively costly, and as such, previous simulation work has rarely exceeded a microsecond \[73–75\].

This is of particular interest to biomedical science because many diseases can result from mutations in myosin genes \[65, 76–79\]. Let us take, as an illustrative example, the most common of these diseases: hypertrophic cardiomyopathy (HCM), affecting roughly 1:500 healthy young adults \[80\]. The second most common etiology of familial HCM is mutations in \(\beta\)-cardiac myosin \((MYH7)\) \[81\], and in some cases, \(\beta\)-cardiac myosin motor function is specifically implicated: a few common variants have directly linked disease to motor overperformance \[82–85\]. Interestingly, mutations in the same protein can also give rise to a different clinical phenotype, dilated cardiomyopathy (DCM), requiring different clinical management \[86\]. Even more confusingly, this gene is also highly variable among healthy human genomes \[87\], making the interpretation of variants in this gene exceptionally difficult.

A recent model proposed that about half of known pathogenic myosin variants fall into two classes: 1) disruptors of the ‘interacting heads motif’ (IHM) \[88\], which is an alternative, quiescent myosin conformation, or 2) disruptors of enzymatic function \[89\]. Mutations causing HCM were statistically enriched in regions thought to be involved in the IHM, while
mutations causing DCM were enriched in regions likely to be involved in catalysis. No prediction is made for the remaining pathogenic variants, since they neither lie at the IHM interface nor in the active site. However, if HCM and DCM are caused by disruption of the IHM and catalysis, respectively, then it seems likely that other pathogenic variants act by allosterically disrupting these same functions. The carefully regulated allosteric networks that allow for healthy communication between effector binding sites [90] then become the mechanism by which disease is “transmitted” to these sites.

This protein is both an outstanding technical challenge, because of its size and the slowness of its motions, and an interesting biomedical problem. If we are to understand the atomic basis for disease—which one might reasonably expect to speed therapeutic development and improve clinical decision making—it is necessary to understand how these allosteric networks regulate function. To this end, in this thesis, I describe a methodology for using computer simulations to model the allosteric effects of motor sequence on two important kinetic properties of human (and chicken) myosin isoforms. We hypothesized that, much like cryptic allostery, the excited states that particular myosin isoforms sample ought to report on their kinetic behavior. In Chapter 4, we showed that by analyzing the kinetics and thermodynamics of the S1 fragment in solution, we can predict duty ratio and ADP release rate.

1.8 Thesis aims

Aim 1: Develop technologies that allow construction and analysis of MSMs at the terabyte and millisecond scale. With the advent of terabyte- and millisecond-scale simulations, analysis of molecular dynamics trajectories will require new tools and careful optimization of existing tools. As the first aim of my thesis, I developed enspara, a suite of
tools for analyzing datasets of this size.

**Aim 2: Examine cryptic pockets, switches, and allosteric coupling in large MSMs.** Even once the tools have been developed to produce and begin handling enormous simulation datasets with traditional tools, larger proteins’ and simulation sets yield larger conformational landscapes, which must in turn actually be analyzed to produce insight. Even once simulations are fully processed, it is often not obvious what happened in the simulation. To this end, in this thesis, I propose a method for segmenting proteins into smaller parts that show cooperative solvent exposure changes. We name these groups “exposons” to distinguish them from other clustering applications. This is motivated first by the observation that most important functions proteins carry out are manifest at the surface, since they must interact with the world, and the hypothesis that non-functional uncorrelated changes are rare, since this non-functional allostery is likely to be destroyed over time by genetic drift.

**Aim 3: Apply the results of Aims 1 and 2 to myosin motors to understand kinetic diversity of myosin motors.** Finally, I apply the mechanisms for analyzing large datasets to approximately 2 ms of data for various isoforms of the 600-800 residue myosin motor domain. I use tools to analyze these conformational landscapes and produce insight into the way these proteins are tuned by evolution for their specific function. This serves as a test for the hypothesis that function in these systems is encoded by the accessibility and favorability of excited states, rather than gross structural properties of the ground state. This is a resolution to the apparent paradox that the crystal structures of these kinetically divergent myosin motor domains are substantively indistinguishable.
Chapter 2

Modeling molecular ensembles at scale

The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble. It therefore becomes desirable that approximate practical methods of applying quantum mechanics should be developed, which can lead to an explanation of the main features of complex atomic systems without too much computation.

Paul Dirac

The work presented in this chapter was originally published as Porter et al. [91].
2.1 Abstract

Markov state models (MSMs) are quantitative models of protein dynamics that are useful for uncovering the structural fluctuations that proteins undergo, as well as the mechanisms of these conformational changes. Given the enormity of conformational space, there has been ongoing interest in identifying a small number of states that capture the essential features of a protein. Generally, this is achieved by making assumptions about the properties of relevant features—for example, that the most important features are those that change slowly. An alternative strategy is to keep as many degrees of freedom as possible and subsequently learn from the model which of the features are most important. In these larger models, however, traditional approaches quickly become computationally intractable. In this paper, we present enspara, a library for working with MSMs that provides several novel algorithms and specialized data structures that dramatically improve the scalability of traditional MSM methods. This includes ragged arrays for minimizing memory requirements, MPI-parallelized implementations of compute-intensive operations, and a flexible framework for model construction and analysis.

2.2 Introduction

Markov state models (MSMs) [23, 92–94] 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 [16, 47, 95, 96] and for producing insights about the mechanism of protein conformational transitions [97–100]. And, with the rise of special-purpose super-computers [18, 101], distributed-computing platforms [19], 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 [102] and MSMBuilder3 [103–105] offer 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) [106–108], but also through deep learning[109, 110] or explicit state-merging [111–114], 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 [55, 57], 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 [16], enzyme catalysis [47], and biochemical properties [55]. 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.

2.3 Results & Discussion

2.3.1 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 [19]. 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 $3n_{\text{atoms}}$), which assumes uniform trajectory length [37, 102].

To represent non-uniform trajectory lengths, a number of approaches exist. One approach, found in MSMBuilder2 [104], 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 [115], 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[105] 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 (Fig. 2.1a). 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 (Fig. 2.1b), such as the square array used in MSMBuilder2 while retaining the useful properties of an array which are lost in an list-of-arrays representation.
Figure 2.1: 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 [99].
2.3.2 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 [116–118]. The most popular clustering algorithms for use in the MSM community are $k$-means [119] (generally composed of $k$-means++ [120] initialization and Lloyd’s algorithm [121] for refinement) for featurized data, and $k$-hybrid [104] (composed of $k$-centers [122] initialization and $k$-medoids [123] 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 2.3.4), 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 [124] 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 [37] so that only every $n$th frame is used. Another approach is to reduce the number of features by including only certain atoms (as in Zimmerman et al. [16], Schwantes and Pande [125], Bowman and Geissler [126]), using a dimensionality reduction algorithm like principal components analysis (PCA) [127, 128], or creating a hand-tuned set of order parameters (e.g. specific, relevant pairwise atomic distances). Yet a third approach is to use tICA [106, 107] 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) [106].
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) [129] 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 [31, 130] (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 [104]. 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 [122] 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 `scatter`ing 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 Fig. 2.2a 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, Zhao, Yutong et al. [117]). 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
Figure 2.2: 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.

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 Fig. 2.2c. 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.
2.3.3 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 (Fig. 2.3a–c). This clustering application can take trajectories in formats accepted by MDTraj (Fig. 2.3a) or numpy arrays of numerical features (Fig. 2.3c), supports several different distance metrics, provides easy support for clustering different topologies into shared state spaces (Fig. 2.3b), and supports execution under MPI.

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 Fig. 2.3d, 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 [55, 57]. 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
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 3Å, and with 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 distance between vectors of residue sidechain solvent accessible surface area, whereas solid lines use backbone RMSD. Blue traces used $k$-centers and red traces used $k$-hybrid. The experimental measurement is a previously-published\cite{55} 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 is deterministic and has no error bars).
large amounts 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. 2.2). 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 (Fig. 2.4). For purposes of reference, clustering of the TEM-1 data set used all 2026 protein heavy atoms across 90.5 microsecond total simulation time saved every 100 picoseconds and the G_q dataset used all 2655 protein heavy atoms across 20.5 microsecond saved every 10 picoseconds. All these values trade off against one another, however, meaning that if every 10th frame were used to cluster the G_q dataset, 205 microsecond of data could be clustered on a single node (and up to 1.03 ms on 5 nodes using MPI).

2.3.4 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, MSMBuilder3 [37, 104, 105] and PyEMMA [102]. 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 [131].

To address the computational challenges posed by traditional arrays, enspara has been
Figure 2.4: 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 32GB because, on the computer system tested (see Methods), the operating system allocates double the necessary RAM to enspara. Where MSMBuilder runs out of RAM loading ~16GB, enspara is capable of using almost all of the available 64GB 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 26GB of data loaded is a consequence of the slowdown in state discovery as new data is added. For a and b, error bars represent the standard deviation of three trials.
engineered to support sparse arrays wherever possible. Sparse arrays have been supported by MMMBuilder 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 non-zero 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 (Fig. 2.5a). By implementing routines that support scipy’s sparse matrices, it becomes possible to keep much larger Markov state models in memory (Fig. 2.5b) and analyze those models much more quickly (Fig. 2.5c).

### 2.3.5 Fast and MSM-ready information theory routines

Recent work [52, 53, 132] 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

\[ MI(X, Y) = \sum_{y \in Y} \sum_{x \in X} P(x, y) \log \frac{P(x, y)}{P(x)P(y)} \]  

(2.1)

where \( P(x) \) is the probability that random variable \( X \) takes on value \( x \), \( P(y) \) is the probability that random variable \( Y \) takes on value \( y \), and \( P(x, y) \) 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 \( P(x, y) \) is estimated by counting the number of times that
Figure 2.5: The performance characteristics of sparse and dense matrices 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 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.
combination of features appeared in each frame [52, 53]. 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 \( P(x, y) \) is estimated by \( \sum_{s \in S} \pi(s) \), where \( \pi(s) \) is the equilibrium probability of state \( s \) 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 [53, 133] either do not provide the option to compute MI with weighted observations or require a specific object-based framework to do so [134].

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 (e.g. Singh and Bowman [53]), 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 [135] and OpenMP [136] 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 [137].

Additionally, in this package, we include a reference implementation of Correlation of All Rotameric and Dynamical States framework (CARDS) [53]. 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 2.3.5.

2.3.6 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 2.3.5) 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 [115].) 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. [21].

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,[133] 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 flow 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 [21, 35–43]. Perhaps the simplest procedure to estimate the transition probability matrix, $T$, is to row-normalize the transition count matrix, $C$,

$$T_{ij}^{\text{normalize}} = \frac{C_{ij}}{\sum_k C_{ik}} \quad (2.2)$$

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 \( \hat{c} \) before estimation,

\[
T_{ij}^{\text{pseudo}} = \frac{C_{ij} + \hat{c}}{\sum_k (C_{ik} + \hat{c})}
\]

(2.3)

which ensures ergodicity.[21] A more dramatic conditioning comes when forcing the counts matrix to obey detailed balance by averaging forward and reverse transitions:

\[
C_{ij}^{\text{transpose}} = \frac{C_{ij} + C_{ji}}{2}
\]

(2.4)

\[
T_{ij}^{\text{transpose}} = \frac{C_{ij}^{\text{transpose}}}{\sum_k C_{ik}^{\text{transpose}}}
\]

(2.5)

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 [37, 93]. Framed as a Bayesian inference, the transition probabilities are solved as the most likely given a transition counts matrix, such that,

\[
T_{ij}^{\text{MLE}} = \arg\max P (T_{ij}^* | C)
\]

(2.6)

Additionally, there exist more sophisticated schemes of estimation, such as those that draw on inspiration from observable operator models [39], and projected MSMs [138]. 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
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 2.6: 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.

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 (Fig. 2.6a). 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 (Fig. 2.6b). 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.
2.4 Methods

2.4.1 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 function's 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.

2.4.2 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 [139–141] 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 [118].

2.4.3 Simulation data

For example simulation data, we used a previously-published 90.5 microsecond TEM-1 β-lactamase dataset [99] and a 122.6 microsecond Gq dataset [142]. As described previously, simulations were run at 300 K with the GROMACS software package[31, 130] using the Amber03 force field [143] and TIP3P [144] explicit solvent. Data was generated using the Folding@home distributed computing platform [19].

2.4.4 Residue labeling analysis

Residue labeling behavior for residues A150, L190, S203, A232, A249, I260, and L286 was measured in Bowman et al. [55] and for S243 in Porter et al. [57]. “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 [57]. In brief, sidechain atoms’ solvent exposure to a 2.8 Å probe was calculated (using the Shrink-Rupley [145] algorithm implemented by MDTraj [118]) for the representative structure for each MSM state, and the residue was called as exposed if its exposed area exceeded 2Å.
2.5 Conclusion

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.
Chapter 3

Cooperative changes in solvent exposure identify functional motions

Pensar es olvidar diferencias, es generalizar, abstraer. En el abarrotado mundo de Funes no había sino detalles, casi inmediatos.

To think is to forget a difference, to generalize, to abstract. In the overly replete world of Funes, there were nothing but details.

Funes el memorioso, J.L. Borges

The work presented in this chapter was originally published as Porter et al. [57].

3.1 Abstract

Proteins are dynamic molecules that undergo conformational changes to a broad spectrum of different excited states. Unfortunately, the small populations of these states make it dif-
ficult to determine their structures or functional implications. Computer simulations are an increasingly powerful means to identify and characterize functionally-relevant excited states. However, this advance has uncovered a further challenge: it can be extremely difficult to identify the most salient features of large simulation datasets. We reasoned that many functionally-relevant conformational changes are likely to involve large, cooperative changes to the surfaces that are available to interact with potential binding partners. To examine this hypothesis, we introduce a method that returns a prioritized list of potentially functional conformational changes by segmenting protein structures into clusters of residues that undergo cooperative changes in their solvent exposure, along with the hierarchy of interactions between these groups. We term these groups exposons to distinguish them from other types of clusters that arise in this analysis and others. We demonstrate, using three different model systems, that this method identifies experimentally-validated and functionally-relevant conformational changes, including conformational switches, allosteric coupling, and cryptic pockets. Our results suggest that key functional sites are hubs in the network of exposons. As a further test of the predictive power of this approach, we apply it to discover cryptic allosteric sites in two different β-lactamase enzymes that are widespread sources of antibiotic resistance. Experimental tests confirm our predictions for both systems. Importantly, we provide the first evidence for a cryptic allosteric site in CTX-M-9 β-lactamase. Experimentally testing this prediction did not require any mutations, and revealed that this site exerts the most potent allosteric control over activity of any pockets found in β-lactamases to date. Discovery of a similar pocket that was previously overlooked in the well-studied TEM-1 β-lactamase demonstrates the utility of exposons.
3.2 Introduction

Proteins are highly dynamic molecules that are capable of accessing a wide variety of excited conformations. Many of these excited states have important biological functions. For example, many proteins predominantly adopt an off state until they interact with a binding partner that stabilizes a higher energy on state. However, the most common tools for structural biology, such as x-ray crystallography and cryoelectron micrography, typically only provide a static picture of one (or a few) low energy states.

Computer simulations, because of their excellent spatiotemporal resolution, are a promising means to identify functionally-relevant excited states and conformational changes [146]. However, simulations have historically faced severe limitations. In particular, the inability to capture slow processes, such as large-scale conformational changes, has hampered the routine discovery of physiologically-relevant excited states using computer simulations. However, enormous advances in computer hardware and simulation algorithms have made it possible to capture processes that occur on tens to hundreds of milliseconds, finally giving access to this physiologically-important timescale for many proteins [45, 124]. One successful approach has been to combine many parallel simulations executed on commodity hardware into a single model of protein dynamics using Markov state models (MSMs) [23, 41, 44]. MSMs are network models of protein energy landscapes composed of many conformational states and the probabilities of hopping between them. Because they are able to integrate information from many independent simulations, they are capable of reaching timescales many orders of magnitude larger than any of the individual simulations used to build the model.

The growing availability of long-timescale simulations has revealed a new major challenge: extracting meaningful insights from the resulting colossal datasets. These datasets are not
only composed of hundreds of millions or billions of timepoints, but are also embedded in
tens or hundreds of thousands of dimensions. Numerous methods have been developed to ad-
dress this challenge. One method, projecting simulation data onto specific order parameters
is a valuable means to test hypotheses, but this approach requires detailed foreknowledge of
which parameters are important to avoid obscuring important features [147]. For data sets
where foreknowledge is not available, unsupervised methods have been developed to learn
what degrees of freedom are important. For example, principle component analysis (PCA)
[148] highlights large geometric changes. Unfortunately, larger conformational changes are
not necessarily more functionally relevant. For example, the large variance in the atomic
positions of a disordered loop can easily dwarf a more subtle, but more functionally rele-
vant, conformational change. Another approach is to leverage the variational principle [149],
typically operationalized in the form of time-lagged independent component analysis (tICA)
[150, 151] instead focuses on slowly varying dimensions. Slowness, however, does not nec-
essarily imply functional relevance. For instance, the process of flipping a phenylalanine
about its ring axis may be slow, but the exchange-symmetry of atoms on either side of the
ring means the process does not alter the conformation at all. Indeed, practitioners have
increasingly begun to move toward a combination of hand-tuned features combined with
tICA [152] which, while focusing the featurization, is labor-intensive and requires detailed
foreknowledge of the system to avoid omitting important features.

To address these challenges, we hypothesized that functionally-relevant conformational changes
are likely to result in large, cooperative changes to the surfaces of a protein that are available
to interact with potential binding partners. This hypothesis was inspired, in part, by the
fact that surface chemistry is an especially important feature of most proteins, since it is how
the protein interacts with other objects, including any substrates and binding partners. Fur-
thermore, we reasoned that functionless cooperativity at protein surfaces is rare. Physically,
the folded state is bombarded by thermal noise which—in the absence of a specific design constraint—will tend to decorrelate any arbitrary pair of features. Genetically, sequence drift would be expected to eliminate cooperativity that is not selected for over time; much as early protein redesigns sometimes inadvertently destroyed cooperative folding [153]. This assumption is also the basis for sequence-based methods that use patterns of conservation and covariance to infer pairs of residues that are in direct contact in a protein’s three-dimensional structure, or that are allosterically coupled [154–156].

To make testing this hypothesis tractable, we developed a method that returns a prioritized list of potentially functional conformational changes by segmenting protein sequences into clusters of residues that undergo cooperative changes in their solvent exposure and uncovers the hierarchy of interactions between these groups. We term these clusters of mutually-correlating residues exposons to disambiguate them from other forms of clustering found in this work and in the literature. To identify exposons and the structural motions that give rise to them, we present an efficient, MSM-based approach. For a concrete example of the utility of identifying cooperative changes in solvent exposure, consider cryptic pockets. Cryptic pockets are transient concavities on protein surfaces that open when the protein fluctuates to an excited state [157, 158]. Pocket opening concomitantly increases the solvent exposure of surrounding residues, and closing a pocket simultaneously reduces their exposure. Thus, these residues undergo correlated changes in their solvent exposure and are likely to form an exposon.

To establish the value of exposons, we demonstrate that they naturally identify a variety of functionally-relevant conformational changes without any foreknowledge of what structural features are important for a given system. First, we show that exposons identify cryptic pockets and allostery in the enzyme TEM-1 β-lactamase. We then show that they detect a conformational switch in the Ebola virus’ nucleoprotein (eNP) and allostery in the catabolite
activator protein (CAP). Then we use exposons to prospectively discover cryptic allosteric sites in two different $\beta$-lactamase enzymes with less than 40% sequence identity, TEM-1 and CTX-M-9, and perform *in vitro* biochemical experiments to test our predictions.

### 3.3 Methods

#### 3.3.1 Simulations

As described previously [159], simulations were run at 300 K with the GROMACS software package [160] using the Amber03 force field [161] and TIP3P [162] explicit solvent. $\beta$-Lactamase simulations were deployed on the Folding@home distributed computing platform [19], while simulations of eNP and CAP were performed on NVIDIA P100 GPUs on our local cluster. For our retrospective work, we used previously published datasets including 90.5 µs of simulation of TEM-1 $\beta$-lactamase with the M182T substitution [99], 28.0 µs of simulation of eNP [163], and 1.5 µs of simulation of CAP [53]. For our prospective work on CTX-M-9 $\beta$-lactamase, we ran 76.0 µs of aggregate simulation on Folding@home.

#### 3.3.2 Solvent exposure featurization

To generate a solvent exposure featurization of each dataset, we computed the solvent-accessible surface area (SASA) of each residue’s sidechain in each simulation frame to a drug fragment-sized probe using the Shrake-Rupley [145] algorithm, as implemented in MDTraj [164]. The result is a set of $t$ vectors of length $n$, where $n$ is the number of residues and $t$ is the number of frames. A probe size of 2.8 Å was chosen because previous work suggests this value identifies pockets that can accommodate a drug-sized molecule [55]. Jug [137] was
used to organize the parallel execution of many independent tasks, including parallelizing solvent-accessibility calculations across many cores.

### 3.3.3 Markov state models

We defined MSM microstates by clustering the sidechain solvent-accessible surface area featurized representation. Clusters were discovered using \( \mu \)-centers, which adds new cluster centers until the maximum within-cluster distance dropped below a threshold value, which was 2.6 nm for TEM-1, 3.5 nm for CTX-M-9, 5.0 nm for eNP, and 3.7 nm CAP. These values were chosen based on the implied timescales test (Fig. A.1). Then, five rounds of \( \mu \)-medoids updates were performed, where updates were accepted if the largest distance to the nearest medoid decreased. Then, to estimate transition probabilities from assignments of frames to clusters, we first constructed a transition count matrix, where the element \( C_{ij} \) is the number of transitions observed from state \( i \) to state \( j \). Then we added a pseudocount of \( 1/n \) (where \( n \) is the number of states) to each element of the transition counts matrix, and row-normalized this matrix to obtain a transition probability matrix, as suggested in Zimmerman et al. [21] and Prinz et al. [93]. The lag times were 0.1 ns (TEM-1), 0.1 ns (CTX-M-9), 0.5 ns (eNP), and 0.4 ns (CAP), which were chosen by the implied timescales test (Fig. A.1). The highest flux pathways between two sets of states were then extracted using transition path theory [165, 166].

### 3.3.4 Exposon calculation

Beginning with the featurized representation for the representative conformation for each MSM state, we classified each sidechain in each state as exposed or buried using a fixed threshold. We chose a fixed threshold rather than a continuous threshold to reduce the
number of parameters (a sigmoid, for example, would require a step width and a step mid-
point) and because previous work suggested that mutual information performs better when a
smaller number of bins are used [53]. Our fixed threshold in this paper was 2.0 Å, but choices
in the range 2.0–5.0 Å, as well as formulated as a fraction of maximum possible sidechain
exposure in the 1-3% range, gave similar results. This invariance implies that our algorithm
does not erroneously favor larger residues due to their larger maximum possible SASA. The
end result is a featurization of each MSM state, wherein each snapshot is represented by a
binary vector with one entry per residue that contains a one for exposed residues and a zero
for buried residues.

We then calculate the mutual information between each pair of residues. Mutual information
(MI) is a measure of the statistical interdependence of two random variables. It is given by
the equation,

$$\text{MI}(X, Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left( \frac{p(x, y)}{p(x)p(y)} \right)$$  \hspace{1cm} (3.1)

where $X$ and $Y$ are any pair of residues and $x$ and $y$ represent the solvent accessibility
states (i.e. buried, exposed) of the corresponding residue. The probability $p(x)$ is the
probability that a residue is observed in state $x$ and $p(x, y)$ is the joint probability of $x$
and $y$. These probabilities are the equilibrium probabilities calculated during MSM fitting.

Other methods could be used in place of MSMs to identify a set of representative structures
and their equilibrium probabilities. However, MSMs are advantageous as they provide a
facile means to extract the motions that give rise to an exposon. Another key advantage of
MSM technologies is that they often better estimate true equilibrium probabilities on sets of
trajectories of finite length. To compute mutual information matrices for CAP, we leveraged
its dimer symmetry to improve sampling of solvent exposure states. If $A_i$ and $B_i$ are the
random variables representing the exposure states of residue $i$ of chain A and B, respectively, then by the chemical identity of the two chains, at equilibrium $p(A_i, B_j) = p(A_j, B_i)$. To exploit this fact to enhance our sampling of the state space and make our predictions more robust to sampling error, we take the mean of these two distributions when computing mutual information.

Exposons are the cluster assignments computed by affinity propagation [167]. Our choice of this algorithm was motivated by affinity propagation’s ability to use a similarity matrix (rather than distances), lack of a need for its similarities to not satisfy the triangle inequality (as MI does not satisfy this inequality), and use of relatively few tunable parameters. We use the affinity propagation implemented in scikit-learn 0.19.0 [133] and zero initial affinities. The one parameter that must be chosen is the so-called “damping parameter” which usually (but not always) causes the algorithm to produce fewer clusters at higher values. In practice, however, the results are generally similar throughout most of the range of valid choices of the damping parameter (from 0.5 to 1.0), though the results will vary for damping parameters very near to 1.0 or 0.5. The damping parameters were 0.9 for TEM-1, CTX-M-9 and eNP and 0.95 for CAP, generally chosen to be as high as possible (creating a low number of exposons) without causing the algorithm to converge on a single exposon for the entire protein. This typically generates 10-50 exposons, of which we visualize the top 3-10. Affinity propagation is robust to the choice of damping parameter, giving similar values for much of the range of valid choices (Fig. A.2).

Coarse-grained exposon graphs are network models of the communication between each pair of exposons. In this model, each node represents an exposon and each edge represents the communication between a pair of exposons. The weight of each edge in the coarse-grained
network is calculated by,

\[ \text{MI}_{\text{coarse}}(A, B) = \frac{\sum_{i \in A} \sum_{j \in B} MI(i, j)}{C(A, B)} \]  

(3.2)

where \( A \) and \( B \) are any pair of exposons (sets of residues) and \( C \) is the channel capacity, which is the maximum possible mutual information between two pairs of exposons [53]. In this case, the channel capacity is given by \( C(A, B) = \min(|A|, |B|) \) or one bit per residue up to the number of residues in the smallest exposon. When visualizing coarse grain exposons, we omit self-edges and very low-valued edges (< 0.015 for TEM, < 0.075 bits for eNP, and < 0.2 bits for CAP).

Eigenvector centrality calculations were performed with NetworkX [168].

### 3.3.5 Labeling rate predictions

We predict time-dependent labeling behavior using the MSM we fit as described above. Labeling rate predictions are made first by identifying all states in which the residue of interest is exposed and converting them to sink states by zeroing out the rows in the transition probability matrix. Then, iteratively multiplying the equilibrium probability distribution by this new matrix gives a monotonically decaying fraction of ‘unlabeled’ probability density as density flows into the sink states. Finally, we fit the unlabeled fraction as a function of time to a single exponential to yield a rate. In the limit of a perfectly good fit, this rate is equivalent to a mean first passage time [169]. An implementation of this simple procedure is provided in a Jupyter notebook (see ‘Code Availability’). We used SciPy [170] version 0.19.1 for curve fitting.
3.3.6 Protein expression and purification

TEM-1 was purified from the periplasmic fraction of BL21(DE3) cells (Agilent Technologies) using both cation exchange and size exclusion chromatography. The full protocol is described in previous work [47].

We subcloned the CTX-M-9 gene into the multiple cloning site of pET9-a vector. Plasmids were 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 for 5 hours at 37°C. The cells were then centrifuged and the pellet was frozen at -80 °C.

CTX-M-9 cells were resuspended in 20 mM sodium acetate, pH 5.5, sonicated, and then centrifuged. The protein was purified from the insoluble cytoplasmic fraction. The pellet was unfolded in 9 M urea 20 mM sodium acetate, pH 5.5 and centrifuged. CTX-M-9 was then refolded in 20 mM sodium acetate, pH 5.5, purified by both cation exchange and size exclusion chromatography and stored similarly to TEM-1.

All cysteine mutations were introduced with Quik-Change mutagenesis.

3.3.7 Thiol labeling

We observe the change in absorbance over time of DTNB (Ellman’s reagent, Thermo Scientific), a small molecule that changes its absorbance as it covalently binds reduced cysteine sidechains [55]. We used a SX20 stopped-flow instrument (Applied Photophysics) with a dead time of 1.5 ms. Measurements were taken over time in 20 mM Tris, pH 8 1% DMSO, followed at an absorbance of 412 nm (ε_{412} = 14,150 M⁻¹ cm⁻¹), and fit by a single-exponential (Fig. A.2). Our previous work with thiol labeling was performed using manual mixing in a standard UV-Vis spectrophotometer [55], but in this work we used a stopped-flow instru-
ment that gives access to faster timescale motions and improves the quality of fits because it offers a dead time that is much shorter than the time scale of our experiments. It also allows for the use of lower DTNB and protein concentrations.

The labeling rate at a given DTNB concentration can be described by the Linderstrøm-Lang model, originally derived for hydrogen-deuterium exchange [171]:

\[
\begin{align*}
\text{closed} & \quad \xrightleftharpoons[k_{cl}]{} \quad \text{open} \\
\Rightarrow & \quad \text{labeled}
\end{align*}
\]

The observed rate is given by:

\[
k_{obs} = \frac{k_{\text{op}}k_{\text{int}}[\text{DTNB}]}{k_{\text{op}} + k_{\text{cl}} + k_{\text{int}}[\text{DTNB}]}
\] (3.4)

which is a nonlinear function that approaches a linear dependence on [DTNB] at low concentrations and [DTNB] independence at high concentrations. In the limiting case where \( k_{\text{cl}} \ll k_{\text{int}}[\text{DTNB}] \) called the EX1 regime, the observed rate of labeling reduces to

\[
k_{\text{obs}}^{(\text{EX1})} = k_{\text{op}} \quad (k_{\text{cl}} \ll k_{\text{int}}[\text{DTNB}])
\] (3.5)

In the limiting case where \( k_{\text{cl}} \gg k_{\text{int}}[\text{DTNB}] \) called the EX2 regime, the observed rate of labeling reduces to

\[
k_{\text{obs}}^{(\text{EX2})} = \frac{k_{\text{op}}}{k_{\text{cl}}} k_{\text{int}}[\text{DTNB}] = K k_{\text{int}}[\text{DTNB}] \quad (k_{\text{cl}} \gg k_{\text{int}}[\text{DTNB}])
\] (3.6)
where $K$ is the equilibrium constant between the open and closed forms. In the intermediate regime where $k_{cl} \approx k_{int}[\text{DTNB}]$, called the EXX regime, one must fit to the full expression (given in equation 3.3). We found that over the concentrations of DTNB used, TEM-1 S243C labeling was in the EX2 regime (linear dependence on [DTNB]) and CTX-M-9 labeling was in the EXX regime (nonlinear dependence on [DTNB]).

The three regimes differ in terms of the controls required to demonstrate that labeling is not occurring predominantly in the unfolded state. In the EX1 regime, the observed labeling rate for a pocket must be faster than the rate of global unfolding. Neither of the pockets we test in this paper labeled in this regime, but we have previously observed labeling rates in pockets that showed this behavior [55]. In the EX2 regime, the equilibrium constant for pocket opening must be greater than that for the unfolded state (Equation 3.5). To determine these quantities for TEM-1 S243C, we measured the $K$ of unfolding (Table A.1) and the intrinsic rate of labeling for the denatured protein.

To determine the intrinsic rate of labeling, $k_{int}$, our labeling assay was repeated with the addition of 6 M urea. In the EXX regime, the observed labeling rate for a pocket must be greater than the maximum expected labeling rates of the unfolded state in either the EX1 or EX2 regimes (derivation in SI). Thus, to test that CTX-M-9’s labeling rate is not consistent with global unfolding alone, we measured both its rate of unfolding (Fig. A.4) and its thermodynamic stability (Table A.2). We then combined that with the fit value of $k_{int}$ (Table A.1) to produce a piecewise function that is an upper bound for labeling from the unfolded state. In this case, however, we found that the unfolding rate is the relevant control for all DTNB concentrations used here—the population of unfolded enzyme is relevant only at DTNB concentrations less than about 170 nM, the DTNB concentration where $k_{cl} = k_{int}$.
3.3.8 Urea melts and unfolding kinetics

Equilibrium stabilities and unfolding kinetics were acquired on a Chirascan circular dichroism spectrometer (Applied Photophysics) at a temperature of 25 °C. Protein denaturation was observed by measuring the average ellipticity over 60 s at 222 nm as a function of urea concentration (Fig. A.5, Table A.2). Samples of 35 µg ml$^{-1}$ protein were equilibrated in 50 mM potassium phosphate pH 7 and varying concentrations of urea overnight prior to data collection.

To determine the global unfolding rate, we used a linear extrapolation model [172] fit the log observed unfolding rates as a function of urea concentration at concentrations above the concentration at which it is half folded and half unfolded (the $C_m$, Table A.2) and extrapolated back to 0 M urea (Fig. A.4). Concentrations were between 4 and 5.5 M urea for TEM-1 M182T and between 1.8 and 2.8 M urea for CTX-M-9.

3.3.9 Activity measurements

Activity measurements were performed on both labeled and unlabeled protein. In order to measure the activities of the labeled proteins, 10 µM S243C and 5 µM CTX-M-9 were each incubated with excess DTNB for one hour, giving ample time for both proteins to fully label prior to the activity measurements. The proteins were then separated from excess DTNB using size exclusion chromatography.

Enzyme activities against nitrocefin (Cayman Chemical Company) were monitored at 482 nm ($\epsilon_{482} = 15,000$ cm$^{-1}$) using a Cary 100 UV-vis spectrophotometer (Agilent Technologies). Reactions were measured in 50 mM potassium phosphate, 10% glycerol (v:v), 2% DMSO pH 7.0 at 25 degreeCelsius using 2 nM enzyme. Initial velocities were plotted as a function of
nitrocefin concentration and fit to a Michaelis–Menten model to extract $k_{\text{cat}}$ and $K_m$ values (Fig. A.6, Table A.3).

### 3.3.10 Visualizations

Protein structures were visualized using PyMOL 2.2 [173]. Graphs were embedded with Fruchterman-Reingold algorithm [174] in NetworkX [168].

### 3.3.11 Code availability

Library code is available on GitHub as bowman-lab/enspara [91]. MSM weights and state representative structures, along with a Jupyter notebook demonstrating the analysis described in this chapter, can be found at https://wustl.box.com/v/2018-exposons.

### 3.4 Results

#### 3.4.1 Exposons simultaneously capture conformational changes and allostERIC communication at protein surfaces

To identify exposons, we first construct an MSM (see Methods). In this work, we defined states using the Euclidean distance between vectors of sidechain solvent accessible surface areas, but in general, other methods can also be used. Each state in the MSM is then represented by a binary vector that characterizes the solvent exposure of each residue in the cluster center for that state (i.e. element $i$ is zero if residue $i$ is buried or one if it is exposed) (Fig. 3.1a). Based on previous work [55], sidechains are classified as exposed if
Figure 3.1: A schematic outline of our method for identifying exposons. 

a, A Markov state model composed of variably populated states (circles, population is indicated by circle diameter) and transitions between states (single-headed arrows, probability is indicated by arrow length). Each state is associated with a binary exposed/buried classification for each residue, indicating whether the residue is exposed or buried in that state (column of black and white boxes, white denoting buried and black denoting exposed). 

b, An all-against-all pairwise mutual information (MI) matrix that is calculated from (a). Exposons, indicated by the colored groups in the margins, are clusters of residues with mutually high pairwise mutual information. 

c, The residue-level network representation of a mutual information matrix. Residues are indicated by double-edge circles, and are colored by their exposon membership. Exposons are indicated by dashed-lined circles. The mutual information between residues is indicated by straight lines between double-edged circles, and the weight of the line represents the strength of the correlation.
their surface area exceeds 2 Å. We then compute the mutual information (MI) between each pair of residues, giving a square MI matrix (Fig. 3.1b). Mutual information—defined in Eq. 3.1—is a nonlinear measure of the statistical interdependence of two random variables that has been previously used in studies of protein allostery \cite{52, 53}. A particularly useful property of the mutual information is that residues that never change their solvent exposure (i.e. have zero entropy) have zero mutual information. Finally, we cluster this mutual information matrix using affinity propagation \cite{167} to assign residues to exposons. The resulting mutual information matrix can be visualized as a network with nodes colored according to their exposon assignment (Fig. 3.1c). The list of exposons can be prioritized for further analysis based on the total information of each exposon, which is the sum of the non-diagonal elements of each exposon’s row in the coarse-grained exposon MI matrix \cite{53}. This tends to identify larger exposons with more communication, which we reasoned are more likely to be functionally relevant and less susceptible to noise and errors introduced by finite sampling.

Once exposons have been computed, we typically wish to identify—in terms of atomic coordinates and protein conformations—which motion or motions give rise to an exposon. MSMs, which contain kinetic as well as thermodynamic information, are a natural source of this information. An MSM’s top eigenmodes capture how much each conformational state participates in the slowest motions observed in a simulation. In each eigenmode, each state is assigned a value in the interval \([-1, 1]\). The absolute value of each assignment represents the degree of participation of the state in the eigenvector, and the sign is arbitrary but groups states into opposite ends of the motion. That is, states with low values in the eigenmode are slowly interconverting with states with high values of the eigenmode. Therefore, we reasoned that an MSM’s top eigenmodes provide a facile means to identify the dominant motions contributing to an exposon \cite{23}. To identify which eigenmode reports on a particular
exposon, we first compute the degree to which changes in an exposon’s solvent exposure are correlated with each eigenmotion. Specifically, for residue pair, we choose the eigenvector that maximizes the Pearson’s R correlation coefficient between the eigenvector’s component and a vector of solvent accessibilities for each state. More formally, we compute,

$$\arg \max_j R(\nu_j, S_i)$$

(3.7)

where $R$ is the function that calculates Pearson’s correlation coefficient, $\nu$ is the $m \times m$ matrix of the eigenvectors of the transition probability matrix, and $S$ is the $n \times m$ matrix of state exposures for an MSM with $m$ states of a protein with $n$ residues. This approach is similar to dynamical fingerprinting [96]. We then choose the eigenmode that maximizes this correlation and extract the structures of the conformers at the extremes of this motion.

This conceptual framework has several important advantages over more traditional geometric approaches. First, it does not make any assumptions about which types of surfaces are most interesting—instead, any surfacial rearrangement that shows cooperativity will be detected. Second, this approach explicitly considers the entire sampled ensemble and uses this information to prioritize the most interesting features of the ensemble, rather than relying on structural features of particular conformers. Third, because exposons exist in sequence space, the results are insensitive to structural alignments and can be easily compared with experimental techniques that provide a read-out at the primary structural level, including thiol labeling. Consequently, this tool is applicable to a wide variety of conformational changes and scientific questions, as we demonstrate below.

Our approach is predicated on the assumption that interesting features are those that change at the surface. Our motivation for this assumption was that most interesting protein behavior ultimately is a consequence of the protein’s ability to interact with other objects, which occurs
at the surface. Any cooperative rearrangement that does not substantially alter a protein’s solvent exposure will not be detected. For instance, a rotameric transition that creates geometry necessary for catalysis may not entail any change in solvent exposure. Likewise, any cooperative changes occurring exclusively in the protein core will not be detected. Allosteric coupling between two surface sites that occurs through the core will be detected, but the mechanism will not be apparent since exposons will only be sensitive to the end-points. Another potential limitation of our approach is imposed by the use of mutual information, which is only sensitive to features that change. A concavity at a protein surface that never changes its conformation will not be identified by this method—this situation is much better suited to the many excellent geometrical pocket detection methods proposed over the years [58–60, 175].

3.4.2 Retrodiction of a cryptic allosteric site in TEM-1

As a first test of our model, we examined its ability to identify cryptic allosteric sites. A cryptic allosteric site is a pocket that is absent in available structures but is present in excited states and can exert allosteric control over a distant functional site, such as an enzyme’s active site. Cryptic pockets are a particularly interesting class of excited states because identifying new cryptic sites could offer new druggable sites on established drug targets, provide a means to inhibit targets that are currently considered undruggable, or even enable the enhancement of desirable activities [175, 178]. Therefore, a systematic means to identify functionally-relevant conformational transitions to excited states in the absence of stabilizing interactions could provide biophysical insight and new therapeutic opportunities. We expect the formation of a cryptic pocket to result in an exposon because, as explained above, the opening and closing of a pocket should result in cooperative changes in the solvent exposure of surrounding residues. Furthermore, for a cryptic allosteric site, we expect the allosteric
Figure 3.2: Exposons for TEM-1 β-lactamase. a, The coarse-grained exposon network for TEM-1. Edge weights are proportional to the total correlation between each pair of exposons and node sizes are proportional to their eigenvector centrality. Self-edges and edges with very low values are omitted. b, The highest total information exposons superimposed on a crystal model of unliganded TEM-1, 1JWP [176]. Residue colors match the exposon colors in (a). Note the spatially contiguous exposons centered about the active site (dark blue), Horn allosteric site (light blue), a previously identified cryptic pocket (beige), and the Ω-loop (light and dark green). c, A representative structure of the open state from the light blue exposon (teal) overlaid on a ligand-bound crystal structure of the Horn allosteric site (grey, 1PZO) [177].
coupling to give rise to correlations between the pocket exposon and residues around the relevant functional site.

We chose to test our approach on the enzyme TEM-1 β-lactamase because it is known to contain several cryptic allosteric sites [55, 99]. It is also an important source of antibiotic resistance, so new inhibitors could provide a valuable means to restore the efficacy of existing antibiotics. In pursuit of new inhibitors, allosteric modulators have been discovered for at least one of these sites, which is created when a short alpha helix undocks from the protein, exposing a ligand binding site [55, 56, 177]. To distinguish this site from other putative allosteric sites on this protein, we will refer to this site as the Horn pocket, or the Horn allosteric site, after the author who first reported this pocket [177].

As expected, we identify exposons corresponding to known cryptic pockets in TEM-1 β-lactamase (Fig. 3.2a). To visualize this, we mapped high total information exposons onto a crystal model of the TEM-1 ground state (Fig. 3.2b). In this format we observe a small number of spatially-condensed clusters of residues that are distant in sequence space, recapitulating our expectation that spatially (but not necessarily sequentially adjacent) objects are more likely to act cooperatively. The exposon with the highest total information (dark blue) corresponds to the active site. The exposon with the second-highest total information (light blue) corresponds to the Horn site [99, 177], shown in Fig. 3.2c, grey structure. Yet another exposon (beige, 4th highest total information) reports on a second cryptic pocket that we reported previously [99]. Each of the exposons corresponding to a cryptic pocket has substantial inter-exposon communication (i.e. at least 90th percentile of all exposon-exposon edges) with the active site, suggesting the potential for perturbations to these pockets to exert allosteric control over activity.

To assess the effectiveness of using an MSMs’ eigenmodes to identify the motions that induce
a particular exposon, we compare the structures identified in this way with known crystal models for the relevant ligand-bound state. In the case of the Horn cryptic allosteric site, a crystallographic model for the ligand-bound, open state is available [177]. We then compare this structure with the structures at the extremes of the eigenmotion that best correlates with this exposon’s exposure state, as described in the Methods section. In this case, one extreme of the configuration resembles the ligand-free crystal structure, and the other is similar to the bound crystal structure (Fig. 3.1a, teal structure). The fact that the open structure from our model is somewhat more open than the ligand-bound structure is consistent with previous evidence that the pocket opens even further in solution than is seen in the crystal structure [99].

As an even more stringent test of our model, we assessed the consistency of our model’s predictions with an *in vitro* measurement of the kinetics of solvent exposure. Specifically, we used a thiol-labeling approach which we have improved from our previous work [55] by the use of a stopped-flow instrument (see Methods). In brief, this assay uses a drug-sized labeling reagent, DTNB (Ellman’s reagent) that changes absorbance upon covalently reacting with solvent-exposed reduced cysteines, providing a time-resolved measurement of residue-level solvent exposure with millisecond resolution. If a cysteine is not natively present at a position of interest, then one can be introduced via mutation (see Methods). To make the comparison between our MSM and our thiol labeling experiment, we also developed a method for labeling rate predictions (see Methods) that gives, as a function of time, the fraction of the population that has ever exposed the relevant sidechain to solvent.

As predicted, experimentally-confirmed pocket positions (S203, A232, L286) expose at intermediate rates in our model (Fig. A.7). Furthermore, positions that do not label in our experiments (L190, I260) remain buried in our simulations. Similarly, a surface control (A150) labels immediately in our experiments and never buries in our simulations. Addi-
tionally, rank is preserved: residues that label faster in vitro also label faster in silico. The main discrepancy between predicted and experimental labeling occurs at S249, which labels very slowly in vitro but does not label in silico, likely because finite sampling prevented us from ever observing the slow process that leads to exposure of this residue. It is worth noting, however, this residue is located just “beneath” (i.e. deeper toward the core of the protein) an exposon that reports on cryptic pocket opening at this position (Fig. 3.2b, beige), suggesting that the exposon analysis may be somewhat robust to sampling error. As discussed previously [55], the fact that we can place cysteines at positions where they remain buried is reassuring that we have not introduced pockets where they did not exist before. Furthermore, the strong correlation between our predicted and observed labeling rates supports the conclusion that we have not erroneously created pockets. In the future, a more precise understanding of the labeling reaction’s geometric requirements could enable quantitative predictions of labeling rates. For now, our model’s ability to correctly order pocket opening rates demonstrates its utility for identifying and characterizing pockets.

3.4.3 Retrodiction of a conformational switch in nucleoprotein

As a subsequent test of our model, we assessed its ability to retrodict a conformational switch that was previously identified by Su et al [179]. Proteins must frequently act as switches, altering their behavior in response to some signal. For example, many signaling proteins undergo conformational changes in response to specific stimuli that either increase or decrease their propensity to interact with downstream binding partners. We expect these concerted changes to manifest as exposons in our analysis.

As a test of the hypothesis that functional conformational switches at protein surfaces induce exposons, we analyzed Ebolavirus’s nucleoprotein (eNP), a conformational switch that
Figure 3.3: Exposons for eNP. 

a The distribution of the highest total information exposons superimposed upon a crystal model of monomeric eNP [179]. Note the spatially non-contiguous nature of some of the exposons, especially the dark blue exposon.

b The coarse-grained exposon network of eNP. Node colors match exposon colors in (a). Edge weights are proportional to the correlation between each pair of exposons and node sizes are proportional to their eigenvector centrality. Self-edges and edges with very low value are omitted.

c The extremes of the eigenmotion best correlating with the highest total information exposon (blue exposon in a).
controls access to and replication of the viral genome. Understanding and manipulating this conformational switch is of interest because Ebola was the causative agent in several recent, high case-fatality epidemics in sub-Saharan Africa [180] and is a pathogen for which very limited treatment options are available. Therefore, an improved biophysical understanding of this virus's lifecycle may prove useful in understanding how to therapeutically target it.

In one state, eNP oligomerizes and encapsidates the viral genome to package it for transport and protect it from degradation [181]. In a second state, eNP exists as a monomer, releasing RNA to allow transcription of the viral genome [181]. Recent evidence suggests that oligomerization is controlled by the curling of C-terminal helices of eNP into the RNA-binding cleft [179]. We then expect an exposon to be formed by the residues in this groove and by the residues in the C-terminal tail that transiently occupies it.

Consistent with our expectation that the surficial rearrangements required for eNP function result in exposon formation, the highest total information exposon in eNP (Fig. 3.3a, dark blue) spans the residues in the C-terminal polymerization domain and the RNA binding groove. This is an interesting case in which an exposon is not be predicted to be composed of residues that are spatially contiguous when mapped to a crystal model of the ground state. This exposon is also at the center of the network of exposons (Fig. 3.3b). Extracting the motion that induces this exposon, shown in Fig. 3.3c, reveals that this exposon reports on the very same collective curling of the terminal helices into the RNA-binding cleft identified previously [182]. Crucially, this dynamic process is consistent with hydrogen-deuterium exchange data that cannot be accounted for using available cryoelectron microscopy models [179]. Manipulating this conformational equilibrium with small molecules or peptides could provide a powerful means of modulating the Ebola lifecycle. Indeed, a peptide that binds this interface has already been found to inhibit viral replication [182].
3.4.4 Retrodiction of allosteric coupling between domains in CAP

As a further test of our model, we investigated its capacity to identify allosteric coupling between binding sites. Wherever an element of conformational selection is present, a binding site will sample both its bound and unbound configurations, and whenever the bound and unbound configurations differ in their pattern of solvent exposure, an exposon is expected to form. Because bound and unbound configurations presumably expose a different pattern of surface chemistry—one association-compatible and the other association-incompatible—we expect that differing patterns of solvent exposure might be a near-rerequirement. Furthermore, if these sites are allosterically coupled, they may even cluster into the same exposon.

Catabolite activator protein (CAP) is a homodimeric transcriptional activator in *E. coli* that allosterically couples cAMP binding to sequence-specific DNA association [184]. This allosteric coupling between the cAMP binding domains (CBDs) and DNA binding domains (DBDs) is realized by a dramatic swiveling motion of the DBDs [183, 185], which changes the pattern of solvent accessibility on both the CBDs and DBDs, potentially producing one or more exposons. Besides coupling between the CBDs and DBDs [186], CAP also exhibits strong negative cooperativity between the two cAMP binding sites [187]. Since these binding sites show different solvent exposure in cAMP-free and doubly cAMP-liganded crystal models, we expect these sites induce exposons as well. Since previous computational work on this protein suggests that evidence of this coupling is present in equilibrium simulations of the unliganded state [53], we expect to observe exposons that encompass residues in these regions.

As expected, the two highest total-information exposons computed from simulations of CAP in the unliganded state (Fig. 3.4a) are a symmetric pair stretching from the cAMP binding site in each monomer’s CBD to both DBDs. There is very strong communication between
Figure 3.4: Exposons for CAP. 

(a). The distribution of the highest total information exposons superimposed upon a crystal model of unliganded, dimeric CAP [183]. Purple circles indicate cAMP binding sites, and the cAMP-binding domains (CBDs) and DNA-binding domains (DBDs) are labeled. 

(b). The coarse-grained exposon network of CAP in graph form. Node colors match exposon colors in (a). Edge weights are proportional to the correlation between each pair of exposons and node sizes are proportional to their eigenvector centrality. Self-edges and edges with very low value are omitted.
these two exposons, consistent with the negative cooperativity between the CBDs [184, 186].

The third-highest total information pair of exposons (Fig. 3.4a beige and orange) is centered about the individual cAMP binding sites, and they show less communication with one another than the larger DBD/CBD exposons (Fig. 3.4b). One explanation for the fact that these sites cluster separately from the rest of the cAMP binding site is that they are responsible primarily for substrate recognition, rather than allostery. The two highest total-information residues in this exposon, Q80 and R82, are two of the only four residues in the cAMP binding cassette that reduce their dynamicity upon binding [188]—opposite to the trend of the rest of the molecule and opposite to the hypothesized entropy-driven mechanism of allostery in this system. Furthermore, R82 is predicted to form a salt bridge with the cAMP phosphate and its mutation strongly affects binding [189].

To understand the motions that create the larger exposons reporting on interdomain and intermolecular allostery in this system, we examined the eigenmotion that best correlates with the highest total-information exposon we identified in this system (dark blue in Fig. 3.4a). Two extreme states of this eigenmotion indicates that this exposon represents a see-saw motion of the DBDs coupled to the closing of one cAMP site and the opening of the other. This is consistent with structural evidence [185] that the coupling between CBD and DBD involves large, rigid-body displacements of the two DBDs. This immediately suggests a testable hypothesis for how the negative coupling between cAMP binding sites might be achieved. This hypothesis could be further refined and dissected using methods like CARDS, as we have done previously [53], or experimental methods.
Figure 3.5: Eigenvector centrality of exposons for TEM-1, eNP, and CAP. Exposons are numbered from highest to lowest total information. In each case, the central exposon or exposons are associated with the primary function of the molecule they are found in. a In TEM-1, the most central exposon is at the active site. b In eNP, the central exposon is associated with a curling motion crucial to protein function (Fig. 3.3c). c In CAP, the pair of central exposons report on allosteric coupling between the DBDs and CBDs.

3.4.5 Functional sites are exposon graph hubs

Exposons are a network model and consequently provide facile access to a protein’s allosteric topology. Because we have segmented the sequence into disjoint sets, this allows us to coarse grain our original mutual information matrix—which represents the sparse communication graph between all residues—into a much smaller graph representing communication between exposons. To calculate the communication between two exposons, we simply sum all edges that begin in one exposon and end in the other, and normalize by the channel capacity [53]. The channel capacity is a measure of the maximum information that could possibly be transmitted between exposons, given the number of nodes they each contain (see Methods). Normalizing by this quantity allows for an intuitive comparison between the strength of communication between different pairs of exposons.

All exposon networks we examined had a hub-and-spoke architecture, with the exposon(s) with the highest total information serving as a hub and having a clear functional role.
In TEM-1, the active site exposon (colored dark blue in Fig. 3.2b), including the active site serine, is visually central to the exposon graph (Fig. 3.5a), and each other node has its strongest connection with this node. We formalize this intuition by calculating each exposon’s eigenvector centrality (Fig. 3.5a–c) [190]. Eigenvector centrality is a measure of the amount of time a random walker would spend at a particular node if transitions between nodes were distributed according to edge weights. Hence, nodes with higher-weighted or more connections to other nodes have a higher eigenvector centrality. In this case, we also find two groups of exposons attached to the hub but that are relatively uncorrelated with each other. Interestingly, one is a set of exposons that are under and around the Ω-loop, which is a critical modulator of substrate specificity and activity. In eNP, we also found that the exposon with the highest total information is a hub (Fig. 3.5b). As discussed previously, this exposon captures the dramatic curling motion that has been proposed to mediate RNA binding [179]. In CAP, we find that the two exposons with the highest total information both have high centrality (Fig. 3.5c). These exposons appear to couple the ligand and DNA-binding domains of that protein.

The fact that functionally-relevant conformational changes result in exposons with high total information and high centrality in three completely unrelated proteins furnishing wildly different functions is consistent with our motivating hypothesis that cooperativity does not arise at random.

### 3.4.6 Discovery of the first known cryptic allosteric site in CTX-M-9

To demonstrate how the exposon model can be used to generate hypotheses and design experiments, we applied it to predict cryptic pockets in the enzyme CTX-M-9 β-lactamase.
Figure 3.6: Exposons identify novel cryptic pockets in the CTX-M and TEM β-lactamases.

a The extremes of the eigenmotion for the exposon containing C69 identify closed (left) and open (right) conformations of a cryptic pocket under the Ω-loop. Residues within 7 Å of C69 are shown as spheres and residues participating in the same exposon as C69 are shown as red sticks. Residue C69 is colored in yellow. 

b The observed labeling rates (solid green line) are in the EXX regime. The labeling rates expected for the global unfolding process (dashed line) are much slower. 

c The extremes of the eigenmotion best correlating with the exposon containing S243 identifies closed (left) and open (right) conformations of a cryptic pocket under and behind the Ω-loop in TEM-1. Residues within 7 Å of S243 are shown in spheres and residues participating in the same exposon as S243 are colored in dark green and as sticks. Residue S243 is colored in yellow. 

d The observed labeling rates (solid blue line) of a cysteine introduced at position 243 are in the EX1 regime. The labeling rates expected for the global unfolding process (dashed line) are much slower. In (b) and (d), standard deviations across three experiments were on the order of $10^{-5}$ and $10^{-4}$, respectively, and are not included for visual clarity.
CTX-M-9 is interesting because, to the best of our knowledge, no cryptic pockets have been reported in this protein. It has less than 40% sequence identity with TEM-1, so it is not obvious whether or not it is likely to have similar cryptic pockets.

Examining the exposons for CTX-M-9 revealed that one of them contains the protein’s single native cysteine, C69 (Fig. 3.6a, yellow). This cysteine is completely buried in the apo crystal structure. Examining the motion that gives rise to this exposon reveals that C69 is exposed to solvent by a displacement of the Ω-loop (Fig. 3.6a), a structural element conserved among many β-lactamases and containing residues absolutely required for enzymatic activity [191] and that has significant conformational heterogeneity [47]. The open structure of this pocket appears to be well-structured, as opposed to disordered, making it a potentially viable drug target. Therefore, we expect a small molecule that binds this pocket and displaces the Ω-loop would be a potent inhibitor while a drug that stabilizes the closed conformation would increase activity. The exposure of C69 in particular is of great interest because our thiol labeling assay can be applied without having to introduce a cysteine. Therefore, unlike previous applications of this method, there is no concern that the introduction of a cysteine created a pocket where none existed before.

We examined the labeling of C69 using our thiol labeling assay. The single exponential labeling that we observe is consistent with our prediction that C69 lines the first cryptic pocket to be identified in CTX-M-9 (Fig. 3.6b). C69’s labeling rate is much faster than the rate of the global unfolding process (Fig. 3.6b, dashed lines) measured by circular dichroism (Fig. A.5), supporting our prediction that it is exposed by a fluctuation within the native state.

To ensure that exposon participation is a bona fide signal of pocket formation, we assayed the labeling rate of a residue that is buried in the crystal model but does not participate
in an exposon, S123. Therefore, according to our model, a cysteine at this position should not show labeling. Consistent with this prediction, the S123C variant of CTX-M-9 does not show significant labeling.

To assess the allosteric potency of this site, we also measure the catalytic efficiency of the label-conjugated enzyme. In this case, after incubating CTX-M-9 with DTNB, which TNB-labels C69, we measure the rate at which it degrades nitrocefin, a $\beta$-lactam substrate (Fig. A.7). TNB conjugation acts as a proxy for the binding of a drug. However, owing of TNB’s small size and hydrophilicity, this assay could easily underestimate the effect a true drug could have. We found an approximately 15-fold reduction in the catalytic efficiency (Fig. A.7). By comparison, this same assay applied to previously-identified cryptic pockets in TEM-1 showed a less than threefold change in activity [55], making our newly identified site the most potent site in either TEM-1 or CTX-M-9.

Taken together, this newly-predicted pocket is the most attractive cryptic drug target site found to date in either TEM-1 or CTX-M-9. The fact that no mutation was required to perform thiol labeling of C69 also makes the results presented here some of the most compelling support for the predictive power of exposons in particular and MSMs in general.

### 3.4.7 Prediction of a novel cryptic allosteric site in TEM-1

In light of our results for CTX-M-9, we examined the exposon graph for TEM-1 to see if a similar cryptic pocket may arise due to a displacement of the $\Omega$-loop. Since TEM-1 has been extensively studied for the purpose of identifying cryptic allosteric sites, discovery of a new pocket in this molecule would be strong evidence for the utility of exposons for pocket discovery.
Two exposons (Fig. 3.2b, dark and light green), showing strong communication with one another, map onto the $\Omega$-loop. The best-correlating MSM eigenmode revealed that S243 is significantly exposed by the opening of this pocket (Fig. 3.6c), and that the open form also appears well-structured and druggable. This conclusion is supported by quantitative druggability scores from fpocket [192] (Fig. A.8). Interestingly, in our previous work, we were unable to detect this pocket because it frequently forms a channel-like connection with the active site, causing it to be combined with the active site pocket by pocket clustering methods [55, 99].

As position 243’s participation in an exposon predicts, the S243C variant labels at an intermediate rate that is slower than the near-instant labeling of a surface residue but substantially faster than the global unfolding process (Fig. 3.6d), which is on the order of hours [55]. Once again, we also measured the catalytic function of the TNB-labeled enzyme. Somewhat surprisingly, the TNB adduct had a 3.75-fold increase in catalytic efficiency—the ratio of $k_{\text{cat}}$ over $K_m$—driven primarily by a ~4-fold decrease in $K_m$ (Fig. A.6, Table A.3). This is consistent with recent evidence suggesting that both activation and inhibition are possible at the same allosteric site [56], and suggests that TNB may pack into the $\Omega$-loop in such a way as to stabilize the closed conformation. Examination of crystal models of the closed state [176] reveals a void under the $\Omega$-loop into which TNB might plausibly pack.

The fact that exposons identify a new cryptic allosteric site even in TEM-1—a protein that has been studied for many years by many groups, including intensively by our group with the specific goal of locating these sites—highlights the value of our approach for identifying functionally-relevant conformational changes. It also supports the hypothesis that the paucity of known cryptic allosteric pockets may stem more from technical limitations in locating them than from a low prevalence.
3.5 Conclusion

We have demonstrated that exposons provide a powerful conceptual framework for identifying functionally-relevant conformational transitions. Exposons retrodict cryptic pockets, retrospectively identify conformational switches, and identify allosteric coupling between domains. We also showed that exposons are able to make *bona fide* predictions by discovering two new cryptic allosteric sites and experimentally verifying their existence. One of these sites is in a protein, CTX-M β-lactamase, that was not known to have any cryptic pockets, and in which no mutations were required to experimentally test our prediction. The other is in an enzyme that has been the target of an extensive search for cryptic pockets, so discovering a new site is a surprising testament to the power of exposons. Taken together, these results are compelling evidence for the utility of exposons.

Because many proteins’ most biologically interesting behavior involves changes at their surfaces, we expect our methodology to serve as a powerful first step in the analysis pipeline for proteins with complex, allosteric functions. Our results applying exposons to cryptic pockets, for example, demonstrates this method’s potential as the first step of a drug development pipeline targeting cryptic sites. Since the motions giving rise to exposons are substantially more diverse than simply pocket formation, exposons may also serve as a nearly automatic, high-throughput mechanism for dissecting allostery at protein surfaces either to refine an existing hypothesis or to identify potential alternative hypotheses.

Finally, the apparent ubiquity of the centrality of important functional surfaces in informational graphs suggests graphs for all four of the systems studied in this work is provocative. It may be, for example, that a general feature of protein evolution creates this behavior: that genetic drift destroys functionless cooperativity, or that the allostery incurs a thermodynamic penalty and is hence selected against. It remains to be seen, however, if this is
a general physical or biological principle in the organization of proteins, or if this finding generalizes to proteins of other sizes and with other functions. Whatever the case, exposons’ value for rapidly analyzing conformational ensembles is clear, and we expect this method may have the capacity to detect even larger allosteric changes, such as folding-upon-binding events.
Chapter 4

Isolated myosin motor domains
conformational ensembles encode
their mechanochemical properties

Shut up and calculate!

David Mermin

The work presented in this chapter is, at time of writing, under review as a peer-reviewed publication.

4.1 Abstract

Myosin motor domains perform an extraordinary diversity of biological functions despite sharing a common mechanochemical cycle. Motors are adapted to their function, in part, by
tuning the thermodynamics and kinetics of steps in this cycle. However, it remains unclear how sequence encodes these differences, since biochemically distinct motors often have nearly indistinguishable crystal structures. We hypothesized that sequences produce distinct biochemical phenotypes by modulating the relative probabilities of an ensemble of conformations primed for different functional roles. To test this hypothesis, we modeled the distribution of conformations for twelve myosin motor domains by building Markov state models (MSMs) from an unprecedented two milliseconds of all-atom, explicit-solvent molecular dynamics simulations. Comparing motors reveals shifts in the balance between nucleotide-favorable and nucleotide-unfavorable P-loop conformations that predict experimentally-measured duty ratios and ADP release rates better than sequence or individual structures. This result demonstrates the power of an ensemble perspective for interrogating sequence-function relationships.

4.2 Introduction

Myosin motors (Figure 4.1A) perform an extraordinary diversity of biological functions despite sharing a common mechanochemical cycle. For example, myosin-II motors power muscle contraction, whereas myosin-V motors engage in intracellular transport. This diversity is in part due to differences in myosins’ tails and light chain-binding domains, which influence properties like localization and multimerization [64]. However, some of this diversity is encoded in the motor domains themselves [71]. These differences stem from variations in the tunings of the thermodynamics and kinetics of the individual steps of the myosins’ conserved mechanochemical cycle, which couples ATP hydrolysis to actin binding and the swing of a lever arm [72].

Two important and highly variable parameters for motor function are the rate of ADP release,
which sets the speed of movement along actin, and the duty ratio, which is the fraction of
time a myosin spends attached to actin during one full pass through its mechanochemical
cycle. For example, in muscle, myosin-II motors are arranged into multimeric arrays called
thick filaments and the individual motors typically have a strong preference for the actin free
state (i.e., low duty ratio). These motors quickly detach after pulling on the actin filament
to avoid creating drag for other motors in the array, much as a rower quickly removes their
oar from the water to minimize drag. In contrast, individual myosin-Va motors have high
duty ratios (i.e. prefer the actin-bound state), helping them to processively walk along actin
filaments in intracellular transport. Similarly, the speed of myosin movement along actin
(in the absence of opposing forces) is set by the rate of ADP dissociation [72], and it varies
by four orders of magnitude from \(~0.4\ \text{s}^{-1}\) for non-muscle myosin-IIb [193] to \(>2800\ \text{s}^{-1}\) for
myosin-XI [194].

Unfortunately, inferring the relationship between a motor’s sequence and its biochemical
properties is not trivial. For example, one cannot simply predict the duty ratio or ADP
release rate of a motor based on phylogeny. Myosin-V family members contain both high
duty ratio motors, like myosin-Va, [195] and low duty ratio motors, like myosin-Vc [196].
Similarly, ADP release rates within the myosin-II family vary from \(~0.4\ \text{s}^{-1}\) (non-muscle
myosin-IIb) [193] to \(>400\ \text{s}^{-1}\) (extraocular myosin-II) [67, 197]. Insertions and deletions in
the myosin motor domain sequence also convey useful, but typically incomplete, information.
For instance, pioneering biochemical work [198][198] demonstrated a correlation between the
length of loop 1 and ADP release rates in myosin-II motors. However, this observation does
not explain how other myosin isoforms that have virtually the same loop 1 lengths have
ADP release rates that differ by an order of magnitude [69]. It is also difficult to predict the
effects of mutations implicated in human disease, as the effects cannot be easily predicted
from the location of the mutation. For example, in human $\beta$-cardiac myosin, an A223T
mutation causes a dilated cardiomyopathy [199] while an I263T mutation has the opposite effect, resulting in a hypertrophic cardiomyopathy [200], despite being separated by less than 6 Å [201].

Structural studies have provided detailed pictures of many key states in the mechanochemical cycle, but have yet to enable the routine prediction of a motor’s biochemical properties from its sequence. For example, high-resolution structures have illuminated many shared features of myosin motor domains, such as the lever arm swing [202] and conformational rearrangements associated with changes in nucleotide binding [203, 204]. They have also revealed the strain-sensing elements of myosin-I motors [205–207] and the binding modes of many small molecules [201, 208, 209]. However, the structures of motor domains with vastly different biochemical properties are often nearly indistinguishable. Similarly, computer simulations have begun to reveal aspects of motor function [73–75, 210]. However, simulating an individual motor domain (~700 residues) is a huge computational expense, so most simulation studies have been based on less than a microsecond of data. Thus, adding binding partners like actin to simulate the full mechanochemical cycle and infer properties like duty ratio is currently infeasible, especially if one wanted to compare multiple isoforms to infer sequence-function relationships.

Here, we investigate the possibility that the distribution of structures that an isolated motor domain explores correlates with its biochemical properties, allowing the prediction of sequence-function relationships. This hypothesis was inspired by a growing body of work showing that protein dynamics encode function [3, 211], even in the absence of relevant binding partners [47, 126, 212]. In the case of myosin, we reasoned that as sequence changes modulate motors’ preferences for different states of the mechanochemical cycle, they likely also have a systematic effect on the distribution of conformations explored by the motor, even in the absence of binding partners. Therefore, comparing the distribution of confor-
mations that isolated motor domains sample in solution should reveal signatures of their biochemical differences.

To test this hypothesis, we ran an unprecedented two milliseconds of all-atom, explicit solvent molecular dynamics (MD) simulations of twelve myosin motors with diverse but well-established biochemical properties (Figure 4.1B, Table S1 and S2). Such simulations are adept at identifying excited states, which are lower-probability conformational states that are often invisible to other structural techniques. Indeed, our simulations reveal a surprising degree of conformational heterogeneity, particularly in the highly conserved P-loop (or Walker A motif), a common structural element for nucleotide binding that is highly conserved across myosin motor domains [213]. Because of its high conservation, we reasoned that the P-loop would report on the conformation of the nucleotide binding site while still being comparable between motors with otherwise differing sequences. To enable quantitative comparisons, we constructed Markov state models (MSMs) from the MD data for each motor. MSMs are network models of protein free energy landscapes composed of many conformational states and the probabilities of transitioning between these states. They are a powerful means to capture phenomena far beyond the reach of any individual simulation by integrating information from many independent trajectories [22, 23]. Analyzing our MSMs, we find they capture sufficient information about myosin motor domains’ thermodynamics and kinetics to produce reasonable estimates of duty ratio and ADP release rates. Thus, MD and MSMs constitute a powerful platform for identifying relationships between the sequence of individual motor domains and their mechanochemical cycles.
Figure 4.1: The conserved myosin motor domain fold across a diverse phylogeny of motors. (A), A crystal structure (PDB ID 4PA0) [209] of Homo sapiens β-cardiac myosin motor domain as an example of the conserved myosin motor domain fold. We note the structural elements most relevant to our work here (loop 1, in purple backbone sticks, and the P-loop, in orange sticks), along with the actin binding region (blue spheres). For orientation, we include the location of the lever arm (black line) and, to indicate the active site, the estimated location of ADP (yellow sticks). (B) The phylogenetic relationship the various myosin motor domains examined in this work. Except MYH11, all genes are from Homo sapiens. Gene names in blue indicate high duty ratio motors and red indicates low duty ratio. Common protein names are indicated as parentheticals to the left of each gene name. Phylogenetic relationships were inferred from the sequence of the motor domain using k-mer distances [214].
4.3 Materials & Methods

4.3.1 Preparation of homology models

For simulations, the initial structure of each myosin motor domain was prepared by first obtaining the full-length protein’s sequence from PubMed Protein, trimming the sequence down to include only the motor domain using crystal structure 4PA0 of MYH7 as a guide, and submitting that sequence to SWISS-MODEL for homology modeling [215]. Templates were chosen with a preference for those that were high-resolution, high sequence similarity, and in the rigor state. A complete list of sequences, templates, and motor domains can be found in Table 4.1.

<table>
<thead>
<tr>
<th>Gene</th>
<th>Protein Name</th>
<th>Construct</th>
<th>Species</th>
<th>Template</th>
<th>Agg. Sim µs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYH13</td>
<td>Extraocular</td>
<td>4–781</td>
<td><em>H. sapiens</em></td>
<td>4PA0 [209]</td>
<td>271.9</td>
</tr>
<tr>
<td>MYH7</td>
<td>β-cardiac</td>
<td>2–780</td>
<td><em>H. sapiens</em></td>
<td>4PA0 [209]</td>
<td>276.2</td>
</tr>
<tr>
<td>MYH10</td>
<td>Nonmuscle IIb-B2</td>
<td>8–791</td>
<td><em>H. sapiens</em></td>
<td>4PD3 [216]</td>
<td>323.0</td>
</tr>
<tr>
<td>MYO1B</td>
<td>Myosin-Ib</td>
<td>5–703</td>
<td><em>H. sapiens</em></td>
<td>4L79 [207]</td>
<td>282.3</td>
</tr>
<tr>
<td>MYO5A</td>
<td>Myosin-Va</td>
<td>2–762</td>
<td><em>H. sapiens</em></td>
<td>1W8J [203]</td>
<td>297.5</td>
</tr>
<tr>
<td>MYO6</td>
<td>Myosin-VI</td>
<td>2–770</td>
<td><em>H. sapiens</em></td>
<td>2BKI [217]</td>
<td>295.0</td>
</tr>
<tr>
<td>MYO7A</td>
<td>Myosin-VIIa</td>
<td>3–742</td>
<td><em>H. sapiens</em></td>
<td>1OE9 [218]</td>
<td>130.9</td>
</tr>
<tr>
<td>MYO10</td>
<td>Myosin-X</td>
<td>3–740</td>
<td><em>H. sapiens</em></td>
<td>2AKA [219]</td>
<td>126.2</td>
</tr>
<tr>
<td>MYH11</td>
<td>Chicken gizzard</td>
<td>wt/2–782</td>
<td><em>G. gallus</em></td>
<td>4PD3 [216]</td>
<td>6.0</td>
</tr>
<tr>
<td>MYH11</td>
<td>Chicken gizzard</td>
<td>alanine</td>
<td><em>G. gallus</em></td>
<td>4PD3 [216]</td>
<td>6.4</td>
</tr>
<tr>
<td>MYH11</td>
<td>Chicken gizzard</td>
<td><em>Xenopus</em></td>
<td><em>G. gallus</em></td>
<td>4PD3 [216]</td>
<td>16.5</td>
</tr>
<tr>
<td>MYH11</td>
<td>Chicken gizzard</td>
<td>loop 1</td>
<td><em>G. gallus</em></td>
<td>4PD3 [216]</td>
<td>10.5</td>
</tr>
</tbody>
</table>
Table 4.1: Summary of simulations performed for this study. Gene names are those found in PubMed Gene for the appropriate organism, and residue numbers are those used in the given template.

<table>
<thead>
<tr>
<th>Gene</th>
<th>Protein Name</th>
<th>Construct</th>
<th>Species</th>
<th>Template</th>
<th>Agg. Sim µs</th>
</tr>
</thead>
</table>

4.3.2 Preparation of example myosin conformation

In Figure 4.1A, the position of ATP is based on ligand-bound crystal structure 1MMA [216]. The actin binding region was defined by all atoms within 10 Å of the actin filament after alignment to 6BNP chain K [220].

4.3.3 Sequence alignments

All sequence alignments were performed with MUSCLE 3.8.1551 [221] using default parameters. Phylogenetic trees were inferred with the neighbor joining method using these alignments. Distances between sequences were k-mer distances [214].

4.3.4 Molecular dynamics simulations

GROMACS [31, 130] was used to prepare and to simulate all proteins. The protein structure was solvated in a dodecahedron box of TIP3P water [144] that extended 1 nm beyond the protein in every dimension. Thereafter, sodium and chloride ions were added to produce a neutral system at 0.1 M NaCl.

Each system was minimized using steepest descents until the maximum force on any atom decreased below 1000 kJ mol$^{-1}$ nm$^{-1}$. The system was then equilibrated with all atoms...
restrained in place at 300 K maintained by Bussi-Parinello thermostat [222]. After these equilibration runs, the restraints on heavy atoms were removed.

Molecular dynamics were performed using the AMBER03 force field [143]. All covalent bonds involving hydrogen were constrained using LINCS [223]. Virtual sites were used to allow for a 4 fs time [224].

Production simulations were performed on a mixture of Folding@home [19] and an in-house supercomputing cluster. A mix of Tesla K20, Titan Xp, Tesla P100, and Quandro RTX 6000 GPUs were used and Intel Xeon E5-2650 v2, Intel Xeon E5-2630 v3, Intel Xeon E5-2690 v4, Intel Xeon Gold 6148 CPUs clocked at 2.4–2.6 GHz were used.

4.3.5 Markov state models

Fine-grain, whole-motor domain Markov state models were constructed first by defining microstates using the \( k \)-hybrid clustering algorithm with five rounds of \( k \)-medoids refinement using the Euclidean distance between residue sidechain solvent accessible surface area (scSASA) as a distance metric. This approach first appeared in Porter et al. [212] and was chosen because it scales well for extremely large datasets compared to traditional RMSD clustering. The reasons for this are discussed in Porter et al. [91] but, briefly, although scSASA calculations are initially expensive, they realize substantial performance gains in clustering because each frame’s scSASA need only be computed once. Each frame can be computed independently, allowing for massive parallelization. It also reduces the size of the input data size, since only a single floating point number represents an entire residue, and allows the use of a cheaper distance metric (Euclidean distance rather than RMSD).

Markov state models were then fit for each variant by applying a \( 1/n \) pseudocount to each el-
lement of the transition counts matrix and row-normalizing, as recommended in Zimmerman et al. [21]. Lag times were chosen by the implied timescales test and by examining the equilibrium probability distribution for unrealistically overpopulated states (suggesting insufficient sampling of a particular transition or internal energy barriers). Important hyperparameters are listed in Table 4.2.

<table>
<thead>
<tr>
<th>Simulation Set</th>
<th>No. of States</th>
<th>Cluster Radius [nm$^{-1}$]</th>
<th>Lag Time [ns]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYH13</td>
<td>14102</td>
<td>7.4</td>
<td>0.4</td>
</tr>
<tr>
<td>MYH7</td>
<td>5128</td>
<td>7.34</td>
<td>0.5</td>
</tr>
<tr>
<td>MYH10</td>
<td>7746</td>
<td>8.0</td>
<td>1.5</td>
</tr>
<tr>
<td>MYO1B</td>
<td>6458</td>
<td>6.6</td>
<td>0.8</td>
</tr>
<tr>
<td>MYO5A</td>
<td>4728</td>
<td>7.25</td>
<td>0.4</td>
</tr>
<tr>
<td>MYO6</td>
<td>4193</td>
<td>6.9</td>
<td>0.9</td>
</tr>
<tr>
<td>MYO7A</td>
<td>8737</td>
<td>6.9</td>
<td>0.4</td>
</tr>
<tr>
<td>MYO10</td>
<td>9273</td>
<td>6.9</td>
<td>0.4</td>
</tr>
<tr>
<td>MYH11, wild-type</td>
<td>8050</td>
<td>4.9</td>
<td>1.5</td>
</tr>
<tr>
<td>MYH11, alanine sub.</td>
<td>7822</td>
<td>4.9</td>
<td>1.5</td>
</tr>
<tr>
<td>MYH11, <em>Xenopus</em></td>
<td>12804</td>
<td>5.2</td>
<td>1.5</td>
</tr>
<tr>
<td>MYH11, loop 1</td>
<td>8925</td>
<td>5.0</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Table 4.2: Parameters of whole-motor Markov state models used in this study.

Fitting coarse-grained P-loop MSMs used the same procedure, but assignments based on P-loop state were used, rather than assignments to whole-motor SASA states. $P(A \rightarrow B)$ is a parameter of these MSMs. In all cases for coarse-grained P-loop MSMs, a lag time of 37.5 ns was used.

Clustering and Markov state model routines are implemented in *enspara*, git revision f874ba.
Solvent accessibility, atomic distance, and RMSD calculations were performed with MDTraj [118].

We made extensive use of jug [137] and GNU Parallel [225] for task-level parallelization and management of dependencies between tasks.

### 4.3.6 Construction of the P-loop free energy surface

Pairwise interatomic distances in the P-loop were computed using MDTraj [118], selecting all possible pairs of a backbone amide nitrogen and a backbone carbonyl oxygen atom in the GESGAG portion of the Walker A motif (i.e., the conserved P-loop sequence) that makes up the P-loop.

Principle components analysis (PCA) was performed on the 36-dimensional pairwise atomic distance vectors for each MSM microstate using the PCA implementation in sklearn [133]. No whitening was employed and the full SVD was calculated.

The surface was then estimated by constructing a weighted two-dimensional histogram in the PC1/PC3 plane with 50 bins between the minimum and the maximum data in each direction. The resulting array of probabilities was then converted into free energies of units $kT$ by taking the natural logarithm of each value. It was then convoluted with a gaussian of variance 0.3 per grid cell using scipy’s gaussian_filter method [226]. The resulting array was then level-set into six level sets.
4.3.7 Selection of myosin motor domain PDB crystal structures

We selected crystal structures to map on to the P-loop free energy landscape by querying the PDB [7] for all structures with sequence identities to the motor domain of \textit{Hs MYH7} greater than 10\%, resolution $\leq$ 5.0 Å and a BLAST E-value less than $10^{-10}$. We then selected the largest chain in each crystal structure, used MUSCLE [221] to align that chain’s sequence to the motor domain of \textit{Hs MYH7}, and used the resulting alignment to identify the P-loop. P-loop distances were computed and projected into the low-dimensional space as described above. Sequence bookkeeping and I/O relied heavily on scikit-bio (\url{github.com/biocore/scikit-bio}).

Crystal structures were classified as bound to a nucleotide or nucleotide analogue if they contained a residue with the name ADP, ATP, ANP, MNQ, MNT, ONP, PNQ, DAE, DAQ, NMQ, AGS, AD9, AOV, or FLC.

4.3.8 Hierarchical clustering of the P-loop

The five coarse-grained MSM microstates for MYH7 were learned using agglomerative clustering on the four-dimensional P-loop features learned by PCA for the free energy surface. Ward linkage and a Euclidean distance metric were used. Briefly, the states are recursively combined in a way that minimizes the within-cluster variance in a until the specified number of clusters is reached. The number of clusters were increased until no obvious internal free energy barriers were seen in the four PC dimensions. Agglomerative clustering was implemented by sklearn 0.21.2 [133].
4.3.9 Assignment of new conformations to P-loop states

P-loop state assignments for conformations of motors other than Hs MYH7 were made using a $k$-nearest neighbors [133] approach. In this approach, a query conformation is assigned to a cluster based on the assignments of nearest $k$ points in the labeled dataset (i.e., MYH7). In other words, the nearest $k$ points to the query point “vote” on the assignment of the query point to a cluster. In our case, $k$ was 5, but we did not appreciate any differences for values of $k$ from 3 to 15.

Implementation of $k$-nearest neighbors was from sklearn 0.21.2. A ball tree was used to speed the search for neighbors [227].

4.3.10 Estimation of equilibrium probability of P-loop states

For each motor, the probability of a P-loop state was calculated by summing the equilibrium probabilities of all states in the whole-motor MSM assigned to that P-loop state.

4.3.11 Duty Ratios of Crystal Structures

While many myosin motors’ duty ratios have been well characterized, some constructs’ unloaded duty ratio have not been measured. For these motors, it was therefore necessary to infer duty ratios from phylogeny. For our analysis of duty ratio and P-loop crystal position, we considered only the 29 ligand-free structures, namely: 4DBP, 2MYS, 3I5H, 2Y0R, 2BKH, 6I7D, 1DFK, 1OE9, 3I5I, 2OS8, 4P7H, 5V7X, 4ZLK, 1MNE, 1FMV, 2AKA, 3MYL, 2EC6, 4L79, 3L9I, 2BKI, 2Y9E, 1KK7, 1W8J, 2X51, 4PA0, 4PD3, 3I5G, and 1SR6. Based upon previous biochemical experiments, myosin-Is and IIs were assumed to have low duty ratios.
Myosin-VIs were assumed to have high duty ratio. Myosin-Va and Vb from all organisms were assumed to have high duty ratios and Myosin-Vc was assumed to have a low duty ratio. *Plasmodium falciparum* MyoA (6I7D) has been shown to have a high duty ratio [228].

Myosin class was inferred as follows. Where a roman numeral was given in the PDB description (e.g. Myosin-II) this classification was used. Otherwise, if “muscle” or “striated" was appeared in the PDB polymerDescription field, the myosin was classified as a myosin-II. Finally, in the absence of other indicators, myosins from *Doryteuthis pealeii*, *Placopecten magellanicus*, and *Argopecten irradians* were classified as Myosin-IIIs, and myosins from *Plasmodium falciparum* were classified as Myosin-XIVs.

### 4.3.12 Visualization

Proteins structures were visualized and rendered with PyMOL. Data plots were constructed with matplotlib [229]. Free energy surface colormaps were constructed with the cubehelix color system [230].

### 4.3.13 Code and model availability

MSMs and starting conformations for each of the myosin constructs studied in this have been uploaded to the Open Science Framework as project ID 54G7P, along with the parameters for the PCA used in Figures 2 and 3. This OSF project also includes a CSV that lists the P-loop definition, P-loop RMSD from the reference state, and assignment to P-loop state A-E for each crystal structure. This data repository will be made public upon the publication of this manuscript, but can be viewed privately without authentication at https://osf.io/54g7p/?view_only=0f563027able4fa18f527019d56cad2 for the purposes of peer review.
4.4 Results & Discussion

4.4.1 In simulation, the P-loop adopts conformational states that are rare in crystal structures.

We reasoned that any differences between myosin motor domains in nucleotide handling—ADP release rate or duty ratio, for instance—must somehow be manifest at the active site to have an effect. The P-loop is a highly conserved element of the myosin active site that plays an important role in interacting with the phosphates of the ATP substrate [231]. Consequently, we reasoned that the P-loop would report on the conformation of the nucleotide binding site while still being comparable between motors whose sequences differ elsewhere in the protein. To assess the degree of conformational heterogeneity captured by crystal structures, we first analyzed structures deposited in the PDB (Figure 4.2A). We queried the PDB [7] for myosin motor domains (see Methods), yielding 114 crystal structures. Using sequence alignments (see Methods) we identified the P-loop in each of these models and computed the backbone root mean square deviation (RMSD) of each of these models to a reference structure (β-cardiac myosin, PDB ID 4PA0) [209]. We found very little structural diversity among crystal structures, which rarely sample any conformations with P-loop backbone RMSD > 0.6 Å away (Figure 4.2A).

Then, to assess the capacity of the P-loop to adopt conformations not observed in crystal structures, we used molecular dynamics to simulate the myosin motor domain. These simulations of human β-cardiac myosin (Hs MYH7) were performed in the actin-free, nucleotide-free state for roughly a quarter-millisecond in all-atom explicit-solvent detail used to construct an MSM (see Methods). All simulations were conducted using the same force fields and conditions that we have previously used to analyze other systems’ conformational distributions,
including β-lactamases [16, 55, 57], *E. coli* catabolite activator protein [53], Ebola virus nucleoprotein [179], and G-proteins [142]. Then, using the MSM, we computed the distribution of backbone RMSDs of the P-loop relative to the reference crystal structure.

In contrast to the relative uniformity among crystal structures, simulations revealed extensive conformational heterogeneity in the P-loop (Figure 4.2B). Where crystal structures rarely sampled conformations with RMSD >0.6 Å, in simulation we observe broad sampling (i.e. high-probability density) in regions from 0.2 Å RMSD all the way to ~1.5Å RMSD from the starting structure. Only 10 of 114 (9%) crystal structures’ conformations were >0.6 Å RMSD from the reference conformation, whereas fully 58% of the distribution observed *in silico* is above 0.6 Å RMSD from the reference conformation. These results suggest our simulations may provide mechanistic insight not previously accessible from crystal structures alone.

4.4.2 Simulations suggest that the nucleotide-free motor explores distinct nucleotide-favorable and nucleotide-unfavorable states.

We reasoned that P-loop conformations identified by our simulations might have important implications for motors’ nucleotide handling. For example, modulating the relative probabilities of these conformations would provide a facile mechanism by which sequence variation might tune the mechanochemical cycle.

To assess the nucleotide compatibility of the P-loop conformations we observe in simulation, we sought to systematically compare these conformations with crystal structures with and without nucleotide. To do this, we built a map of P-loop conformational space using the dimensionality reduction algorithm Principal Components Analysis (PCA) to learn a low-dimensional representation of the pairwise interatomic distances between P-loop atoms that retains as much of the geometric diversity in the input as possible (see Figure 4.2) [127]. We
Figure 4.2: The P-loop conformational distribution observed \textit{in silico} is substantially broader than that found in crystal structures. (A) P-loop conformations in the PDB are largely restricted to backbone RMSD $\leq 0.6$ Å to a reference conformation (PDB ID 4PA0). \textit{Inset}, the 114 myosin crystal structures superimposed, with the P-loop shown as sticks. (B) P-loop conformations from simulations of \textit{Hs $\beta$-cardiac} myosin frequently explore conformations that are rare or not seen in crystal structures. \textit{Inset}, the 114 most probable P-loop conformations extracted from our simulations of \textit{Hs $\beta$-cardiac} myosin.
then projected the states of our MSM built from our MYH7 simulations onto two principle components (PCs) to visualize the free energy surface sampled by our simulations (Figure 4.3A, green level sets). Using the same PCA, we then projected each crystal structure’s P-loop conformation into this space, plotting each as a point (Figure 4.3A, points). Points labeled with PDB IDs represent crystal structures with P-loops >0.6 Å backbone RMSD away from the reference structure 4PA0 used above. We also classified each structure (see Methods) as nucleotide-bound (yellow points) or nucleotide-free (purple points). Then, we compared the frequency at which nucleotide-bound and nucleotide-free P-loop conformations were found in various conformations.

This analysis revealed two dominant conformational states that likely constitute nucleotide-favorable and nucleotide-unfavorable states (Figure 4.3A and B). Once the distribution of P-loop conformations is projected onto two PCs (the green level sets in Figure 4.3A), we observe two broad minima in the P-loop conformational landscape. We refer to these apparent minima as the upper and lower basin for brevity but recognize that other minima may exist and be obscured by the projection of a high-dimensional space into a low-dimensional space. The lower basin (<0.6 Å RMSD from the reference structure) contains 91% of crystal structures (104/114) and, because 80% (84/105) of these structures are bound to nucleotide, it is highly likely to represent a nucleotide-compatible conformation. In contrast, despite being populated roughly equally in simulation, regions outside the lower basin (0.6 Å RMSD) contain only 9% (10/114) of crystal structures. And, because only one (11%) of these structures is nucleotide bound, these regions are significantly depleted in nucleotide-bound structures (odds ratio = 0.03, p < 1.3x10^-5 by Fisher’s exact test), strongly implying that they are less or not at all nucleotide compatible. Interestingly, this single exception (PDB ID 2Y8I, Dictyostelium discoideum myosin-II G680V) is a highly perturbed motor that has been shown to have low ATPase activity, low motility and a disordered allosteric network.
[232, 233], potentially contributing to its aberrant conformation.

To characterize the structural differences between nucleotide-favorable and nucleotide-unfavorable states captured in the simulations, we coarse-grained our MSM into a model with just five states, called A-E. We used hierarchical clustering to group the thousands of states explored by Hs MYH7 into five states based only on their P-loop conformations (see Methods). Then, using the assignment of each frame from our simulations to one of these five states, we fit a five-state MSM (Figure 4.3D, node sizes indicate equilibrium probabilities, arrow weights indicate transition probabilities). The most probable single state is the A state (49%), which encompasses the entire lower basin and, as we will see below, appears to form favorable interactions with nucleotide based on the conformation of the P-loop. The excited, apparently nucleotide-disfavoring conformations in the upper basin are split into 3 states, B-D, which together account for 50% of the equilibrium probability. Thus, β-cardiac myosin spends about equal time in nucleotide-favorable (state A) and nucleotide-unfavorable states (states B-D) in simulations. Finally, state E (1%, too low to be seen clearly in Figure 4.3A), involves a condensation of the P-loop into an extension of the HF helix, similar to the crystal structure 4L79 [207]. The reduced number of states in this MSM allowed us to inspect a small number of high-probability conformations near the mean of each P-loop state, which we took as exemplars of each of the five P-loop states.

Comparing the states of our MSM reveals that the dominant geometrical difference between nucleotide-favorable and nucleotide-unfavorable P-loop states is the orientation of the peptide bond between S180 and G181 (Figure 4.3C). In the nucleotide-favorable state A (Figure 4.3D, lower right inset), the S180 backbone carbonyl (shown in pink sticks with a white arrow) is oriented away from the phosphates of the nucleotide, enabling the nucleotide to bind to the active site. In contrast, nucleotide-disfavoring states (labeled B-D in Figure 4.3D) orient the S180 backbone carbonyl toward the phosphate groups of the nucleotide. This positions the
carbonyl oxygen in a way that appears to sterically clash with the phosphates of nucleotide. It also orients the negative end of the carbonyl bond’s electric dipole toward the nucleotide binding site and the negatively charged phosphates of ADP and ATP. Taken together, our observations about the geometry of the excited, nucleotide-disfavoring state in the upper basin are consistent with a lowered capacity for nucleotide binding.

4.4.3 The balance between nucleotide-favorable and nucleotide-unfavorable P-loop states predicts duty ratio.

We reasoned that motors with a higher probability of adopting nucleotide-favorable P-loop conformations in isolation are likely to have an increased affinity for nucleotide and, therefore, spend more time in nucleotide-bound states of the mechanochemical cycle. Our reasoning is that motors that prefer nucleotide-favorable P-loop conformations in isolation pay a lower energetic cost to adopting these same nucleotide-favorable conformations when they form a complex with nucleotide. Supporting this logic, it has been observed that, absent load, a large free energy difference between ADP-bound and nucleotide-free states is associated with a low duty ratio [70, 234]. Thus, we hypothesized that a preference for the nucleotide-favorable A state should correlate with low duty ratio.

To test if differences in the probability of excited states encodes information about duty ratio, we simulated an additional seven myosin isoforms of differing duty ratio for a total of ~2 ms of aggregate simulation in all-atom, explicit solvent detail. Specifically, we simulated four human low duty ratio myosin motor domains (from myosin-II genes MYH13, MYH7, MYH10, and myosin-I gene MYO1B) and four human high duty ratio myosin motor domains (from genes MYO5A, MYO6, MYO7A, and MYO10), for between 125 and 325 µs each (see Methods). These motors were selected because extensive kinetic characterization [66, 67, 69,
Figure 4.3: Excited P-loop states are less compatible with nucleotide than the states preferred in crystal structures. (A) The P-loop conformational space projected onto two principal components (PCs) reveals two distinct free energy basins (green level sets). Yellow and purple points represent crystal structures with and without ligand, respectively. Structures farther than 0.6 Å from the β-cardiac myosin structure (red empty circle) are labeled with their PDB ID. (B) Proximity to the β-cardiac myosin reference conformation is associated with the presence of a nucleotide in crystal structures ($p < 1.3 \times 10^{-5}$ by Fisher’s exact test), suggesting that the ligand stabilizes the A state. Error bars represent the 95% confidence interval of 1000 bootstrap realizations. (C) The re-orientation of the S180 backbone carbonyl accounts for the split between upper and lower basins. Points represent P-loop conformations from each state in the β-cardiac whole-motor MSM projected onto the same PCs as in panel A. Points are sized by their probability from the MSM and colored by the angle between the backbone carbonyl bond vectors of S180 and K184. (D) Center, each of the five states of the P-loop MSM are indicated as nodes in a network, sized by their equilibrium probability and connected by arrows with line width proportional to the transition probabilities between them. Surrounding the model, insets show example configurations of the P-loop in sticks colored to match the state they represent. State A is associated with a conformation of the S180 (pink sticks) carbonyl bond vector (white arrow) directed away from the nucleotide binding pocket, whereas states B-D are associated with the opposite orientation of the S180 backbone carbonyl bond vector. The A state conformation is the conformation found in most crystal structures. For reference, PDB 1MMA is shown in grey sticks and the crystallographic position of ATP is shown in semi-opaque grey sticks. For all states, important interactions with the Switch-I loop are shown as two-dimensional sketches for visual clarity. An interaction between R237 and E179 is specific to state A, whereas various interactions with S242 are indicative of other states (Figure 3).
has revealed very diverse kinetic tuning, providing a robust test of our hypotheses. Because no crystal structure of the human sequence was available for any of these proteins except MYH7, homology models were built in each case and used as starting points for simulations (see Methods and Table S1). To allow for direct comparisons between motors, we used the same PCA and state definitions as described above for MYH7.

As expected, high duty ratio motors have a stronger in silico preference for nucleotide-favoring P-loop states than low duty ratio motors (Figure 4.4A). Figure 4.4A shows an example of this effect on the P-loop conformational distributions of high duty ratio motor MYO6 and low duty ratio motor MYH7. The low duty ratio motor explores both upper and lower basins (Figure 4.4A, left) while the high duty ratio motor strongly prefers the lower basin (Figure 4.4A, right). Provocatively, when motors are crystallized without ligand, only motors with low unloaded duty ratios have been crystallized with P-loops outside the nucleotide-favorable conformation (Figure 4.4A, red and blue points). Of 29 unliganded crystal structures, 8/20 (40%) of low duty ratio motors’ P-loops crystallized outside the A state, whereas 0/9 (0%) high duty ratio motors’ P-loops crystallized outside state A (p < 0.034 by Fisher’s exact test, see Methods).

Given this trend, we reasoned that the relative free energies of the nucleotide-favorable state and the nucleotide-disfavoring excited states would provide a useful predictor of a motor’s duty ratio. We assigned every whole-motor MSM state to one of the five P-loop states and used these assignments to compute the free energies of each of the five states for each of the eight motors (see Methods). We then took the difference in free energy between states A and B, which are the two best sampled states and therefore give statistically robust results. Numerical values and references for these experimental values can be found in Table S3.

As expected, we find a strong correlation between motors’ duty ratios and their preferences
for the nucleotide-favorable A state over the nucleotide-unfavorable B state (Figure 4.4B). Specifically, high duty ratio motors have a strong preference for the A state (negative free energy difference) while low duty ratio motors spend more time in state B (positive free energy difference). Decreased stability of the nucleotide-favorable conformation in these low duty ratio motors could explain this observation.

4.4.4 Simulations predict ADP release rates better than loop 1 length does by capturing sequence-specific effects.

Because ADP release allows a motor to adopt nucleotide-incompatible P-loop conformations, we reasoned that the rate at which a motor can transition to these conformations in silico might correlate with in vitro ADP release kinetics. While we expect a correlation, we acknowledge that the absolute rates will almost certainly differ, since the rates themselves likely differ in the presence and absence of nucleotide. To test for a correlation, we first focus on data sets that examine several motors under the same experimental conditions. Identical conditions are important because in vitro biochemical rates depend strongly on experimental conditions such as salt and temperature [236, 238, 239]. We focus on low duty ratio motors, since their frequent transitions to nucleotide-unfavorable states make it possible to estimate their transition rates with confidence. In contrast, in high duty ratio motors, transitions between these states are sufficiently rare that their rates cannot be estimated with confidence.

An especially useful dataset for comparing relative ADP release rates was created by Sweeney et al. [198], which carefully dissected the effect of variation in loop 1 length and sequence on ADP release rates using the same experimental conditions. These authors established a positive relationship between loop 1 length and ADP release rate using engineered constructs
Figure 4.4: The free energy landscape of the P-loop encodes duty ratio. (A) Free energy landscapes in the PC1/PC3 plane demonstrate that the upper basin is well sampled by an example low duty ratio motor (MYH7, left) and poorly sampled by an example high duty ratio motor (MYO6, right). Ligand-free crystal P-loop conformations from high and low duty ratio motors are shown as blue and red points, respectively. (B) Experimental duty ratio (x-axis) is correlated with the simulated free energy difference between nucleotide favorable and nucleotide-unfavorable states (y-axis, more negative values mean higher probability of the nucleotide-favorable A state). Error in simulated free energy differences were estimated by jackknife resampling and were too small to be visualized as error bars.
of chicken gizzard myosin-II (shown in Figure 4.5A, henceforth Gg MYH11). A notable exception, however, was the myosin with wild-type loop 1, which had an ADP release rate more than three times faster than predicted by the length-based model (Figure 4.5B). This deviation from a purely length-driven ADP release rate led these authors to hypothesize that there must also be sequence-specific effects of loop 1 on ADP release rate. They then identified an alanine mutant that ablated the sequence-specific effects of the wild-type loop (henceforth Gg MYH11-ala).

To assess the capacity of in silico P-loop kinetics to capture the experimentally measured ADP release rates in the constructs investigated by Sweeney et al. [198], we simulated and analyzed four Gg MYH11 constructs. These constructs are a subset of the variants considered by Sweeney et al. [198]. We selected the wild-type loop (Gg MYH11-wt) because it was the primary outlier in their length-only model. We selected the alanine mutant (Gg MYH11-ala) because it, with just 5 mutations, shifted the wild type loop in line with the length-only model proposed by Sweeney et al. [198]. Then, we selected the extreme points that were well fit by the loop length-only model: the loop 1 deletion (Gg MYH11-loop1) and the construct using the loop 1 from Xenopus non-muscle myosin (Gg MYH11-xeno). We simulated these four constructs for 6-16 µs each beginning from a homology model (see Methods and Table S1) and built whole-motor MSMs which, as before, were used to compute five-state P-loop MSMs. Each P-loop MSM contains a parameter P(A→B) which captures the probability that a conformation in state A transitions to state B within a fixed period of time (known as the lag time of the model). We then compared P(A→B) to ADP release rates measured in vitro for these four constructs.

As expected, there is a strong positive relationship (Pearson’s R=0.99) between the P(A→B) fit by our MSMs and in vitro ADP release rate (Figure 4.5C). This is stronger than the equivalent correlation for the length-based model (Pearson’s R=0.72). Importantly, the rank
order of the four isoforms is correct, whereas using a loop 1 length-only model dramatically underestimates the ADP release rate for the wild-type motor. Together, the fact that the sequence change is small (only five residues differ between wild type and the alanine mutant) and the change is distant (~25Å) from the P-loop indicate that our model is exquisitely sensitive to sequence, even at sites distant from the active site.

**P-loop kinetics in silico correlate with ADP release rates across conditions.**

To further assess the generalizability of our model, we considered several additional datasets that relax constraints placed on data sets in the previous section. First, we relaxed the constraint that motors differ by just one structural element (loop 1). Specifically, we considered several skeletal myosin isoforms, including MYH7 and MYH13 that Johnson *et al* (35) studied under the same conditions (Figure 4.5D and E, yellow points). These motor domains are an interesting case because, at 80% sequence identity, their sequences differ much more than constructs, and these differences are distributed throughout the protein. Crucially, and despite having roughly the same loop 1 length, their ADP release rates differ by about an order of magnitude (59 s\(^{-1}\) vs 400 s\(^{-1}\)). Owing to the fact that Johnson *et al*’s data were collected under different experimental conditions than Sweeny *et al*’s data (5 mM MgCl\(_2\) at 25 °C vs 1 mM MgCl\(_2\) at 20 °C with different light chains), we only expect a general trend to hold, since motors’ properties are very sensitive to magnesium, temperature, and light chain identity [236, 238, 240]. Second, we assessed the trend in two human non-muscle motor domains, MYO1B and MYH10 with measurements carried out under different conditions. Notably, because they both release ADP very slowly, they test our model’s capacity to evaluate very slow ADP release rates.

Consistent with our expectations, and despite the diverse experimental conditions, we still observe a reasonable correlation between P(A→B) and ADP release across all data sets.
(Figure 4.5E, Pearson’s R = 0.75). This dramatically improves on the length-based model (Pearson’s R = 0.14). Importantly, under the matched experimental conditions for MYH7 and MYH13 we still find the correct order of ADP release rates (Figure 4.3C, yellow points), suggesting that this method generalizes well to the larger phylogenetic distances between myosin isoforms. Furthermore, MYO1B and MYH10 are correctly identified as very slow releasers of ADP, although the point estimates appear to be quite noisy. MYH10 is known to be exquisitely sensitive to light chains [240], so it is not surprising that it is one of the greatest outliers given that we did not include these in our simulations.
Figure 4.5: The probability of transitioning from nucleotide-favorable to nucleotide-unfavorable P-loop conformations (P(A→B)) predicts ADP release rates for motors with low duty ratios. (A) Loop 1 sequences and lengths considered in this work. Residues mutated to alanine in the wild-type chicken gizzard MYH11 (wt Gg MYH11) are bolded in the appropriate row. (B) For the Sweeney dataset, there is a moderate relationship between loop 1 length and ADP release rate (Pearson’s R = 0.75) but, (C) there is a much stronger correlation between P(A→B) and ADP release rate (Pearson’s R = 0.99). (D) Across all datasets, the relationship between loop 1 length and ADP release rate is weak (Pearson’s R = 0.14), and (E) there is a much stronger correlation between P(A→B) and ADP release rate (Pearson’s R = 0.75). Error in MSM parameters was estimated by jackknife resampling and errors in ADP release rates are those reported in the relevant original publication, where available.
4.5 Conclusion

In this work, we used computer simulations of isolated myosin motor domains to predict the in vitro ADP release rate and duty ratio of unloaded myosin motors. To do this, we identified systematic shifts in the distribution of conformations that a motor explores that correlate with changes in biochemistry, rather than by directly simulating the biochemical processes themselves, which would have been prohibitively expensive. While binding partners (actin and nucleotide, for instance) and structural elements outside the motor domain almost certainly affect the distribution of conformations, our results demonstrate that it is nevertheless possible to extract reasonable estimates for at least some unloaded biochemical properties from only the isolated motor domain’s conformational distribution. The ability of the isolated motor domain’s fluctuations to predict these parameters likely stems from a link between the isolated and bound conformational distributions. In other words, because the motor domain active site must adopt certain key conformations during its functional interactions with binding partners (i.e., nucleotide and actin), it is nearly guaranteed to at least transiently sample those conformations even in the absence of those binding partners. Importantly, our simulations only require a reasonable homology model as a starting point, so our methods should be applicable to a broad range of motor variants, including mutations implicated in disease.

Given the high degree of structural conservation of the myosin motor domain, it was not previously possible to directly predict the duty ratio or kinetics for a given myosin isoform from the sequence or structure of a motor domain alone. Our studies demonstrate that the duty ratio and the rate of ADP release are not captured by a single structural element, but rather by the distribution of conformations that the motor explores in solution. Throughout our simulations, we observed that the distribution of P-loop conformations is sensitive
to relevant sequence changes, both large and small, throughout the myosin motor domain. Presumably, these changes are allosterically propagated through the myosin motor domain through complex networks of coupled motions. Thus, capturing the difference between the wild-type and alanine-substituted chicken gizzard myosins (Figure 4.5C), for instance, required the model to capture the allosteric perturbation induced by a change of a few dozen atoms in a molecule of ~12,500 atoms at a distance of ~25 Å (Figure 4.1A). Meanwhile, classifying the duty ratio of diverse myosin motors requires the P-loop to integrate signals from across the molecule into a single overall conformational preference. This underscores a key advantage of physics-based simulations, which is the ability to represent these allosteric networks by modeling in detail the complex, nonlinear couplings throughout the molecule.

One tantalizing interpretation of the excited states of the P-loop we observe in silico is that they may be related to the biochemically-observed “open” and “closed” states that nucleotide-free myosin motors populate in vitro [241]. In our simulations, we see that the P-loop fluctuates between conformations that are nucleotide-compatible and conformations that probably are not. In biochemical experiments, at least some myosin isoforms in the nucleotide-free actin-bound state fluctuate between a state that binds nucleotide and a state that does not. It has also been shown that the equilibrium between these two biochemical states (K), correlates with duty ratio and the transition rate from the nucleotide binding incompetent state to the nucleotide binding competent state (k+) correlates with the ADP release rate [70]. Similarly, we showed that the equilibrium between nucleotide-favorable and nucleotide-disfavorable conformations predicted duty ratio, while the rate of transition predicted ADP release rate. A simple explanation for these similarities is that there may be a correspondence between these biochemical states and the structural states that we observe in our MSMs in silico.

Finally, our results highlight the general capacity of computational modeling to link sequence
and function. One immediate application of our work here is to estimate *in silico* the biochemical parameters of new or difficult-to-study myosins. In the near term, constructing such models could help us learn more about the atomic basis for healthy functional diversity in myosin motors, and how small changes can give rise to malfunction and disease. Indeed, in the coming years it may prove possible to use these models as a tool for studying patient-specific mutations by understanding the atomic basis for diseases caused by dysfunction of myosin motors or to aid in developing therapeutics. Finally, because we find no reason to believe our approach’s applicability is limited to myosin motors, we expect the techniques we have presented here to be of use for any protein where the physics that maps sequence to biochemistry is not straightforward.
Chapter 5

Main findings and future directions

5.1 Main Findings

5.1.1 Summary

In this thesis, I presented the development of a systematic approach for extracting insight from simulation datasets of enormous size. It is compatible with at least milliseconds of sampling of proteins at least 800 residues in size. Three key developments form the basis for this approach.

First, in Chapter 2, my colleagues and I tackled the challenge of constructing Markov state models on data at the terabyte and millisecond scale. This required first a careful analysis of the scaling failures of existing technologies and then, careful implementation of several key features. Some problems were solved by straightforward software engineering, such as careful memory management and low-level programming. A few more problems were solved by using pre-existing well-suited data structures, such as the ragged array and the
sparse matrix. Finally, some problems required the development of novel algorithms, like a low-communication, parallelized version of the classic $k$-centers and $k$-medoids clustering algorithms.

Second, in Chapter 3, we proposed and demonstrated the usefulness of exposons, an unsupervised segmentation algorithm. Exposons, by grouping residues into units that show cooperative solvent exposure, provide an overview of an entire conformational landscape in a single picture. While this necessarily omits some information, it focuses attention on the areas of a protein’s surface that are most likely to be relevant to function. Using this approach, we identified a new cryptic pocket in the very-carefully studied TEM-1 β-lactamase and the first known cryptic pocket in the CTM-M-9 β-lactamase.

Third and finally, in Chapter 4, we applied these methods to a large system of biological, historical, and medical interest. We started by collecting more than 2 ms of sampling for eight human and four chicken myosin isoforms. We then built Markov state models for each using the technologies outlined above. Then, we applied these technologies and others to make substantial inroads into a longstanding question about myosin biochemistry: why do isoforms with dramatically different kinetic properties not differ in their crystal structure?

Our hypothesis was that kinetic differences between myosins are accounted for by differences in the favorability and kinetics of higher-energy states. Because these states are generally not observed in crystal structures, it has not been possible to make predictions about myosins’ kinetics from crystal structures alone. Ultimately, we identified molecular motions in simulation that predicted important kinetic properties of these molecules, providing strong support for this hypothesis and demonstrating the value of large computation for addressing biochemical questions.
5.1.2 Key Conclusions

One important conclusion from this is that large amounts of computation is extremely valuable for addressing questions of protein dynamics. As discussed in Chapter 1, protein conformational space is gigantic, and much of it is not accessible without large data sets. For instance, the MYH7 P-loop is apparently able to access the conformation populated by MYO1B in its crystal state, but it does so with such infrequency that without large computation, it is likely to be overlooked. Likewise, the opening of most of the pockets discovered in Chapter 3 are rare events (indeed, even our large datasets appear to have missed at least one type of exposure event). Furthermore, because it is the relative favorabilities of these high-energy states (not their existence or absence) that allowed biochemically-relevant predictions, there is the additional need for good statistics over and above mere observation for these states.

Another key finding is the value of solvent accessibility as a rotation- and translation-invariant, highly compressed order parameter. Compressing atomic coordinates for an entire residue into a single 32-bit floating point by representing the residue as sidechain SASA achieves a roughly 50-fold compression of data while—apparently—still preserving the ability to learn good Markov state models. This computation is embarrassingly parallel (i.e. \( p \approx 1.0 \)) and \( O(N) \) in the number of frames in a dataset, and so can be easily scaled to the size of the dataset. Importantly, it reduces the size of the data that must be clustered, which is an inseparable computation for which \( p \ll 1.0 \) and for which scaling is least \( O(N^2) \) in the number of frames in both time and space. While this particular featurization was especially useful for discovering pockets (Chapter 3), these properties made it useful for other applications as well (Chapter 4).

This thesis is also additional support for the increasingly-less-controversial claim that the conformational ensemble—or at least much of it—determines much about function. In the
case of β-lactamases, we saw that excited states can be targeted directly to modulate function and, hence, are potential drug targets. In the case of myosin, we found that the relative populations of these high energy states contained information about overall biochemistry of each motor—tempting one to speculate that it is in fact these states (or something like them) that are tuned by sequence.

5.2 Future Directions

As parallelization increasingly becomes the rule, and the cost of individual parallel processors continues to decline, the size of data sets, and the size of systems that can be studied with them, will continue to increase. Indeed, the day will likely come the datasets cannot be easily manipulated on a simple NFS on an HPC, as is expected for work in Chapter 2. As data sizes increase—and we have begun to see the beginning of this already—it will become increasingly infeasible to store data on the same (traditional, physically co-located) filesystem, because of various physical limitations and limitations on the size of traditional (i.e. RAID-style) disk arrays. Instead, technologies developed for web-scale computation, like append-only filesystems and map/reduce style parallel computation, among many others, will become increasingly necessary. A general feature of these technologies is that they move code to data (executing relevant analysis on whatever system is storing the data) rather than moving data to computation (as is the case with the traditional HPC environment using an NFS). This will further necessitate the reconstruction of traditional algorithms, and likely the development of new algorithms, to tackle the challenges of whichever parallel computing paradigm is best suited to the challenges of large datasets going forward. It is my hope that, when or if that day comes and the specific algorithms outlined in Chapter 2 are no longer useful, the general approach I outlined there will be of guidance.
As these datasets of increasing size become tractable through improvements in computation, new and more advanced parallel programming frameworks, and the cleverness of computational scientists, the next challenge will be actually transforming these models into human insight. While Markov state models and allied technologies faithfully represent a conformational landscape as a series of free energy basins separated by energy barriers, this landscape is sufficiently complex and multifaceted that these models are exceptionally difficult to decipher manually. While the exposons framework we proposed in Chapter 3 is effective at analyzing simulations of the size we presented in this thesis, it may not be as we are able to examine these conformational landscapes more completely. For instance, for simulations that capture full folding events, every protein might have only one exposon: the core, which buries cooperatively. An interesting elaboration of the exposons framework would be to recast exposons as a hierarchy, such that exposons are split into “subexposons” that are even more strongly connected among themselves than the other residues in the exposon. In addition to elaborations of the exposons framework, additional methods for making conformational distributions more accessible to the human mind are necessary, particularly because the internal (i.e. subsurfacial) mechanics of proteins are also clearly important, but are ignored by exposons.

Another question that is left open by this thesis is how sequence actually creates the differences we observe in conformational ensembles. In our study of \( \beta \)-lactamases, for instance, we found that TEM-1 and CTX-M-9, while sharing a virtually indistinguishable folded conformation, have very different patterns of cooperative solvent exposure. In this case, it wasn’t clear if these solvent exposure differences are incidental—perhaps the consequence of genetic drift—or actually integral to the protein’s function and differing spectra of antibiotic degradation. Similarly, in the case of the eight human myosin isoforms, the distribution of P-loop conformational states was strongly affected by isoform type, despite total identity of
the P-loop sequence. In that case, however, it seems likely that the differences in the con-
formational distribution arise at least because of the sequence differences that give rise to
differing molecular phenotype, and potentially themselves even cause those differences. This
question—what are the underlying rules that allow differing sequences to produce differing
phenotypes without altering topology?—are not answered in a detailed way in this thesis.
However, it remains possible that no such answer that is simpler than the molecular dynam-
ics forcefield itself exists. Proteins are *bona fide* complex systems, with all the challenges
to the human intuition and computational tractability that creates. The answer, then, may
really be just to “shut up and calculate.”

Finally, and most importantly, it is my view that the purpose of my research is to help
people. The most obvious and most exciting next steps to bring my research to the bedside
would be as a parts of a toolchain for personalized medicine. If simulation can be used to
predict if a myosin will be high duty ratio or low duty ratio—as we have done in this thesis—it
seems not so far away that these technologies could distinguish between healthy and diseased
versions of a myosin motor or any other protein. The number of potential applications for
such a technology are clearly numerous. They range from stratifying patients based on
risk of a genetic disease to personalized predictions for drug efficacy and kinetics. In their
final form, a more complete understanding of the way genotype influences molecular (and
eventually organism-level) phenotype is equivalent to a more complete picture of ourselves
as individuals, and of all of us collectively.
Appendix A

A.1 Implied timescales

We built our Markov state models using the discretization discovered by the $k$-hybrid clustering algorithm on SASA feature vectors (see Methods). We then verified our choice of $k$-centers stopping condition (2.6 nm for TEM-1 and 3.0 nm for CTX-M-9) and chose our lag times (4 ns for TEM-1 and 0.6 ns for CTX-M-9) using the implied timescales test.
Figure A.1: Implied timescales tests for TEM-1 (top left), CTX-M-9 (top right), CAP (bottom left), and eNP (bottom right). The implied timescales test is a test for Markovianity of an MSM. The model is fit with a variety of lag times (x-axis) and the slowest few motions’ speed is computed. In a good model, the timescales are relatively insensitive to lag time, and so typically a lag time will be chosen that minimizes the derivative of implied timescale with respect to lag time [23, 92].
A.2 Affinity clustering is stable to damping parameter changes
Figure A.2: Affinity clustering is stable across most values of the damping parameter in TEM-1. This figure is a parameter scan across the range of possible damping parameters for the clustering algorithm that assigns residues to exposons. In this figure, the y-axis represents choice of damping parameter, the x-axis represents the protein sequence. The color of at any position denotes the exposon to which a residue was assigned at that particular choice of damping parameter. Thus, the appearance of vertical bars is a consequence of the fact that residues generally do not change which exposon they are assigned to. Some residues are assigned to the same cluster for all damping parameter choices < 0.95.
A.3 Time-resolved DTNB thiol labeling by stopped-flow

To measure the labeling rate of any particular residue, we prepared the relevant cysteine mutant (see 3.3) and ran triplicate DNTB labeling experiments both with and without protein. We subtracted the baseline absorbance of DTNB in buffer from the labeling trace and fit to a single exponential. Each point in Fig. 3.2b and e represents the results of such a procedure. A representative fit to the data for TEM-1 at 500 µM DTNB is shown below.

<table>
<thead>
<tr>
<th></th>
<th>TEM-1 M182T/S243C</th>
<th>CTX-M-9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{\text{int}}$</td>
<td>$6.83 \pm 1.18 \text{ mM}^{-1} \text{s}^{-1}$</td>
<td>$71.5 \pm 5.3 \text{ mM}^{-1} \text{s}^{-1}$</td>
</tr>
<tr>
<td>$K$</td>
<td>$1.10 \times 10^{-2} \pm 1.9 \times 10^{-3}$</td>
<td>$2.34 \times 10^{-4} \pm 7.8 \times 10^{-5}$</td>
</tr>
<tr>
<td>$k_{\text{op}}$</td>
<td>N/A</td>
<td>$1.22 \times 10^{-2} \pm 2.05 \times 10^{-3} \text{ s}^{-1}$</td>
</tr>
<tr>
<td>$k_{\text{cl}}$</td>
<td>N/A</td>
<td>$51.3 \pm 14.4 \text{ s}^{-1}$</td>
</tr>
<tr>
<td>$K \times k_{\text{int}}$</td>
<td>$7.5 \times 10^{-2} \pm 1.5 \times 10^{-3} \text{ mM}^{-1} \text{s}^{-1}$</td>
<td>$1.67 \times 10^{-2} \pm 5.70 \times 10^{-3} \text{ mM}^{-1} \text{s}^{-1}$</td>
</tr>
</tbody>
</table>

Table A.1: Parameters of Linderstrøm-Lang model of DTNB labeling. Error is the standard deviation from 100 trials of bootstrapping.
Figure A.3: A representative trace of absorbance over time for a sample of 9 µm TEM-1 S243C mixed with 500 µm DTNB along with a single exponential fit (top) and the residuals to that same model (bottom). For the top figure, red is raw data and dashed black is the fit. For the bottom figure, red represents the raw residuals and black represents a Gaussian convolution of that data.
A.4 Global unfolding rates of TEM-1 S243C and CTX-M-9
Figure A.4: Fig S4: Rates of unfolding of TEM-1 S243C (left) and CTX-M-9 (right) as a function of urea concentration. A linear fit is used to extrapolate to the rate of global unfolding of each protein to the rate in the absence of urea. The rate of unfolding for TEM-1 S234C is $1.054 \times 10^{-5} \pm 1.371 \times 10^{-5}$ s$^{-1}$ whereas the rate of unfolding for CTX-M-9 is $1.308 \times 10^{-5} \pm 2.274 \times 10^{-5}$ s$^{-1}$. Error is estimated using the standard deviation from 100 rounds of bootstrapping.

A.5 Protein stability measurements

\[
\text{CD}(\Theta) = \frac{\Theta_u + \Theta_n \exp\left(-\frac{G_{\text{un}} + m_{\text{un}}[\text{urea}]}{RT}\right)}{1 + \exp\left(-\frac{G_{\text{un}} + m_{\text{un}}[\text{urea}]}{RT}\right)}
\]  

(A.1)

where $\Theta_u$ and $\Theta_n$ are the circular dichroism signals at 222 nm for the unfolded and native states. $G_{\text{un}}$ is the extrapolated free energy change in 0 M urea between the unfolded and native states, and $m_{\text{un}}$ is a proportionality term related to the steepness of the linear fit of the unfolded to native state transition [242].

<table>
<thead>
<tr>
<th></th>
<th>$G_{\text{un}}$ (kcal mol$^{-1}$)</th>
<th>$m_{\text{un}}$ (kcal mol$^{-1}$ M$^{-1}$)</th>
<th>$C_m$ (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEM-1 M182T/S243C</td>
<td>5.0 ± 0.3</td>
<td>1.12 ± 0.07</td>
<td>4.5 ± 0.4</td>
</tr>
<tr>
<td>CTX-M-9</td>
<td>5.5 ± 0.2</td>
<td>3.0 ± 0.1</td>
<td>1.83 ± 0.09</td>
</tr>
</tbody>
</table>

Table A.2: Table S2. Equilibrium Fit Parameters. Errors are standard deviations.
Figure A.5: Circular dichroism as a function of urea concentration (solid circles) fit to a two-state model (equation S1) of unfolding for TEM-1 S243C (left) and CTX-M-9 (right). Data were collected in triplicate.

### A.6 Activity of labeled enzyme

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$k_{\text{cat}}$ (s$^{-1}$)</th>
<th>$K_M$ (µM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEM-1 M182T/S243C</td>
<td>354 ± 13</td>
<td>88 ± 5</td>
</tr>
<tr>
<td>Labeled TEM-1 M182T/S243C</td>
<td>337 ± 9</td>
<td>22 ± 2</td>
</tr>
<tr>
<td>CTX-M-9</td>
<td>254 ± 10</td>
<td>28 ± 3</td>
</tr>
<tr>
<td>Labeled CTX-M-9</td>
<td>65 ± 8</td>
<td>114 ± 21</td>
</tr>
</tbody>
</table>

Table A.3: Parameters for Michaelis-Menten model of enzyme activity. Error is the standard deviation from 100 trials of bootstrapping.
Figure A.6: Figure S6: Activity of enzyme before (blue) and after (red) addition of covalent DTNB label for TEM-1 S243C (left) and CTX-M-9 (right). Blue points are unlabeled enzyme, red points are labeled enzyme. Points were taken in triplicate. Fits are to a Michaelis-Menten model.

A.7 Synthetic labeling of TEM-1 residues
Figure A.7: Synthetic labeling traces of various TEM-1 residues indicate the fraction of the population (y-axis) that is expected to be have been exposed to solvent after a particular amount of time (x-axis). The rank order here recapitulates the \textit{in vitro} rates' order reported in [47].
A.8 Estimation of druggability of S243 pocket

We used fpocket [192] to estimate the druggability of every frame associated with a microstate where S243 is classified as exposed. We then filtered pockets for S243 gamma oxygen involvement and for the lack of involvement of the active site serine, S70. As noted in the main text, traditional pocket detection algorithms have a tendency to combine this pocket with the active site, as they often form a channel-like connection, despite being geometrically distinct. We expect the druggabilities noted here to be lower bound estimates for druggability, the druggability score was trained on crystal structures of ligands, which are typically adopt a more closed conformation created by contributions of induced fit, whereas in simulation these same pockets are often much more open [99].
Figure A.8: Figure S8: Druggability and equilibrium probabilities for pockets involving S243. Top, a violin plot of the distribution of FPocket druggability score for both microstates with exposed S243. Bottom, the equilibrium probability of each of those states. These data suggest a minor population of a very-likely druggable conformation (state 1271, blue).

A.9 $k_{\text{obs}}$ is bounded above by $k_{\text{obs}}^{(\text{EX1})}$ and $k_{\text{obs}}^{(\text{EX2})}$.

As defined in Section 3.3, the observed labeling rates of the three regimes are defined as:

$$k_{\text{obs}} = \frac{k_{\text{op}} k_{\text{int}}[\text{DTNB}]}{k_{\text{op}} + k_{\text{cl}} + k_{\text{int}}[\text{DTNB}]}$$

$$k_{\text{obs}}^{(\text{EX2})} = \frac{k_{\text{op}}}{k_{\text{cl}}} k_{\text{int}}[\text{DTNB}] = K k_{\text{int}}[\text{DTNB}]$$

$$k_{\text{obs}}^{(\text{EX1})} = k_{\text{op}}$$

We want to show that $k_{\text{obs}} < k_{\text{obs}}^{(\text{EX2})}$ and $k_{\text{obs}} < k_{\text{obs}}^{(\text{EX1})}$.

**Hypothesis I:** $k_{\text{obs}} < k_{\text{obs}}^{(\text{EX2})}$
\[ k_{\text{obs}} < k_{\text{obs}}^{(\text{EX2})} \]

\[
\frac{k_{\text{op}}k_{\text{int}}[\text{DTNB}]}{k_{\text{op}} + k_{\text{cl}} + k_{\text{int}}[\text{DTNB}]} < \frac{k_{\text{op}}}{k_{\text{cl}}} k_{\text{int}}[\text{DTNB}]
\]

\[
\frac{1}{k_{\text{op}} + k_{\text{cl}} + k_{\text{int}}[\text{DTNB}]} < \frac{1}{k_{\text{cl}}}
\]

\[ k_{\text{op}} + k_{\text{cl}} + k_{\text{int}}[\text{DTNB}] > k_{\text{cl}} \]

\[ k_{\text{op}} + k_{\text{int}}[\text{DTNB}] > 0, \text{ which is true, since each term is greater than zero individually.} \]

**Hypothesis II:** \( k_{\text{obs}} < k_{\text{obs}}^{(\text{EX1})} \)

\[
\frac{k_{\text{op}}k_{\text{int}}[\text{DTNB}]}{k_{\text{op}} + k_{\text{cl}} + k_{\text{int}}[\text{DTNB}]} < k_{\text{op}}
\]

\[
\frac{k_{\text{int}}[\text{DTNB}]}{k_{\text{op}} + k_{\text{cl}} + k_{\text{int}}[\text{DTNB}]} < 1
\]

\[ k_{\text{int}}[\text{DTNB}] < k_{\text{op}} + k_{\text{cl}} + k_{\text{int}}[\text{DTNB}] \]

\[ 0 < k_{\text{op}} + k_{\text{cl}}, \text{ which is true, since each term is greater than zero individually.} \]

Thus, since \( k_{\text{obs}} < k_{\text{obs}}^{(\text{EX2})} \) and \( k_{\text{obs}} < k_{\text{obs}}^{(\text{EX1})} \) we can conclude that \( k_{\text{obs}} < \min \left\{ k_{\text{obs}}^{(\text{EX2})}, k_{\text{obs}}^{(\text{EX1})} \right\}, \)

*i.e.* that the EX1 and EX2 observed rates serve as strict upper bounds to the EXX observed rate.
References


[31] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C Smith,
Berk Hess, and Erik Lindahl. GROMACS: High performance molecular simulations
through multi-level parallelism from laptops to supercomputers. SoftwareX, 1-2:19–25,
September 2015.

events and reconstruct the free energy in biophysics, chemistry and material science.

[33] Rafael C. Bernardi, Marcelo C.R. Melo, and Klaus Schulten. Enhanced sampling tech-
niques in molecular dynamics simulations of biological systems. Biochimica et Biophys-
ica Acta (BBA) - General Subjects, 1850(5):872–877, May 2015. ISSN 03044165. doi:


[36] Philipp Metzner, Frank Noé, and Christof Schütte. Estimating the sampling error:
distribution of transition matrices and functions of transition matrices for given tra-
2009.

135


[81] Sara L Van Driest, Steve R Ommen, A Jamil Tajik, Bernard J Gersh, and Michael J


[117] Zhao, Yutong, Sheong, Fu Kit, Sun, Jian, Sander, Pedro, and Huang, Xuhui. A fast


[134] Carlos X Hernández and Vijay S Pande. MDEntropy: Information-Theoretic Analyses


Lukas Burger and Erik van Nimwegen. Accurate prediction of protein-protein interactions from sequence alignments using a Bayesian method. *Molecular systems biology*,


152


[199] Zoltan Ujfalusi, Carlos D Vera, Srbolujub M Mijailovich, Marina Svicevic, Elizabeth Choe Yu, Masataka Kawana, Kathleen M Ruppel, James A Spudich, Michael A Geeves, and Leslie A Leinwand. Dilated cardiomyopathy myosin mutants have reduced


[221] Robert C Edgar. MUSCLE: multiple sequence alignment with high accuracy and high


[228] Julien Robert-Paganin, James P. Robblee, Daniel Auguin, Thomas C. A. Blake, Carol S. Bookwalter, Elena B. Krementsova, Dihia Moussaoui, Michael J. Previs, Guillaume Jousset, Jake Baum, Kathleen M. Trybus, and Anne Houdusse. Plasmodium myosin A drives parasite invasion by an atypical force generating


