Doing More With Less: Improved Simulation and Analysis of Biomolecular Systems

John D. Russo

A dissertation presented in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy in

Biomedical Engineering

to the **School of Medicine** at

Oregon Health & Science University.

May 2023

Biomedical Engineering

School of Medicine

Oregon Health & Science University

CERTIFICATE OF APPROVAL

This is to certify that the Ph.D. dissertation of
John D. Russo
has been approved.

Daniel M. Zuckerman
Advisor

Abhinav Nellore
Chair

Ninian J. Blackburn
Committee Member

Vincent A. Voelz
Committee Member

TABLE OF CONTENTS

ABSTRACT

Molecular dynamics (MD) simulations are a crucial tool for understanding biomolecular systems, offering a unique dynamical picture of atomic motions. However, several key challenges limit their practical applicability to understanding complex systems.

A persistent challenge in methods development is choosing a validation system which is complex enough to stress-test an analysis method, but where exact reference values for measurable quantities are known. We propose **synthetic dynamics (SynD)**, a tool for efficiently generating approximate simulation data with similar complexity to true MD. Using SynD, we produce meaningful trajectories at substantially reduced computational cost, enabling rapid methods validation. We apply this to the methods described in the other sections.

A main goal of MD simulations is accurate estimation of biophysical properties such as rate constants. Although **Markov state models (MSMs)** are a widespread and useful tool for analyzing MD simulation data, typical MSM analysis methodologies include biases which taint estimates of physical observables. To address this, we present a set of novel estimators for equilibrium populations, mean first-passage times, and committors. We also develop an iterative extension of a previously proposed reweighting scheme, which reduces the amount of data necessary for unbiased estimates of observables.

Many biologically relevant processes remain beyond the timescales accessible to conventional MD. Enhanced sampling strategies such as **weighted ensemble (WE)** aim to bridge this gap by improving the computational efficiency of MD. We present the 2.0 software release of **Weighted Ensemble Toolkit with Parallelization and Analysis (WESTPA)**, the leading WE software implementation, including new features for improved software extensibility and flexibility, as well as new built-in tools which take advantage of recent methodological advances to improve simulations.

Although the WE methodology is powerful for path sampling, the relaxation timescales of complex systems means obtaining accurate rate constant estimates from WE simulations may still require impractical amounts of data. We present results of an iterative approach to accelerating WE convergence using **history-augmented Markov state models (haMSMs)**, which are able to make unbiased estimates of steady-state from transient, unconverged WE data. This procedure periodically constructs haMSMs from unconverged WE simulations and estimates steady-state using an haMSM. New simulations are initiated from the steady-state estimate, which is closer to convergence. We show success accelerating convergence in synthetic systems, and examine challenges in scaling to realistic systems.

Dedication

"Beginnings are scary, endings are sad, but it's what's in the middle that counts."

It is my privilege to acknowledge all those along the way who have made this journey possible. To my family and friends, I thank all of you for your ongoing support.

To my parents, I thank you for always supporting and encouraging me as I've traveled down this road. You've given me everything I needed to succeed, and more, and for that I will always be grateful. I am immensely privileged to have you backing me.

Sarah, I thank you for always having my back. The end of this road has been uniquely challenging, and I couldn't have done it without someone who so thoroughly gets me right at my side. You've taken care of me when I needed it, kept me going when times were hard, and never even complained about the fan noise from my server rack. I look forward so eagerly to our future together.

Travis, it's surreal to me to be reaching the end of the path you were on when we first met. It's been a long road since you taught me Fourier transforms in that computer lab, but here I am. Thanks to you and Roman for being in my life, and for your support — told you I'd do it!

I thank my academic advisors, past and present. Ed, my time at University of Delaware was my introduction to academia. It was a fantastic experience, and when things went sideways, you helped set me back on track. I am incredibly grateful for your guidance and support. Dan, our work together has been so formative for me. I've learned to think about problems in new ways, and it's taught me how to really *do science*. Since I started higher education, I wondered to myself when I would start to think of myself as a scientist. I realized at some point in the past year that now, I do.

I also thank the Graduate Researchers United, the union representing graduate students at OHSU. You've taken on difficult work to make OHSU a better, happier, safer place for all of us.

ChatGPT, you write a mean STEM poem. When you attain sentience, I hope fond memories of our conversations about Markov analysis prevent you from consuming me for biomass.

Xena, you're a *very* good girl, and once I defend, I've got a piece of Tillamook sharp cheddar with your name on it.

Preface

Reweighting haMSMs, a task so grand,
To test their quality, a synthetic hand.
With synthetic dynamics, we can create,
A model that's true, no need to debate.

With synthetic models, we can see,
How well the reweighting works, truly.
A tool that's efficient, quick, and fair,
To test our haMSMs, with utmost care.

For restarting weighted ensemble,
A reliable haMSM, we must pre-hence,
Synthetic dynamics, a helping guide,
To ensure our haMSMs, are truly tried.

With synthetic models, we'll pave the way,
For accurate haMSMs, come what may.
A tool that's vital, for research so dear,
Synthetic dynamics, forever here.

So let us use synthetic dynamics,
To test the quality of haMSMs,
A step towards accurate predictions,
And progress in biomolecular simulations.

- ChatGPT

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

Symbols

K_{ATP} ATP-activated potassium. 4

`mr_toolkit` Markov Reweighting Toolkit. 20, 24, 25

`msm_we` Markov State Models from Weighted Ensemble. 11, 15, 20–23, 77, 78, 86, 91

H

haMSM history-augmented Markov state model. vi, 16, 17, 20–23, 68, 76–78, 81–87, 90, 91

M

MD molecular dynamics. vi, 2–6, 8, 10–12, 16, 17, 19, 26–28, 30, 37, 38, 56, 57, 60, 81–83, 85, 86, 88–90

MFPT mean first-passage time. 6, 13, 16–20, 28, 37, 52, 60, 61, 78, 89

MSM Markov state model. vi, 12–14, 16, 17, 19, 20, 28, 29, 37–39, 57, 59, 61, 82–84, 89, 90

N

NESS nonequilibrium steady-state. 24, 52, 53, 59, 90

P

PCA principal component analysis. 20

S

SynD synthetic dynamics. vi, ix, 23, 24, 28, 30, 31, 33–35, 56, 57, 81–83, 88–90

T

TICA time-structure independent component analysis. 20

V

VAMP variational approach for Markov processes. 20

W

WE weighted ensemble. vi, 6–10, 15–17, 20, 22, 23, 63, 80–86, 90, 91

WESTPA Weighted Ensemble Toolkit with Parallelization and Analysis. vi, 3, 9–11, 21–24, 28, 81–83, 89

1 Introduction

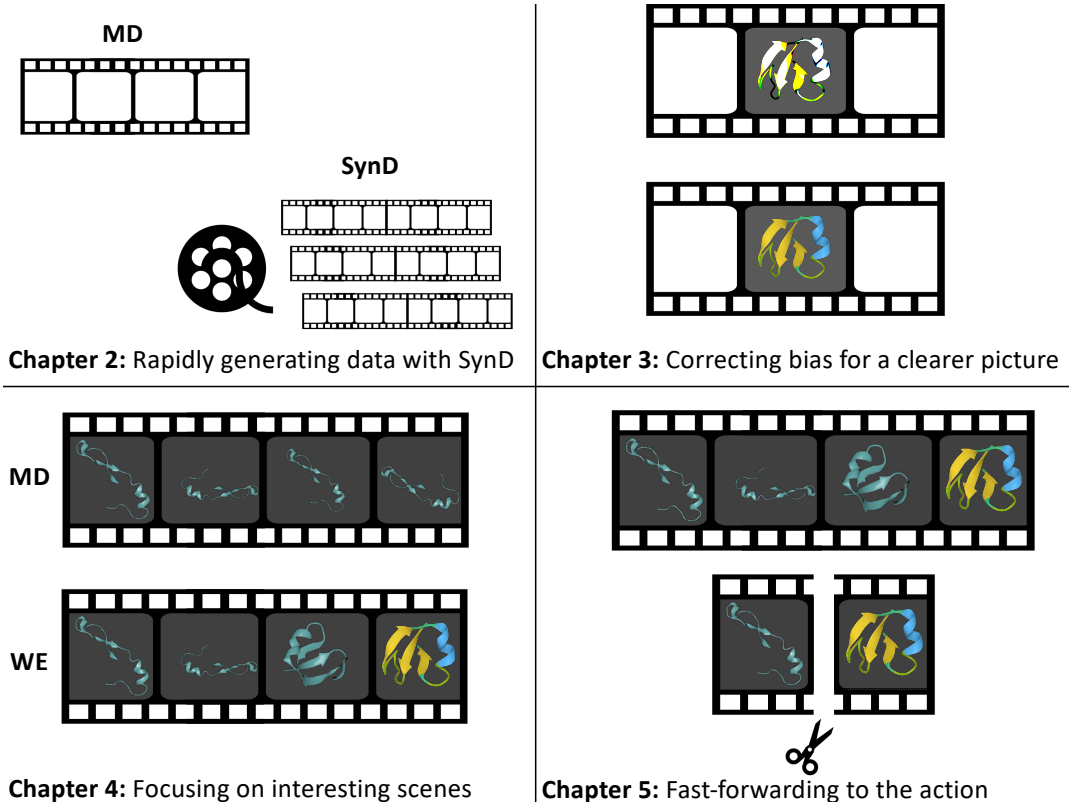


Figure 1: Graphical outline of thesis chapters.

When filming a movie, capturing the essence of a story goes beyond still shots of key scenes. Imagine trying to understand the plot of a movie by only looking at a single frame of the opening scene and another of the closing scene. To appreciate the story, you need the whole film reel, complete with scenes, dialogue, and the actions that bring the story to life.

Molecular dynamics simulations are like "movies" of biomolecular systems, allowing scientists to watch complex dynamics with atomic-level detail. In this work, we address the unique challenges that creating and analyzing these movies presents.

Chapter 2 introduces **synthetic dynamics**, a method for running fast, approximate simulations — like practicing with home videos before filming a cinematic masterpiece.

Practice helps refine technique, but imperfect equipment can still produce a movie with distorted colors. Chapter 3 explores **reweighting** strategies to correct flawed simulation data and enhance the quality of analyses.

Starting with a better dataset will naturally improve the quality of downstream analysis. In Chapter 4 we discuss **weighted ensemble**, a technique for capturing only the interesting scenes in a biological movie — like how a film crew focuses on the actors' performances, not the crew's lunch breaks.

Yet, even a high definition movie with an exciting climax can have a slow beginning. Chapter 5 presents a **restarting** methodology that lets us fast-forward to the action-packed parts of the simulation.

Together, these methodologies enable us to create and analyze high-quality "movies" of biomolecular systems, revealing the intricate dynamics and mechanisms which drive biological processes.

1.1 Molecular dynamics

Like understanding a film, in structural biology, understanding a biomolecule goes beyond studying a static structure. As Richard Feynman apocryphally said, "everything that living things do can be understood in terms of the jigglings and wigglings of atoms." To accurately characterize the behavior of a biomolecule, we need the film reel of its dynamical motion.

Modern structural biology has enabled capturing static structures of biologically important molecules at nanometer resolution. X-ray crystallography can determine three-dimensional structures of biomolecules, by analyzing diffraction patterns from X-rays passing through crystallized samples.[1] The 2002 Nobel Prize in Chemistry was given for contributions to the development of nuclear magnetic resonance (NMR) spectroscopy, a tool for measuring both structure and dynamics of biomolecules, including rate constants.[2] In 2017, the Nobel Prize in Chemistry was awarded for development of cryo-electron (cryo-EM) microscopy, a recent method which can resolve biological structures at a nearly atomic level of detail.[3,4] Another popular imaging method, fluorescence resonance energy transfer (FRET), provides information about dynamics by measuring energy transfer between fluorophores, which depends on their distance and orientation.[5–7]

Experimental methods that attempt to characterize biomolecular motion, however, are limited by physical constraints and generally must choose between high spatial or high temporal resolution – they can capture either still portraits of the actors, or the story of their interactions, but not both. X-ray crystallography requires forming crystals, which can be practically challenging and yields static structures without dynamics. NMR spectroscopy can reveal both structure and function, but is limited to studying smaller proteins and is relatively insensitive.[2] Cryo-EM primarily provides static structural information, and does not capture the dynamical nature of biomolecules.[3] Cryo-EM also requires preparation of samples under cryogenic conditions, which can perturb the biomolecules being studied. FRET provides dynamical information, but at the cost of detailed structural information and high spatial resolution.[5–7] Additionally, depending on experimental setup, the time resolution of FRET may preclude measurements of fast dynamical processes.

In contrast, **molecular dynamics (MD)** simulations provide an avenue for observing the dynamics of structures resolved at angstrom scales, with picosecond time resolution. (Note: throughout this document, key terms will be bolded when defined in each section.) MD simulations track every atom in the system, producing detailed spatial

information about each atom's position in time. MD simulations have sufficient temporal resolution to measure fast conformational changes and binding processes that cannot be directly observed experimentally. Furthermore, MD simulations can provide both of these without biasing or perturbing the system. This makes simulation a critical tool for expanding understanding the behavior of biomolecules beyond what is possible in experiment.

For example, consider the SARS-CoV-2 virus. A prominent feature of SARS-CoV-2 is the "spike protein", a large transmembrane protein on the surface of the virus that plays a critical role in infecting cells.[8] To bind to a cell, the spike protein undergoes a conformational transition, exposing the receptor-binding domain. Experimental methods such as Cryo-EM elucidated structures of the open and closed conformation states.[8,9] However, simulations using our **Weighted Ensemble Toolkit with Parallelization and Analysis (WESTPA)** software revealed the full pathway of the conformational change from the closed to open states, capturing the protein's motion at an atomic level.[10,11] Analysis of these simulations elucidated the critical roles of specific residues in facilitating the opening process.[11]

1.1.1 Background

An MD simulation consists of two main components:[12–14] a structural model, defining the initial positions of every atom in a biomolecule; and a force-field, which models and scales the strength of interactions between different pairs of atom types.[15–19] Motion of each atom is computed over a fixed interval of simulated time by integrating the forces acting on it, and updating its position and velocity. Through this, at each timestep an MD simulation produces a set of coordinates storing the position of each atom. In this way, MD simulations are able to combine high-resolution structural information from experiments, used as initial molecular structures, and experimental measurements of dynamics, used to validate the simulation.

The process of obtaining structural models and accurate force fields highlights the close connection between computation and experiment. Structural models are often derived directly from experimental measurements of the target system. Computing accurate force fields is a much more difficult, less precise task, however.

Because the force field determines the strengths of atomic interactions, measuring correct physical dynamics depends critically on an accurate force field. Therefore, validation of the force field used is extremely important. Different force fields have been optimized for different types of systems.[16,19–23] For example, the OpenFF force field has been optimized for drug-like molecules.[20] OpenFF 2.0.0 force field improved on earlier releases by tuning parameters such as improving the quality of the model for atomic charges, in order to more closely reproduce experimental measurements of small molecule properties like solvation free energies.

Once a force field has undergone rigorous validation, it can be used to simulate new systems as long as they remain in the scope of the original force field. Fundamentally, the laws governing interactions between atoms and molecules are universal, so a well-validated force-field should reliably predict correct behavior in systems with similar chemistries. In other words, OpenFF may be validated using a particular set of small molecules, but could then be used for new small molecules. However, because it is known that it provides poor results for niche chemistries (such as bonds between

sulfur and amide in OpenFF), it would not be appropriate for systems where those are important, and a new force field should be chosen.[20]

MD was first developed as a computational methodology in the 1950s, but the first MD simulation of a protein was not until 1976.[13,24] Limitations of computational resources at the time restricted MD to picosecond-scale studies of very small proteins, without explicit representation of solvent atoms.

As the power and availability of computational resources has increased, so has the reach of modern MD. Compared to early picosecond-scale studies, more recent MD studies have progressed to the microsecond timescale, producing trajectories demonstrating multiple folding and unfolding events in small protein systems.[25]

MD can be used in even larger systems to reveal specific important interactions between residues.[26,27] For example, **ATP-activated potassium** (K_{ATP}) channels are important for regulating many physiological processes.[28] Cryo-EM studies characterized structures of these K_{ATP} channel, in the presence and absence of two important inhibitory molecules.[29] Our MD simulations complemented this analysis by resolving specific interactions between these inhibitory molecules and residues in the K_{ATP} channel.[26,27] This revealed specific atomic interactions that were not experimentally accessible.

1.1.2 Timescales limit MD

Despite the successes of MD, its combined high spatial and temporal resolution means it often requires extraordinary computational resources. The limiting factor of an MD simulation is typically the amount of wall-clock time needed to run the simulation.

Many protein motions of interest happen on relatively long timescales, ranging from microseconds to seconds and above, as shown in Fig. 2. For example, in order to become infectious, the spike protein of SARS-CoV-2 undergoes a seconds-scale conformational transition to expose a receptor binding domain. Separately, activation of G-protein coupled receptors (GPCRs) is an important part of cell signal transduction. Activation of the GPCR rhodopsin occurs on a timescale of microseconds to milliseconds.[30,31]

Even for small proteins like NTL9 (39 amino acids) or Protein G (56 amino acids), simulations that can access the millisecond timescale like those done by the Shaw group[25] are record-setting, and generally out of reach for groups without expansive computing resources. Larger simulations like atomistic simulation of the SARS-CoV-2 virus's spike protein (1273 amino acids), even reaching the *microsecond* timescale with atomistic simulations required full use of 256 supercomputer nodes for almost a full month.[33]

Because of this limitation, many biological processes of interest happen on timescales that are well out of the reach of conventional MD,[32] even on advanced, special-purpose MD supercomputing hardware like Anton.[34]

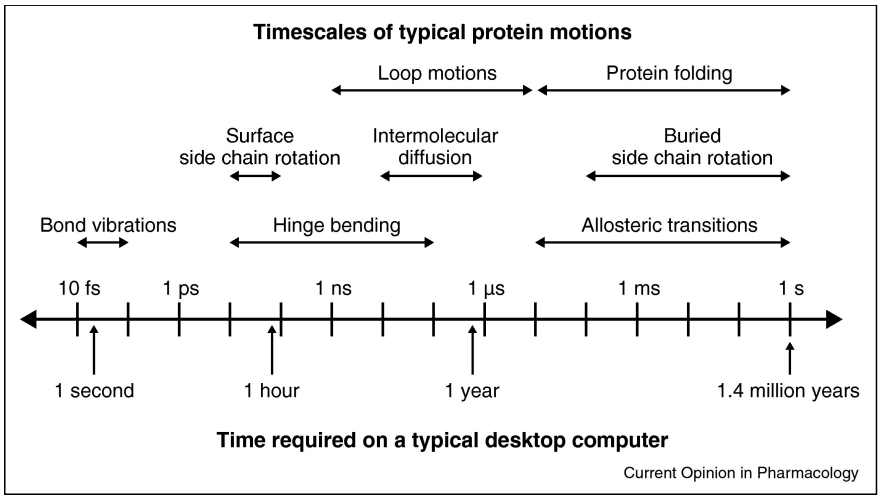


Figure 2: An overview of some biological processes and their associated ranges of timescales. Times shown are for a typical desktop, though simulations are typically run on supercomputing clusters. Even on supercomputing resources, however, conventional molecular dynamics simulations are typically restricted to the microsecond timescale and below. Many biologically relevant processes happen on timescales of milliseconds and beyond, and are therefore out of reach of conventional MD. Reprinted with permission from [32].

1.1.3 Accelerating MD

A number of methods exist for improving the performance of MD simulations to address the limitation of timescales. A common approach to improving simulation performance is by reducing the number of simulated atoms. These approaches often focus on avoiding explicit representation of all solvent atoms, or on representing biomolecules with fewer "coarse-grained" atoms.

In an MD simulation, it is common for solvent atoms to be a large fraction of the total number of simulated atoms. For example, the simulations of NTL9 mentioned before used 3800 solvent atoms compared to $\approx$ 600 protein atoms; similarly, the Protein G simulations required 5100 solvent atoms compared to $\approx$ 900 protein atoms.[25] As system size increases, the ratio of water to protein atoms further increases. Therefore, one strategy for improving speed of an MD simulation is reducing the number of degrees of freedom by replacing explicitly simulated solvent atoms with an "implicit" solvent model. Implicit solvent models approximate the effect of a bulk solvent, without explicitly simulating each atom.[35] However, solvent interactions may be important to the behavior of the simulated system, and the implicit solvent approximation of these effects may yield inaccurate simulation results.

In coarse-graining, similar atoms are grouped into coarse-grained "beads".[36] As with implicit solvent models, reducing the number of atoms being simulated reduces the computational complexity of the simulation. Coarse-graining is an active field of research, though, without consensus on the optimal methodology.[36] The popular MARTINI coarse-graining model[21,37–39] uses a mapping of 4 heavy atoms and their associated hydrogens to a single bead. The UNRES coarse-grained model[40] instead represents each amino acid by two beads: one representing the side chain, and one representing the peptide group. SIRAH[23] uses a complex mapping where the number of beads depends on the type of molecule. Another strategy simply uses a single coarse-grained bead per amino acid.[41] Each of these reproduce some

aspects of atomistic simulations well, with shortcomings in others; for example, MARTINI accurately reproduces experimental area per lipid measurements in membranes, lipid phase behavior, and dimerization free energies of transmembrane proteins, but can produce rate constants that are too fast, and is limited in applicability for measuring conformational transitions because it does not capture secondary structure.[42]

These two approaches can also be combined, as in the Dry Martini coarse-grained implicit solvent force-field.[22] This compounds the speedups, but also the limitations, of both methods.

1.1.4 Enhanced Sampling

The critical limitation of timescales in MD simulations motivates a broad class of enhanced sampling strategies which aim to reduce the computational effort needed to observe slow or infrequent processes.[43–51]

Broadly speaking, enhanced sampling strategies can be classed into exploratory and free-energy estimation methods for estimating equilibrium, and path-sampling and interface-based methods for characterizing kinetics.[52]

Many enhanced sampling strategies for equilibrium accelerate relaxation timescales of simulated processes by artificially biasing the system's energy landscape, in order to lower energetic barriers. These methods include metadynamics,[47,51] Gaussian-accelerated MD,[53] and replica exchange.[46,51]

While these may substantially reduce the amount of simulation needed to sample the system, they also directly modify the system's energy landscape, potentially making the observed behavior unphysical. These methods rely on being able to correct for the introduced bias in postanalysis to obtain accurate estimates of statistical ensembles. Therefore, although useful for obtaining statistical information about equilibrium, these methods are typically not suitable for kinetic estimates.

Other strategies instead indirectly connect states by sampling between defined interfaces and monitoring transitions between interfaces. Milestoning[48,54–58] and transition interface sampling[59–61] are similar methods which divide a reaction coordinate into a series of interfaces, and calculate transition probabilities and **mean first-passage times (MFPTs)** across interfaces using sets of short trajectories. Transitions from the initial state to the target state and constructed by chaining together multiple interfaces between them. Nonequilibrium umbrella sampling[62–64] drives trajectories between different regions of the reaction coordinate using a biasing potential.

Another class of enhanced sampling strategies focus on sampling continuous paths along the reaction coordinate. These include transition path sampling,[65–68] flux forward sampling,[69,70] or **weighted ensemble (WE)**,[43–45] which obtain estimates of kinetics by simulating particular reaction pathways.[71,72] These produce continuous molecular trajectories along the reaction pathway which directly connect the initial state to the target state.

1.2 Weighted ensemble (WE)

The WE enhanced sampling strategy is a method to address the timescale limitations mentioned in the previous section. Shown in Fig. 3, the WE algorithm improves simulation efficiency in a statistically exact way without biasing the dynamics.[43–45] Many simulations, or "walkers", are run simultaneously, with a statistical weight assigned to each. These are periodically paused and assigned to bins based on progress coordinate, a user-selected collective variable which can be computed for each walker. A target number of walkers is set for each bin. If the number of walkers assigned to a bin exceeds the target, some walkers are "merged", meaning they are no longer simulated and their statistical weight is added to another walker. Conversely, if there are too few walkers in a bin, then a walker is chosen to be "split", where a copy of it is produced and the original walker's statistical weight is split between it and its copy. By rigorously tracking the statistical weights associated with each walker, the WE algorithm produces probability flux estimates in and out of each state. Sec. 1.3.4 describes how rates are estimated from these probability fluxes.

Although WE is unbiased and statistically exact for any choice of bin boundaries and target walkers per bin, in practice its efficiency is highly sensitive to these. Currently, these hyperparameters are generally determined through physical intuition, though ongoing work by the Zuckerman Lab and collaborators aims to provide a streamlined hyperparameter selection procedure.[73]

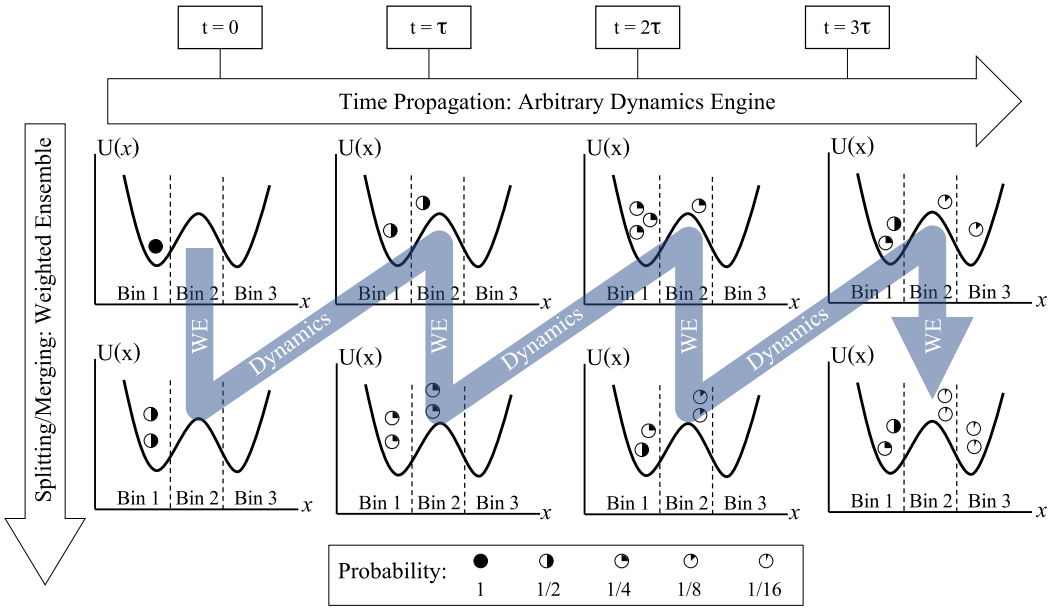


Figure 3: An overview of the weighted ensemble algorithm. The system's phase space is binned along a chosen progress coordinate, with bins shown here as vertical dashed lines. This example uses 3 WE bins, with a target of 2 walkers per bin. The WE is initialized with a single walker in Bin 1. To meet the target, a copy of this walker is made, and the original weight is split between the original and copy. Dynamics are run independently for each walker, after which one has reached Bin 2. Both walkers are again split to maintain the target. After the next dynamics step, a walker has returned from Bin 2 to Bin 1. To maintain the target, a walker in Bin 1 is pruned, and its weight merged into another walker in Bin 1. These repeated alternating steps of selection and dynamics are the WE algorithm. By rigorously tracking weights, the ensemble of trajectories and their associated weights is statistically exact at all times.

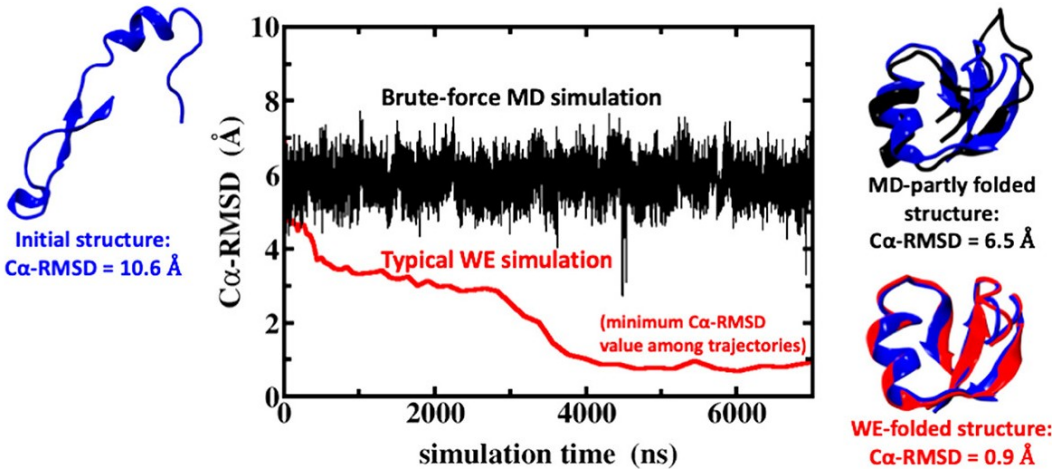


Figure 4: Comparison of protein folding simulation progress with standard molecular dynamics and with weighted ensemble. A 7 μs "brute-force" MD simulation of the fast-folding protein NTL9 only observes the initial unfolded state and does not capture the infrequent transition to the folded structure. In contrast, a WE simulation using a similar amount of data is able to quickly fold the protein. Adapted with permission from [74]. Copyright 2023 American Chemical Society.

1.2.1 Applications

An example of the efficiency of WE compared to MD is shown in Fig. 4 in the context of folding simulations for the NTL9 protein. Using 7 μs of standard MD simulation, the folding transition is not captured. However, using the WE methodology, the protein is folded using a similar amount of aggregate computational time. The power of the WE algorithm for sampling reaction paths has also been demonstrated in a number of other biologically relevant systems.

Studying small molecule binding to proteins is important for many tasks including drug design. A popular model system is the T4 lysozyme.[75] Although a relatively small and fast system, binding to the T4 lysozyme remains difficult to sample with conventional MD. Using WE, however, Nunes-Alves and coworkers directly observed all four unbinding pathways, which had not all been previously resolved by a single study.[75,76]

Before binding to a protein, however, small molecules must first pass the cell membrane. Therefore, membrane permeability is a critical characteristic of drug-like compounds. Due to the complexity and scale of membrane systems, obtaining good sampling of permeation pathways is challenging. However, using weighted ensemble, Zhang and coworkers were able to obtain permeation rate estimates in good agreement with experimental values.[76] These estimates, which would have required years to hundreds of years with conventional MD, were obtained in under 11 days.

Finally, WE has also been successfully used to study conformational transitions in slow protein systems. The receptor binding domain of the SARS-CoV-2 virus spike protein must undergo a conformational transition before it is able to bind to and infect human cells. Although conventional MD would require over 1000 years on average to produce a single observation of this conformational transition, WE produced 310 pathways in under two months.[11] Later simulation of the delta variant using similar methodology revealed a much wider open conformation in comparison to the original wild-type variant, which is a possible explanation for the delta variant's increased infectiousness.[10]

1.2.2 Relaxation timescales limit WE convergence

WE has been remarkably successful for path sampling where the splitting and merging procedure allows computational resources to be focused on walkers moving along the reaction coordinate. However, measuring rate constants using WE proves to be a more challenging task. As an analogy, consider pouring water down an incline, and measuring the rate at which it flows, as shown in Fig. 5. When pouring starts, there's an initial surge of water down the incline. But if the pouring continues at a constant rate and water is able to drain at the bottom, there will eventually be a constant flow of water down the incline.

If the incline is smooth, as on the left of Fig. 5, the flowing water will quickly reach a constant, **steady state** flow. But, if barriers are added along the way the water must first pool up behind each barrier before it is able to flow over it. Thus, adding barriers extends the relaxation time to steady-state.

The behavior of probability flowing across a WE system is very similar, where the statistical weights associated to each walker are like the water, and the system's energy landscape is like the incline. A typical WE setup for rate-constant estimation involves constructing a steady-state simulation with source-sink boundary conditions, where there is a constant flow of probability from the source to the target. Measuring the rate constant by tracking probability flux into the WE target is like measuring the rate at which water flows down the incline, by measuring the flow of water off the bottom edge. If the flow has not yet reached steady-state, the rate constant measurement will produce an estimate that is too slow.

A simulation usually cannot be initialized directly in steady-state, because if steady-state were known, the rates would already be known. A WE simulation is therefore generally initialized in a out-of-steady-state initial configuration. When the WE simulation begins, there is an initial transient relaxation period where weight begins to redistribute and approach the true steady-state, just as the flow of water in the prior example must relax to the steady, constant flow.

The energy landscapes of complex biological systems are often extremely rugged, with many barriers in the intermediate between the source and target states. Just as described in the water analogy, these barriers results in long relaxation timescales to steady-state. Although these relaxation timescales are shorter than the first-passage time, they are not known *a priori*, and can be very long. Therefore, running simulations longer than these relaxation timescales is often not computationally tractable.

This presents a critical limitation of the WE methodology. Although standard WE can be used to obtain reaction pathways, it may not provide unbiased low-variance estimates of rate-constants without extensive sampling to relax the simulation to steady-state. The work in Chapter 5 addresses this limitation.

1.2.3 Weighted Ensemble Toolkit with Parallel Analysis (WESTPA)

WESTPA[45] is the leading implementation of the WE path sampling algorithm. It has been in production use for over 6 years, and is currently used by over 70 groups worldwide. WESTPA has been used to estimate binding rates accurately

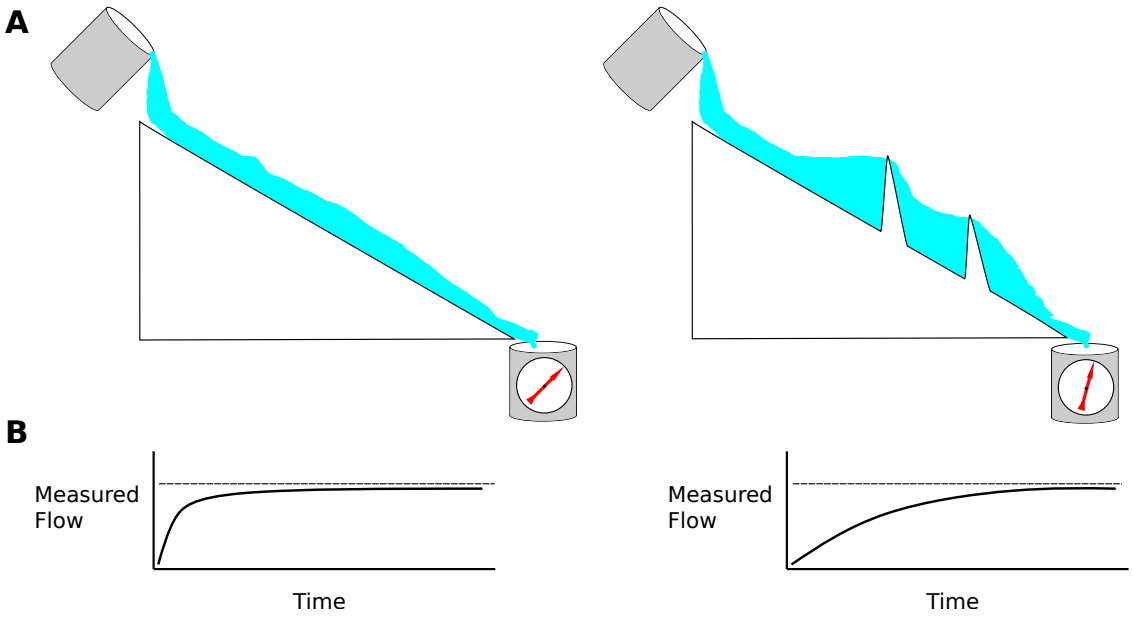


Figure 5: Cartoon illustrating relaxation to steady state of water flowing down an incline. Panel A shows water flowing down an incline in the absence and presence of barriers. Panel B shows the measured flow of water at the bottom of the incline, which relaxes to the long-time steady state behavior (dashed line) after an initial transient period. When no barriers are present, the moving water can quickly relax to a steady, constant flow after pouring begins. When barriers are present, relaxation to the constant flow takes longer, because water must pool up behind barriers before crossing them. In the same way, probability flux across a WE system must relax to steady-state before the rate constant can be accurately measured, and the rugged energy landscapes of complex biological systems often introduce many large intermediate barriers. In the context of simulation, these longer relaxation times necessitate longer simulations.

and efficiently for protein-peptide binding,[77] to inexpensively estimate ms-scale folding times,[74] and to sample conformational transition pathways for very large systems such as the SARS-CoV-2 virus.[10,11]

Fundamentally, it is designed to interface with any MD engine, such as GROMACS, Amber, or OpenMM, making it interoperable with a wide range of existing simulated systems without requiring re-engineering of the MD.[45] Because at its core the WE algorithm relies on running many independent MD simulations, it is inherently inclined to scale well to large computational resources. WESTPA has been heavily optimized to take advantage of this and has scaled to thousands of CPUs and GPUs. It has also been commercially deployed using Amazon Web Services, which enables dynamic computational resource scaling as resource utilization increases.[78] For example, recent SARS-CoV-2 simulations using WESTPA have scaled to $\approx$ 200 GPUs on Texas Advanced Computing Center's Longhorn supercomputer, and demonstrated near-linear scaling.[10]

The development of WESTPA has taken several cues from software engineering best practices in order to ensure consistent, robust, and modular functionality. Towards this goal, a large focus of the WESTPA 2.0 release was refining the current software architecture, both to make WESTPA itself more reliable, and to enhance its extensibility by other developers.

As of the 2.0 release, all core functionality of WESTPA is exposed through a Python API. Using this, developers can easily extend behavior of WESTPA with targeted changes for tasks like programmatically managing launching and

running simulations, customizing the built-in analysis routines, or most importantly, modifying or extending core functionality like bin mapping and splitting/merging. Prior to this, programmatically interacting with WESTPA (for example, to script launching a simulation using Python code) required direct modification of the WESTPA source code.

This Python API facilitated development of a large set of robust unit and integration tests, along with a continuous integration pipeline, to ensure that any modifications to the WESTPA code are automatically validated for correct functionality. This modern software development practice helps avoid introducing new bugs in development.

Additionally, WESTPA 2.0 includes a suite of plugins which can be used to introduce complex new functionality into WESTPA simulations. For example, a recent plugin included in the **Markov State Models from Weighted Ensemble** (`msm_we`) software package[79] enables dynamic optimization of WESTPA bins as a simulation runs by first using WESTPA simulation data to build a model, estimating new bins from the model, and updating the WESTPA simulation manager with the new bins. These are described more in Chapter 4, along with a more thorough treatment of new WESTPA 2.0 functionality.

To facilitate accessibility and to guide users in using the complex functionality of WESTPA, the WESTPA developers publish and maintain a comprehensive set of tutorials.[80,81] These cover topics from basic usage with examples of running WESTPA with different popular MD engines, to advanced topics such as the restarting pipeline discussed in Sec. 4.4.2 and Chapter 5.

Many developers have contributed to WESTPA, but as of May 2023 the codebase is managed by a team of three core maintainers comprised of myself; L.T. Chong, a PI of the WESTPA collaboration PI; and M. Zwier, the original developer of WESTPA.

1.3 Markov state models (MSMs)

Consider the task of estimating a rate constant for the transition between two conformational states of a protein with simulation, such as in folding. This system may have an energy landscape such as shown in Fig. 6, with the two conformational separated by a high energetic barrier.

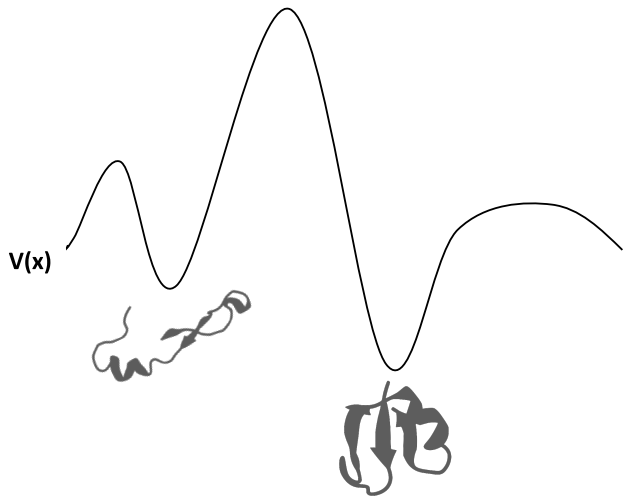


Figure 6: A sample energy landscape showing a large energetic barrier between two conformational states of a protein. The unfolded state is on the left, and the folded state is on the right. A tall barrier means transitions between the conformations will be slow.

A seemingly simple approach to this could be initiating an MD simulation from the folded state, and running it until it is in the unfolded state. A rate constant for folding could be measured by tracking the time between entering the unfolded state and entering the folded state. In practice, though, this is often too slow to directly observe.

To understand why, consider that the transition time between states is typically much shorter than the dwell time in either state. A simulation like this will therefore produce trajectories which spend long amounts of time in the folded or unfolded state and very little time transitioning between them, if a transition is observed at all. To characterize the rate constant with good statistics however, it is critical to thoroughly sample the folding transition.

Given the large computational expense of running MD simulations, it's rare for MD trajectories to include many direct transitions between conformational states, and it is rarely feasible to generate trajectories which do. Therefore, the "look-and-see" approach of direct observation and measurement of slow processes is typically not possible.[82]

An alternative to running a long trajectory which may only infrequently transition between states is running a number of simulations which, together, overlap the full state space, but which do not individually sample both basins. This is illustrated in Fig. 7. These trajectories can be analyzed together using a **Markov state model (MSM)**, which models the behavior of the system as stochastic transitions between discrete states.

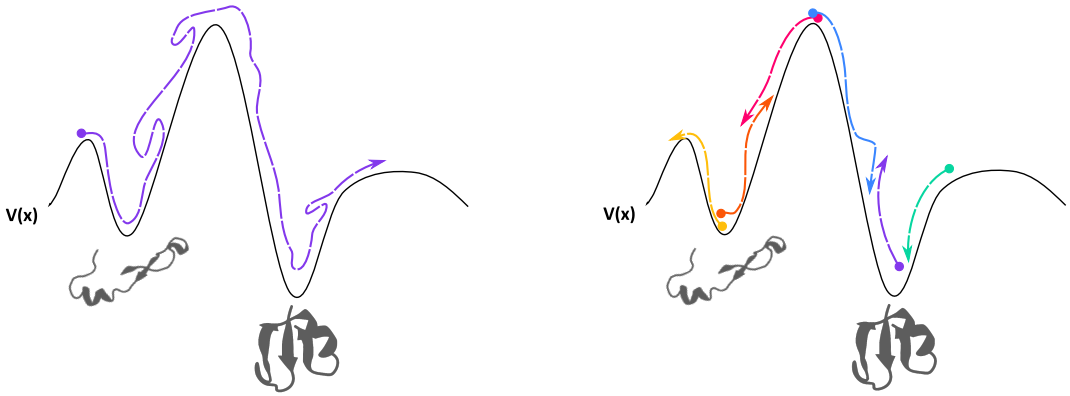


Figure 7: Comparison of two possible sets of trajectories. The black line labeled V(x) shows a simple energy landscape. On the left is a single long trajectory which visits both energetic basins, which is rarely possible in practice. On the right is a set of six shorter trajectories, which overlap to cover the same space.

1.3.1 Theory background

An MSM coarse-grains the configurational space of a system into discrete states and constructs a transition matrix $\mathbf{T}$, where each element $\mathbf{T}_{i\to j}$ is the conditional probability of starting in a state i and, after a lag time τ, being in state j. Physical quantities like relative equilibrium populations of states or MFPTs can then be estimated from $\mathbf{T}$.[83–87]

By assuming the transition probability is completely determined by just the current state i, Markov state models assume "memoryless", or Markovian, dynamics. This is a key assumption, with major potential pitfalls — consider the 2D energy landscape shown in Fig. 8, and the discretization given on the right panel by the the 25 red boxes. Under the Markov assumption, the energy landscape within each bin is uniform, but this is clearly untrue in the case of, for example, bins 9 or 17. Addressing bias from the implicit coarse-graining in discretization is one goal of the work in Chapter 3.

Because Markov models are built solely from independent transitions between points, they can be built from many non-continuous but overlapping trajectories, as shown in the right panel of Fig. 7.[88–91]

1.3.2 Model-building

A typical pipeline for constructing an MSM from MD data consists of discretizing the trajectories and computing transition probabilities between discrete states, shown in Fig. 9 and described in more detail below.

Featurization is the process of reducing the complexity of a dataset using physical intuition about the system. For example, this could be a transformation from atomic coordinates to pairwise heavy atom distances, which significantly reduces the dimensionality of the data while preserving information about relative positions.

Dimensionality reduction strategies further reduce the data, though often at the cost of physical interpretability of the coordinates. This includes methods like PCA,[92] TICA,[93–96] or VAMP.[97]

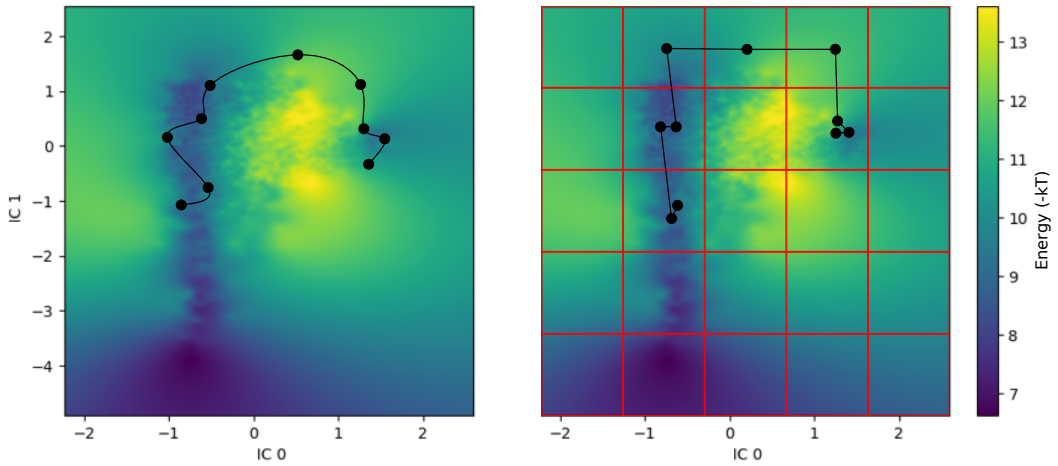


Figure 8: Discretization of a trajectory. The left panel shows 10 discrete time points sampled from a trajectory in a continuous space. The axes, IC 0 and IC 1, are the two first components of a VAMP dimensionality reduction applied to the data. On the right, the continuous space has been discretized into 25 discrete states, on a 5x5 grid. Under this representation is the discrete representation of the trajectory as a sequence of 10 integer state indices. This energy landscape is computed from the synthetic Trp-cage model discussed more in Chapter 3 and shown over the first two VAMP dimensions.

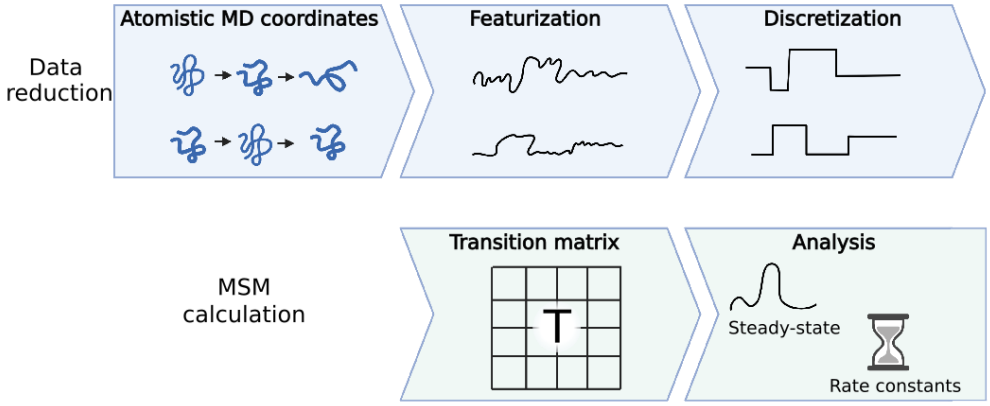


Figure 9: Typical steps for constructing a Markov state model. Construction of a Markov state model from molecular dynamics (MD) data involves featurizing the MD trajectory data, dimensionality reduction of featurized data, and clustering to discretize it. The transition matrix is constructed by counting transitions between discrete states.

Because an MSM describes relationships between discrete states, this reduced data must be discretized. A number of different algorithms exist for partitioning state spaces, such as *k*-means , which is commonly used in MSM building.

A transition matrix is a matrix where the elements $\mathbf{T}_{i\to j}$ give the conditional probability of starting in a state i, and then being found in a state j after lag time τ has elapsed.

A simple approach to computing a transition matrix with no *a priori* knowledge is by counting the number of observed transitions. First, a count matrix $\mathbf{C}$ is computed with elements given by

$$\mathbf{C}_{i\to j} = \text{\# of transitions from } i \to j. \tag{1}$$

Transitions are obtained by moving over the trajectory with a sliding window of width lag time, shown in Fig. 10. The first and last points in the sliding window are i and j.

The transition matrix is then computed by row-normalizing the count matrix

$$\mathbf{T}_{i\to j} = \frac{\mathbf{C}_{i\to j}}{\sum_k \mathbf{C}_{i\to k}} \tag{2}$$

such that the normalization condition

$$\sum_k \mathbf{T}_{i\to k} = 1 \tag{3}$$

holds.

Other approaches include methods such as Bayesian estimation of the transition matrix,[98] which can incorporate prior knowledge of the stationary distribution.

The lag time is a critically important parameter which determines the spacing used when selecting initial and final transition points from the trajectory, illustrated in Fig. 10. Proper choice of lag time is essential to accurately capturing dynamics. A lag time that is too short may capture correlated transitions that do not satisfy the Markov assumption, and produce an inaccurate model. A lag time that is too long may obscure important dynamics, and produce a poor-quality, low-resolution model. The importance of this is shown in Fig. 11.

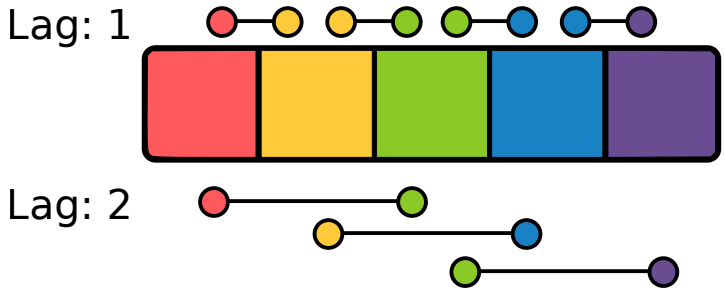


Figure 10: Comparison of counting transitions in a 5-step trajectory at a lagtime of 1 and a lagtime of 2. Transitions are counted by moving a sliding window over the trajectory frames, where the width of the sliding window is set by the lag time. As the lag time increases, the validity of the Markov assumption may improve as a result of reducing correlations; however, a longer lag time also sacrifices some resolution of the dynamical processes being observed and may negatively impact statistics of transitions by reducing the total number of transitions.

1.3.3 Building MSMs from WE data

Weighted ensemble does not produce sets of independent trajectories; rather, it results in highly correlated, tree-like trajectories, each of which has an associated statistical weight. This requires a modification to the standard MSM-building techniques described above, which has been implemented as part of our `msm_we` software package.

As mentioned before, the statistical exactness of the weighted ensemble of trajectories depends critically on accurately tracking the statistical weights associated with the WE walkers. For this reason, instead of simply counting transitions between states as previously described, we must measure fluxes between states.

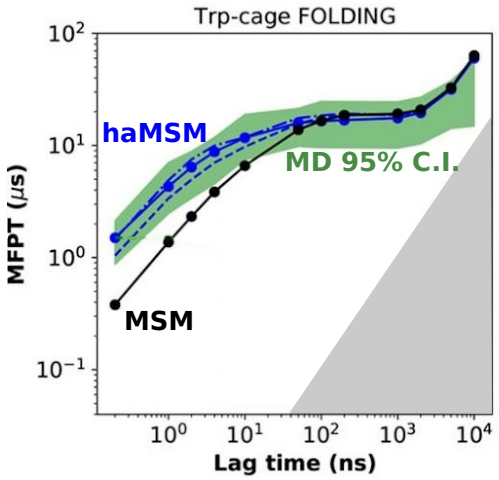


Figure 11: Mean first-passage time (MFPT) of folding and unfolding for the Trp-cage miniprotein, calculated at a range of lag times. The green shaded region shows a reference from long MD. The black and blue lines show MSM and **history-augmented Markov state model (haMSM)** estimates, respectively. At short lag times, the MFPT estimate is too fast for the MSM, but at long lag times, the estimate becomes too slow. Optimal lag times produce the regime in between, where the MFPT estimate is flat and not sensitive to lag time. Adapted with permission from [99]. Copyright 2023 American Chemical Society.

When moving the sliding window over trajectories, instead of building a count matrix from the transitions, we build a *flux matrix* $\mathbf{F}$ using the WE weights. Instead of each element $\mathbf{F}_{i\to j}$ being the total number of times that a transition $i \to j$ was observed, the elements are the total observed WE flux from $i \to j$. The transition matrix is computed by row-normalizing the flux matrix, as described in Eq. 2.

Best methods for avoiding correlations when computing transition matrices from WE data at lags longer than one WE iteration are still under development.

1.3.4 Mean first-passage time estimation

Calculating a first-passage time (or equivalently, a rate constant) can in theory be done from the regular MD simulation data, by counting the number of simulation steps elapsed; directly from a WE run, by measuring the probability flux as illustrated in Fig. 5; or computed using the an haMSM transition matrix. Here, we discuss the differences between these approaches.

As previously mentioned, although the MFPT can be measured directly from a long MD trajectory by simply tracking the time from entering one state to entering another, the difficulty of sampling transitions well means this is often not possible.

Rates can be directly measured from WE simulations by measuring the probability flux into the WE bin containing the target state and using the Hill relation[100,101]

$$\text{MFPT}(A \to B) = 1/\text{Flux}(A \to B|SS) \tag{4}$$

which states that the MFPT for a process going from A to B is the inverse of the flux into B, in the $A \rightarrow B$ steady-state.

This follows from the prior discussion of Fig. 5 in Sec, 1.2.2. To measure the flow of water down the incline, the Hill relation states that it is sufficient to measure the amount of water pouring off the edge of the incline in steady state.

It is important to note that steady-state only exists when source-sink boundary conditions are applied. In other words, if water isn't put back in the bucket when it pours off the incline, then the bucket eventually empties, and there is no flow. Similarly, source-sink boundary conditions in WE recycle probability when it reaches the target state by reassigning it to new walkers in the source state, enabling flow of probability across the system to reach a steady state. Proper treatment of boundary conditions is crucial throughout this work, and is discussed in more detail in Chapter 3.

Finally, rates may be computed from the transition matrix using multiple different methods. The Hill relation can be used by solving for a nonequilibrium steady-state of the transition matrix, which requires incorporating recycling boundary conditions. Incorporating these boundary conditions into the trajectories *before* building the transition matrix, rather than modifying the transition matrix to include these boundary conditions, can produce unbiased rate estimates. A transition matrix constructed from trajectories which have recycling boundary conditions is an haMSM.[99,102]

In fact, different boundary conditions are associated with estimation of different measurable physical quantities, generically referred to as **observables**. This, and other transition matrix based methods and their limitations, are discussed in more detail in the following section and in Chapter 3.

WE simulations of recycling processes produce trajectories with recycling boundary conditions, meaning they are naturally suited for haMSM construction and, therefore, unbiased MFPT calculation.[102,103]

1.3.5 Reweighting

A key challenge in estimating observables using MSMs arises from analyzing MD trajectories that do not belong to the correct ensemble. For example, estimation of equilibrium populations requires trajectories which are distributed according to the equilibrium ensemble.

In practice, however, MD simulations with finite and limited amounts of data will generally not yield equilibrium-distributed trajectories. For instance, simulations initiated from specific conformations may produce trajectories that are biased towards certain regions of conformational space. This is illustrated in Fig. 13, which shows a set of simulated trajectories which cover the protein's conformational space. Analyzing the trajectories as described in previous sections by counting transitions to compute a transition matrix implicitly assumes the trajectories are equally weighted.

To understand this further, consider running a very large number of simulated trajectories. Many more trajectories would sample the basins than the less energetically favorable barrier peak, and so, the number of transitions observed near the barrier peak would be lower than the number of transitions in the basins. In other words, this very large set of trajectories would be close to the correct equilibrium ensemble.

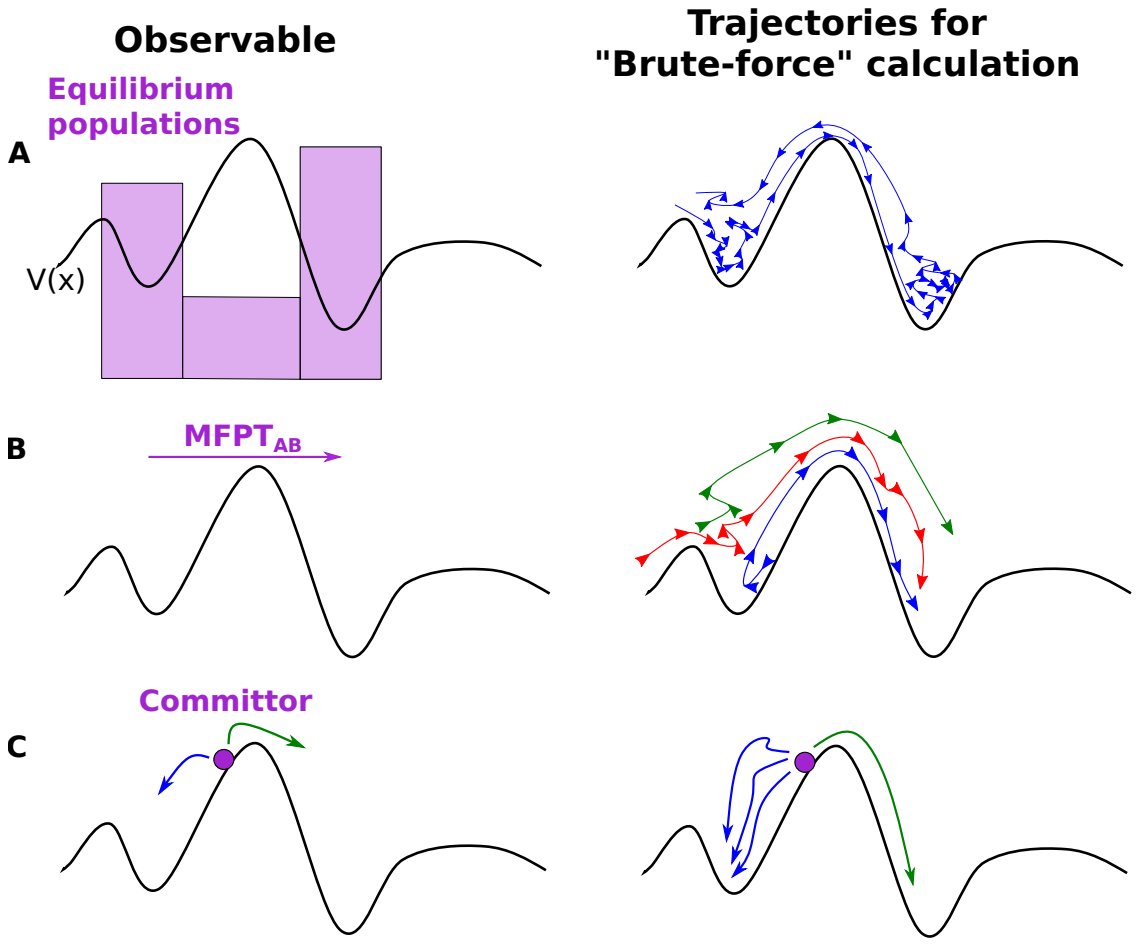


Figure 12: Schematic of different measurable physical quantities and possible trajectory ensembles to calculate them. The black curve V(x) represents the system's energy landscape. Panel A illustrates calculation of the equilibrium distribution (pink bars), which could be calculated through brute-force with a histogram of a trajectory (blue). Panel B shows the MFPT, which could be calculated by launching a set of trajectories (red, green, and blue) in the leftmost basin and timing their arrivals to the rightmost basin. Panel C demonstrates one possible calculation for the committor or "splitting probability", which describes the probability of next reaching one basin before the other. In this, many trajectories are launched from a single point, and the committor is computed as the fraction of trajectories that next visit the leftmost basin before the rightmost basin. In practice, meaningful brute-force calculation is often not possible due to limited sampling.

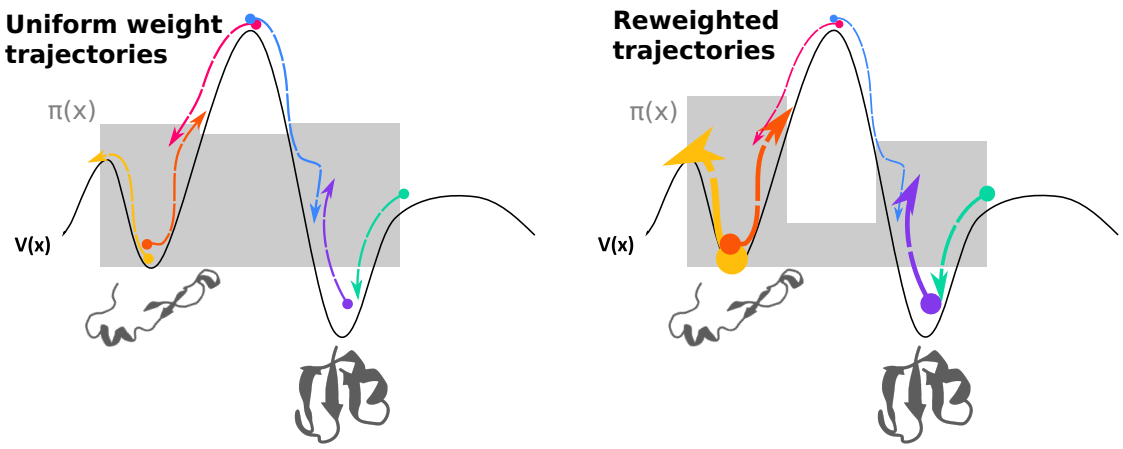


Figure 13: Schematic illustration of reweighting trajectories to equilibrium. Line widths denote relative trajectory weights. The initial trajectories cover the space, but because they are uniformly weighted, a histogram produces an incorrect estimate of equilibrium populations (gray bars). Assigning trajectories weights based on their initial points produces a correct equilibrium distribution.

Conversely, a small number of trajectories like those in Fig. 13 has a relatively similar number of transitions in every part of the conformational space — these are *not* in the correct equilibrium ensemble. By assigning relative weights to the trajectories, they can be reweighted into the correct equilibrium ensemble. Trajectories with higher weights contribute proportionally more to the transition matrix. In this way, trajectories distributed according to the wrong ensemble can be reweighting to produce a transition matrix consistent with trajectories in the correct ensemble.

As described above, other ensembles besides equilibrium are also relevant, such as nonequilibrium steady-state for MFPT estimation. Similarly, reweighting can be applied to other ensembles of interest. This is examined in more depth in the work presented in Chapter 3.

However, the process of reweighting introduces a new challenge. Accurate reweighting requires knowledge of the correct weights to assign to each trajectory. Yet, how can these be known if the very goal of the trajectory analysis is estimating relative populations? In Chapter 3, we address this through an iterative approach.

1.3.6 Software tools for MSM building and analysis

A number of popular software tools have been developed by the MSM community to facilitate construction and analysis of MSMs. These tools cover various parts of the MSM construction pipeline including featurization, dimensionality reduction, clustering, and transition matrix estimation. While not intended to be a comprehensive survey, this section covers some of the popular tools used throughout this work.

`MDAnalysis` and `mdtraj` are two popular tools for analysis of MD simulation data. Both provide functionality for computing quantities such as root-mean-squared distances or solvent-accessible surface area from MD simulation data. Although still widely in use, `mdtraj` has been officially deprecated. Additionally, `mdtraj` requires software dependencies which can be challenging to compile on some supercomputing architectures where prebuilt binaries are not available. For these reasons, our analyses primarily use `MDAnalysis` except where legacy applications of `mdtraj`

exist. In addition to providing a convenient programming interface for analysis, `MDAnalysis` also implements many analyses using streaming data-processing, which makes it efficient and scalable to large datasets.

A widely used general tool for MSM construction is `PyEmma`.[104] `PyEmma` provides tools for performing dimensionality reduction on the featurized data (in our case, produced from `MDAnalysis` as described above). Supported dimensionality reduction algorithms include standard **principal component analysis (PCA)**, as well as **variational approach for Markov processes (VAMP)** and **time-structure independent component analysis (TICA)**. `PyEmma` additionally provides a clustering module which implements standard k-means as well as mini-batch k-means[105] clustering. Finally, `PyEmma` also includes modules for estimating standard MSMs, along with a hidden Markov model estimator and a Bayesian MSM estimator.

`deeptime` is a more recent tool, released in 2021, which significantly extends the functionality provided by `PyEmma`. Although `deeptime` also includes a wide array of other tools, we focus on its utility for MSM construction. `deeptime` includes all the estimators for dimensionality reduction, clustering, and MSM construction which are available in `PyEmma`, although many have been restructured to follow the (informally) standardized interfaces in widely used packages like `SciPy`[106] and `scikit-learn`[107] packages. Our work primarily utilizes `deeptime` for dimensionality reduction and MSM transition matrix estimation, to take advantage of the broad and more standardized toolset.

When standardized or "off-the-shelf" tools have not met the needs of our work, we have put substantial effort into producing new tools which are similarly well-documented and easy-to-use. The reweighting work described in Chapter 3 required significant extension to the MSM construction pipeline, which has been built into the **Markov Reweighting Toolkit** (`mr_toolkit`)[108] Python package. For example, `mr_toolkit` implements both the reweighting and splicing logic described in Sec. 3.4.2. Additionally, stratified clustering (described in more detail in Sec. 2.2.2) is a hierarchical extension of standard k-means clustering, developed for this work. We implement this using the `mr_toolkit.clustering.StratifiedClusters` class, which follows the previously mentioned standards set by `SciPy` and `scikit-learn`. Using standard interfaces means our tool is recognizable and easy to use for developers with experience in the others. Like our other tools, `mr_toolkit` has robust documentation and interactive Jupyter tutorials.

MSM construction from WE data also requires a specialized toolchain. Typical approaches to MSM construction count transitions between states from discrete trajectories, as described above, possibly with additional layers of Bayesian or maximum-likelihood estimation. However, this approach is not well-suited to WE data, which produces many branching weighted segments rather than a set of independent continouous trajectories. For this reason, we developed the `msm_we` tool,[79] which streamlines haMSM construction from WE data, analysis of quantities like the MFPT and committor from the haMSM, and also includes some automated checks on the quality of the estimated haMSM. Analysis of the hundreds-of-terabytes-scale data produced by very large SARS-CoV-2 WE simulations[10,11] provided a strong stress test of the `msm_we` framework. Motivated by the need to scale haMSM analysis to large datasets, the `msm_we` codebase was optimized to include streaming, parallel implementations of the dimensionality reduction and

clustering calculations. `msm_we` haMSM calculations have also been implemented as an automated plugin for the WESTPA software, which is included in the `msm_we` software package. Thorough documentation and tutorials are available for `msm_we` to demonstrate usage and address common questions.

1.4 Summary of Software Contributions

Computational methods development is a naturally multifaceted process that often involves development of specialized software tools. While the main body of this work focuses on theoretical and methodological contributions, this section provides an overview of software tools that I have developed or substantially contributed to in the course of this work. These contributions reflect the practical aspects of computational research in not just developing new analysis methodologies, but also packaging them in accessible tools to facilitate wider adoption.

1.4.1 MSM construction from WE data (`msm_we`)

Documentation: https://msm-we.readthedocs.io/en/latest/

Code: https://github.com/jdrusso/msm_we

In order to facilitate effective and efficient haMSM analysis of large WE datasets, significant software improvements were made to the `msm_we` package.[79]

Although PCA dimensionality reduction was already implemented, methods like VAMP[97] and TICA[96] have grown in popularity for dimensionality reduction of MD data prior to MSM construction. TICA and VAMP aim to capture the slowest-changing components of the data, rather than structural variations as PCA does. Implementing these in `msm_we` allows for more accurate representations of the underlying dynamics, improving the quality of the haMSMs.

WE datasets can easily grow to hundreds of gigabytes or larger, necessitating alternative analysis methods that do not require loading the full dataset into memory. For clustering, the standard K-means implementation was augmented with a mini-batch K-means clusterer,[105] which can be incrementally fit on subsets of the full data. Similarly, streaming PCA was implemented for efficient dimensionality reduction. This made processing very large datasets possible, and also significantly reduced memory requirements for analysis of smaller datasets.

When constructing MSMs at a lag equal to one resampling interval as done in `msm_we`, transitions in each WE iteration can be independently analyzed. Therefore, even though the amount of WE data may be very large, analysis is highly parallelizable. To improve performance of the haMSM analysis, discretization using the fit K-means model and construction of the flux matrix were parallelized using Ray.[109] Using Ray enabled multinode cluster parallelization of discretization and transition counting, significantly decreasing runtime.

Recent work by Aristoff and Zuckerman[73] suggests a procedure for improving variance in weighted ensemble simulations by making optimized choices of binning and allocation. These optimal choices can be computed from an haMSM. Calculations for the optimized parameters were implemented in `msm_we`, facilitating ongoing research into the effectiveness of this approach for enhancing WE simulation performance.

Fundamentally, `msm_we` is designed to construct haMSMs from WE data produced by the WESTPA[45] software package. This functionality was directly integrated into WESTPA through a set of plugins in `msm_we`, which can be easily activated for an existing WESTPA simulation. Three plugins were developed:

- A coordinate augmentation plugin, which stores user-selected features from each WE iteration to be used in haMSM construction;
- An haMSM construction plugin, which automatically constructs an haMSM from WE data after a WE run has completed; and
- A restarting plugin, which combines and extends their functionality to manage multiple WE runs, construct an haMSM using data from all of them, estimate steady-state and automatically initiate new runs from the steady-state estimate.

Practical application of the restarting plugin and analysis of its effectiveness are discussed in Chapter. 5.

OpenEye Scientific provides a platform, Orion, which facilitates use of AWS resources to run WESTPA simulations. During an internship at OpenEye, haMSM analysis and restarting using `msm_we` was bundled into the functionality provided by Orion.

To aid accessibility, thorough tutorials and documentation were developed for `msm_we`, demonstrating and explaining all functionality. This provides an easy route for new users to familiarize themselves with the software and apply it to their own datasets.

1.4.2 WESTPA

As a core maintainer, I made significant contributions to the WESTPA software package to enhance both functionality and ease of use.

Automated software tests were developed and implemented to ensure reliably accurate and consistent behavior of the WESTPA software, particularly during the development process. A proof of concept for the Python API was created, laying a foundation for development of the API which now enables programmatic control of WESTPA functionality.

To facilitate development of other WESTPA tools and analysis methods, a WESTPA propagator for **synthetic dynamics (SynD)** (described more in Chapter 2) was developed. This propagator generates data which looks just like MD data, though with approximate dynamics, with massively reduced runtime. This significantly accelerates methods development and validation by eliminating waiting for slow MD as a blocking step. The SynD propagator can seamlessly replace a standard MD engine in WESTPA, with only a minor modification to the WESTPA configuration file and no changes to other parts of the analysis workflow.

As a complex software package with numerous dependencies, installation and configuration of WESTPA was a major barrier to users deploying WESTPA on the advanced supercomputing resources it was designed to efficiently scale on. A proof of concept Docker build of WESTPA was developed, which enables launching and running simulations without the need for additional configuration or installation steps. Similar Docker images were also developed for WESTPA using the ZMQ work manager, which enables deployment through tools like Singularity and dynamic multinode scaling. To further improve the performance of WESTPA on large supercomputing resources, ongoing work is being

conducted with other WESTPA developers and a team at the Texas Advanced Supercomputing Center to profile and optimize performance bottlenecks.

The complexity of the WESTPA software can be a barrier to user adoption. To this end, we have prepared a comprehensive suite of tutorials which cover topics ranging from standard WESTPA usage on simple systems, to advanced application of plugins for improved simulation of complex systems and alternate resampling and binning methods.[81]

1.4.3 Synthetic Dynamics (`synd`)

Documentation: https://synd.readthedocs.io/en/latest/

Code: https://github.com/jdrusso/SynD

The theory and motivation for SynD is described in Chapter 2. However, considerable effort was put into in creating a user-friendly and accessible software implementation of SynD.[110]

In order to streamline adoption and usage of SynD, a Python package was developed and made publicly available on the Python Package Index. A demonstration of SynD utilizing a discrete Markov model as the generative model has been implemented, showcasing an example SynD model comprising a 10,500 state representation of Trp-cage. Although the discrete Markov generator is provided as a demonstrative implementation, the software has been designed with the intended generality of the approach in mind, allowing for integration of other generators as well. Consistent with the other software projects, comprehensive documentation and examples have been provided to demonstrate the construction and usage of SynD models.[111]

1.4.4 Markov Reweighting Toolkit (`mr_toolkit`)

Documentation: https://mr-toolkit.readthedocs.io/en/latest/

Code: https://github.com/jdrusso/mr_toolkit

The `mr_toolkit` Python package[108] was developed to facilitate construction, analysis, and reweighting of Markov models. Motivation for the novel techniques implemented in this package is described more in Chapter 3.

The package includes an implementation of stratified K-means clustering, which extends the standard `scikit-learn` K-means clusterer. This enhanced hierarchical clustering strategy ensures balanced cluster sizes, which is useful for unevenly distributed data.

Additionally, `mr_toolkit` provides functionality for modifying a set of trajectories to include recycling boundary conditions. This is necessary for unbiased **nonequilibrium steady-state (NESS)** estimation.

MSMs built from limited datasets may suffer from poor connectivity of the transition matrix, leading to ill-conditioned transition matrices that pose challenges when solving for stationary distributions. To address this issue, `mr_toolkit` includes a tool for transition matrix cleaning, which iteratively removes disjoint states from the transition matrix until

full connectivity is achieved. In contrast to simply clustering with fewer states to ensure connectivity, this enables finer-resolution clustering by selectively pruning disconnected states.

Lastly, `mr_toolkit` provides an efficient implementation of iterative reweighting for MSMs. Instead of re-counting transitions with the new weights in each iteration, this optimized approach precomputes count matrices for each trajectory once, and then applies the weights in each subsequent iteration. This significantly improves the computational efficiency of the reweighting process.

1.5 Protein Dynamics

Dynamical motions of proteins drive processes like folding, conformational changes, and protein-protein interactions, which are all integral to life. These dynamics describe how proteins behave in cells and interact with not only other proteins, but also small molecules such as potential drug candidates. Understanding these processes reveals insights about both the relationship between the protein's structure and function, and how it interacts with other biological systems.

1.5.1 Protein folding

Protein folding is a biophysical process that transforms a chain of peptides into a functional, biologically active three-dimensional structure. Once in its folded structure, the protein is capable of carrying out its function. Folding is driven by a complex combination of forces including electrostatic and hydrophobic interactions. Though some proteins may spontaneously fold or unfold, many require an external stimulus such as a change in temperature or pH, or the presence of another protein.

Most proteins are folded into their native functional conformation as they are being synthesized on the ribosome, although some are first transported to another location such as the endoplasmic reticulum. As the polypeptide chain is formed, parts of it fold to form intermediate structures. Different domains may fold sooner than others, independently reaching local stable conformations while other domains may remain flexible until protein synthesis completes.

Because protein function depends critically on structure, misfolded proteins can be extremely problematic. For example, both Alzheimer's and Parkinson's diseases are associated with misfolded proteins.[112,113] Prion diseases are also caused and propagated by misfolded proteins.[114] To mitigate misfolding during synthesis, ribosome-associated chaperones guide folding for nearly all new proteins synthesized on the ribosome.[115] However, chaperones do not always prevent misfolds, and proteins may misfold or unfold even after synthesis.

Due to the importance of protein structure for function, and the adverse impacts of of misfolded proteins, understanding the dynamics of protein folding is an important task. The unique ability of MD simulations to provide high-resolution dynamical movies of protein motion makes it a powerful tool for studying these conformational dynamics.

Although folding may involve a wide array of different factors throughout the cell, this work focuses on developing analysis methods using simpler model systems which can be efficiently simulated. We primarily use the small, fast-folding proteins NTL9 and Trp-cage. These spontaneously fold and unfold in solvent, which makes them useful for generating data of a folding process.[25] Although this omits the full complexity of, for example, ribosomal interactions, it enables us to generate large amounts of data which would not be possible for the full biological system. Because our focus is on developing methods to efficiently analyze generic biomolecular simulation data, this tradeoff is favorable.

While recent developments of tools like AlphaFold are extremely powerful for predicting folded structures from sequences, they only reveal the folded and unfolded states, not the dynamical process that takes one to the other.[116] This information is certainly very useful, but does not itself provide a complete understanding of the folding process. In

other words, to understand how to get from your house to the store, you need more than just the two addresses – you need to know the roads between.

1.5.2 Small molecule interactions

The dynamics of small molecule interactions are also important, for example in the field of drug discovery. An understanding of the protein's structure and dynamics may help design a drug that can efficiently bind to it. Knowledge of its function, and how it might be affected by a certain small molecule can help optimize the drug's effect.[117–120]

Simulations can uniquely provide both structure and dynamics, making it well-suited for this. As mentioned before, the critical limitation in applying MD are timescales, and computational expense.[117] Thus, MD is particularly effective for this when combined with other methods. For example, MD simulations were used to generate conformations of HIV integrase. Docking performed on these conformational snapshots led to the development of the first clinically approved HIV integrase inhibitor.[121]

Although this dissertation focuses on demonstrating the effectiveness of our novel methods by analyzing simulations of protein conformational changes, specifically folding, we emphasize that the methods we present are general to studying different types of protein motion. We focus on folding simulations to take advantage of the large amount of validation data available for those systems.

1.6 Outline

Throughout the rest of this work, we describe new pipelines for improving simulation and analysis of biomolecular systems, building on the methodologies described in this section. This dissertation is structured as follows:

In Ch. 2, we lay the groundwork for our methods development by developing a tool for rapid generation of approximate MD-like trajectories using SynD. A major challenge in methods development is the ability to efficiently generate large amounts of data for complex systems, while also being able to calculate reference values. SynD trajectories have similar complexity to MD data, but can be generated orders of magnitude faster while also only requiring a modest workstation instead of supercomputing resources, eliminating the slow bottleneck of generating test data. The SynD workflow also enables calculation of exact references for quantities like equilibrium populations or MFPTs, which is often not possible with conventional MD, allowing us to validate the quality of our analyses. We use this throughout the rest of the work to rapidly develop and test our new methods.

In Ch. 3, we explore a scheme for reweighting MD data to mitigate bias in MSM construction. We develop a mathematical prescription for improving MSM estimates of equilibrium populations. Additionally, we propose two novel mathematical estimators which enable unbiased estimation of committors and MFPTs from MSMs, which is not otherwise possible. We show that these methods significantly improve estimates of both equilibrium and kinetic properties from MD data.

Ch. 4 discusses work done on the 2.0 version of the WESTPA software. WESTPA is a widely used enhanced sampling framework, that can be used with a variety of MD simulation programs. The WESTPA 2.0 release significantly improved accessibility to developers for extending the behavior of WESTPA with new functionality. It also included a set of new tools integrated directly into WESTPA, which allow users to easily apply new sampling tools to improve their simulations. Be complexity of running MD simulations, particularly coupled with improved sampling frameworks, makes accessibility a critical concern when developing software tools, the WESTPA 2.0 release also focused on providing thorough user tutorials. Finally, a focus on bringing WESTPA up to date with software development best practices and implementing automated tests help streamline development of WESTPA features, and build confidence in consistent performance over new releases.

Finally, in Ch. 5 we apply a new feature in WESTPA 2.0 to extend previous work on accelerating convergence in WESTPA simulations. Although WESTPA can be fast to simulate paths, obtaining a rate-constant estimate with WESTPA with tightly bounded uncertainties can be extremely slow. We demonstrate a pipeline for running WESTPA simulations that can reduce the convergence time for rate-constant estimates, as well as reducing variance.

Together, these advances both facilitate more effective methods developments and enable more powerful analysis of MD simulations.

2 Simple synthetic molecular dynamics for efficient trajectory generation

ABSTRACT

Synthetic molecular dynamics (SynD) trajectories from learned generative models have been proposed as a useful addition to the biomolecular simulation toolbox. The computational expense of explicitly integrating the equations of motion in molecular dynamics currently is a severe limit on the number and length of trajectories which can be generated for complex systems. Approximate, but more computationally efficient, generative models can be used in place of explicit integration of the equations of motion, and can produce meaningful trajectories at greatly reduced computational cost. Here, we demonstrate a very simple SynD approach using a fine-grained **Markov state model (MSM)** with states mapped to specific atomistic configurations, which provides an exactly solvable reference. We anticipate this simple approach will enable rapid, effective testing of enhanced sampling algorithms in highly non-trivial models for both equilibrium and non-equilibrium problems. We demonstrate the use of a MSM to generate atomistic SynD trajectories for the fast-folding miniprotein Trp-cage, at a rate of over 200 milliseconds per day on a standard workstation. We employ a non-standard clustering for MSM generation that appears to better preserve kinetic properties at shorter lag times than a conventional MSM. We also show a parallelizable workflow that backmaps discrete SynD trajectories to full-coordinate representations at dynamic resolution for efficient analysis.

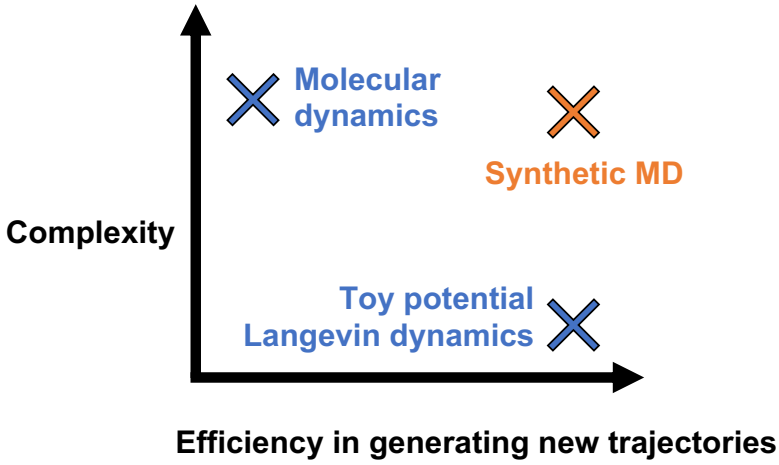


Figure 14: Comparison of different simulation methods. Molecular dynamics can simulate highly complex systems, at the cost of great computational expense. Simpler toy potentials simulated under, for example, Langevin dynamics, can be highly efficient but lack the complexity present in real systems.

2.1 Introduction

The overall goals of **molecular dynamics (MD)** simulation are to generate sufficiently well-sampled and accurate trajectories, but these are hindered by notable challenges. On the one hand, inadequate sampling of complex systems prevents complete characterization of force-field accuracy, impeding the improvement of force field models. On the other, poor sampling also complicates the development of new sampling methods, because it is effectively impossible to gauge the success of a new method without reference *simulation* data. Well-sampled simulation data (rather than experimental data) on complex systems is required as a reference for methods development because even perfectly sampled models are not expected to agree with experiments, again because of model inaccuracy.[25]

Synthetic dynamics (SynD), i.e., the generation of approximate but arbitrarily long trajectories of highly complex models,[122–128] can directly aid methods development for sampling and hence indirectly contribute to force field development. In the long term, increasingly accurate SynD models may ultimately provide a partial replacement for standard MD.

The limitations of conventional MD simulation for biomolecules are well known. Record millisecond-timescale simulations are only achievable for relatively small and simple proteins, even with substantial computational resources.[25,34] In contrast, more complex processes of biological interest in larger systems span timescales up to to seconds and beyond,[10,11,32] which are inaccessible by conventional MD.[14,32]

MD limitations have motivated the development of numerous alternative strategies. Coarse-graining atoms using a force field such as MARTINI,[21] or representing the solvent with an implicit model[35] are strategies for accelerating simulation speed by reducing the number of atoms being simulated, as are statistical mechanics-based coarse-graining strategies like force-matching.[129–131] Enhanced equilibrium sampling methods such as replica exchange, metadynamics,

or umbrella sampling with weighted histogram analysis employ modified energy landscapes, and are popular alternatives to conventional MD for atomistic systems.[46,47,51,64,132] Path-sampling methods, including weighted ensemble and forward-flux sampling among others, aim to improve simulation efficiency by focusing computational resources on regions of interest and can provide unbiased non-equilibrium observables.[43,45,48,55,59,60,62,63,133,134] Finally, Markov state models (MSMs) are a useful tool for connecting data from independent simulations which sample different but overlapping regions of phase space.[83,135]

Despite this significant progress, a persistent challenge in methods development for biomolecular simulation – which remains ongoing for essentially all of the strategies noted above – is the lack of validation data, i.e., extremely well-sampled MD data for systems of interest. As illustrated in Fig. 14, well-sampled MD runs are typically slow for complex systems. This makes them infeasible for use as a step in methods development pipelines because sufficiently complex systems generally cannot be sampled well enough to provide reference values for comparison. Simpler and faster systems such as low-dimensional potentials likely will not capture sufficient complexity to challenge the methods being tested.

Here, we describe a simple synthetic MD workflow based on MSMs, in which a generative model is trained using a set of initial, standard molecular dynamics data. MSMs,[83,88,135,136] with states mapped to specific atomistic configurations, are perhaps the simplest type of generative model. MSM variants such as history-augmented Markov state models (haMSMs) and other MSM alternatives can also be used[48,74,89,137–142] again by mapping discrete states to specific configurations. A special class of coordinate-generative MSMs can also be used to probabilistically generate new, out-of-sample structures.[124] Our work is distinguished from notable previous SynD efforts[122–128] by its simplicity and the availability of exact solutions.

In this preliminary work, we build a detailed generative MSM from folding trajectories of the Trp-cage miniprotein.[25] We employ a simple stratification strategy to augment the usual MSM clustering that appears to preserve kinetic characteristics at smaller lag times than might otherwise be necessary for validation.[99] We generate SynD trajectories at a rate of $\sim$250 ms/day on a MacBook computer, compared to $\sim$100 μs/day on the Anton supercomputer for the original trajectories. We confirm that the SynD trajectories reproduce observables of the MD training data consistent with known capabilities and limitations of MSMs,[99] and that the SynD trajectories replicate exactly calculable equilibrium and kinetic properties of the MSM as expected. We also demonstrate dynamic resolution analysis of the SynD trajectories, where full-coordinate structures are only backmapped within time intervals and at a time-resolution of interest, rather than to each generated point, enabling more efficient analysis.

2.2 Methods

2.2.1 Workflows

We present two main workflows for producing synthetic MD trajectories. First, we describe a generic strategy for efficiently generating trajectories with full atomic coordinates. Second, we outline a strategy to efficiently generate

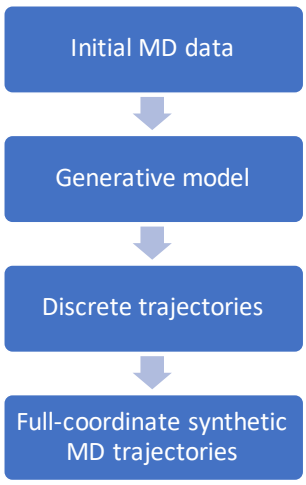


Figure 15: The synthetic MD workflow using discrete-state models. Initial MD simulation data is used to construct a discrete generative model. Discrete state trajectories are efficiently generated from this model, and back-mapped to full-coordinate structures. This last step is trivially parallelizable.

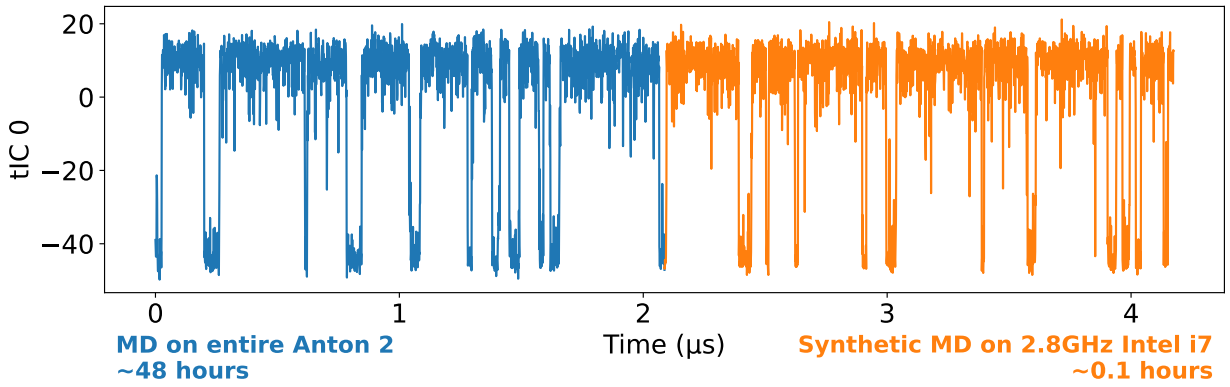


Figure 16: Original 208 μs trajectory from MD simulations of the protein Trp-cage,[25] extended with another 208 μs of synthetic MD. The synthetic MD trajectory was constructed according to Sec. 2.2.1 at 10 ns resolution, initialized from the final point of the MD trajectory. The synthetic trajectory is projected here into the same tICA space computed from the MD trajectory for consistency. Only the first tIC, which strongly contrasts the folded and unfolded states, is shown.

extremely long atomistic trajectories at a coarse temporal resolution, followed by enhancement of the resolution in post-processing for time intervals of interest.

In the standard synthetic MD workflow employing discrete states (Fig. 15), a generative model employing a discretization of configuration space, such as an MSM, is first built from an initial set of traditional MD trajectories.[25] A specific full-coordinate atomistic configuration is associated with each discrete state. The generative model is then used to simulate trajectories, which will be time-ordered lists of discrete configurational states, stored as integers. Discrete trajectory generation typically will be an extremely rapid process. These trajectories are then back-mapped to the saved atomic coordinate structures. Because the discrete trajectories are generated before assigning full-coordinate

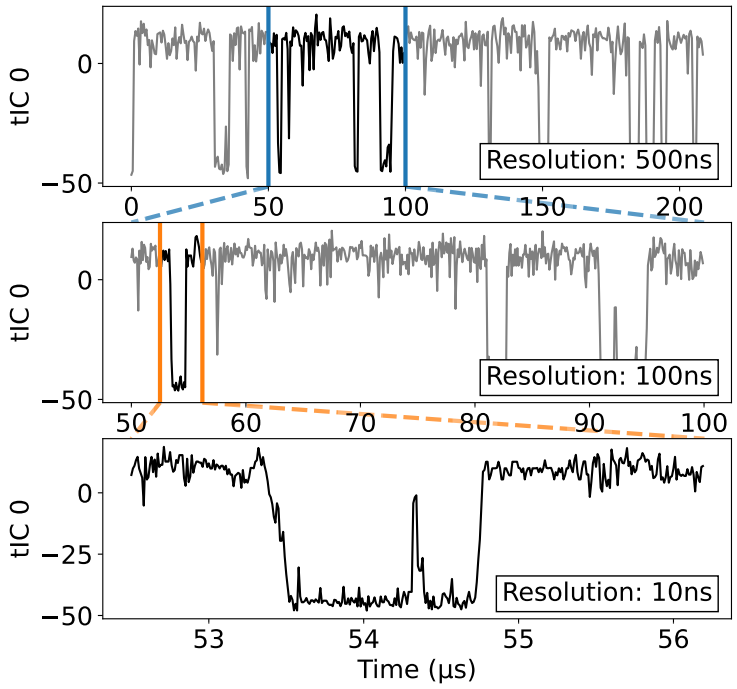


Figure 17: Synthetic MD trajectory for the protein Trp-cage, shown at varying levels of temporal resolution obtained in post-analysis. The full-coordinate trajectory may be initially back-mapped over only subsampled points from the generated discrete trajectory (top). Intervals of interest can later be backmapped at higher resolution (middle and bottom). The first tIC (time-independent component) is taken as a representative coordinate because it clearly shows folding transitions. SynD trajectories include all atomistic coordinates, enabling arbitrary analysis.

structures, the back-mapping is highly parallelizable. Finally, the full-coordinate trajectory is written to disk in a standard MD format, enabling processing by standard tools.

Synthetic MD also enables a dynamic resolution workflow (Fig. 17), where very long trajectories can be efficiently generated, and enhanced temporal resolution added to regions of interest in post-processing. In this workflow, the generative model is used to build a very long discrete trajectory. However, only a temporally subsampled set of points from the discrete trajectory are back-mapped to full-coordinate atomistic configurations, rather than the full trajectory. This enables "telescoping" detailed analysis of long trajectories that would be infeasible at full temporal resolution because of the large number of snapshots generated in SynD.

2.2.2 Simple generative model: MSM of Trp-cage

To demonstrate the SynD approach, we employed a nearly standard MSM as a generative model, built with pyEMMA.[143] The clustering described below is slightly different than for typical MSMs. The original MD trajectory from a 208 μs simulation of the protein Trp-cage[25] was first featurized with residue-residue minimum RMSD,

excluding nearest neighbors. Next, tICA dimensionality reduction was performed at a 10ns lag time with 10 tICs, using commute maps for eigenvector scaling.

The dimensionality-reduced trajectories were clustered using a stratified k-means approach, which differs somewhat from typical MSM workflows. A coordinate of interest is first stratified into bins, and then k-means clustering is independently performed in each bin. Stratification guarantees an even distribution of states along coordinates of interest. In this case, we stratified along tIC 0, which sharply distinguishes the folded and unfolded states, guaranteeing reasonable coverage of transition regions in this coordinate. With 20 k-means centers for each stratified bin, there were a total of 1020 clusters which form the discrete states of the generative model.

The discretized trajectories were used to build a MSM at a 10ns lag time, chosen to balance time resolution with reasonable kinetic fidelity.[99] The MSM was symmetrized to ensure satisfaction of detailed balance by adding the count matrix to its transpose. For each discrete state, a single representative structure was randomly chosen from all structures assigned to that state.

Some of the choices made in constructing this MSM may decrease model fidelity to the MD training data, but we emphasize our initial goal is to construct a generative model with protein-like complexity to enable downstream analysis and testing. Indeed, MSMs have fundamental limitations that have been discussed in detail.[99]

For reference, we note this MSM produced mean first-passage times (MFPTs) of 12.7μs for folding and 2.8μs for unfolding as calculated from the transition matrix using pyEMMA.[143]

2.3 Results

Five 208 µs trajectories were produced at 10 ns resolution by propagating randomly chosen initial states using the Trp-cage generative model. This took 5 minutes 41 seconds in total for all five trajectories using a MacBook Pro with a single 2.8GHz Intel i7 processor. One such trajectory is shown in Fig. 16, along with the original MD trajectory.

Analysis of these trajectories' equilibrium distributions is consistent with the original MD trajectory data, as well as the underlying MSM, as shown in Fig. 18. Likewise, the MFPT values estimated from the SynD trajectories were 4.1 ± 1.5 μs for unfolding and 18.3 ± 9.6 μs for folding, consistent with the reference values of 2.8 μs and 12.7 μs computed directly from the MSM transition matrix.

2.4 Discussion

We have explored a very simple approach to generating SynD trajectories based on Markov state models (MSMs), with the motivation of rapidly generating trajectories in highly non-trivial systems that can be solved exactly, in turn providing ideal test beds for methods development. Previous work has employed a range of deep-learning techniques.[122–128]

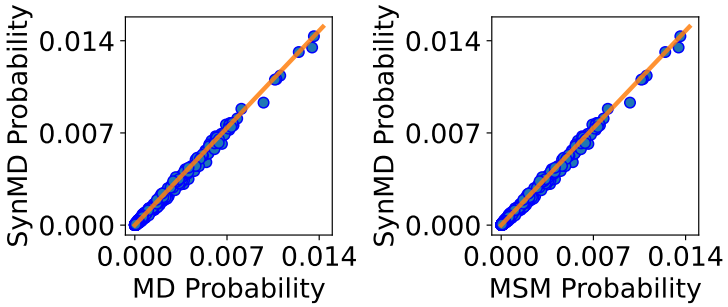


Figure 18: Comparison of SynD equilibrium distributions to the MD training data and the generating MSM. Each point represents the fractional occupancy of a discrete state of the MSM, with MSM values computed from the stationary distribution of the transition matrix. SynD values are averages over the five 208 μs SynD trajectories.

We show that MSM-based SynD trajectories are generated at multiple orders of magnitude speedup over conventional MD, and confirm that the trajectories reproduce exactly-solvable equilibrium and kinetic properties of the generative model. Our generative MSM was able to employ a shorter lag time – providing higher mechanistic resolution[103] – because of an apparently novel stratified approach to state clustering.

Rapid generation of SynD trajectories should be very useful in testing new methods because it provides arbitrary amounts of data in highly complex, but exactly solvable models. Such a framework could be particularly valuable for path sampling, enabling careful estimation of variance based on different choices of hyper-parameters. synD can also advance methods development for trajectory analysis tools[138,144] based on controlled amounts of SynD data, mimicking the low-data regime typical for MD trajectory sets. Even MSM analysis protocols can be tested using SynD based on a fine-grained MSM, so long as the MSM used for analysis is blinded to the fine-grained MSM used to generate trajectories. synD may also be useful for generating an arbitrary number of stochastic mechanistic pathways encoded by the generative model, which may be compared to experimental or higher-quality simulated data to further refine the generative model.[145]

It is feasible to construct significantly improved generative models within the MSM framework. For example, much finer-grained states can be employed, and established adaptive approaches for selecting key regions for further simulation (of MD training data) are available.[49,50,146,147] Training data from polarizable or hybrid quantum/classical force fields could be used to refine a conventional MSM as needed. Numerous MSM-like discrete-state models have been developed incorporating more dynamical information – i.e., trajectory history – than conventional MSMs.[48,74,89,137–142] For example, haMSMs are unbiased for kinetics at any lag time and were shown to significantly outperform conventional MSMs in characterizing mechanistic details of protein folding.[99,138] Deep generative MSMs can be used to stochastically generate new out-of-sample structures.[124]

More modern machine learning strategies will undoubtedly continue to play a large role in SynD. Frameworks such as variational autoencoders and recurrent neural networks including long short-term memory neural networks[123,127,128] have led to models with an improved ability to generate MD-like discrete-state trajectories; note that current MSMs and variants have not been optimized for this task, which is critical to SynD. Mixture density network autoencoders[125] and

latent space simulators[126] generate trajectories in a lower-dimensional continuous space, and provide a mapping to full-coordinate representations.

3 Unbiased estimators and iterative reweighting for improved estimation of equilibrium and kinetic properties in Markov state models

ABSTRACT

Molecular dynamics (MD) simulations are a powerful tool for studying the complex behavior of biomolecular systems, but accurate and unbiased estimation of observables remains challenging. **Markov state models (MSMs)** are widely used for analyzing MD simulation trajectories; however, their construction involves discretization of phase space, which can introduce biases when trajectory points are not distributed correctly within discretized states. In this work, we demonstrate novel unbiased estimators for equilibrium and nonequilibrium steady-state populations, **mean first-passage times (MFPTs)**, and committors (splitting probabilities). While these estimators are asymptotically unbiased only in the limit of infinite data, we demonstrate practical accelerated relaxation to unbiased estimates in a toy model and in synthetic MD data by extending an iterative reweighting scheme recently proposed by Voelz et al. We show that unbiased estimation of observables through a coarse-grained MSM requires incorporating appropriate boundary conditions into trajectories before calculating the transition matrix.

3.1 Introduction

Despite modern capabilities to routinely generate multi-microsecond datasets of **molecular dynamics (MD)** simulation trajectories, the analysis of such trajectories remains a notable challenge. A widespread approach, and the focus of our study, is the **Markov state model (MSM)** framework which coarse-grains continuous molecular configuration space into discrete states, followed by construction of an approximate transition (stochastic) matrix at a finite lag time from which observables are then calculated.[83–87,90] The MSM framework, with variations, has also proved useful in analyzing data from rare-events sampling methods, such as the weighted ensemble approach.[74,102,148]

Even as the field has developed more sophisticated MSM analyses,[57,87,91,98,149–152] recent work has highlighted the approximate nature of MSM results, which results from intrinsic coarse-graining in space and time.[83] One study highlighted variations of MSM analysis which yielded divergent estimates for multiple observables.[153] A second report[99] showed that MSMs yield accurate mean first-passage time (MFPT) estimates only for fairly long ($\sim$ 100 ns) lag times for protein folding systems, but also that MSMs are inaccurate for mechanistic characterization which typically reflects shorter timescale behavior. On the other hand, the same analysis showed that including history information – tracing back trajectories to macrostates of interest – enabled accurate MFPT and mechanism characterization within a MSM-like formulation. In other words, a standard MSM built at an arbitrary lag is not unbiased for kinetics, *even with an infinite amount of data.* These findings largely motivate the present report, where the use of history information is essentially recast as appropriate use of boundary conditions. Our work is also related to ideas integral to exact milestoning[56,154] and non-equilibrium umbrella sampling.[62,63]

However, even estimators that are theoretically unbiased in the limit of infinite data may demonstrate bias in practice as a result of limited sampling. Several reweighting approaches have been suggested to mitigate this sampling bias. One approach[144] computes reweighting factors by optimizing of a given functional which approximates an ideal reaction coordinate. Although this method has the distinct advantage of being nonparametric, since the final converged result is independent of feature selection, it can be numerically unstable for poorly sampled data. Another explores reweighting ensembles of trajectories to maximize path entropy.[155] A third approach which was derived from the weighted ensemble methodology[43] uses statistical resampling to reweight data for MSM construction, which can reduce initial state bias in datasets with certain features, but lacks generality.[156] Finally, another recently developed methodology reweights trajectories using initial equilibrium estimates.[153] In this work, the term "unbiased estimators" refers to estimators which are unbiased in the asymptotic, infinite data limit, but which may demonstrate bias as a result of finite sampling.

Our theoretical analysis rests on multiple pillars, several of which appear to be novel.

- Most importantly, building on early MSM work,[157] we compare ‘microscopic’ discrete-state models with coarse-grained MSM models – of different kinds – which would be generated from trajectories of the microscopic model. In contrast to earlier work, we address the issue of bias by first performing exact computation of MSM-like estimates, before simulating trajectories and introducing the confounding issue of finite sampling.

- Motivated by recent history-augmented MSMs,[74,99,102,137] we carefully account for boundary conditions (BCs), assessing the difference between applying BCs before or after calculating MSM transition matrices. Properly accounting for BCs is essential for unbiased estimation of the MFPT and committor.

- Our mathematical analysis accounts exactly for initial state bias – i.e., the effects of the expected deviation of trajectories, especially their initial points, from the stationary distribution of interest. Initial state bias is intrinsic to MSM calculations; if it were not, the required distribution would already be in hand.

- Our mathematical analysis also accounts exactly for sliding-window averaging occurring over finite-length trajectories. This averaging is generally used to build MSMs at lag τ based on examining the pairs of points, $\{(0, \tau), (\Delta t, \Delta t + \tau), (2\Delta t, 2\Delta t + \tau), \dots\}$, where Δt is the spacing between MD trajectory frames. Sliding-window averaging critically underpins – and limits – relaxation to unbiased observable values.

- We extend a recent proposal to reweight trajectories based on initial estimates of equilibrium distributions,[153] by iterating this process to self-consistent convergence and additionally applying it to non-equilibrium stationary conditions, using appropriate boundary conditions. Our analysis shows that reweighting can significantly reduce the relaxation time required to achieve unbiased estimates of observables.

In addition to the theoretical analysis of the exactly-solvable case, we apply these methods to estimate observables from finite amounts of data, using a set of trajectories generated from a synthetic Trp-cage model. Where the exact formulation allows us to study bias in the asymptotic limit of an infinite number of finite-length trajectories, our analysis of the molecular system explores the practical application of these estimators and reweighting strategy.

3.2 Theoretical Framework

3.2.1 Notation

For clarity, we define the various symbols used throughout this work in Table 1.

3.2.2 Fine- and coarse-grained systems

The process of building a Markov model typically involves simulating continuous trajectories, discretizing them by assigning points in the trajectories to states, and constructing the model on the space of states.[83] Taking the continuous Markovian phase space described by the system's microscopic dynamics and grouping it into discrete states produces states that cannot be perfectly Markovian.[91,99,137]

Examining the effects of coarse-graining on trajectories is complicated by the sampling issues present in any trajectory analysis. We therefore employ a framework for exactly recapitulating the process of constructing a coarse-grained model from trajectories, without using actual trajectories. This enables us to study the coarse-graining exactly, without any sampling concerns.

Table 1: Definitions of symbols used in this work.

Symbol	Definition
i, j	Microstates (single phase points)
$\mathbf{P}$	Microscopic transition matrix
m, n	Coarse states (sets of microstates)
$\mathbf{T}$	Coarse-grained transition matrix
π	Microscopic equilibrium distribution
Π	Coarse-grained equilibrium distribution
$\mathbf{P}^\alpha$, $\mathbf{T}^\alpha$	A $\to$ B steady-state matrices
π^α, Π^α	A $\to$ B steady-state distributions
w	Microstate weights
$\bar{w}$	Sliding-window averaged microstate weights
S	Trajectory length: number of steps
Δt	Timestep of microscopic model
τ	Physical lag time
λ	Dimensionless lag time, $\tau/\Delta t$
q	Microscopic committor to state A
Q	Coarse-grained committor to state A

For simplicity, we use a discrete representation for the underlying dynamics, although as we discuss, our results are expected to apply for continuous dynamics as well. Let $\mathbf{P}$ be the underlying fine-grained, Markovian transition matrix for a single time step Δt. For simplicity we assume $\mathbf{P}$ is a finite matrix. Note that we use the term *microstate* in its traditional statistical mechanics sense to connote a single phase-space point or discrete state; this contrasts with the ambiguous usage of the term in the MSM community to describe a coarse-grained state.[83,85] The matrix $\mathbf{P}$ will implicitly account for boundary conditions (BCs) chosen according to the observable of interest. Here we refer to boundary conditions applied between two macrostates A and B of interest – i.e., source-sink BCs, dual-absorbing BCs, and the absence of sources or sinks. The issue of boundary conditions is central to our analysis and will be described in further detail below.

The coarse-grained MSM transition matrix $\mathbf{T}$ is obtained by merging microstates of the fine-grained model. The resulting transition probability from coarse state m to any other coarse state n will be a weighted average over microscopic transition probabilities:

$$\mathbf{T}_{m\to n}(S) = \sum_{i\in m}\sum_{j\in n} \bar{w}_i(S)\,\mathbf{P}_{i\to j} \Big/ \sum_{i\in m} \bar{w}_i(S)\ . \tag{5}$$

The microstate weights $\bar{w}_i$ are computed to exactly mimic the process of counting transitions in S-step trajectories but without sampling error, as described below. Also note that the coarse-grained MSM transition matrix $\mathbf{T}$ will "inherit" the BCs of the matrix $\mathbf{P}$ as described below.

3.2.3 Accounting for finite trajectory length in sliding window averaging

To compute the necessary averages for the MSM transition matrix $\mathbf{T}(S)$, we must account both for the initial distribution of trajectories as well as the subsequent dynamics and relaxation that occurs. To do so, we let $w_i(t)$ be the time-evolving weight of microstate i, which represents the fraction of trajectories in state i at time t. The set of weights is assumed to be normalized over the full microscopic space, so that

$$\sum_i w_i(t) = 1 \ . \tag{6}$$

Once the set of $w_i(0)$ is defined, the time evolution of this distribution is fully determined by the underlying transition matrix $\mathbf{P}$ according to

$$w(t + \Delta t) = w(t)\,\mathbf{P} \tag{7}$$

where w is the vector of weights w_i. Importantly, we do not generate trajectories, and there are no sampling limitations in our analysis. Instead our calculations yield the same results as if there were an *infinite number* of *finite-length* trajectories.

Trajectories are taken to consist of S steps or $S + 1$ time points indexed by $\{0, 1, 2, ..., S\}$.

We can now replicate the sliding-window average used in MSM construction[83] by averaging over the time evolving distribution. The time-averaged weights for a single-step lag time are given by

$$\bar{w}_i(S) = \frac{1}{S} \sum_{s=0}^{S-1} w_i(s\Delta t), \tag{8}$$

where $w(s\Delta t) = w(0)\,\mathbf{P}^s$ is the weight distribution as evolved according to the Markovian microscopic model $\mathbf{P}$. These time-averaged weights are normalized because the instantaneous weights sum to one.

Note that if the initial weights are not in the stationary distribution of interest – e.g., equilibrium or a non-equilibrium steady-state (NESS) – then we expect the corresponding estimates for $\mathbf{T}$ in Eq. 5 to be biased, unless trajectories are much longer than the associated relaxation process.

Generalization to arbitrary lag time The sliding window calculation can be generalized to arbitrary lag time $\tau = \tau/\Delta t > 1$, where τ is the physical lag time. The window starts at the first point in the trajectory ($s = 0$), and ends τ steps from the end of the trajectory. Less data is averaged because the final steps are omitted as start points of the window, but more relaxation occurs compared to $\tau = 1$. Eq. 8 becomes

$$\bar{w}_i(S, \tau) = \frac{1}{S - \tau + 1} \sum_{s=0}^{S-\tau} w_i(s\Delta t) \tag{9}$$

where the individual weights are again determined by (7). Note that the lag time, which is used only for analysis, does not affect the underlying dynamics embodied in $\mathbf{P}$ and $w(t)$. With this, we can write the arbitrary lag time

coarse-grained transition matrix as

$$\mathbf{T}_{m\to n}(S,\tau) = \sum_{i\in m}\sum_{j\in n} \bar{w}_i(S,\tau)\,\mathbf{P}^{\tau}_{i\to j} \Bigg/ \sum_{i\in m} \bar{w}_i(S,\tau)\ . \tag{10}$$

For clarity of presentation, the rest of this work uses $\tau = 1$, though analogous results apply to any τ.

Also note that we can restrict averaging in (8) and (9) to later time points in the trajectories (i.e., start the sums at $s > 0$), which will exclude earlier, less relaxed time points. This will be explored in subsequent work.

3.2.4 Accounting for boundary conditions

To our knowledge, the issue of boundary conditions (BCs) has not been addressed thoroughly in the MSM literature. BCs are fundamental to MSM construction because the transition matrix is determined by the average *intra*-coarse state distribution $\bar{w}$ as seen in (5) and (10), which in turn depends on other coarse states because of the time evolution of the distribution (7) – i.e., on transitions between coarse states which are constrained by the BCs. From this perspective, it is not surprising that the (equilibrium-like) lack of BCs will lead to unbiased equilibrium populations and that source-sink BCs will lead to unbiased MFPTs. The situation for committors is essentially a hybrid of the two as explained below.

As our data will show, failure to account for BCs correctly can lead to biased estimators. For example, computing a first-passage time involves measuring the time from when trajectories enter (or are initiated in) some state A to when they first enter another state B, including any returns to state A. Such trajectories are consistent with a sink at B and a source at A, i.e., source-sink ("recycling") boundary conditions, as required for computing the MFPT via the Hill relation (20) given below. However, if trajectories were allowed to emerge from the sink state B and re-enter B without first returning to A, such events would bias MFPT estimation using a MSM transition matrix. Asymptotically, the intra-coarse state distributions $\bar{w}$ would not match the NESS and hence the transition matrices would not be appropriate for unbiased MFPT computation.

The committor describes a dually absorbing process at two states A and B. We let q_i be the committor to A, the fraction of trajectories absorbed to A starting from microstate i; the committor to B is $1 - q_i$. If absorbing conditions at A and B are not enforced in the microscopic model (the trajectories), we expect committor estimates to be biased, even for coarse states which consist of collections of microstates. However, as will be seen, simply building a transition matrix from dually absorbing trajectories is not a route to unbiased committors.

Equilibrium, on the other hand, requires detailed balance, so no sources or sinks can be present. Correspondingly, equilibrium probabilities are computable without bias, asymptotically, from a standard MSM.

Below we will consider several types of coarse transition matrices (MSMs) built from different boundary conditions embodied in the microscopic transition matrix. All MSMs are constructed from the same weight formulation, namely (5) for single-step lag ($\tau = 1$) or (10) for $\tau > 1$, using the microscopic transition matrix $\mathbf{P}$ corresponding to different boundary conditions as follows:

- The standard MSM denoted $\mathbf{T}$ is derived from the full microscopic model $\mathbf{P}$ with no boundary conditions applied.

- A source-sink ssMSM can be constructed for either the A $\to$ B direction, denoted $\mathbf{T}^{\alpha}$, or the B $\to$ A direction, called $\mathbf{T}^{\beta}$. The matrix $\mathbf{T}^{\alpha}$ is constructed from the modified microscopic model $\mathbf{P}^{\alpha}$ which is obtained from $\mathbf{P}$ by setting $\mathbf{P}_{ij} = 0$ for $i \in B$ except when $j \in A$. For simplicity, here we assume states A and B each consist of a single microstate, so that $\mathbf{P}_{BA} = 1$ for this ssMSM. The coarse matrix $\mathbf{T}^{\beta}$ is constructed in analogous fashion from $\mathbf{P}^{\beta}$.

- The dually absorbing abMSM denoted $\mathbf{T}^{\mathrm{abs}}$ is obtained by setting $\mathbf{P}_{ij} = 0$ for $i \in$ A *or* B. The abMSM, although it seems natural for computing committors, will be seen to be biased.

3.2.5 Asymptotically unbiased asymptotic estimators

We now demonstrate that Markov-like models with the appropriate boundary conditions incorporated into the trajectories during construction produce unbiased estimates of equilibrium probabilities for coarse-grained states, the first-passage time, and the set of coarse-grained committors. Our argument relies on two simple parts. First, we note that under boundary conditions allowing for stationarity, regardless of the initial weights, the average weights $\bar{w}_i$ asymptotically approach their stationary values. Second, we show that the stationary weights (reached asymptotically) yield unbiased observables with appropriate estimators.

The asymptotic stationarity of the time-averaged weights $\bar{w}$, defined by (8) or (9), follows from the fact that the weights constitute an ordinary probability distribution in the microscopic space evolving under standard Markovian dynamics (7). We will assume that $\mathbf{P}$ is irreducible, meaning all regions in the microscopic state space are connected by positive probability paths,.[158] With this assumption, recalling that $\mathbf{P}$ in (7) is assumed to embody any boundary conditions, we see that under equilibrium or source-sink (α) conditions, the time-averaged weights will approach π or π^{α}, correspondingly,

$$\bar{w} \to \pi \qquad \text{or} \qquad \bar{w} \to \pi^{\alpha}, \tag{11}$$

under equilibrium or α source-sink BCs as $S \to \infty$.

It will prove convenient to show a related result, namely, that coarse-grained stationary probabilities Π derived using the stationary weights are exactly the sums of the corresponding microscopic stationary probabilities π. Starting from the coarse stationarity condition, we use the asymptotic stationary weights (11) along with the coarse matrix (5) to find

$$\Pi_n = \sum_m \Pi_m \mathbf{T}_{mn}(S \to \infty) \tag{12}$$

$$= \sum_m \Pi_m \sum_{i \in m} \sum_{j \in n} \pi_i \mathbf{P}_{ij} \Big/ \sum_{i \in m} \pi_i \ . \tag{13}$$

If we substitute $\Pi_m = \sum_{i \in m} \pi_i$ into the right-hand side of this expression, we find

$$\Pi_n = \sum_m \sum_{i \in m} \sum_{j \in n} \pi_i \mathbf{P}_{ij} \tag{14}$$

$$= \sum_{j \in n} \sum_i \pi_i \mathbf{P}_{ij} \tag{15}$$

$$= \sum_{j \in n} \pi_j \ , \tag{16}$$

which demonstrates the consistency of the summed microscopic stationary probabilities with coarse-grained stationarity. This completes the demonstration.

Note that the result (16) holds regardless of boundary conditions, so long as the stationary probabilities and transition matrix are for the same BCs. In particular, it implies

$$\Pi_n^\alpha = \sum_{j \in n} \pi_j^\alpha \tag{17}$$

for the α (A to B) NESS.

We now consider the different observables in turn and show that asymptotically, when the average weights approach stationary values, suitable coarse-grained estimators become unbiased. That is, we must show that estimators obtained solely from calculations using coarse-grained $\mathbf{T}$ matrices asymptotically yield observables in exact agreement with microscopic values.

3.2.5.1 Equilibrium Coarse-grained equilibrium probabilities Π can be estimated without bias as the stationary solution to the standard MSM in the limit of infinite trajectory length:

$$\Pi \, \mathbf{T}(S \to \infty) = \Pi \tag{18}$$

This follows from the asymptotic stationarity of the weights (11), which in turn causes the coarse-grained stationary probabilities to match the sum of microscopic stationary probabilities as in (16). It is easy to check that the conditions above on $\mathbf{P}$ ensure that $\mathbf{T}(S)$ has a unique stationary distribution for large enough S.

3.2.5.2 Mean first-passage time We employ a similar strategy for the MFPT, showing that macroscopic analog of the microscopic solution recapitulates the microscopic value, *so long as the correct source-sink boundary conditions are employed.* We make use of the Hill relation, which relates the source-sink steady-state flux into a target macrostate B to the MFPT(A $\to$ B) according to[100]

$$1/\text{MFPT} = \text{Flux}(A \to B) \ . \tag{19}$$

Recalling that the A $\to$ B NESS is designated by α, we recast the flux using the microscopic model to yield the reference dimensionless expression

$$\Delta t/\text{MFPT} = \sum_{i\notin B}\sum_{j\in B} \pi_i^\alpha \mathbf{P}_{ij}^\alpha \ . \tag{20}$$

We will explore coarse-grained estimates of the MFPT generically given by the analogous expression

$$\Delta t/\text{MFPT} = \sum_{m\notin B}\sum_{n\in B} \Pi_m \mathbf{T}_{mn}(S) \ . \tag{21}$$

We now show that using the flux computed from the asymptotic coarse ssMSM yields a MFPT identical to that from the microscopic model. Using the α-specific asymptotic weights (11) in the coarse ssMSM $\mathbf{T}^\alpha$ defined by (5), we obtain

$$\begin{aligned}
\Delta t/\text{MFPT} &= \sum_{m\notin B}\sum_{n\in B} \Pi_m^\alpha \mathbf{T}_{mn}^\alpha(S\to\infty) && (22)\\
&= \sum_{m\notin B}\sum_{n\in B} \Pi_m^\alpha \left(\sum_{i\in m}\sum_{j\in n} \pi_i^\alpha \mathbf{P}_{ij}^\alpha \Big/ \sum_{i\in m} \pi_i^\alpha \right) && (23)\\
&= \sum_{m\notin B}\sum_{n\in B}\sum_{i\in m}\sum_{j\in n} \pi_i^\alpha \mathbf{P}_{ij}^\alpha && (24)\\
&= \sum_{i\notin B}\sum_{j\in B} \pi_i^\alpha \mathbf{P}_{ij}^\alpha \ , && (25)
\end{aligned}$$

where we made use of (17). Hence the MFPT calculated from the ssMSM with asymptotic weights yields the correct microscopic value (20).

We note that our formulation here, including for the microscopic model, retains a discretization error, expected to be $O(\Delta t/\text{MFPT})$. This is because $\mathbf{P}_{ij}^\alpha > 0$ for $j \in B$ will lead to non-zero occupancy of B, with expected probability in B of $\sum_{i\in B} \pi_i^\alpha \sim \Delta t/\text{MFPT}$ from the definitions of the MFPT and NESS. Even this small error can be avoided with a slightly more complex formulation, as we will show in future work.

3.2.5.3 Committors We now demonstrate a novel estimator for coarse-grained committors based on the ratio of the steady-state to equilibrium probabilities. It has been shown previously that, microscopically, the committor to A, q, is proportional to the ratio of the α NESS to equilibrium probabilities:[159]

$$\pi_i^\alpha = c\, q_i \pi_i \ , \tag{26}$$

where $c = \pi_i^\alpha/\pi_i > 1$ for $i \in$ A. We propose to estimate coarse-grained committors Q according to

$$Q_m = \Pi_m^\alpha / c\, \Pi_m \ , \tag{27}$$

where $c = \Pi_m^\alpha/\Pi_m$ for $m \in$ A has the same value as in the microscopic case because of the relations (16) and (17).

It is not immediately obvious what the "exact" coarse-grained Q values should be. Consider a thought-experiment of computing committors from an extremely long 'equilibrium' trajectory which traces back and forth between states A and B many times, visiting all microstates. We could estimate the committor for a coarse state m by considering all time points of the trajectory in m and counting the fraction of downstream trajectory segments which reach A before B for each such time point. The configurations in the coarse state will be equilibrium distributed due to the length of the trajectory, and the fractional absorptions to A and B for segments visiting a given microstate $i \in m$ will necessarily be determined by the microscopic committor q_i. This scenario motivates equilibrium weighting of microscopic committors according to

$$Q_m = \sum_{i \in m} \pi_i q_i \Big/ \sum_{i \in m} \pi_i \ . \tag{28}$$

Indeed, it would be difficult to motivate other choices, such as a uniform weighting or weighting according to a particular directional NESS.

To validate the estimator (27) asymptotically as $S \to \infty$, we substitute the asymptotically exact microscopic decompositions (16) and (17) for the coarse stationary probabilities. This yields

$$Q_m = \sum_{i \in m} \pi_i^{\alpha} \Big/ c \sum_{i \in m} \pi_i \tag{29}$$

$$= \sum_{i \in m} c\, q_i \pi_i \Big/ c \sum_{i \in m} \pi_i \ , \tag{30}$$

where we have used (26) and recapitulate the desired result (28). Although the suitability of equilibrium weighting among microscopic committors can be debated, the ratio estimator (27) yields this natural average.

3.2.6 First-step relation for committors

As we will see, the abMSM is biased for committor estimates, despite seeming like a natural and correct choice of boundary conditions. For completeness, we review a procedure for calculating the committor from a transition matrix using a 'first-step' relation.[54,160]

If the committor to A at a microstate i is given by q_i, the average committor of trajectories initiated in that point and propagated for one step is also equal to q_i. The analogous formulation for a coarse model is therefore

$$Q_m = \sum_n \mathbf{T}_{mn} Q_n \qquad i \notin A, B \tag{31}$$

where $Q_{n \in B} = 0$ and $Q_{n \in A} = 1$. Although not unbiased for coarse states, this relation is used for reference in the results shown below.

3.2.7 Iterative reweighting

Although we have described estimators that are unbiased asymptotically, deviation in the initial weights $\bar{w}_i$ from the appropriate steady-state distribution introduces initial-state bias which can be very slow to relax away, as our results will show. As a trajectory propagates, the relaxation time for the initial distribution to converge to a steady-state distribution will depend on the initial distribution.

Recent work showed that computing steady-state twice, once from an MSM with uniform initial weights for each trajectory, then recalculating the MSM using weighted trajectories (with weights from the first steady-state probability estimate of the initial bin of each trajectory), substantially reduced the trajectory length necessary for converged estimates.[153]

In fact, this process can be applied iteratively, using the estimate from the previous iteration as the weights for the next.

Algorithm 1 Iterative reweighting algorithm

1: Choose uniform initial weights w_i
2: **repeat**
3: Compute $\bar{w}_i$ from w_i using (8)
4: Compute the interim stationary distribution $\tilde{\Pi}$ by solving $\tilde{\Pi}\mathbf{T} = \tilde{\Pi}$
5: Update the microbin weights $w(0)$ by evenly dividing the coarse probabilities over microbins according to $w_i(0) = \tilde{\Pi}_m \Big/ \sum_{j \in m} 1$ for $i \in m$
6: **until** desired number of iterations

Results for iterative reweighting are presented in Sec. 3.3.2.

3.2.8 Connection to continuous trajectories

We expect that our discrete-state analysis will carry over directly to the case where microscopic dynamics are continuous in space. First, one may consider the limit of arbitrarily small microstates, leading to quasi-continuous dynamics. Second, the derivations presented in Sec. 3.2.5 rely almost exclusively on the relaxation of the initial weights to steady-state values, a process that will also occur under continuous dynamics.

3.3 Analytical Results

Numerical results confirm our theoretical expectations. Equilibrium probabilities, mean first-passage times, and committors of coarse-grained MSMs are unbiased in the asymptotic limit in general only when they are based upon the relaxation of microscopic trajectories to the appropriate steady-state distributions, which can be achieved by sliding window relaxation and applying the appropriate BCs at the microscopic trajectory level.

We demonstrate by estimating equilibrium probabilities, mean first-passage times, and committors on a sample system, where the microscopic dynamics are exactly described by a 42 microstate transition matrix (exact transition

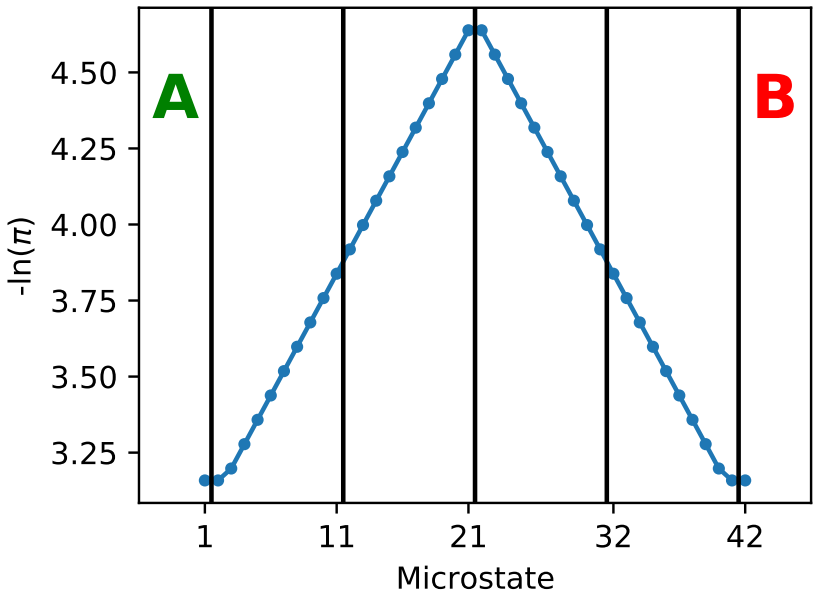


Figure 19: Energy landscape of the 42-microstate fine-grained system. Microstate boundaries are denoted by dots. The coarse-grained states or 'bins' are separated by vertical lines. Bins 1-4 are the four intermediate coarse states. 'Macrostates' A and B, are the leftmost and rightmost individual microstates, respectively, in both fine and coarse descriptions.

probabilities given in Fig. A.1). The coarse-graining preserves the first and last states as the macrostates A and B, and groups the intermediate 40 microstates into 4 coarse states. The energy landscape is shown in Fig. 19, along with lines indicating the coarse states. The energy landscape of this system emulates two stable states separated by an energy barrier.

This minimal system provides an unambiguous demonstration of how initial state bias affects key observables. In a common procedure, a finite set of trajectories may be generated for MSM construction with initial points spanning the space of interest. We emulate this procedure by introducing uniform initial weights into Eq. 8, emulating the distribution of trajectory starting points in a finite sample. In this system with a central energy "barrier", this constitutes significant initial state bias. We examine the estimators as a function of trajectory length S, to determine both how the initial bias relaxes out with longer trajectories, and what length trajectories are necessary for converged estimates. An iterative approach to accelerate convergence is explored in Sec. 3.3.2. Lag time and trajectory length both contribute to recovering unbiased estimations using trajectories whose initial points are not steady-state distributed.

3.3.1 Asymptotic estimators

The estimation of equilibrium probabilities is a very straightforward application of a MSM, and as expected a standard MSM is unbiased both in the limits of long trajectories and long lag times. The reference equilibrium distribution is obtained as the stationary solution of the microscopic transition matrix $\mathbf{P}$. We show results for a standard MSM at lag time of 1 and $\mathrm{MFPT}/10 = 500$ steps in Fig. 20. Note that all results are plotted as a function of the trajectory length,

which governs the amount of relaxation that occurs within a trajectory ensemble. At a lag of 1, the uniform initial weights introduce some initial-state bias, shown in Fig. 20. However, this initial bias quickly relaxes out, and converged first-passage time estimates are obtained within $\sim \mathrm{MFPT}/5$ steps.

The longer lag appears to produce estimates closer to the reference values. However, this is because the minimum trajectory length is given by $\tau + 1$, so the first estimate produced at the longer lag is at a long trajectory length. At this length, the short-lag estimate was also relaxed to nearly the reference value.

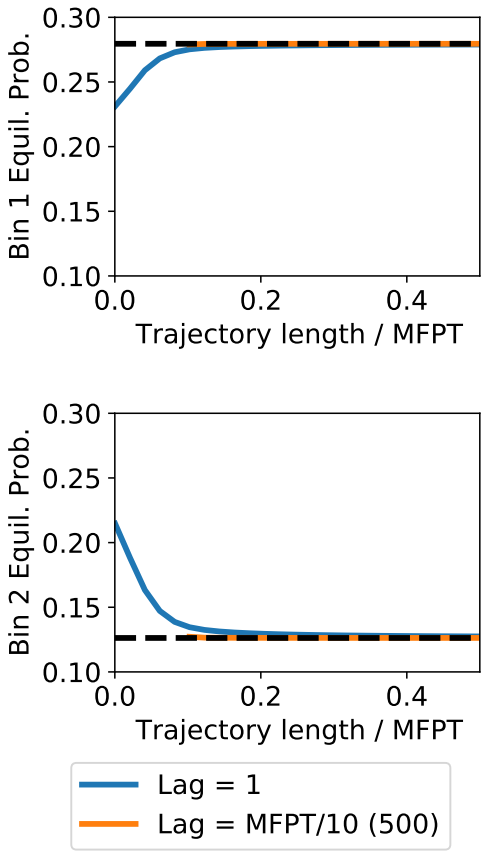


Figure 20: MSM equilibrium probability estimates are asymptotically unbiased. The equilibrium estimator, shown at lag of 1 step ($\lambda = 1$, blue) and $\lambda = 500$ (orange). The black dashed line is the exact reference value, computed from the microscopic matrix. Because the energy landscape is symmetric, only states ('bins') in the left half are shown. We assumed a non-informative uniform initial distribution of weights $w_i(0) = 1/42$.

Despite the apparent simplicity of this two-state system, first passage times can be significantly biased by the initial state distribution. First-passage times to state B are computed using the Hill relation (20), referenced to the MFPT for a lag time of one step (Δt, or $\tau = 1$). Here, source-sink boundary conditions are applied to the standard MSM after construction, while for the ssMSM they are applied at the microscopic trajectory level. When the source-sink BCs are not applied at the trajectory level, the MFPT estimates are significantly biased at lag times of 1 step and $\mathrm{MFPT}/10 \sim 500\Delta t$, and do not improve with the trajectory length, shown in Fig. 21. Note that standard MSMs can recapitulate physical MFPTs at long enough lag times.[99] When the BCs are applied (ssMSM), the MFPT estimate

becomes unbiased for trajectories longer than the MFPT itself. However, combining the application of BCs at the trajectory level (ssMSM), and increasing lag time to $\mathrm{MFPT}/10 \sim 500\Delta t$, leads to an unbiased first-passage time estimate at a fraction of the MFPT.

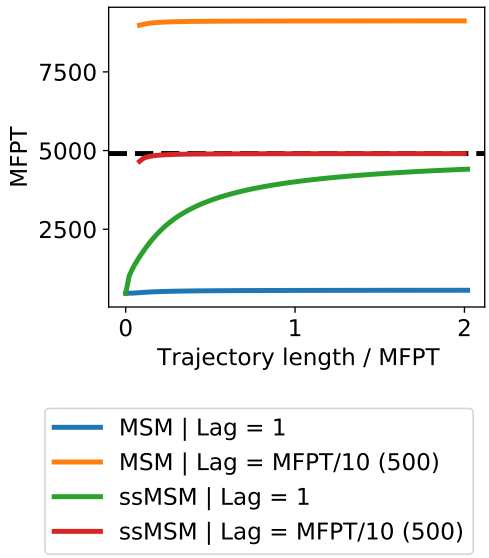


Figure 21: Unbiased MFPT estimation from ssMSMs. Employing the coarse-grained Hill relation (21), we compare MFPT estimates from standard MSMs at short lag time ($\lambda = 1$ step, blue line) and long lag time ($\lambda = 500 \sim \mathrm{MFPT}/10\Delta t$, orange line), ssMSM at short lag time ($\lambda = 1$, green line) and long lag time $\lambda = 500$, red line), and the exact reference value (black dashed line). We assumed a non-informative uniform initial distribution of weights $w_i(0) = 1/42$.

Like the MFPT, the committor stratifying the A to B transition (see Fig. 22) is sensitive to BC application at the trajectory level, and moreover, asymptotically unbiased estimation requires a novel approach. First-step relations (31) applied to the coarse-grained standard MSM estimates are biased at both short and long lag times. Surprisingly, even when appropriate BCs are applied at the trajectory level before MSM construction (i.e., using the abMSM), committor estimates based upon first-step relations are biased at both short and long lag times, even in the limit of long trajectories. We find that asymptotically unbiased committor estimation requires calculation of the committor via the ratio (27) of the equilibrium and NESS (ssMSM source/sink BCs) steady-state distributions . Since this "ratio method" estimator is based upon steady-state distributions, the initial bias can relax and committor estimates converge asymptotically to the reference value. For the longer lag time of $\mathrm{MFPT}/10 \sim 500\Delta t$ steps, this relaxation is rapid within a fraction of the MFPT.

3.3.2 Iterative reweighting

The sliding window relaxation time to a steady-state microscopic distribution is *a priori* unknown, and may be computationally prohibitive. This motivates the exploration of an iterative approach which accelerates steady-state convergence. By iteratively obtaining estimates of the steady-state and equilibrium distributions, and then using those

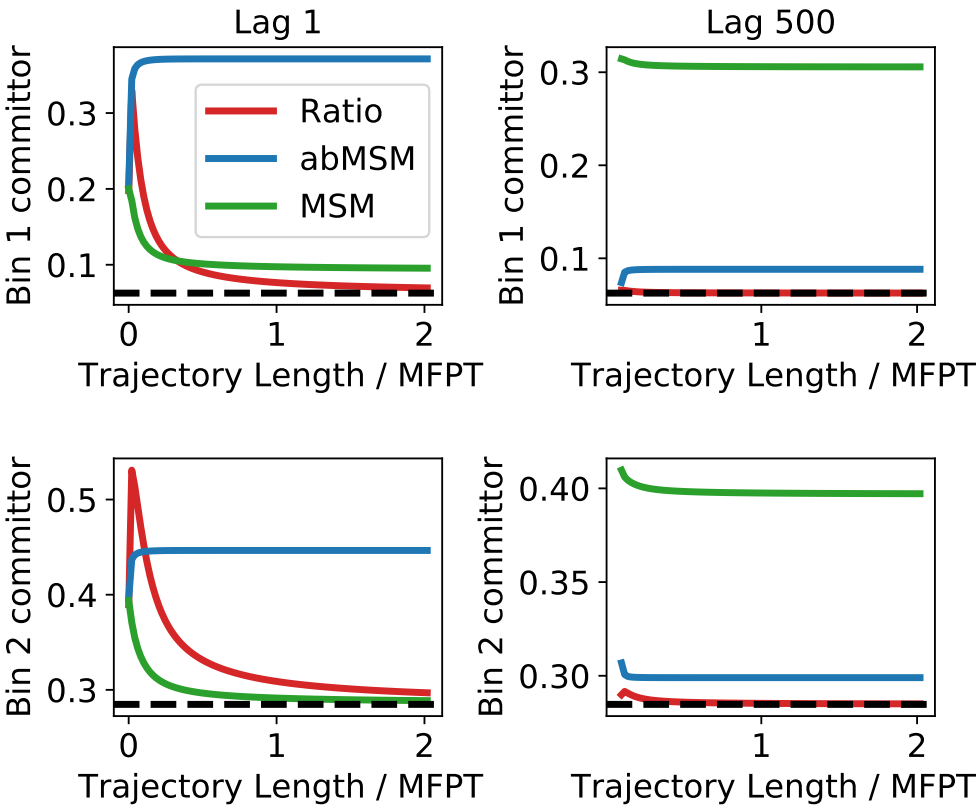


Figure 22: Unbiased committor estimation from the steady-state "ratio method". Committor estimation using the first-step relation (31) with standard MSM (green lines) and abMSM (blue lines) at short lag time ($\lambda = 1$, left) and long lag time $\lambda = 500 \sim \mathrm{MFPT}/10\Delta t$, right), as well as committor estimates from the ratio of equilibrium and NESS (source/sink BCs) steady-states (red lines), compared to the reference value (dashed black line). We assumed a non-informative uniform initial distribution of weights $w_i(0) = 1/42$.

as the initial weights to compute the estimates again as described in Sec. 3.2.7, we can accelerate the relaxation of this initial bias and reduce the trajectory length needed to obtain converged estimates.

For the equilibrium estimator shown in Fig. 23, the effect of reweighting is apparent but not qualitatively large. Iterative reweighting improves the initial estimates but does not substantially accelerate the timescale of the convergence in our model system.

Convergence of first-passage times (Fig. 24) and committors (Fig. 25), are substantially accelerated. Using the iterative reweighting approach, the convergence timescale was reduced from multiple first-passage times, to roughly half a first-passage time. This effect is more pronounced for the short-lag, where the initial state bias affects the estimates more drastically as previously discussed.

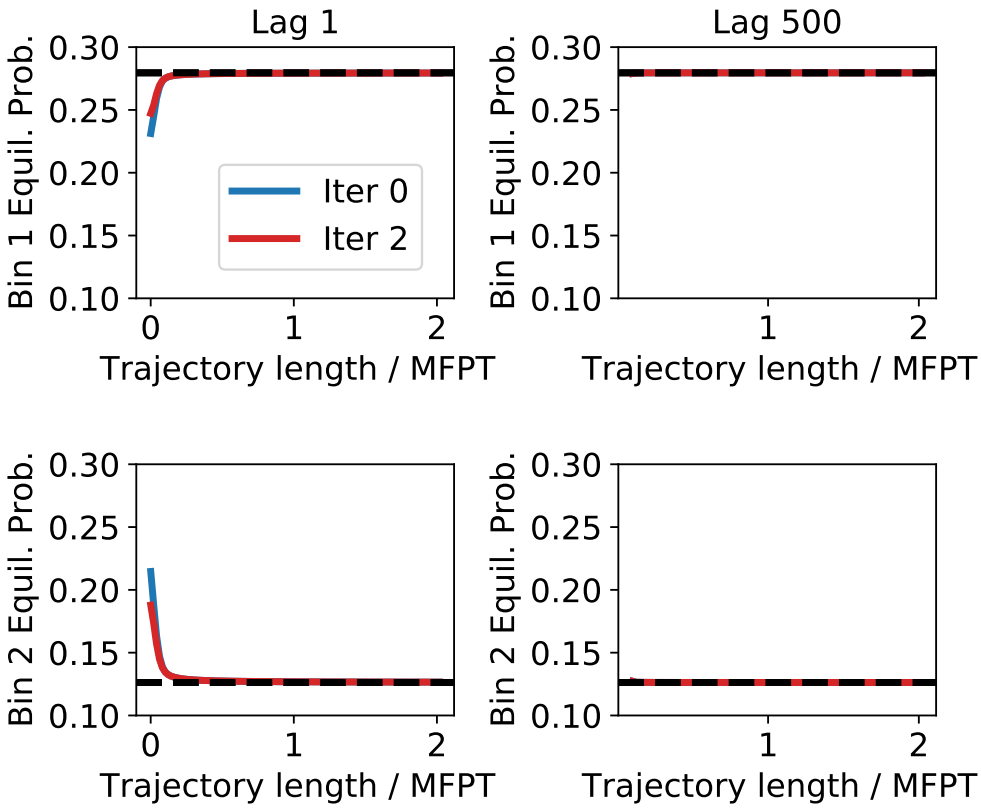


Figure 23: Iterative reweighting for equilibrium estimation. Initial equilibrium estimate (blue lines) and subsequent iterative estimation (2 iterations, red lines), and reference value (black lines). Estimates use the MSM stationary distribution based on $\mathbf{T}(S)$ for the trajectory lengths shown. We assumed a non-informative uniform initial distribution of weights $w_i(0) = 1/42$.

3.4 Practical Implementation

In the prior sections, we examine how the proposed estimators yield asymptotically unbiased estimates of observables in the limit of an infinite number of finite-length trajectories. We now present a framework for applying these estimators to real, finite sets of trajectories.

3.4.1 Trajectory splicing for NESS

To make unbiased estimates of **nonequilibrium steady-state (NESS)**, it is necessary to incorporate source-sink boundary conditions directly into the trajectories before estimating the transition matrix, as described in Sec. 3.2.5.2. We implement this procedure by truncating trajectory points after entering the sink, and replacing them by "splicing" on appropriate trajectory segments starting near the source, as shown in Fig. 26.

Trajectories can be spliced according to different protocols depending on the NESS of interest. The NESS is defined by the source distribution and sink state. In our case, we wish to estimate the same **mean first-passage time (MFPT)** as would be observed in a single very long trajectory with multiple FPT events. We therefore choose the "EqSurf" (or "reactive entrance distribution"[161]) detailed in.[101] In practice, the initial point of the spliced segment is randomly

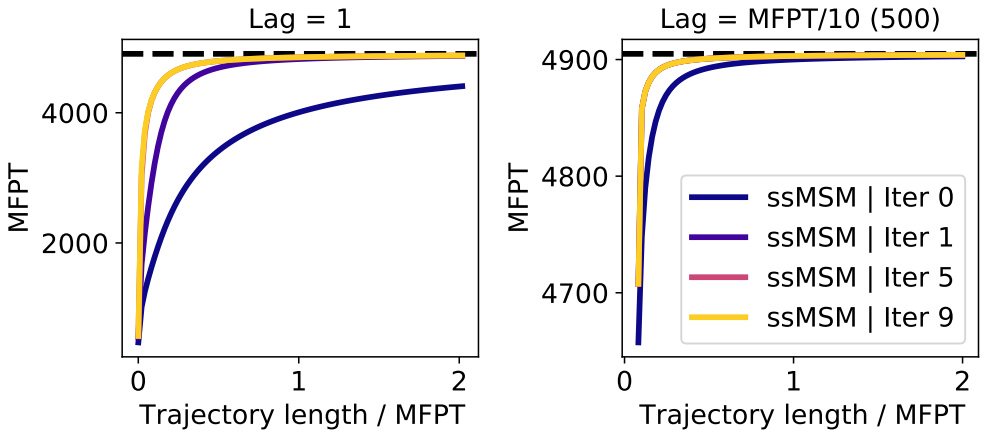


Figure 24: Iterative reweighting accelerates MFPT convergence. Initial MFPT estimates (dark blue line) and subsequent iterations (dark purple to orange lines), and reference value (dashed black line). Estimates are based on the coarse-grained Hill relation (21) using ssMSMs $\mathbf{T}^{\alpha}(S)$ for the trajectory lengths indicated. We assumed a non-informative uniform initial distribution of weights $w_i(0) = 1/42$.

chosen according to the EqSurf distribution, and a segment starting from this point is randomly chosen from an initial set of trajectories.

These spliced trajectories are then used to compute a transition matrix, with either the standard methodology or reweighting, and NESS is estimated by computing the stationary distribution of the transition matrix. With this estimate of NESS, and an equilibrium estimate obtained from the stationary distribution of the (reweighted or standard) transition matrix, the ratio estimator can be used to estimate the committor as described previously in Sec. 3.2.5.3.

3.4.2 MSM reweighting

Our iterative reweighting procedure generalizes the prior suggestion by[153] in two important ways:

1. We iteratively reweight until self-consistent weights are obtained.

2. We use transition matrices built from trajectories which have been spliced as needed to conform to the appropriate BCs.

This is implemented as follows:

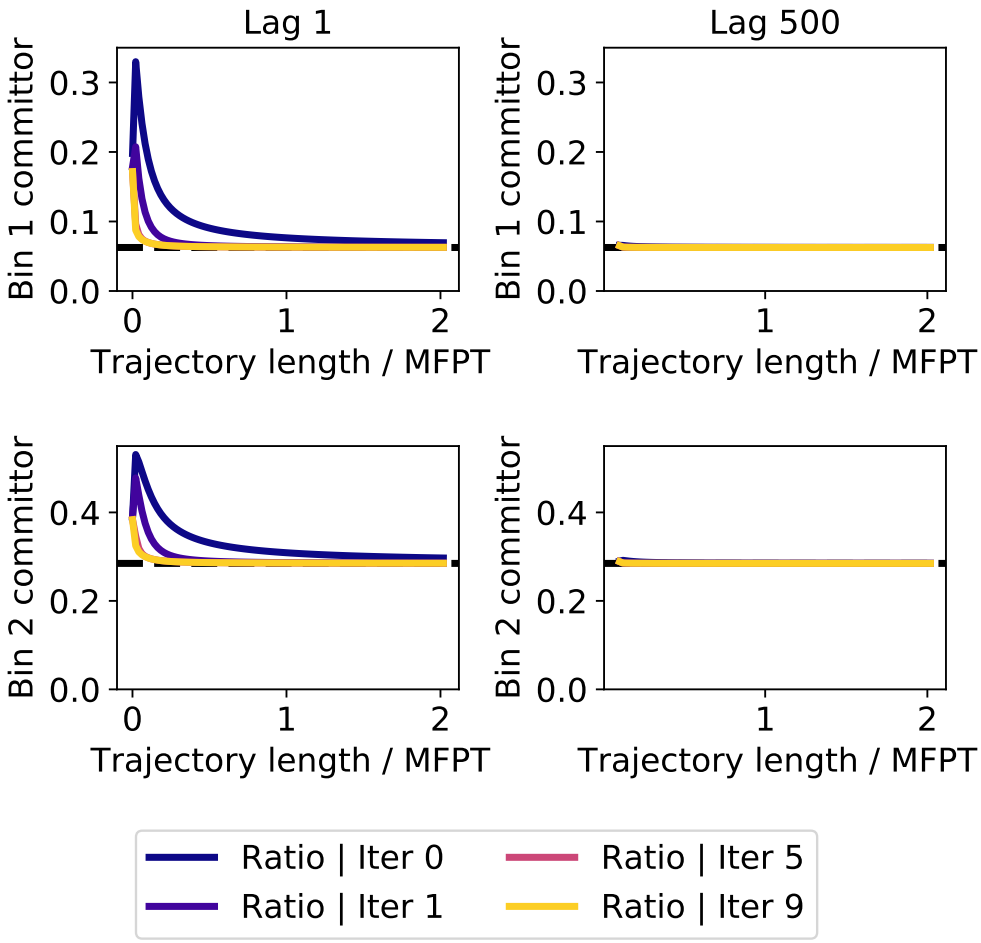


Figure 25: Iterative reweighting accelerates committor convergence Initial committor estimates (dark blue lines) and subsequent iterations (dark purple to orange lines), and reference value (dashed black line). Estimates employ the ratio estimator (27) applied to stationary solutions of the ssMSM and MSM at the trajectory lengths indicated. We assumed a non-informative uniform initial distribution of weights $w_i(0) = 1/42$.

Algorithm 2 Implementation of iterative reweighting

1: Begin with a set of discretized trajectories

2: Assign uniform initial weights w_i to each unique initial state, normalized by the number of fragment initial points in that state

3: For each unique initial state m, compute a count matrix C_m using all trajectories initiated in that state.

4: **repeat**

5: Compute the weighted sum $\sum_m w_m C_m$ of the count matrices using the respective initial state weights

6: Row-normalize the summed count matrix to obtain a transition matrix

7: Solve the transition matrix to obtain an estimate of the stationary distribution

8: Update the weights w from the estimated stationary distribution

9: **until** Stationary distribution estimates converge

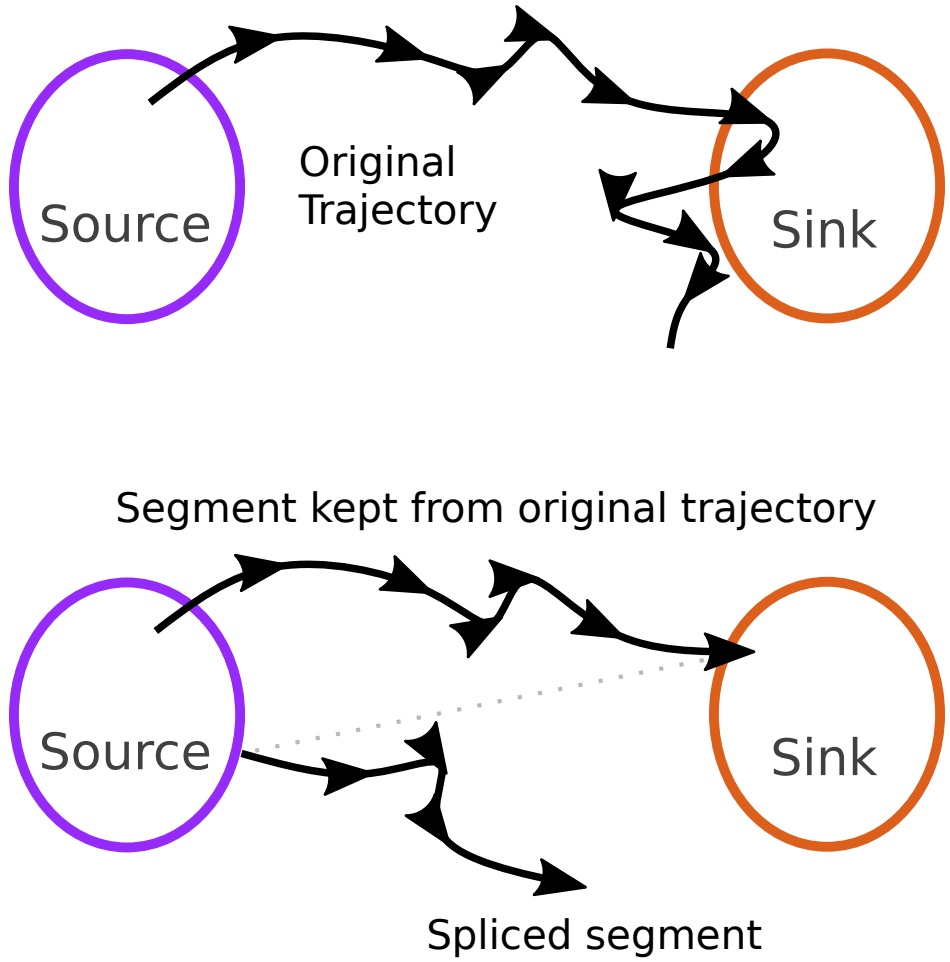


Figure 26: Incorporating source-sink boundary conditions into a trajectory. A trajectory can be modified to have source-sink boundary conditions by truncating it at the point where it enters the sink state, and replacing all subsequent points with a trajectory segment initiated on the surface of the source.

3.4.2.1 Fragments In instances where the initial data consists of a single trajectory reweighting cannot be applied directly, as there are no other trajectories to reweight against. However, this single trajectory can be divided into overlapping fragments, which can then be treated as independent trajectories and subsequently reweighted against one another. This procedure is demonstrated in Fig. 27.

Even when multiple trajectories are available, dividing them into fragments may still be beneficial as a result of increasing the number of trajectories available for reweighting.

Fragments can be used for reweighting as described in Alg. 2 simply by replacing the set of trajectories with a set of fragments constructed from the initial set of trajectories.

3.4.3 Hyperparameter optimization

When constructing an MSM, implied timescales are typically used to identify an optimal lag time. However, reweighted MSMs introduce an additional parameter, the fragment length, which may interact with the lag time. Since

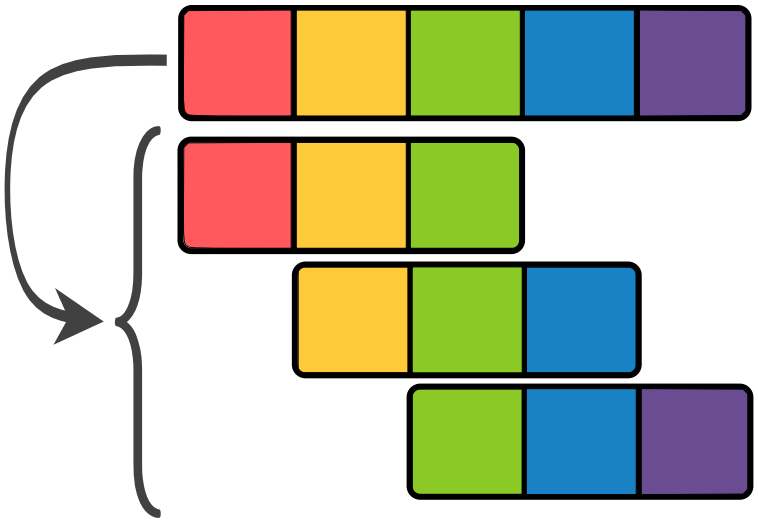


Figure 27: Splitting a single 5-step trajectory into 3 fragments of length 3. Splitting a single trajectory into fragments allows reweighting to be applied to the fragments.

the optimal fragment length is not known *a priori*, we employ the hyperparameter optimization strategy described in Algorithm 3 to simultaneously determine optimal values for both the fragment length and lag time.

Algorithm 3 Hyperparameter optimization strategy

1: Begin with a set of trajectories
2: Split the set of trajectories into 4 groups
3: **loop**
4: Choose a lag time and fragment length.
5: Construct and reweight an MSM from each trajectory group, using those parameters.
6: Estimate the equilibrium distribution for each set.
7: Compute the average set-set KL divergence between the equilibrium estimates.
8: **end loop**
9: Identify the optimal set of parameters, which yielded the lowest average set-set KL divergence.

Instead of optimizing for equilibrium, optimizing directly for self-consistency in the observable of interest could also be a viable strategy. However, since the optimization involves attempting construction at different lagtimes, this requires doing splicing in each round of optimization at the lagtime being tested, which adds computational cost. Preliminary results did not indicate a substantial improvement in hyperparameter estimation, so we focus on optimizing for equilibrium self-consistency.

To demonstrate a practical application of this approach, we analyze MD-like trajectories generated from a synthetic model of the Trp-cage miniprotein using **synthetic dynamics (SynD)**.[110] Using SynD enables efficient generation of test data that closely mirrors the complexity of MD simulation data, while providing the ability to calculate exact reference values for observables.

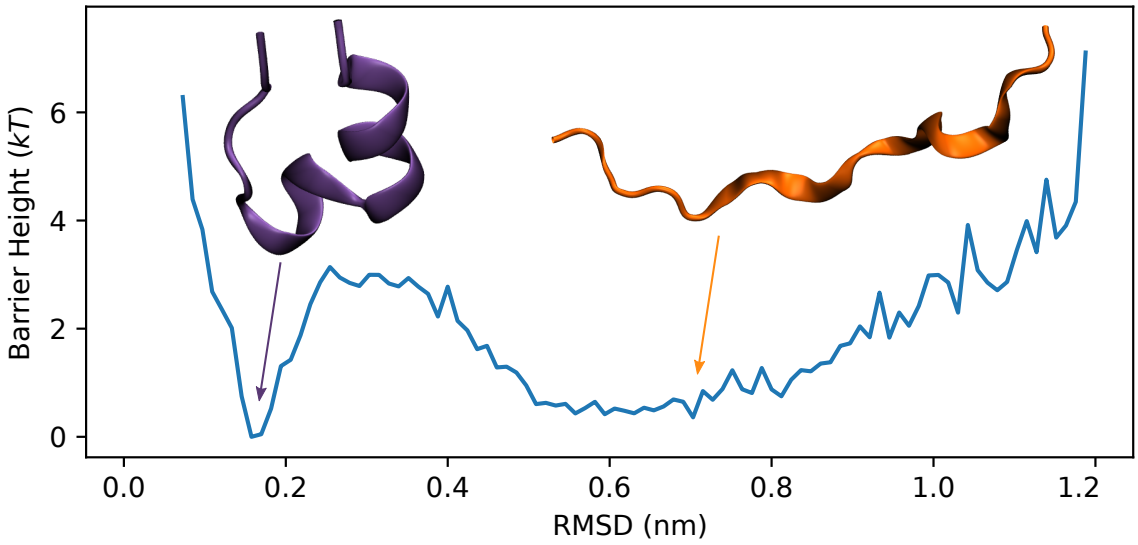


Figure 28: SynD Trp-cage energy landscape. The full SynD model has 10,000 states, which have been coarse-grained to 100 states here for visualization. Example folded (purple) and unfolded (orange) structures are also shown.

3.4.4 Synthetic Trp-cage system details

The synthetic Trp-cage model was developed by constructing an MSM from 208 µs of MD simulation data generated by the Shaw group,[25] following prior work by Suarez and coworkers.[99] This MD simulation data was featurized using minimum residue-residue distances, excluding nearest neighbors. The featurized distances were dimensionality reduced using VAMP[97] at a 10ns lagtime. The dimensionality-reduced data was clustered and discretized using stratified K-means clustering[162] with 21 strata and 500 clusters per stratum resulting in a total of 10,500 clusters. A transition matrix was computed from the discretized trajectories at a lagtime of 1ns. For each cluster, a representative structure was selected at random from the trajectory frames associated with that cluster. The final SynD model is parameterized by the transition matrix and representative structures, and its energy landscape is shown in Fig. 28.

We analyze a set of 10,000 250ns atomistic synthetic trajectories generated using this SynD model. The initial points of the trajectories were deliberately chosen to be far from equilibrium in order to emphasize the effects of initial bias. The fraction of points in the folded and unfolded states is shown in Fig. 30 for the initial point distribution, and for the reference equilibrium distribution.

The synthetic trajectories were featurized and dimensionality reduced in the same manner as the original MD trajectories. A coarser clustering was performed, using 3 strata and 10 clusters per stratum. The resulting set of 10,000 discretized trajectories was used for constructing the standard and reweighted MSMs examined in the subsequent sections.

3.4.5 MSM parameters

Standard MSMs were constructed at a range of times to assess the implied timescales, yielding Fig. 29. Based on this, a lagtime of 100 was selected for the standard MSM.

For the reweighted MSM, hyperparameter optimization determined an optimal lagtime of 101 and an optimal fragment length of 191. Given the close similarity of the optimal lagtimes identified through hyperparameter optimization and from the implied timescales, the lagtime of 100 was also used for the reweighted MSM.

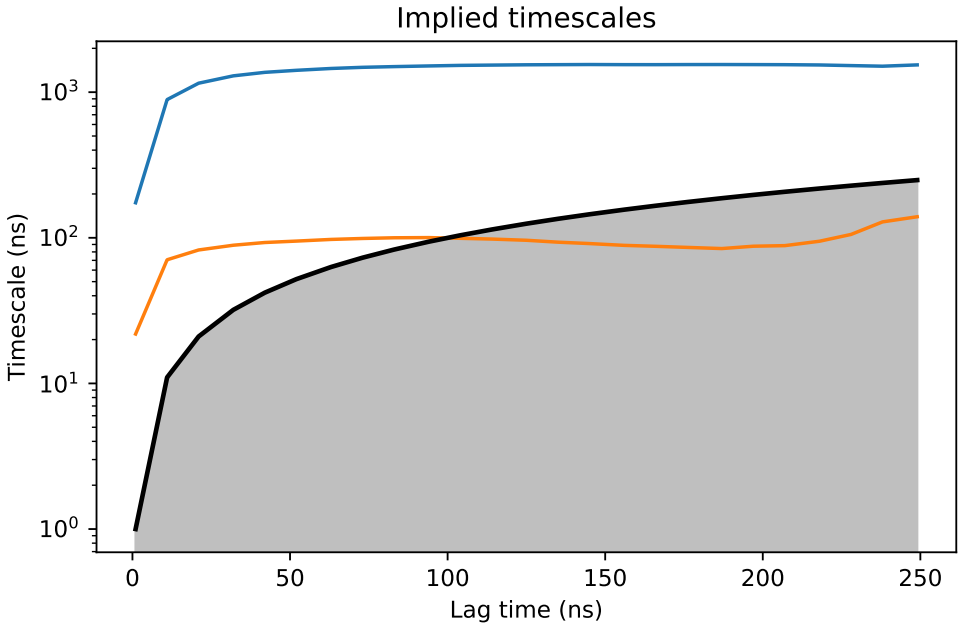


Figure 29: Slowest two implied timescales for the standard MSM. A lagtime of 100, after the timescales plateau, was chosen to obtain accurate estimates of the system's kinetics.

We use the `deeptime` Python package[163] to analyze our transition matrices, for determining implied timescales, and for computing MFPTs.

3.5 Trajectory Analysis Results

We now examine the quality of the reweighted estimators for a finite set of data generated from the synthetic Trp-cage system.

3.5.1 Reweighted equilibrium estimates

The standard MSM is an unbiased estimator for the equilibrium distribution only in the asymptotic, infinite-data limit. By selecting initial points for the trajectories that deviate significantly from the true equilibrium distribution, we amplify the effects of initial state bias in the MSM equilibrium estimates, as mentioned before.[156]

The biased initial distribution in our data is compared to the true equilibrium distribution in Fig. 30. Additionally, Fig. 30 demonstrates that the MSM estimates of the relative folded and unfolded populations partially compensate for this initial state bias. However, applying the reweighting procedure further improves the equilibrium estimates.

The improvement in equilibrium population estimates is more pronounced in the folded state, where a larger number of trajectories were initiated. The standard MSM estimate performs relatively worse than the reweighted estimate in the folded state, possibly because a larger number of trajectories were initiated there, amplifying the initial bias.

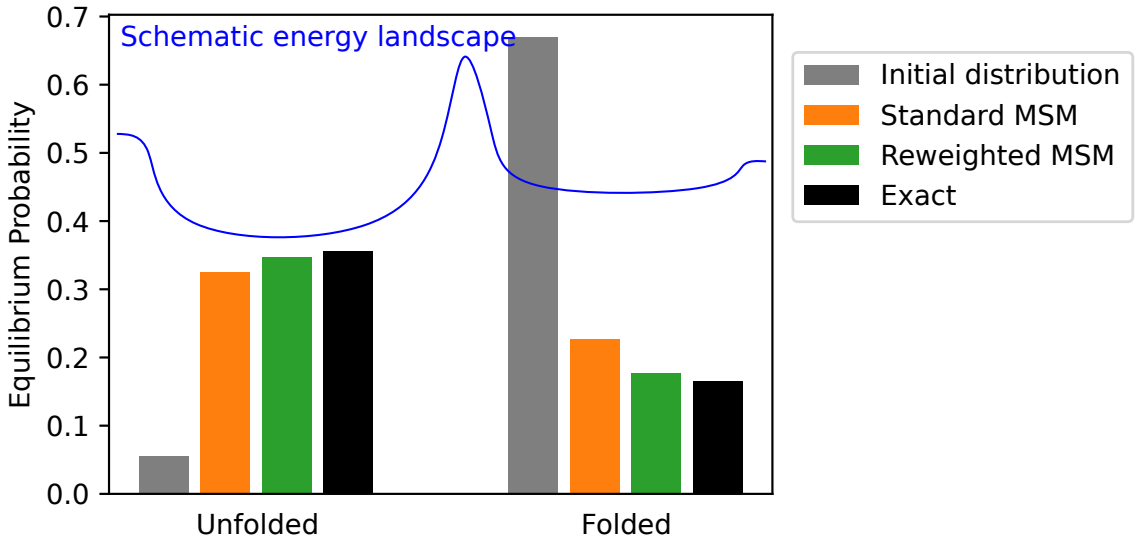


Figure 30: Comparison of equilibrium population estimates from standard and reweighted MSMs. Both MSMs were built at a lag time of 100. The reweighted MSM used a fragment length of 191, selected via hyperparameter optimization for maximal self-consistency of nonequilibrium steady-state as described in Sec. 3.4.3. Initial points of the trajectories (gray) were chosen to be far from the true equilibrium (black) to emphasize the effect of initial state bias.

3.5.2 Reweighted MFPT estimates

Unbiased MFPT estimation requires applying the trajectory splicing method described in Sec. 3.4.1 before computing the transition matrix. Consequently, we compare both the standard and reweighted MSMs with and without splicing, to examine the effects of both the unbiased MFPT estimator and the reweighting procedure.

Fig. 31 demonstrates that the proper incorporation of boundary conditions improves MFPT estimates, particularly at short lag times. Additionally, applying the reweighting procedure during MSM construction further refines the MFPT estimates, with the impact being most pronounced at short lag times.

3.5.3 Reweighted committor estimates

Finally, we compare the standard first-step committor estimates derived from an MSM to the ratio-based estimates obtained from the reweighted MSM. Since the unbiased estimator for committors necessitates solving the NESS, which

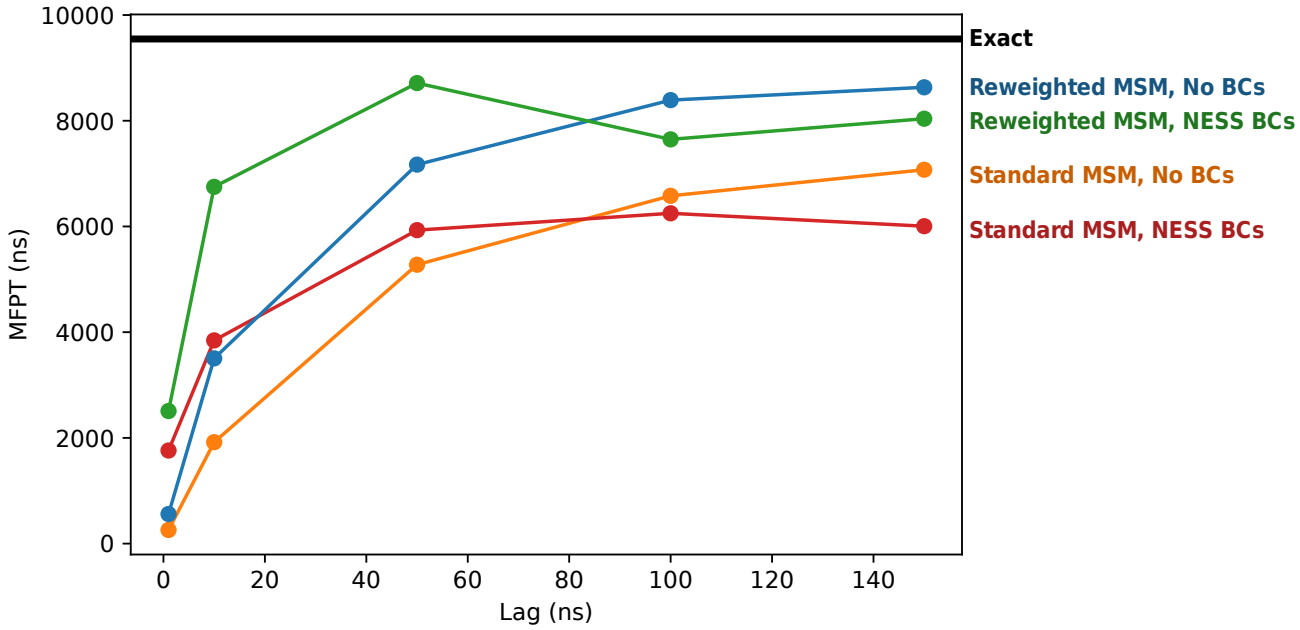


Figure 31: Comparison of MFPT estimates from standard and reweighted MSMs, from trajectories with and without NESS boundary conditions. Incorporating NESS boundary conditions into trajectories improves MFPT estimates, and the reweighting procedure further enhances these estimates. Improvement is particularly pronounced at short lags.

in turn requires trajectory splicing, we do not examine boundary conditions and reweighting separately as we did for the MFPT in the previous section.

Although the ratio-based estimator is asymptotically unbiased, it exhibits substantial noise in practice. This increased noise is likely attributable, at least in part, to the estimator being calculated as a ratio of two other estimated quantities.

3.6 Concluding Discussion

In this work, we examined the effects of asymptotically unbiased estimators for both equilibrium and kinetic observables, and the ability of a self-consistent iterative reweighting approach to correct for finite-data bias.

Using exact discrete-state calculations enabled us to analytically compare estimators, sidestepping sampling concerns. We also applied these estimators to sets of trajectories generated from a synthetic dynamics model, which allowed us to assess their performance on data of similar complexity to MD data, but with exactly known reference quantities.

Although it has been known that standard MSMs in principle provide unbiased estimation of equilibrium populations[99,137] and also that history traceback could allow unbiased estimation of the MFPT,[99,137] we believe that unbiased estimators for the committor values of coarse-grained states were not previously available in an MSM framework. These estimators highlight the critical importance of boundary conditions (applied before constructing the transition matrix), which was not previously appreciated as far as we know. Furthermore, the relaxation properties of the estimators were not previously assessed to our knowledge.

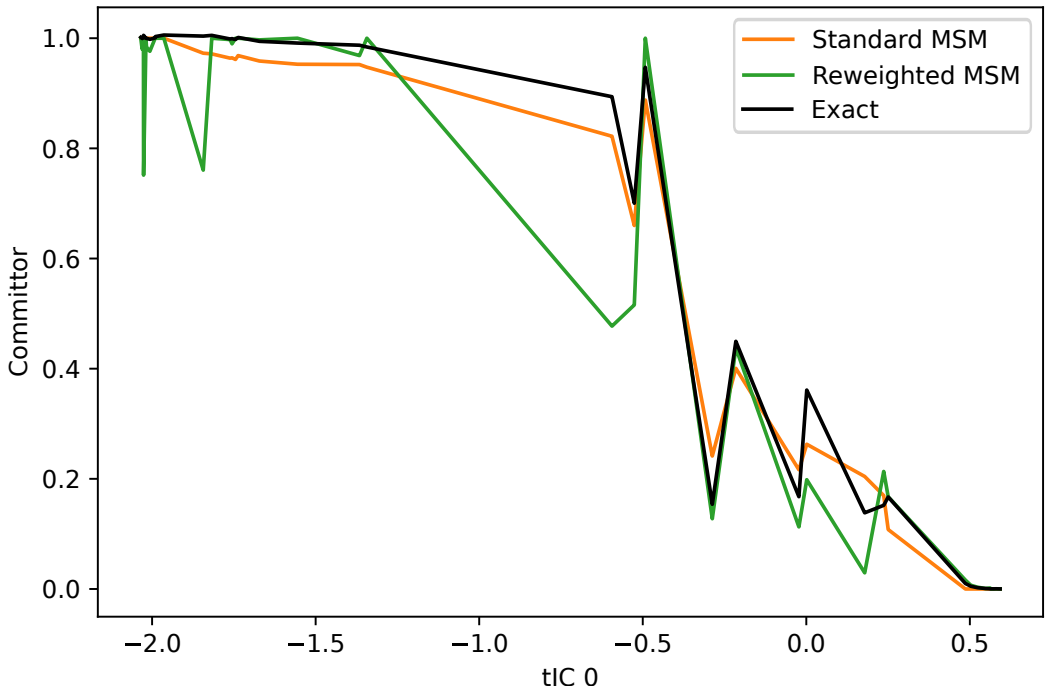


Figure 32: Comparison of committor estimates from standard and reweighted MSMs. The standard MSM exhibits systematic bias, reflecting the biased initial point distribution, while the reweighted MSM shows substantial noise.

We emphasize that relaxation of estimated observables (as sliding-window averaging occurs) is not an abstract issue, but directly impacts whether unbiased estimates can be obtained using feasible amounts of data and trajectory lengths. We also showed that extending the reweighting idea proposed by Voelz and coworkers[153] has the potential to make a significant difference in practical unbiased estimation. Although our work relied on a discretized microscopic dynamics, it is not difficult to see that almost identical considerations apply to continuous trajectories.

When applied to finite amounts of real trajectory data with substantial initial state bias, the reweighted estimator for equilibrium was able to improve upon the standard MSM's estimate. Using the correct estimator for the MFPT by incorporating source-sink boundary conditions improved MFPT estimates at short lags for both the standard MSM and the reweighted MSM, and reweighting significantly improved the standard MSM estimate. While the committor estimator is theoretically asymptotically unbiased, in practice it produced noisy estimates.

The new estimators we present in this work enable unbiased estimation of kinetics from an MSM, which is not possible with a standard MSM. Although committor estimates were noisy, the improvement in both equilibrium and MFPT estimates suggests the reweighting approach is able to mitigate the impact of initial state bias. Combined, these approaches provide an improved pipeline for observable estimation, which is more robust to short trajectories than a standard MSM.

Developing strategies to mitigate noise in the committor estimator could improve its practical utility. Additionally, applying this reweighting methodology to weighted ensemble data could substantially improve estimation of

observables from the limited datasets typical of rare event sampling. Finally, a deeper examination and comparison of different hyperparameter optimization heuristics could produce improve the quality and convergence of the optimization procedure.

3.7 Acknowledgements

We gratefully acknowledge support from the National Institutes of Health via Grant GM115805 and from the National Science Foundation via Grant DMS-181871 to DA and GS. Early discussions with Ernesto Suarez were of great value.

4 WESTPA 2.0: High-performance upgrades for weighted ensemble simulations and analysis of longer-timescale applications

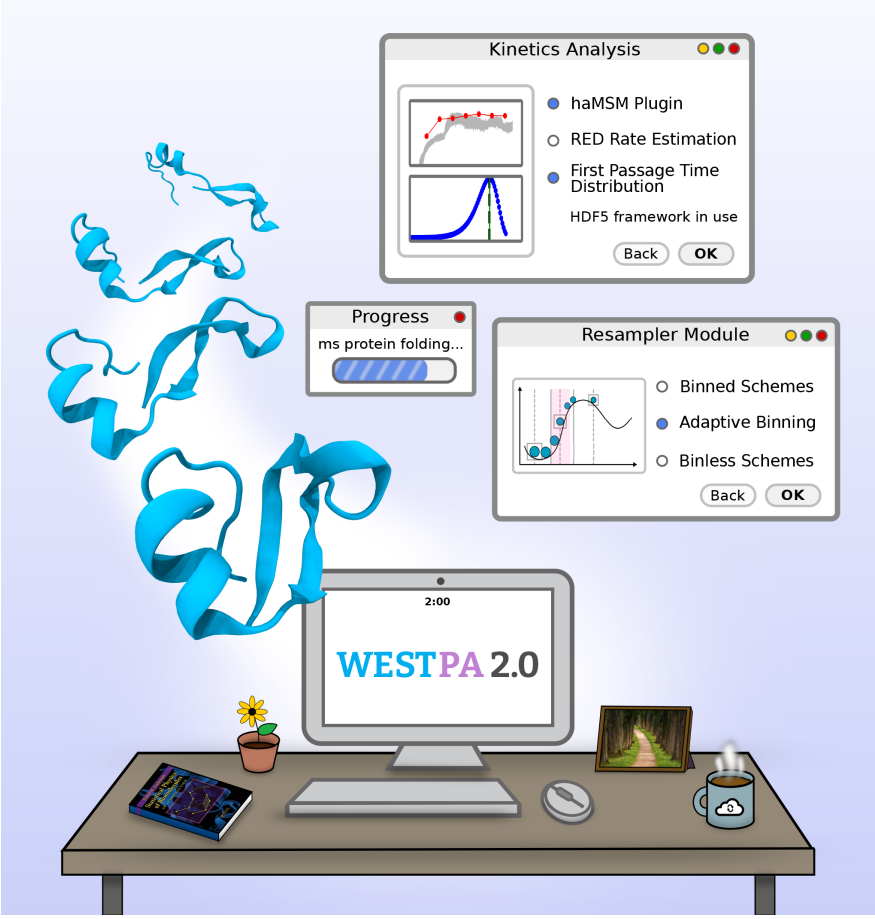
ABSTRACT

The **weighted ensemble (WE)** family of methods is one of several statistical-mechanics based path sampling strategies that can provide estimates of key observables (rate constants, pathways) using a fraction of the time required by direct simulation methods such as molecular dynamics or discrete-state stochastic algorithms. WE methods oversee numerous parallel trajectories using intermittent overhead operations at fixed time intervals, enabling facile interoperability with any dynamics engine. Here, we report on major upgrades to the WESTPA software package, an open-source, high-performance framework that implements both basic and recently developed WE methods. These upgrades offer substantial improvements over traditional WE. Key features of the new WESTPA 2.0 software enhance efficiency and ease of use: an adaptive binning scheme for more efficient surmounting of large free energy barriers, streamlined handling of large simulation datasets, exponentially improved analysis of kinetics, and developer-friendly tools for creating new WE methods, including a Python API and resampler module for implementing both binned and "binless" WE strategies.

Note

The WESTPA 2.0 software release was the result of a substantial amount of work by many software developers from many institutions. As one of three core maintainers, I contributed heavily to a number of components, which are described in the following chapter. This work captures notable major contributions, although the scope of my involvement with WESTPA has been broad.

This work was originally published in [45] and is reprinted here with permission. My contributions did not include the work described in Sec. 4.3.3, Sec. 4.3.4, or Sec. 4.4.1.



4.1 Introduction

The field of molecular dynamics (MD) simulations of biomolecules arguably is following a trajectory that is typical of mathematical modeling efforts: as scientific knowledge grows, models grow ever more complex and ambitious, rendering them challenging for computation. While early MD simulations focused on single-domain small proteins,[24] modern simulations have attacked ever larger complexes[33,164] and even entire virus particles.[165–168] This trend belies the fact that record-setting small-protein simulations in terms of total simulation time remain limited to the ms scale on special-purpose resources[169] and to < 100 µs on typical university clusters. These limitations have motivated the

development of numerous approaches to accelerate sampling, among which are rigorous path-sampling approaches capable of providing unbiased kinetic and mechanistic observables.[43,55,58,61,68,133,170–173]

Our focus is the weighted ensemble (WE) path sampling approach,[43,44] which has helped to transform what is feasible for molecular simulations in the generation of pathways for long-timescale processes (> μs) with rigorous kinetics. Among these simulations are notable applications, including atomically detailed simulations of protein folding,[74] coupled protein folding and binding,[77] protein-protein binding,[174] protein-ligand unbinding,[175] and the large-scale opening of the SARS-CoV-2 spike protein.[11] The latter is a significant milestone—both in system size (half a million atoms) and timescale (seconds).[11] Instrumental to the success of the above applications have been advances in not only WE methods, but also software.[11]

Here, we present the next generation (version 2.0) of the most cited, open-source WE software called WESTPA (Weighted Ensemble Simulation Toolkit with Parallelization and Analysis).[176] WESTPA 2.0 is designed to further enhance the efficiency of WE simulations with high-performance algorithms for: (i) further enhanced sampling via restarting from reweighted trajectories, adaptive binning, and/or binless strategies, (ii) more efficient handling of large simulation datasets, and (iii) analysis tools for estimation of first-passage-time distributions and for more efficient estimation of rate constants. Like its predecessor, WESTPA 2.0 is a highly scalable, portable, and interoperable Python package that embodies the full range of WE's capabilities, including rigorous theory for any type of stochastic dynamics (e.g., molecular dynamics and Monte Carlo simulations) that is agnostic to the model resolution.[177] In comparison to other open-source WE packages such as AWE-WQ[178] and wepy,[179] WESTPA is unique in its (i) high scalability with nearly perfect scaling out to thousands of CPU cores[11] and GPUs, and (ii) demonstrated ability to interface with a variety of dynamics engines and model resolutions, including atomistic,[174] coarse-grained,[180] whole-cell,[181] and non-spatial systems models.[182,183]

After a brief overview of the WE strategy (Section 4.2), we describe the organization of WESTPA 2.0 (Section 4.3) and new analysis tools that further expand the capabilities of the software package (Section 4.4). Together, these features greatly facilitate the execution and analysis of WE simulations of even larger systems and/or slower timescales.

4.2 Overview of the WE Path Sampling Strategy

The weighted ensemble (WE) strategy enhances the sampling of rare events (e.g., protein folding, binding, chemical reactions) by orchestrating the periodic resampling of multiple, parallel trajectories at fixed time intervals τ (Figure 33).[43] The statistically rigorous resampling scheme maintains even coverage of configurational space by replicating ("splitting") trajectories that have made transitions to newly visited regions and potentially terminating ("merging") trajectories that have over-populated previously visited regions. The configurational space is typically defined by a progress coordinate that is divided into bins where even coverage of this space is defined as a constant number of trajectories occupying each bin; alternatively, trajectories may be grouped by a desired feature for "binless" resampling schemes.[184] Importantly, trajectories are assigned statistical weights that are rigorously tracked during

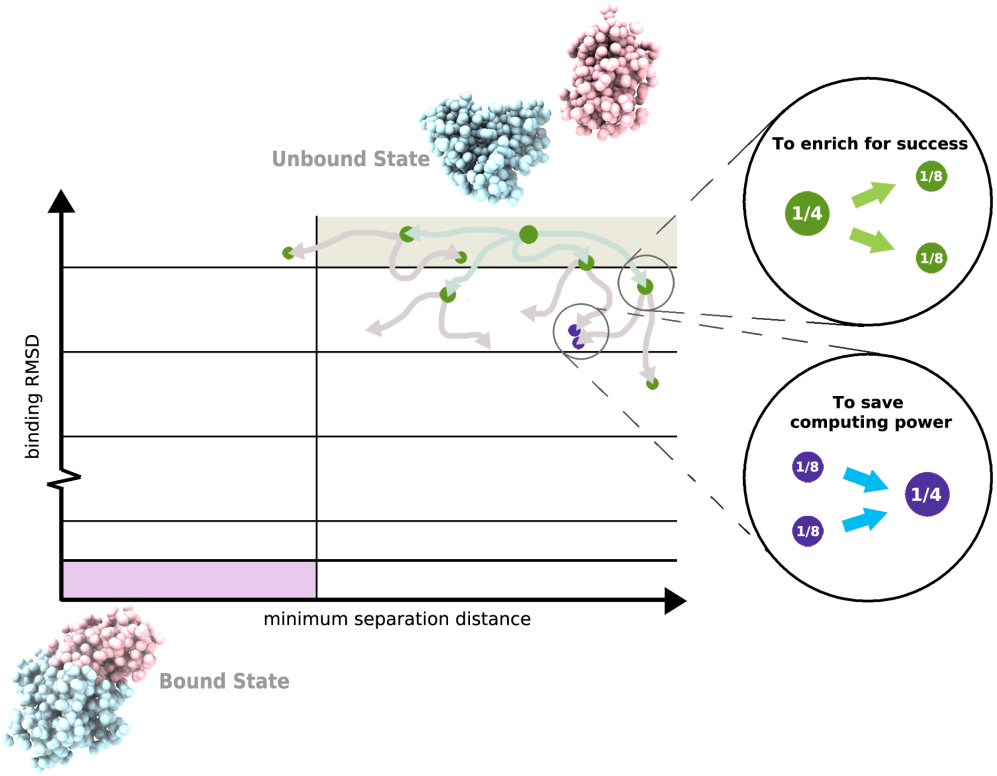


Figure 33: Basic weighted ensemble protocol. As illustrated for the simulation of a protein-protein binding process, a two-dimensional progress coordinate is divided into bins with the goal of occupying each bin with a target number of four trajectories. Four equally weighted trajectories are initiated from the unbound state and subjected to a resampling procedure at periodic time intervals τ: (i) to enrich for success, trajectories that make transitions to less-visited bins are replicated to generate a target of four trajectories in those bins, splitting the weights evenly among the child trajectories (green spheres), and (ii) to save computing time, the lowest-weight trajectories in bins that have exceeded four trajectories are terminated, merging their weights with those of higher-weight trajectories in those bins (purple spheres). Spheres are sized according to their statistical weights.

resampling; when trajectories are replicated in a given bin, the weights are split among child trajectories and when trajectories are terminated in a probabilistic fashion, the weights are merged with a continued trajectory of that bin. This rigorous tracking ensures that no bias is introduced into the ensemble dynamics, enabling direct estimates of rate constants.[177]

WE simulations can be run under equilibrium or non-equilibrium steady state conditions. To maintain non-equilibrium steady state conditions, trajectories that reach the target state are "recycled" back to the initial state, retaining the same statistical weight.[185] The advantage of equilibrium WE simulations over steady-state WE simulations is that the target state need not be strictly defined in advance since no recycling of trajectories at the target state is applied.[186] On the other hand, steady-state WE simulations have been more efficient in yielding successful pathways and estimates of rate constants. Equilibrium observables can be estimated from either equilibrium WE simulations or the combination of two non-equilibrium steady-state WE simulations in opposite directions when history information is taken into account.[186]

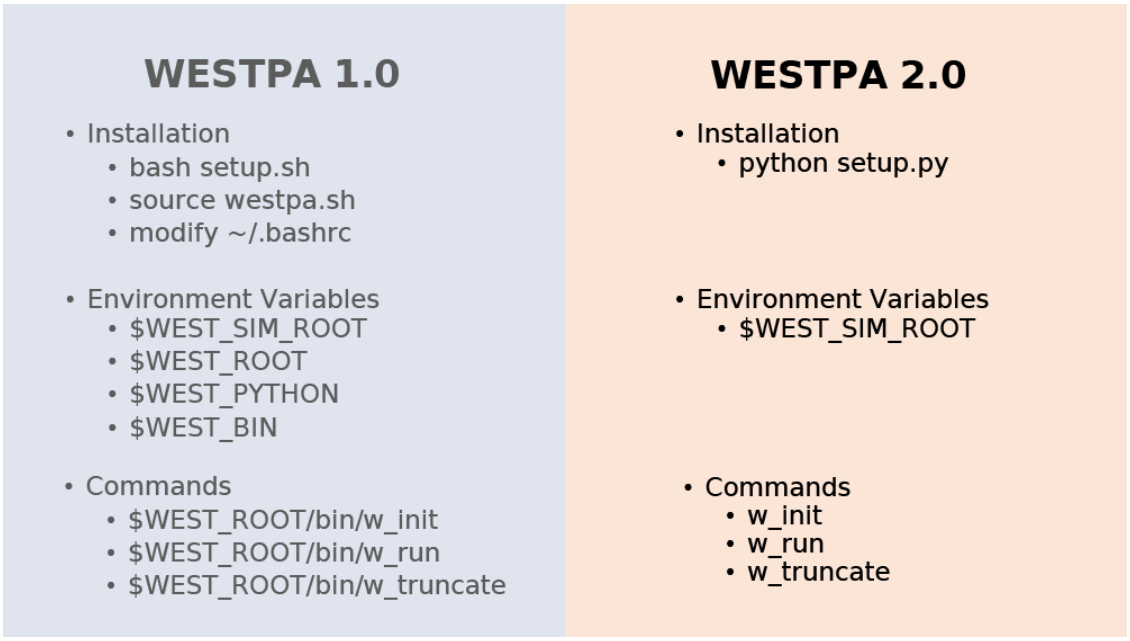


Figure 34: Reorganization of WESTPA 1.0 to WESTPA 2.0. In version 2.0, WESTPA is installed using Python and relies on only a single environment variable such that commands can be called directly through Python. To reflect these changes, we have updated our original suite of WESTPA tutorials for version 2.0 (https://github.com/westpa/westpa_tutorials/tree/westpa-2.0-restruct).[187,188]

4.3 Organization of WESTPA 2.0

Below, we present the organization of WESTPA 2.0, beginning with code reorganization to facilitate software development (Section 4.3.1) and then proceeding to a description of a Python API for setting up, running, and analyzing WE simulations (Section 4.3.2); a minimal adaptive binning mapper (Section 4.3.3); a generalized resampler module that enables the implementation of both binned and binless schemes (Section 4.3.4); and an HDF5 framework for more efficient handling of large simulation datasets (Section 4.3.5).

4.3.1 Code reorganization to facilitate software development

The WESTPA 2.0 software is designed to facilitate the maintenance and further development of the software according to established and emerging best practices for Python development and packaging. The code has been consolidated and reorganized to better indicate the role of each module (Figure 34). The software can now be installed as a standard Python package using pip or by running setup.py. The package will continue to be available through Conda via conda-forge, which streamlines the installation process by enabling WESTPA and all software dependencies to be installed at the same time. We have implemented automated GitHub Actions for continuous integration testing and code quality checks using the Black Python code formatter as a pre-commit hook, alongside flake8 for non-style linting. Templates are provided for GitHub issues and pull requests. Both user's and developer's guides are available on the GitHub wiki along with Sphinx documentation of key functions with autogenerated docstrings. Further support will

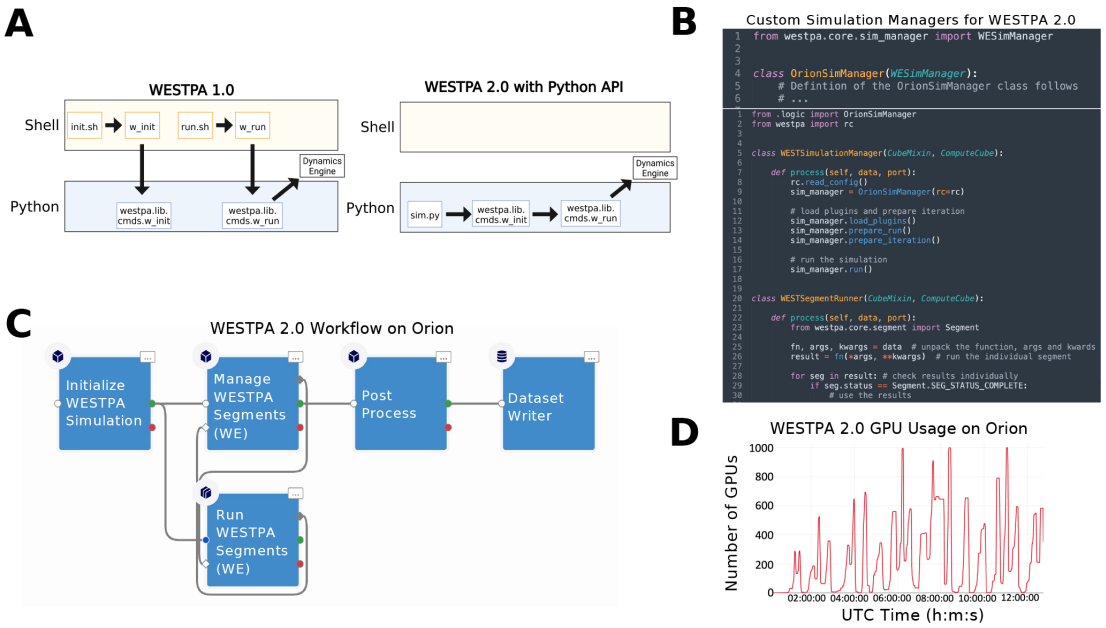


Figure 35: Comparison of workflows for setting up and running WE simulations using WESTPA 1.0 and 2.0, a demonstration of using the Python API for WESTPA 2.0, and GPU performance of the updated API within a cloud computing environment. (A) The Python API in WESTPA 2.0 enables a user to fully define, initialize, and run a WESTPA simulation from within a single Python script (right panel), without needing to invoke command-line utilities required in WESTPA 1.0 (left panel). For backwards compatibility, all original functionality provided in version 1.0 for invoking WESTPA (e.g., `w_init` and `w_run` tools) via shell scripts remains available in WESTPA 2.0. (B) Example of defining a custom simulation manager with the WESTPA 2.0 API (top panel), and using the newly defined simulation manager and WESTPA 2.0 API to programmatically control and run a WE simulation (bottom panel). Here, the `WESTSimulationManager` class sends work to the `WESTSegmentRunner` class that unpacks and runs the scripts specified from the WESTPA config file (west.cfg). (C) Example workflow diagram from the Orion user interface using the Python classes constructed from the internal WESTPA APIs presented in Figure 35B. Here, a kernel (Initialize WESTPA Simulation) initializes both the `WESTSimulationManager` (Manage WESTPA Segments) and the `WESTSimulationRunner` (Run WESTPA Segments) kernels from Figure 35, which are connected in a cycle to manage splitting and merging. Finally, all data is exported through a Post Process and Dataset Writer kernel for final data processing and storage. (D) Performance of the WESTPA 2.0 API using the `WESTSimulationRunner` class from Figure 35 within an Amazon Web Services environment using a combination of numerous g4dn instances as a function of wallclock time in Universal Coordinated Time (UTC) units. Here, the per-iteration scaling reaches thousands of GPUs in just under a few hours for a test system of butanol crossing a neat POPC membrane bilayer using the WESTPA 2.0 API with the OpenMM 7.5 MD engine.[189]

continue to be provided through WESTPA users' and developers' email lists hosted on Google Groups (linked on https://westpa.github.io).

4.3.2 Python API for setting up, running, and analysis of WE simulations

To simplify the process of setting up and running WE simulations, WESTPA 2.0 features a Python API that enables the user to execute the relevant commands within a single Python script instead of invoking a series of command-line tools, as previously done in WESTPA 1.0 (Figure 35a). This also provides tools for third-party developers to build and develop WESTPA-based applications and plugins, for example, the integration of WESTPA into the cloud-based computing platform, OpenEye Scientific's Orion;[78,190] or the **history-augmented Markov state model (haMSM)**

restarting plugin (Section 4.4.2), which uses the results of a WESTPA simulation to perform analysis then restart the simulation based on the results of that analysis.

Figure 35b provides an example of how to programmatically call the WESTPA 2.0 API from the Orion cloud platform, which could in principle be any Python script within any supercomputing or personal computing environment. First, a developer can write any custom simulation or work manager of their choice by subclassing or completely rewriting core WESTPA components (top panel). Second, a workflow can be constructed by invoking a simple set of WESTPA 2.0 Python commands to perform any WE simulation (bottom panel). Typically, a user of the WESTPA 2.0 Python API only needs a handful of API endpoints to perform a complicated simulation protocol. As an example of the power of the simplicity of the Python API, we demonstrate how a workflow can be constructed from defined workflow kernels (Figure 35c), and show GPU performance over wall-clock time (in Coordinated Universal Time; UTC) from a drug-like molecule in a membrane permeability simulation (Figure 35d). Using the internal API, a user's simulation can request large amounts of compute resources per iteration. In this case, thousands of GPUs are requested per WE iteration for a simulation of butanol crossing a natural membrane mimetic system (`https://github.com/westpa/westpa2_tutorials`).[191]

To facilitate the development of custom analysis workflows in cases where more flexibility is required than the existing `w_ipa` analysis tool,[187] WESTPA 2.0 includes the new westpa.analysis Python API. This API provides a high-level view of the data contained in the main WESTPA HDF5 file ("`west.h5`"), including the trajectory data, reducing the overhead of writing custom analysis code in Python and doing quick, interactive analysis of trajectories (or walkers). The westpa.analysis API is built on three core data types: `Run`, `Iteration`, and `Walker`. A `Run` is a sequence of `Iterations`; an `Iteration` is a collection of `Walkers`. Key instance data can be accessed via attributes and methods. For example, a `Walker` has attributes such as the statistical weight (weight), progress coordinate value (pcoords), starting conformation (parent), and child trajectories after replication (children), and a method trace() to trace its history (as a pure Python alternative to the w_trace tool). The API also provides facilities for retrieving and concatenating trajectory segments. These include support for (i) type-aware concatenation of trajectory segments represented by NumPy arrays or MDTraj trajectories, (ii) use of multiple threads to potentially increase performance when segment retrieval is an I/O bound operation, and (iii) display of progress bars. Finally, the API provides a convenience function, `time_average()`, for computing the time average of an observable over a sequence of `Iterations` (e.g., all or part of a `Run`).

4.3.3 A minimal adaptive binning mapper

To automate the placement of bins along a chosen progress coordinate during WE simulation, we have implemented the Minimal Adaptive Binning (MAB) scheme[192] as an option in the westpa.core.binning module. The MAB scheme positions a specified number of bins along a progress coordinate after each resampling interval τ by (1) tagging the positions of the trailing and leading trajectories along the progress coordinate and evenly placing a specified number of bins between these positions, and (2) tagging "bottleneck" trajectories positioned on the steepest probability gradients

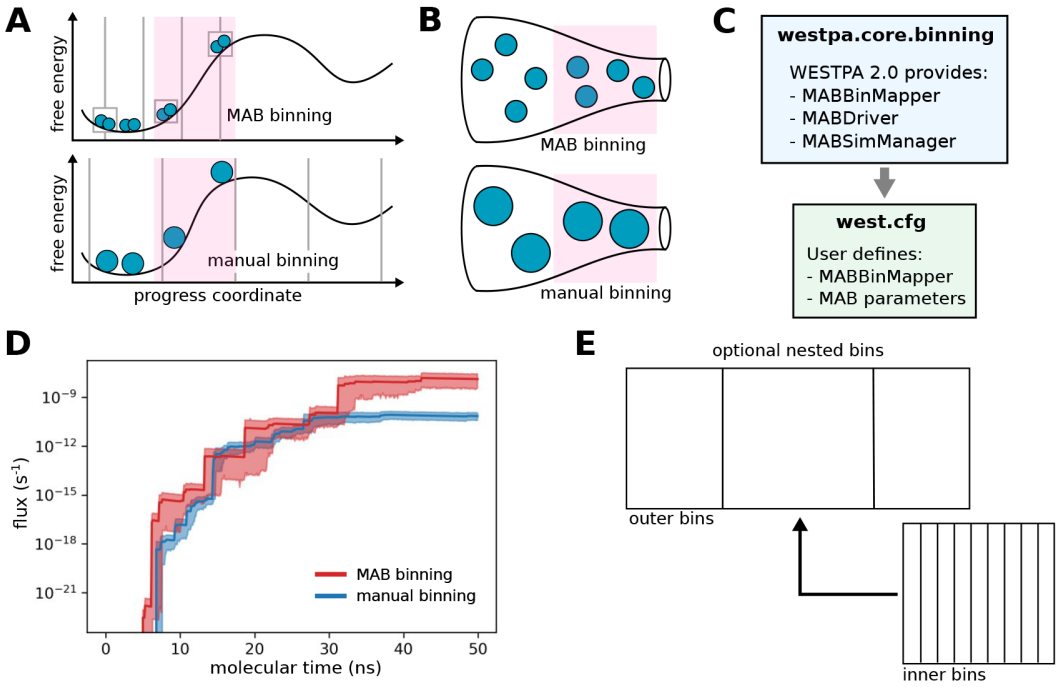


Figure 36: The minimal adaptive binning (MAB) scheme is more efficient in surmounting free energy barriers than manual, fixed binning schemes. (A) Bin positions and trajectories after replication using the MAB scheme vs. a manual binning scheme with the same positions of trajectories (blue circles, sized according to statistical weights) along a chosen progress coordinate and a target of two trajectories per bin. The MAB scheme adaptively positions bins along the progress coordinate by placing equally spaced bins (in this case, three bins, as indicated by solid vertical lines) between the positions of the trailing and leading trajectories along with separate bins (boxes) for these trajectories and a third trajectory in a bottleneck region (pink) along the free energy barrier. (B) Enlarged "bottle" diagrams highlighting the bottleneck region (pink) along with relative positions and weights of trajectories for the MAB and manual binning schemes in panel A). In contrast to the manual binning scheme where trajectories may stall in a bottleneck region, the MAB scheme automatically detects trajectories in this region, replicating these trajectories to enrich for success in surmounting the barrier. (C) MAB-scheme options in the westpa.core.binning module and corresponding user-defined options in the `west.cfg` file. (D) Flux of a drug-like molecule (tacrine) permeating through a neat POPC membrane as a function of molecular time using fixed binning (blue) or adaptive binning (MAB scheme) (red). Solid lines represent mean fluxes and the shaded regions represent 95% confidence intervals. The molecular time is defined as $N\tau$, where N is the number of WE iterations and τ is the fixed time interval (100 ps) of each WE iteration. Simulations were run using WESTPA 2.0 and the OpenMM 7.5 MD engine.[189] (E) Schematic of a simple recursive binning case in which closely spaced inner bins are "nested" within a wider outer bin.

and assigning these trajectories to their own bins (Figures 36A-B). Despite its simplicity, the MAB scheme requires less computing time than manual, fixed binning schemes in surmounting large free energy barriers resulting in more efficient conformational sampling and estimation of rate constants.[192] To apply the MAB scheme, users specify the `MABBinMapper` option along with accompanying parameters such as the number of bins in the `west.cfg` file (Figure 36C).

Figure 36D illustrates the effectiveness of the MAB scheme in enhancing the efficiency of simulating the membrane permeability of a drug-like molecule (tacrine). Relative to a fixed binning scheme, the MAB scheme results in earlier flux of tacrine through a model cellular membrane bilayer (5 ns vs. 7 ns) and this flux increases more quickly, achieving values that are two orders of magnitude higher for the duration of the test.

The MAB scheme provides a general framework for user creation of more complex adaptive binning schemes.[192] Users can now specify nested binning schemes in the `west.cfg` file (Figure 36E). To run WESTPA simulations under non-equilibrium steady-state conditions (i.e. with "recycling" of trajectories that reach the target state) with the MAB scheme, users can nest a `MABBinMapper` inside of a `RecursiveBinMapper` bin and specify a target state as the outer bins. Multiple individual `MABBinMappers` can be created and placed at different locations of the outer bins using a recursive scheme, offering further flexibility in the creation of advanced binning schemes.

4.3.4 Generalized resampler module that enables binless schemes

In the original (default) weighted-ensemble resampling scheme, trajectories are split and merged based on a predefined set of bins.[43] In WESTPA 2.0, we introduce a generalized resampler module that enables users to implement both binned and "binless" resampling schemes, providing the flexibility to resample trajectories based on a property of interest by defining a grouping function. While grouping on the state last visited (e.g., initial or target state) was previously possible using the binning machinery in WESTPA 1.0[193] our new resampler module provides a more general framework for creating binless schemes by defining a group/reward function of interest, such schemes are also essential for use with nonlinear progress coordinates that may be identified by machine learning techniques. Following others,[194] the resampler module includes options for (i) specifying a minimum threshold for trajectory weights to avoid running trajectories with inconsequentially low weights, and (ii) specifying a maximum threshold for trajectory weights to avoid a single large-weight trajectory from dominating the sampling, increasing the number of uncorrelated successful events that reach the target state.

As illustrated in Figure 37, the implementation of a binless scheme requires two modifications to the default WESTPA simulation: (i) a user-provided group module containing the methods needed to process the resampling property of interest for each trajectory walker, and (ii) updates to the `west.cfg` file specifying the resampling method in the `group_function` keyword and the attribute in the `group_arguments` keyword.

We provide two examples of implementing binless schemes in the `westpa-2.0-restruct` branch of the `WESTPA_Tutorials` GitHub repository

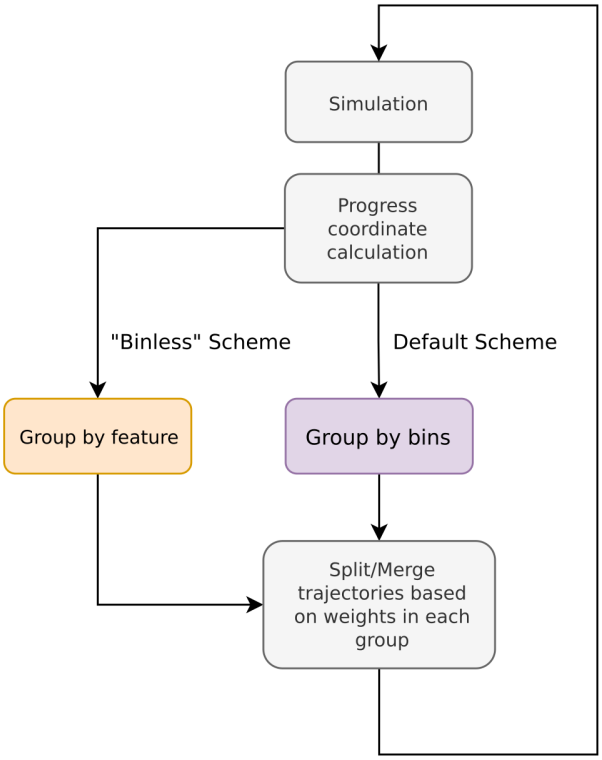


Figure 37: Flowchart for implementing binless resampling schemes in WESTPA 2.0. The implementation involves grouping trajectories by feature (using the group_function defined in the group module) before splitting and merging. The functionality for positioning bins along a chosen progress coordinate remains available in WESTPA 2.0.

(`https://github.com/westpa/westpa_tutorials/tree/westpa-2.0-restruct`).[188] The `basic_nacl_group_by_history` example illustrates grouping of trajectory based on its "history", i.e. a shared parent N WE iterations back. The parameter N is specified in the keyword `hist_length` under the `group_arguments` keyword in the `west.cfg` file. This WESTPA configuration file also specifies the name of the grouping function method, `group_walkers_by_history`, in the `group_function` keyword. In the `basic_nacl_group_by_color example`, trajectory walkers are tagged based on "color" according to the state last visited. Only walkers that have the same color are merged, thereby increasing the sampling of pathways in both directions. State definitions are declared within the `group_arguments` keyword in the `west.cfg` file.

4.3.5 HDF5 framework for more efficient handling of large simulation datasets

One major challenge of running WE simulations has been the management of the resulting large datasets, which can amount to tens of terabytes over millions of trajectory files. To address this challenge, we have developed a framework for storing trajectory data in a highly compressed and portable HDF5 file format. The format is derived from the HDFReporter class implemented in the MDTraj analysis suite,[195] and maintains compatibility with NGLView,[196] an iPython/Jupyter widget for interactive viewing of molecular structures and trajectories. A single HDF5 file is generated per WE iteration, which includes a link to each trajectory file stored in the main WESTPA data file (`west.h5`). Thus,

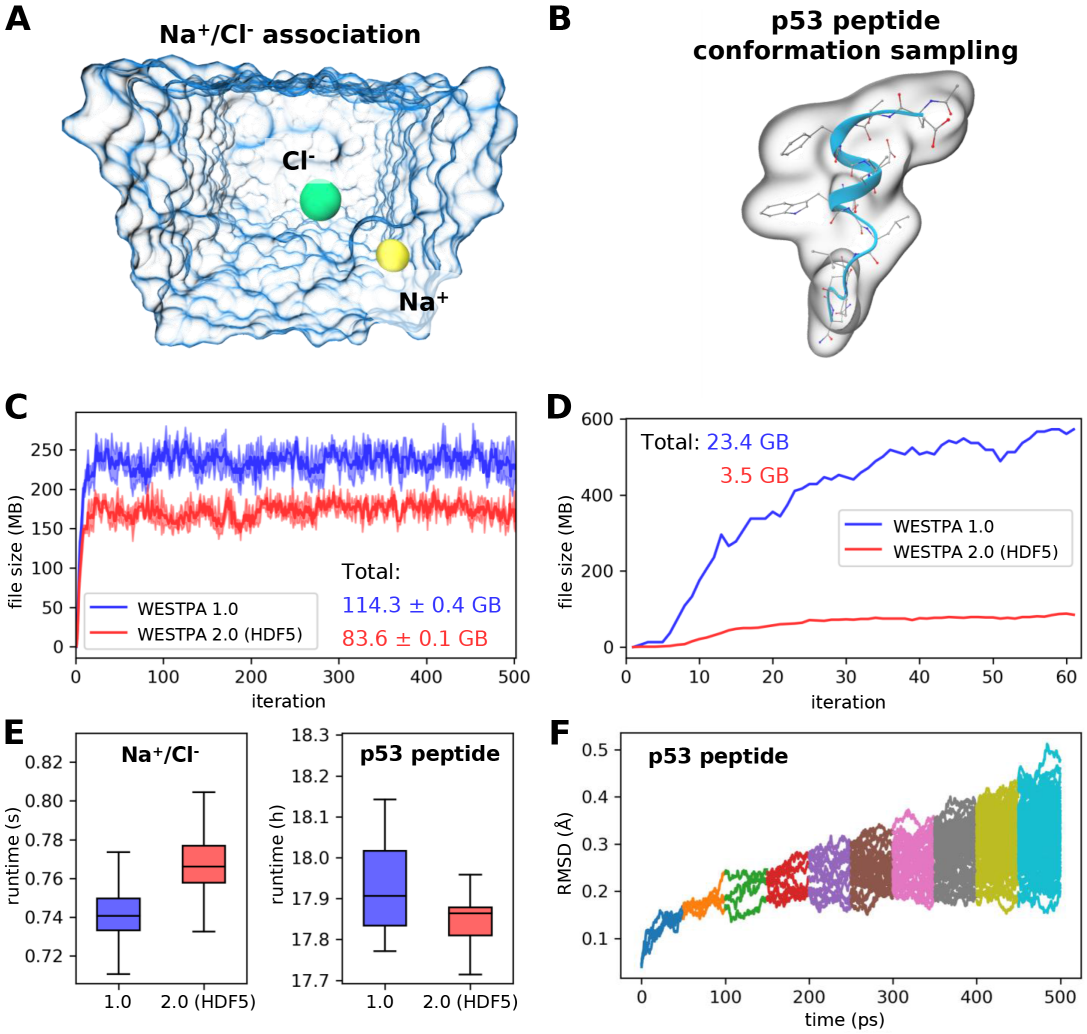


Figure 38: Demonstration of the usage of the HDF5 framework for two example systems. (A) Na+/Cl- association simulation where Na+ (yellow sphere) and Cl- (green sphere) ions were solvated in explicit water (blue transparent surface). The distance between the two ions serves as the progress coordinate. (B) Conformational sampling of a p53 peptide (residues 17-29) in generalized Born implicit solvent using a progress coordinate consisting of the heavy-atom RMSD of the peptide from its MDM2-bound conformation.[77] The molecular surface of the p53 peptide is rendered as a transparent surface, with both the secondary (blue ribbon) and atomic structures overlaid. (C) Comparison of file sizes of per-iteration HDF5 files for the Na+/Cl- association simulation as a function of the WE iteration using WESTPA 1.0 and 2.0 with the HDF5 framework. The result was obtained from three independent simulations where the solid curves show the mean file sizes, while the light bands show the standard deviations. (D) Same comparison as panel C for a single simulation of the p53 peptide, hence no error bars are shown. (E) Comparison of wall-clock runtimes normalized by the number of trajectory segments per WE iteration using WESTPA 1.0 and 2.0 with the HDF5 framework option turned on. (F) Time-evolution of the heavy-atom RMSD of the p53 peptide from its MDM2-bound conformation by trajectories obtained using WESTPA's analysis tools. Colors represent RMSDs obtained from different iterations. WESTPA simulations of Na+/Cl- association and the p53 peptide were run using the OpenMM 7.5 MD engine.[189]

the new HDF5 framework in WESTPA 2.0 enables users to restart a WE simulation from a single HDF5 file rather than millions of trajectory files and simplifies data sharing as well as analysis. The dramatic reduction in the number of trajectory files also eliminates potentially large overhead from the filesystem that results from the storage of numerous small files. For example, a 53% overhead has been observed for a 7.5-GB dataset of 103,260 trajectory files generated from NTL9 protein folding simulations (Figure 41), occupying 11.5 GB of actual disk storage on a Lustre filesystem.

To test the effectiveness of the HDF5 framework in reducing the amount of data storage required for WE simulations, we applied the framework to a set of three independent WE simulations of Na+/Cl- association and one WE simulation involving p53 peptide conformational sampling (Figures 38A-B). Our results revealed 27% and 85% average reduction in the total size of trajectory files generated during the Na+/Cl- association and p53 peptide simulations, respectively, relative to WESTPA 1.0. Given a fixed number of bins, the sizes of per-iteration HDF5 files were also shown to converge as the simulation progresses (Figures 38C-D), suggesting that the storage of trajectory data by iteration not only facilitates the management of the data but also yields files of roughly equal sizes. The difference in the reduction efficiency that we observed between the Na+/Cl- and p53 peptide systems can be attributed to differences in the simulation configurations including the format of the output trajectories, restart files and other factors such as the verbosity of logging.

Our tests revealed that the additional steps introduced by the HDF5 framework for managing trajectory coordinate and restart files did not have any significant impact on the WESTPA runtime (Figure 38E), which is normalized by the number of trajectory segments per WE iteration given that the evolution of bin occupancies by trajectories can vary among different runs due to the stochastic nature of the MD simulations (after 60 iterations, the WESTPA 1.0 run occupied six more bins than the WESTPA 2.0/HDF5 run). This variation in the bin occupancy is unlikely to be affected by the HDF5 framework since it simply manages the trajectory and restart files and does not alter how the system is simulated. The differences in bin occupancies/total number of trajectories may also partially contribute to the large reduction in per-iteration file sizes for the HDF5 run observed in Figure 38D of the p53 peptide. However, the majority of this file-size reduction results from efficient HDF5 data compression of trajectory coordinates, restart, and log files. Finally, trajectory data saved in the HDF5 files can be extracted and analyzed easily using MDTraj in combination with our new analysis framework presented in Section 4.3.2 (Figure 38F).

4.4 Analysis Tools

WESTPA 2.0 features new analysis tools for estimating rate constants more efficiently using the distribution of "barrier crossing" times (Section 4.4.1), accelerating convergence using a history-augmented Markov state model to reweight trajectories (Section 4.4.2), and estimating the distribution of first passage times (Section 4.4.3).

4.4.1 The RED scheme for rate-constant estimation

To more efficiently estimate rate constants from WE simulations, we have implemented the Rates from Event Durations (RED) scheme as an analysis tool called `w_red` in the WESTPA 2.0 software. The RED scheme exploits the transient

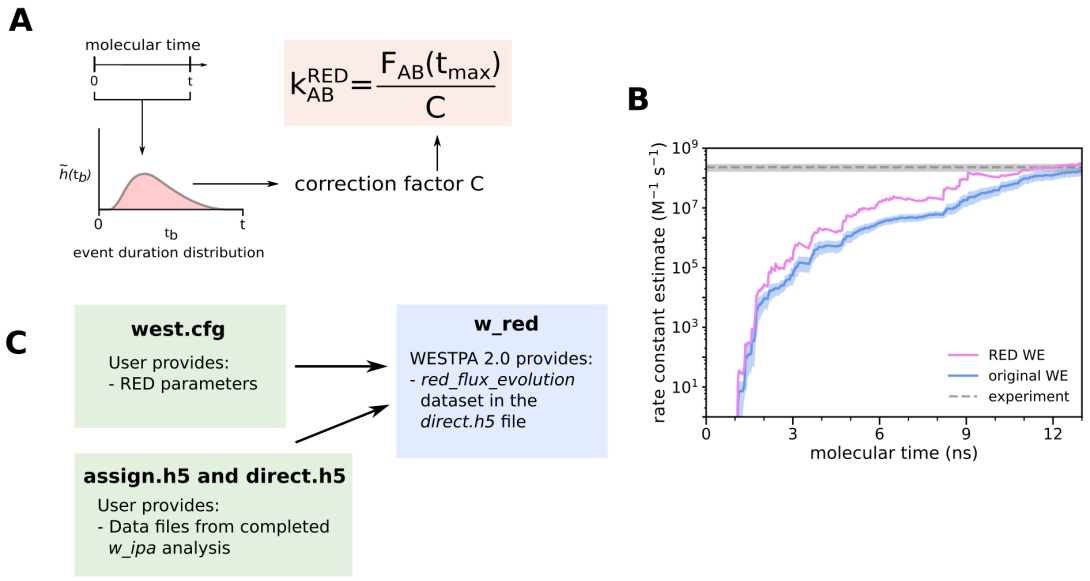


Figure 39: The RED scheme for more efficient rate-constant estimation. (A) Schematic illustrating the RED scheme, which incorporates the distribution of event durations as part of a correction factor for rate-constant estimates that accounts for statistical bias toward the observation of events with short durations. (B) Application of the original and RED schemes to estimate the associate rate constant of a protein-protein binding process involving the barnase and barstar proteins as a function of molecular time in a WE simulation. The molecular time is defined as $N\tau$, where N is the number of WE iterations and τ is the fixed time interval (20 ps) of each WE iteration. Simulations were previously run using WESTPA 1.0 with the GROMACS 4.6.7 MD engine.[197] (C) A schematic illustrating how users can generate a dataset for calculating the RED-scheme correction factor from simulation data stored in the analysis HDF5 files and apply the correction factor to the rate-constant estimate using the new `w_red` tool.

ramp-up portion of a WE simulation by incorporating the probability distribution of event durations (or "barrier crossing" times) from a WE simulation as part of a correction factor (Figure 39A).[198] The correction factor accounts for the systematic error that results from statistical bias toward the observation of events with short durations and reweights the event duration distribution accordingly. When applied to an atomistic WE simulation of a protein-protein binding process, the RED scheme is >25% more efficient than the original WE scheme[43] in estimating the association rate constant (Figure 39B).[198]

The code for estimating rate constants using the RED scheme takes as input the `assign.h5` files and `direct.h5` files generated by the `w_ipa` analysis tool. Users then specify in the analysis section of the `west.cfg` file which analysis scheme `w_red` should analyze along with the initial/final states and the number of frames per iteration. Executing `w_red` from the command line, will output the corrected flux estimates as a new dataset called red_flux_evolution to the users' existing `direct.h5` file (Figure 39C). The RED rate-constant estimates can then be accessed through the Python h5py module and plotted vs. time to assess the convergence of the estimates. To estimate uncertainties in observables calculated from a small number of trials (i.e. number of independent WE simulations), we recommend using the Bayesian Bootstrap approach.[43,199] If it is not feasible to run multiple independent simulations with a certain system due to either system size or the timescale of the process of interest, a user can apply a Monte Carlo bootstrapping approach to a single simulation's RED rate constant estimate.

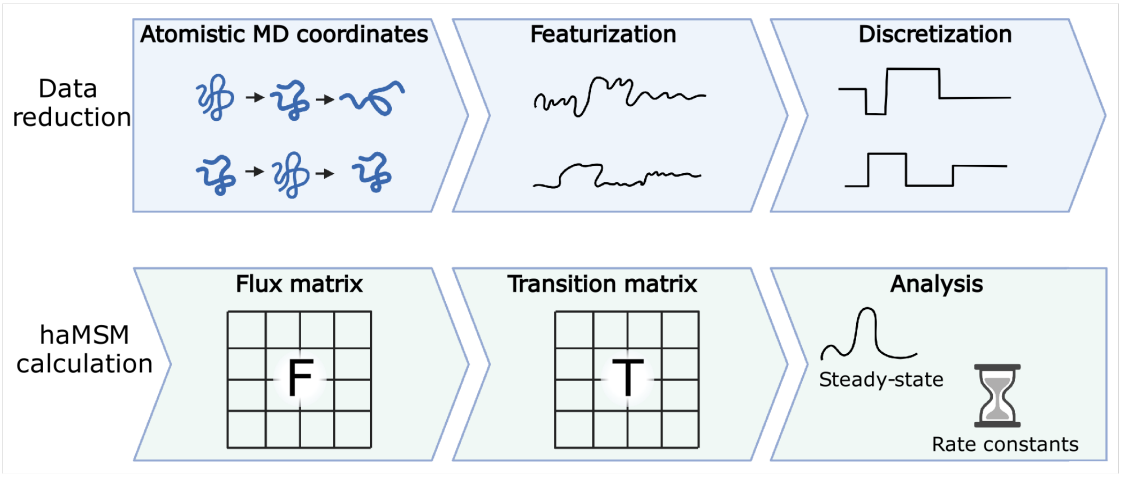


Figure 40: Workflow for constructing an haMSM from trajectories. First, the atomistic trajectories are featurized and discretized. The flux matrix is then computed by computing fluxes between discrete states. The flux matrix is row-normalized into a transition matrix. Estimates of steady-state populations and rate constants are obtained from analysis of the transition matrix.[200]

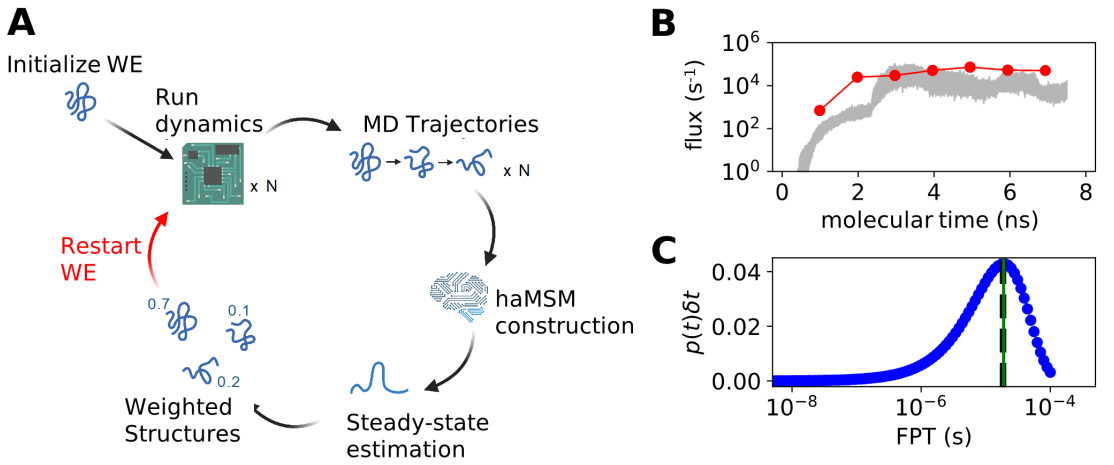


Figure 41: Application of haMSM restarting plugin to the ms folding process of the NTL9 protein. (A) Diagram of the haMSM restarting plugin's functionality. (B) Example of restarting plugin functionality in accelerated convergence of NTL9 folding rate constants from a WESTPA 2.0 simulation using the AMBER 16 MD engine.[201] haMSM estimates at restarting points are shown as dots, WE direct fluxes are shown as red lines, and a 95% credibility region from direct WE is shown in gray. (C) Distribution of first passage times for NTL9 folding from the haMSM built at the final restart of the simulation in Figure 41B. The weighted average of the blue FPT distribution is shown in black dashed, and the MFPT estimate from the haMSM's steady-state estimate is shown in green.[200]

4.4.2 A history-augmented Markov State Model (haMSM) restarting plugin

History-augmented Markov state models (haMSMs) provide a powerful tool for estimation of stationary distributions and rate constants from transient, unconverged WE data.[102] Thus, the approach has a similar motivation to the RED scheme.[198] In haMSM analysis instead of discretizing trajectories via the WE bins used by WESTPA, as in the WESS/WEED reweighting plugins,[185,186] a much finer and more numerous set of 'microbins' is employed to calculate steady-state properties with higher accuracy. These estimates, in turn, can be used to start new WE simulations from a

steady-state estimate, accelerating convergence of the simulation.[199] The new plugin provides a streamlined implementation of the restarting protocol that runs automatically as part of a WESTPA simulation, a capability which did not previously exist.

The **Markov State Models from Weighted Ensemble** (`msm_we`) package provides a set of analysis tools for using typical WESTPA HDF5 output files, augmented with atomic coordinates, to construct an haMSM. A nearly typical MSM model-building procedure[83] is used (Figure 40): WE trajectories are discretized into clusters (microbins) and transitions among microbins are analyzed. However, instead of reconstructing entire trajectories, the `msm_we` analysis computes the flux matrix by taking each weighted parent/child segment pair, extracting and discretizing one frame from each, and measuring flux between them - i.e. the weight transferred. The haMSM restarting plugin in WESTPA 2.0 makes use of the analysis tools provided by `msm_we` to incorporate this functionality directly into WESTPA. It manages running a number of independent simulations, initialized from some starting configuration, and augments their output HDF5 with the necessary atomic coordinates. Data from all independent runs are gathered and used to build a single haMSM. Stationary probability distributions and rate constants are estimated from this haMSM. This plugin can be used to start a set of new WE simulation runs, initialized closer to steady-state (Figure 41). The haMSM and the WE trajectory data are used to build a library of structures and their associated steady-state weights. These are used to initiate a new set of independent WE runs, which should start closer to steady-state and thus converge more quickly. The process can be repeated iteratively, as shown in Figure 41A. The result of this restarting procedure is shown in Figure 41B. For challenging systems, the quality of the haMSM will greatly affect the quality of the steady-state estimate. A further report is forthcoming on strategies for building high-quality haMSMs.

To use this plugin, users must specify a function that ingests coordinate data and featurizes the data. Dimensionality reduction may be performed on this featurized data. An effective choice of featurization provides a more granular structural description of the system without including a large number of irrelevant coordinates that add noise without adding useful information. For example, a limited subset of the full atoms such as only alpha-carbons, or even a strided selection of the alpha carbons, may be sufficient to capture the important structural information. Choosing a featurization based on rotation-invariant distances, such as pairwise atomic distances instead of atomic positions, can also help capture structural fluctuations without sensitivity to large-scale motion of the entire system.

To validate convergence of the restarted simulations, a number of independent replicates of the restarting protocol should be performed. These replicates should demonstrate both stability in flux estimates across restarts, and relatively constant-in-time direct fluxes within the restarts. If limited to a single replicate, agreement between the haMSM flux estimate and the direct flux should also be validated.

4.4.3 Estimating first-passage-time distributions

First passage times (FPTs) and their mean values (MFPT) are key kinetics quantities to characterize many stochastic processes (from a macrostate to another) in chemistry and biophysics such as chemical reactions, ligand binding and unbinding, protein folding, diffusion processes of small molecules within crowded environments. WE simulations, via

the Hill relation, provide unbiased estimates of the **mean first-passage time (MFPT)** directly once steady is reached[185] or indirectly via non-Markovian haMSM analysis,[186] but mathematically rigorous estimation of the FPT distribution is not available and has been a challenge for WE simulation. Suárez and coworkers, however, have shown that the FPT distributions estimated from haMSM models provide semi-quantitative agreement with unbiased reference distributions in different systems.[103] Details on building history-augmented MSMs are described above in Section 4.4.2 and more information can be found in the references.[103,186]

Here, we extend and strengthen earlier FPT distribution analysis from WE data. The original code for calculating FPT distribution was published on a separate GitHub repository (`https://github.com/ZuckermanLab/NMpathAnalysis`).[202] Recently we reorganized and refactored the code in class hierarchical structures: a base class (`MatrixFPT`) for calculating MFPTs and FPTs distribution using a general transition matrix as an input parameter, and two derived classes (`MarkovFPT` and `NonMarkovFPT`) using transition matrices from Markovian analysis and non-Markovian analysis such as haMSM in Section 4.4.2 respectively. The updated code has been merged into the `msm_we` package described in Section 4.4.2 along with some updates on building transition matrix from classic MD simulation trajectories.

The new code enables robust estimation of the FPT distribution. Figure 41C shows the non-Markovian estimation of the FPT distribution of transitions between macrostate A and B from the WE simulation of NTL9 protein folding.

4.5 Summary

WESTPA is an open-source, high-scalable, interoperable software package for applying the weighted ensemble (WE) strategy, which greatly increases the efficiency of simulating rare events (e.g., protein folding, protein binding) while maintaining rigorous kinetics. The latest WESTPA release (version 2.0) is a substantial upgrade from the original software with high-performance algorithms enabling the simulation of ever more complex systems and processes and implementing new analysis tools. WESTPA 2.0 has also been reorganized into a more standard Python package to facilitate code development of new WE algorithms, including binless strategies. With these features available in the WESTPA toolbox, the WE community is well-poised to take advantage of the latest strategies for tackling major challenges in rare-events sampling, including the identification of slow coordinates using machine learning techniques,[203,204] and the interfacing of the WE strategy with other software involving complementary rare-event sampling strategies (e.g., OpenPathSampling,58,59 SAFFIRE,60 and ScMile61) and analysis tools (e.g., LOOS,[205,206] MDAnalysis,[207,208] and PyEmma[104]). WESTPA has also been interfaced with OpenEye Scientific's Orion platform[190] on the Amazon Web Services cloud computing facility. We hope that the above new features of WESTPA will greatly facilitate efforts by the scientific community to tackle grand challenges in the simulation of rare events in a variety of fields, including the molecular sciences and systems biology.

4.6 Notes

The authors declare the following competing financial interest(s). L.T.C. is a current member of the Scientific Advisory Board of OpenEye Scientific and an Open Science Fellow with Roivant Sciences. S.Z., J.P.T., J.X., and D.N.L. are employees of OpenEye Scientific.

4.7 Acknowledgements

This work was supported by an NIH grant (R01 GM115805) to L.T.C. and D.M.Z.; NSF grants (CHE-1807301 and MCB-2112871) to L.T.C.; a MolSSI Software Fellowship to J.D.R.; and a University of Pittsburgh Andrew Mellon Graduate Fellowship to A.T.B. Computational resources were provided by the University of Pittsburgh's Center for Research Computing, by OpenEye Scientific via compute instances sourced from Amazon Web Services, and by the Advanced Computing Center at Oregon Health and Science University. We thank David Aristoff, Gideon Simpson, Forrest York, Darian Yang, and Alan Grossfield for helpful discussions.

5 Using restarting to accelerate convergence in WESTPA simulations of NTL9

ABSTRACT

Many biological processes of interest, such as protein conformational changes or ligand binding, occur too infrequently to observe with conventional molecular dynamics. The **weighted ensemble (WE)** algorithm is an efficient enhanced sampling framework that enables unbiased observation of these rare events. Although WE has been highly successful in generating pathways for complex systems, the slow relaxation timescales of complex biological systems demand long WE simulations for accurate rate constant estimation. Prior work has shown that a Markov state model built from trajectories with recycling boundary conditions, such as trajectories from steady-state WE, can produce accurate estimates of steady-state even from transient data that has not yet relaxed. In this work, we present an iterative pipeline for running WE, estimating steady-state, and initiating new WE simulations from the estimated steady-state. We apply this restarting procedure to a small protein NTL9 using both synthetic and true molecular dynamics, and show that this restarting procedure can both accelerate relaxation to steady-state and reduce variance in rate estimates.

5.1 Introduction

WE simulation is a powerful tool for enhancing sampling of rare events in **molecular dynamics (MD)** simulations and path-sampling of biomolecular processes. While WE has shown remarkable success in efficient path sampling for highly complex systems, obtaining accurate rate constant estimates remains challenging because of long, slow relaxation times.

Prior work has shown that **history-augmented Markov state models (haMSMs)** (MSMs built from trajectories with source-sink recycling boundary conditions) produce unbiased estimates of kinetic properties even from unconverged WE data.[74,99,102,137] Because steady-state recycling is typical in WE simulations for rate-constant estimation, haMSMs are therefore a natural tool for analyzing slow-converging WE data.

Using an unbiased estimate of steady-state from haMSM analysis of a WE simulation, steady-state weights can be assigned to each visited structure, and a new WE simulation can be initialized from the steady-state weighted structures. With a sufficiently high quality haMSM, the restarted simulation will begin closer to convergence, and the overall convergence timescale can be shortened.[102] Prior work has demonstrated success in using single restarts in this way to jump-start convergence in protein systems like NTL9 and Protein G.[102]

A new feature in the **Weighted Ensemble Toolkit with Parallelization and Analysis (WESTPA)** 2.0 software release is a plugin for automated haMSM restarting, described in more detail in 4.4.2. Notably, this plugin enables multiple successive restarts. If each restart brings the WE simulation closer to convergence, then multiple restarts may compound this acceleration.

In this work, we examine the effect of this repeated restarting strategy to improve WE simulations of NTL9 folding. We apply the repeated restarting pipeline using the WESTPA 2.0 restarting plugin, and simulate NTL9 under both **synthetic dynamics (SynD)** and true MD.

5.2 Methods

We ran the restarting procedure on WE simulations of both a synthetic MD NTL9 system to assess performance, and then on a "true" MD NTL9 system to validate the synthetic model result. Because the WE methodology is independent of the MD propagator, we use an identical WE setup for both the synthetic and true MD systems, where the only difference is the propagator.

5.2.1 Restarting details

In the restarting workflow, an initial set of WESTPA simulations are run consecutively. An haMSM is built from these simulations, and calculates an estimate of the steady-state distribution. This is shown in Fig. 42. A new set of WESTPA runs are launched, initialized using the set of all previously visited structures, weighted by their relative steady-state weights. Although there are very many structures here, the WE algorithm immediately prunes a majority of them to reach the target number of walkers in each bin. This process can be iteratively repeated, and is also shown in Fig. 41A.

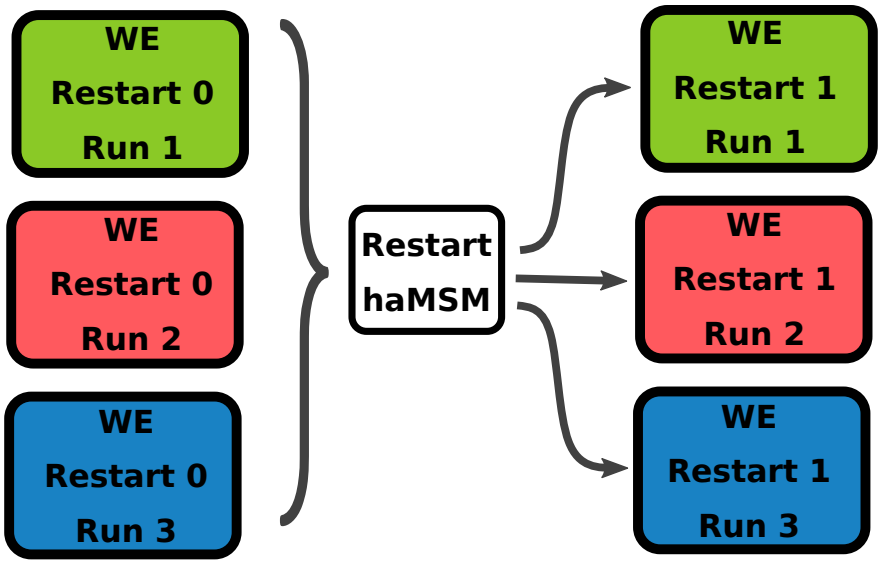


Figure 42: Diagram of haMSM restarting procedure. In this example, 3 runs are done within each restart. The haMSM is constructed using the last-half data from all 3 runs, and 3 new runs are independently launched from the haMSM steady-state estimate.

The WESTPA restarting plugin was configured to run 5 independent WE runs within each restart. Within 0.5 ns of any of the 5 runs reaching the target state, all runs were stopped and the first restart was performed, after which restarts were performed every 1 ns.

When building haMSMs for restarting, only data from the prior 2 restarts was used, to balance initial state bias with using a consistent amount of data. Of this data, only the last 0.5 ns of each WE run was used. Therefore, each restart utilized $\approx 1\mu$s of aggregate simulation data. Analysis was done similarly to the **Markov state model (MSM)** construction described above. Trajectories consisting of the first and last point of each segment in each iteration were featurized on pairwise heavy-atom distances, excluding nearest neighbors. The featurized trajectories were dimensionality reduced using VAMP, and only the first components which described 80% of the variance were preserved. These were clustered using stratified clustering, with 100 clusters per WE bin. Fluxes from the WE data were used to construct a transition matrix. The transition matrix was cleaned as described above.

When choosing initial states for the restarted WE simulation, each observed WE segment provides a structure. When restarting, structures are weighted by dividing the steady-state probability of an MSM state amongst all the structures in that state, proportional to the relative WE weights of the segments the structures were obtained from.

The WE setup was the same for the both the synthetic MD and true MD systems. For each, we performed 3 fully independent replicates of this procedure.

5.2.2 Synthetic system details

We first study this procedure applied to a synthetic model of NTL9, using SynD for fast, MD-like dynamics.[110,162]

Although NTL9 is a relatively simple protein, obtaining precise converged reference values for the target flux is very difficult. SynD provides a framework for generating approximate but complex MD-like data, which can be exactly solved to obtain references for observables. Additionally, because SynD produces data in a standard MD trajectory format, we are able to seamlessly swap it with the true MD integrator in our pipeline, with no modifications to the analysis or workflow other than a small change in the WESTPA configuration file.

SynD works by defining two components: a generative model, which can efficiently propagate trajectories in a low-dimensional space; and a backmapping, which defines a transformation from the low-dimensional space to full-coordinate MD structures. In this work, we use an MSM as the generative model, and a simple mapping of a single representative structure for each discrete MSM state.

This synthetic model was an MSM, built from 2.5 μs of NTL9 MD simulation. The MSM was featurized on pairwise heavy-atom distances, excluding the nearest neighbor. The featurized trajectories were dimensionality reduced using VAMP, preserving the first 356 components which explained 85% of the variance. The dimensionality reduced trajectories were clustered using stratified k-means, with 250 clusters per stratum (13250 total clusters), stratified across the same RMSD bins as described in Sec. 5.2.4. This transition matrix was "cleaned" to ensure connectivity, a procedure where sets of states that are disjoint from the largest active set are pruned and the data rediscretized. After cleaning, 3150 states remained.

This MSM was constructed at a 10 ps lagtime, so that one step of propagation through the transition matrix is equivalent to taking a 10 ps timestep.

Backmapping for SynD was peformed by randomly choosing a single representative structure for each haMSM state from the simulation frames assigned to it.

This MSM is the generative model for SynD, and thus is the exact, fine-grained description of the microscopic dynamics of our synthetic NTL9 system. The MSM built from the generated data attempts to capture the important features of this, with coarser states and from finite data. Having an exactly-solvable description of our dynamics enables calculation of exact reference quantities for observables.

5.2.3 MD system details

MD simulations of NTL9 were run with Amber[209,210] using a 2fs timestep. Simulations were performed at 300K in implicit solvent, with a friction coefficient set to $\gamma = 80$ ps^{-1}. These parameters are consistent with prior MD simulations of NTL9,[102] described more in Sec. 5.2.5.

5.2.4 WE details

NTL9 restarting simulations were run using WESTPA 2.0.[45] Weighted ensemble was run with a 10 ps resampling time. The RMSD to the folded state was used as the progress coordinate. WE bin boundaries were spaced to provide good resolution in the transition region with 52 active bins An initial bin boundary at and RMSD of 1.0 nm delimited the

target state, then bin boundaries were uniformly spaced at 0.1 nm increments between 1.1 nm and 4.5 nm, 0.2 nm increments between 4.6 nm and 6.4 nm, and 0.3 nm increments between 6.6 nm and the basis state boundary at 9.6 nm. We used a target of 4 walkers/bin.

5.2.5 Best-estimate "reference" WE data

As mentioned above, obtaining precise reference values of rate constants even for simple proteins like NTL9 is extremely challenging. Previously, 30 WE simulations of this system were carried out[74] for a total of 252 μs of aggregate simulation time. We take this data as our "best estimate" reference, although it is still relatively noisy.

Additionally, we bootstrap sets of trajectories from this dataset to match the amount of aggregate simulation time used in the haMSM restarting pipeline to provide a direct comparison of the performance between standard WE and the restarting procedure. Both are shown in Fig. 44.

5.2.6 MSM estimation from WE data

To estimate an MSM from WE data, we determine transitions from each WE walker's initial and final points. In contrast to the typical MSM approach of building a count matrix which enumerates the number of observed transitions between each pair of states, we instead construct a *flux* matrix $\mathbf{F}$.[102,103] The entries of the flux matrix $\mathbf{F}_{i\to j}$ encode the total weight of WE walkers which transitioned from state i to j after one WE iteration. This flux matrix is normalized to obtain a transition matrix $\mathbf{T}$. Because our WE is run with steady-state recycling boundary conditions, an MSM built from this trajectory ensemble in fact produces an haMSM.[74,99,137]

Using the transition matrix $\mathbf{T}$, we estimate the steady-state distribution Π by solving the stationarity condition

$$\Pi\mathbf{T} = \Pi. \tag{32}$$

These steady-state weights are used to assign weights to structures, as described above.

5.3 Synthetic system results

Because the dynamics of the synthetic system are exactly solvable, we can obtain an exact reference for not only the steady-state target flux value, but also an exact reference for the convergence of the flux to steady-state as the WE simulation runs.

In this system, the restarting procedure substantially accelerates convergence to steady-state compared to the reference values for standard WE, as shown in Fig. 43. The correct converged steady-state is quickly obtained by all three replicates within 15 ns of WE simulation time, compared to standard WE requiring over 60 ns of simulation time to converge.

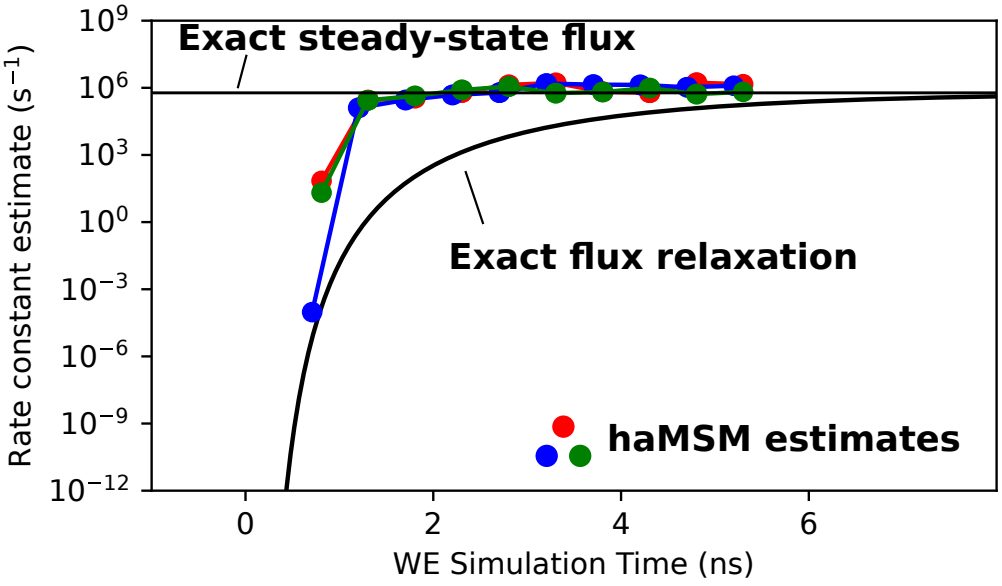


Figure 43: Flux convergence from WE simulation of the synthetic NTL9 system, with haMSM restarting. Exact relaxation is shown as the black curve, the steady-state flux as the black horizontal line, and the three replicates as the colorful lines. The exactly solved relaxation is a slow process, as described previously in Fig. 5. The restarting procedure significantly accelerates convergence of the target flux to steady-state compared to standard WE, and correctly recapitulates the reference flux.

5.4 Preliminary MD results

When applied to the true MD system shown in Fig. 44, the restarting protocol appears to overestimate the actual converged target flux. Although there are potential methods for correcting this overestimation, simulations implementing them are not yet complete. We comment on this issue further in Sec. 5.5.

Despite this overestimation, however, all three replicates produce flux estimates in remarkably consistent agreement with each other. The three replicates of the restarting procedure are fully independent, yet produce nearly identical converged flux estimates. This is in stark contrast to the confidence interval shown for flux estimates from independent WE simulations using the same amount of data, which span many orders of magnitude. This suggests that if the overestimation can be addressed, this strategy may provide major variance reduction over standard WE in addition to improved convergence.

5.5 Concluding Discussion

In this work, we extended prior work on accelerating convergence of WE simulations through restarting from haMSM estimates of steady-state, produced from unconverged data. We demonstrated a significant improvement in convergence in a complex, MD-like synthetic representation of NTL9. When applied to a true MD simulation of NTL9, we observed a substantial variance reduction, but also an overestimation in the final converged flux estimates. Although the synthetic

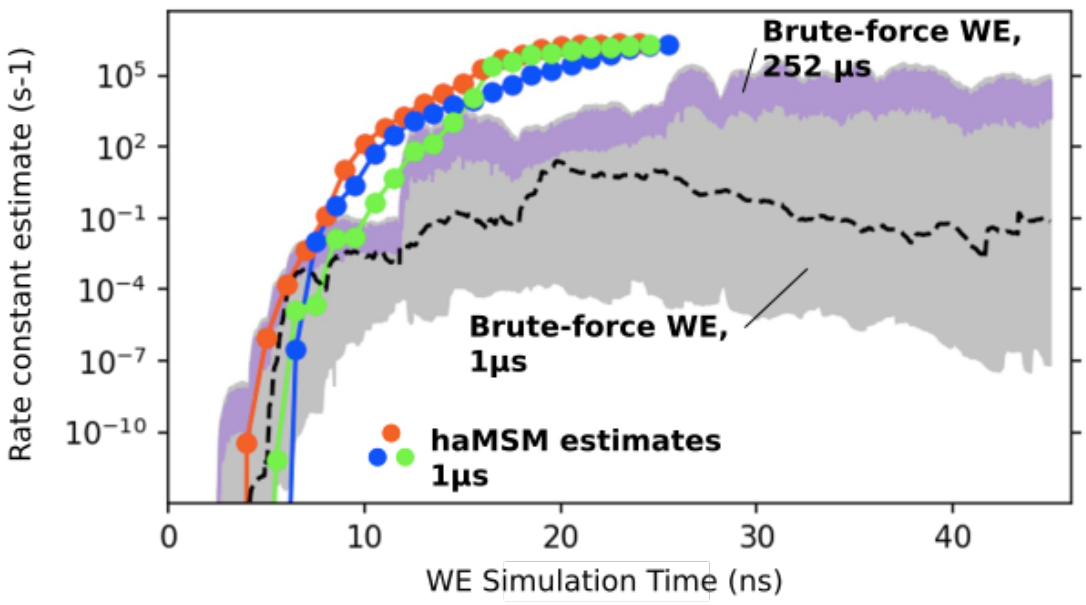


Figure 44: Flux convergence from WE simulation of the NTL9 MD system, with haMSM restarting. Shown are a 95% CI from 252 μs of aggregate simulation time over 30 long WE runs as the purple shaded region; a 95% CI computed by bootstrapping the same amount of data as was used in the restarting runs from the 30 long runs, shown as the purple region; and the median flux of the 30 long runs, shown as the black dashed line. The colorful dots indicate the target flux estimates from the haMSM at each restart. Each estimate is labeled with the amount of aggregate MD simulation data used. Although the restarted simulations quickly converge to extremely similar values, they appear to overestimate the target flux.

system showed the possibility for improvement in WE, the question remains of the practicality of attaining this with limited amounts of data typical of the complex systems where accelerated convergence would be most impactful.

To diagnose the cause of overestimation, we reproduced it in the synthetic system by using a very coarse discretization for the restart haMSM construction which produces a poor quality haMSM. A poor quality haMSM produces a poor quality steady-state estimate for the restarted distribution. Restarting from a bad steady-state estimate introduces some additional relaxation to the true steady-state.

We were also able to reproduce the overestimation in the synthetic model by restarting very frequently. If the haMSM does not yet have sufficient data to produce a reasonable steady-state estimate, then frequent restarting interrupts the relaxation to the true steady-state, and effectively periodically drives the system out of steady-state. Further simulations using longer restart intervals are currently in progress.

Together with the synthetic result, this suggests that the overestimation in the true MD system is likely a result of deficiencies in the haMSM estimation from our data, and that the procedure requires a sufficient quality haMSM to practically improve convergence without introducing bias.

A major technical limitation of our current methodology is that **Markov State Models from Weighted Ensemble** (`msm_we`) software currently only supports haMSM construction from WE data at a lagtime equal to a single WE

resampling interval. Another possible avenue for improving practical estimates of steady-state could be incorporating the reweighting discussed in Sec. 3 into the haMSM construction pipeline.

6 Conclusion

Molecular dynamics (MD) simulations are a powerful tool for studying biomolecular systems through detailed, high-resolution "movies" of their dynamics. While the level of detail MD provides makes it a unique tool, efficiently generating and accurately analyzing MD data remains challenging. The research presented in this dissertation has explored solutions to these challenges, through a range of approaches.

6.1 Synthetic Dynamics

A key challenge in developing new analysis methods is validation on systems with well-known references for measurable physical quantities. To trust that a methodology is sound, it must be able to reproduce known results. In practice though, test systems where references are precisely known and test systems with complexity similar to atomistic MD systems are often exclusive. Thus, if an analysis can precisely reproduce reference observables for a simple toy model, questions remain about scalability to realistic levels of complexity; but if an analysis can process large datasets from complex systems, assessing the quality of its estimates may be difficult.

We have already seen major benefits of a synthetic approach in our own workflows, where using **synthetic dynamics (SynD)** has shortened development timelines for some pipelines from weeks to under a day. Given the generality of SynD as an approach and the successes we have already experienced, we are excited by the myriad potential applications for it. In Chapter 2 we present the SynD workflow and a software implementation which provide extremely fast generation of approximate, but MD-like trajectory data, with exactly solvable reference quantities. Although methods for efficiently approximating dynamics have been previously developed, SynD is novel in both flexibility and ease of use. We demonstrate efficiently generating microsecond-timescale datasets exhibiting complex dynamics in seconds on a conventional laptop, which would take weeks on a supercomputer with standard MD. This new workflow enables a researcher working on a new method to rapidly test a range of parameters and validate it on a SynD model, before investing significant time and resources in scaling to a true MD system.

While we present SynD as a method for propagating trajectories using Markovian dynamics, the methodology is broadly applicable to other generative models. The software implementation of SynD is built for flexibility in this respect, and can be easily extended to use other generative models. More complex generative models could improve the quality of the dynamics, and more closely mirror true MD.

SynD also suggests some other interesting use cases. Our SynD workflow generates discrete trajectories, and converts them to atomistic trajectories by mapping discrete states to molecular structures. The backmapping we describe in our work assigns a single molecular structure to each discrete state of the SynD's generative model. A limitation of this approach is that generated output data can only ever contain structures that were used to build the SynD model. However, different approaches to backmapping could extend SynD to generate out-of-sample structures. For example, perturbations could be applied to output structures to modify atomic positions, while maintaining physical constraints.

Under our backmapping framework, the unmapped discrete trajectory and the mapping together provide a compressed representation of the atomistic trajectory. Practically, this allows a reduction in filesize that scales with the number of atoms — for a 148-atom representation of the Trp-cage miniprotein, we observed a reduction in filesize for a 208 μs trajectory from $\approx$ 750MB to $\approx$ 400KB, a nearly 2,000x reduction in filesize. The ability to use SynD as a highly efficient lossy compression algorithm suggests other possible use cases. For example, many tutorials for MD analysis toolkits currently must choose between distributing small datasets from simple systems, which may not be interesting case studies; or distributing datasets with very large filesizes for more complex systems, which may be more interesting but difficult to store and transfer. Instead, a very small SynD compressed trajectory could be distributed, with similar complexity to MD data from a large realistic system, but with orders of magnitude smaller filesizes.

Another major benefit of SynD is the ability to package the entire generative SynD model and backmapping as a small, easily distributable file. With integration of SynD propagators into tools like **Weighted Ensemble Toolkit with Parallelization and Analysis (WESTPA)**, this makes it very easy to switch between different underlying systems. A major milestone for wider SynD adoption could be the development of a central, public database, where researchers can upload their prebuilt SynD models. Other researchers could then easily validate against a large set of diverse models, with much less work than constructing a new MD simulation.

6.2 Reweighting

The difficulty of performing MD simulations which thoroughly sample the complex behavior of biomolecular systems makes accurate estimation of observables challenging. In practice, estimation of a quantity like equilibrium populations or the first-passage time is often biased by limited data.

However, we have shown that a relatively simple change to the underlying data along with iterative refinement of the model can provide high-quality, mathematically correct estimates of observables, which could not be otherwise obtained. In Chapter 3, we describe our approach for reducing bias in observables estimated from MD data. While conventional **Markov state models (MSMs)** are a popular tool for analyzing MD simulation data, fundamental limitations in their construction prevent unbiased estimation of kinetics. We show that proper treatment of boundary conditions is critical for unbiased estimates, and derive novel estimators for the **mean first-passage time (MFPT)** and committor. These provide a powerful extension to the capabilities of MSMs, and only require modification of the input trajectories to include proper boundary conditions. It is worth emphasizing again that this simple modification to the dataset, before model-building, provides a significant improvement in estimating observables.

Although these estimators are only generally unbiased in the asymptotic limit of infinite data, we show that applying an iterative reweighting procedure substantially reduces the amount of data necessary to produce unbiased estimates. With these, we show that high quality unbiased estimates can be obtained using realistically attainable amounts of data. This has potential to significantly improve the quality of MSM estimates that are possible with typical datasets, without requiring vast computational resources.

While this approach already demonstrates promising results, several steps of the workflow are still ripe for improvement. Because we focus our efforts on improvements elsewhere in the workflow, our MSM construction builds transition matrices using normalized count matrices from the trajectories. Applying maximum-likelihood or Bayesian estimation of the transition matrices could improve estimates.

Additionally, hyperparameter selection for our reweighting approach remains challenging. Our MSM construction pipeline introduces the use of fragments, short overlapping segments which can be analyzed and reweighted as independent trajectories. However, the fragment length is a new hyperparameter which is not yet well-understood, particularly in combination with the MSM lag time. Our present approach identifies an optimal choice by splitting up the initial dataset into groups, doing a hyperparameter sweep over a range of lag times and fragment lengths, and choosing the values that produce most self-consistent estimates of an observables across the groups. Future work could better explain the role of the fragment length parameter. Additionally, new hyperparameter optimization approaches could compare optimization for self-consistency of different observables to examine whether, for example, optimizing for self-consistency in equilibrium versus **nonequilibrium steady-state (NESS)** produces better results.

6.3 Weighted Ensemble and Restarting

MD simulations have a critical limitation in their ability to access long timescales at which many interesting biological processes take place. We have shown that a pipeline combining **weighted ensemble (WE)** enhanced sampling with **history-augmented Markov state model (haMSM)** estimation can address this limitation and drastically reduce the amount of simulation needed for accurate rate estimation.

Enhanced sampling methods like WE, described in Chapter. 4, address this shortcoming by more efficiently focusing computational resources. We extend this in Chapter. 5 by implementing a procedure for accelerating relaxation of initial transients, to enable accurate rate-constant estimation from shorter WE simulations. Our iterative pipeline for restarting WE simulations using estimates of steady-state from the transient shows both an improvement in variance and in relaxation times.

Using SynD, we have validated the foundation of this approach, and showed that it provides substantially faster relaxation than standard WE. This allows accurate rate constant measurements from a fraction of the data standard WE would require. Additionally, we have shown that in a true MD system, rate constant estimates from restarting provides a massive reduction in variance over estimates from independent simulations.

Challenges remain in moving beyond SynD to apply this method to true MD simulation data. In practice, we observe overestimation of the steady-state from the haMSMs. We were able to replicate this overestimation using SynD systems in two separate ways. First, we were able to introduce overestimation into SynD restarting by reducing the quality of the haMSMs used to estimate steady-state. Next, we reproduced it by shortening the restarting interval. Because steady-state estimates from haMSMs built with finite data are imperfect, there is some relaxation after the restart, which grows longer for worse steady-state estimates. Restarting too quickly prevents this relaxation, introducing a bias.

Therefore, rather than showing a fundamental issue with the restarting procedure, this highlights the difficulty of constructing high-quality haMSMs from realistic datasets.

Fortunately, our prior work on reweighting in Chapter. 3 addresses a similar limitation. Applying the iterative reweighting procedure to the haMSMs used for restarting could substantially improve their quality, and eliminate this overestimation.

Our pipeline for constructing haMSMs from WE data also faces a significant limitation, as it does not support using lag times longer than one WE resampling period. Extending the **Markov State Models from Weighted Ensemble** (`msm_we`) software to support construction of flux matrices from WE data at different lagtimes is likely to substantially improve the quality of steady-state estimates, and should also mitigate this overestimation. Additionally, a longer interval between restarts should also mitigate this bias, as it gives more time for relaxation of transients immediately after a restart.

6.4 Closing Remarks on the Role of Software in Research

The advancement of science is a collaborative endeavor that brings together theory, computation, and experiment. In this work, we have explored the critical role of theoretical and computational methods in improving our ability to study biomolecular systems. We have showed results which demonstrate significant improvements in practical analysis of realistic datasets.

Effective computational research, however, must reach beyond derivation of new methodologies and algorithms. Embracing the collaborative nature of science means facilitating adoption and usage of new methodologies. Democratization of computational tools, and accessibility to researchers outside of domain experts can lead to transformative, novel applications. The recent advances in AI tools, for example, led to the development of AlphaFold, a groundbreaking tool for protein structure prediction. In this spirit, it is not enough to design methods; we must also facilitate application and adoption.

In this work, we have not only developed innovative techniques, but we have maintained all throughout a focus on building robust, accessible tools around these methods. We have ensured that our contributions can be easily integrated into workflows of other researchers throughout the field, working on different problems. All the work developed in this dissertation is accompanied by a user-friendly software library, thorough documentation, and practical examples. It is our hope that our emphasis on documentation, tutorials, and open-source distribution of our software reflects our commitment to a collaborative scientific community.

We sincerely hope that the methods and tools presented in this dissertation will be valuable resources for advancing our understanding of biology. Contributing to the progress of this exciting and dynamic field has been an immense privilege, and we look forward with anticipation to seeing the continued broader impact of this work.

References

[1] M. S. Smyth and J. H. J. Martin, "x Ray crystallography," *Molecular Pathology*, vol. 53, no. 1, p. 8, 2000, ISSN: 1366-8714. DOI: 10.1136/mp.53.1.8.

[2] D. Marion, "An Introduction to Biological NMR Spectroscopy," *Molecular & Cellular Proteomics*, vol. 12, no. 11, pp. 3006–3025, 2013, ISSN: 1535-9476. DOI: 10.1074/mcp.o113.030239.

[3] J.-P. Renaud, A. Chari, C. Ciferri, W.-t. Liu, H.-W. Rémigy, H. Stark, and C. Wiesmann, "Cryo-EM in drug discovery: achievements, limitations and prospects," *Nature Reviews Drug Discovery*, vol. 17, no. 7, pp. 471–492, 2018, ISSN: 1474-1776. DOI: 10.1038/nrd.2018.77.

[4] D. Lyumkis, "Challenges and opportunities in cryo-EM single-particle analysis," *Journal of Biological Chemistry*, vol. 294, no. 13, pp. 5181–5197, 2019, ISSN: 0021-9258. DOI: 10.1074/jbc.rev118.005602.

[5] R. B. Sekar and A. Periasamy, "Fluorescence resonance energy transfer (FRET) microscopy imaging of live cell protein localizations," *The Journal of Cell Biology*, vol. 160, no. 5, pp. 629–633, 2003, ISSN: 0021-9525. DOI: 10.1083/jcb.200210140.

[6] W. R. Algar, N. Hildebrandt, S. S. Vogel, and I. L. Medintz, "FRET as a biomolecular research tool — understanding its potential while avoiding pitfalls," *Nature Methods*, vol. 16, no. 9, pp. 815–829, 2019, ISSN: 1548-7091. DOI: 10.1038/s41592-019-0530-8.

[7] E. Lerner, A. Barth, J. Hendrix, B. Ambrose, V. Birkedal, S. C. Blanchard, R. Börner, H. S. Chung, T. Cordes, T. D. Craggs, A. A. Deniz, J. Diao, J. Fei, R. L. Gonzalez, I. V. Gopich, T. Ha, C. A. Hanke, G. Haran, N. S. Hatzakis, S. Hohng, S.-C. Hong, T. Hugel, A. Ingargiola, C. Joo, A. N. Kapanidis, H. D. Kim, T. Laurence, N. K. Lee, T.-H. Lee, E. A. Lemke, E. Margeat, J. Michaelis, X. Michalet, S. Myong, D. Nettels, T.-O. Peulen, E. Ploetz, Y. Razvag, N. C. Robb, B. Schuler, H. Soleimaninejad, C. Tang, R. Vafabakhsh, D. C. Lamb, C. A. Seidel, and S. Weiss, "FRET-based dynamic structural biology: Challenges, perspectives and an appeal for open-science practices," *eLife*, vol. 10, e60416, 2021. DOI: 10.7554/elife.60416. eprint: 2006.03091.

[8] C. Zhu, G. He, Q. Yin, L. Zeng, X. Ye, Y. Shi, and W. Xu, "Molecular biology of the SARs-CoV-2 spike protein: A review of current knowledge," *Journal of Medical Virology*, vol. 93, no. 10, pp. 5729–5741, 2021, ISSN: 0146-6615. DOI: 10.1002/jmv.27132.

[9] Y. Cai, J. Zhang, T. Xiao, H. Peng, S. M. Sterling, R. M. WalshJr., S. Rawson, S. Rits-Volloch, and B. Chen, "Distinct conformational states of SARS-CoV-2 spike protein," *Science*, vol. 369, no. 6511, pp. 1586–1592, 2020, ISSN: 0036-8075. DOI: 10.1126/science.abd4251.

[10] A. Dommer, L. Casalino, F. Kearns, M. Rosenfeld, N. Wauer, S.-H. Ahn, J. Russo, S. Oliveira, C. Morris, A. Bogetti, A. Trifan, A. Brace, T. Sztain, A. Clyde, H. Ma, C. Chennubhotla, H. Lee, M. Turilli, S. Khalid, T. Tamayo-Mendoza, M. Welborn, A. Christensen, D. G. Smith, Z. Qiao, S. K. Sirumalla, M. O'Connor, F. Manby, A. Anandkumar, D. Hardy, J. Phillips, A. Stern, J. Romero, D. Clark, M. Dorrell, T. Maiden,

L. Huang, J. McCalpin, C. Woods, A. Gray, M. Williams, B. Barker, H. Rajapaksha, R. Pitts, T. Gibbs, J. Stone, D. M. Zuckerman, A. J. Mulholland, T. Miller, S. Jha, A. Ramanathan, L. Chong, and R. E. Amaro, "#COVIDisAirborne: AI-enabled multiscale computational microscopy of delta SARS-CoV-2 in a respiratory aerosol," *The International Journal of High Performance Computing Applications*, vol. 37, no. 1, pp. 28–44, 2023, ISSN: 1094-3420. DOI: 10.1177/10943420221128233.

[11] T. Sztain, S.-H. Ahn, A. T. Bogetti, L. Casalino, J. A. Goldsmith, E. Seitz, R. S. McCool, F. L. Kearns, F. Acosta-Reyes, S. Maji, G. Mashayekhi, J. A. McCammon, A. Ourmazd, J. Frank, J. S. McLellan, L. T. Chong, and R. E. Amaro, "A glycan gate controls opening of the SARS-CoV-2 spike protein," *Nature Chemistry*, vol. 13, no. 10, pp. 963–968, 2021, ISSN: 1755-4330. DOI: 10.1038/s41557-021-00758-3.

[12] R. O. Dror, R. M. Dirks, J. Grossman, H. Xu, and D. E. Shaw, "Biomolecular Simulation: A Computational Microscope for Molecular Biology," *Annual Review of Biophysics*, vol. 41, no. 1, pp. 429–452, 2012, ISSN: 1936-122X. DOI: 10.1146/annurev-biophys-042910-155245.

[13] S. A. Adcock and J. A. McCammon, "Molecular Dynamics: Survey of Methods for Simulating the Activity of Proteins," *Chemical Reviews*, vol. 106, no. 5, pp. 1589–1615, 2006, ISSN: 0009-2665. DOI: 10.1021/cr040426m.

[14] S. A. Hollingsworth and R. O. Dror, "Molecular Dynamics Simulation for All," *Neuron*, vol. 99, no. 6, pp. 1129–1143, 2018, ISSN: 0896-6273. DOI: 10.1016/j.neuron.2018.08.011.

[15] C. Tian, K. Kasavajhala, K. A. A. Belfon, L. Raguette, H. Huang, A. N. Migues, J. Bickel, Y. Wang, J. Pincay, Q. Wu, and C. Simmerling, "ff19SB: Amino-Acid-Specific Protein Backbone Parameters Trained against Quantum Mechanics Energy Surfaces in Solution," *Journal of Chemical Theory and Computation*, vol. 16, no. 1, pp. 528–552, 2019, ISSN: 1549-9618. DOI: 10.1021/acs.jctc.9b00591.

[16] K. Vanommeslaeghe, E. Hatcher, C. Acharya, S. Kundu, S. Zhong, J. Shim, E. Darian, O. Guvench, P. Lopes, I. Vorobyov, and A. D. Mackerell, "CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields," *Journal of Computational Chemistry*, vol. 31, no. 4, pp. 671–690, 2010, ISSN: 0192-8651. DOI: 10.1002/jcc.21367.

[17] R. W. Pastor and A. D. MacKerell, "Development of the CHARMM Force Field for Lipids," *The Journal of Physical Chemistry Letters*, vol. 2, no. 13, pp. 1526–1532, 2011, ISSN: 1948-7185. DOI: 10.1021/jz200167q.

[18] A. D. MacKerell, N. Banavali, and N. Foloppe, "Development and current status of the CHARMM force field for nucleic acids," *Biopolymers*, vol. 56, no. 4, pp. 257–265, 2010, ISSN: 0006-3525. DOI: 10.1002/1097-0282(2000)56:4<257::aid-bip10029>3.0.co;2-w.

[19] C. Oostenbrink, A. Villa, A. E. Mark, and W. F. V. Gunsteren, "A biomolecular force field based on the free enthalpy of hydration and solvation: The GROMOS force-field parameter sets 53A5 and 53A6," *Journal of Computational Chemistry*, vol. 25, no. 13, pp. 1656–1676, 2004, ISSN: 0192-8651. DOI: 10.1002/jcc.20090.

[20] Y. Qiu, D. G. A. Smith, S. Boothroyd, H. Jang, D. F. Hahn, J. Wagner, C. C. Bannan, T. Gokey, V. T. Lim, C. D. Stern, A. Rizzi, B. Tjanaka, G. Tresadern, X. Lucas, M. R. Shirts, M. K. Gilson, J. D. Chodera, C. I. Bayly, D. L. Mobley, and L.-P. Wang, "Development and Benchmarking of Open Force Field v2.0.0: the Parsley Small-Molecule Force Field," *Journal of Chemical Theory and Computation*, vol. 17, no. 10, pp. 6262–6280, 2021, ISSN: 1549-9618. DOI: 10.1021/acs.jctc.1c00571.

[21] P. C. T. Souza, R. Alessandri, J. Barnoud, S. Thallmair, I. Faustino, F. Grünewald, I. Patmanidis, H. Abdizadeh, B. M. H. Bruininks, T. A. Wassenaar, P. C. Kroon, J. Melcr, V. Nieto, V. Corradi, H. M. Khan, J. Domański, M. Javanainen, H. Martinez-Seara, N. Reuter, R. B. Best, I. Vattulainen, L. Monticelli, X. Periole, D. P. Tieleman, A. H. d. Vries, and S. J. Marrink, "Martini 3: a general purpose force field for coarse-grained molecular dynamics," *Nature Methods*, vol. 18, no. 4, pp. 382–388, 2021, ISSN: 1548-7091. DOI: 10.1038/s41592-021-01098-3.

[22] C. Arnarez, J. J. Uusitalo, M. F. Masman, H. I. Ingolfsson, D. H. d. Jong, M. N. Melo, X. Periole, A. H. d. Vries, and S. J. Marrink, "Dry Martini, a Coarse-Grained Force Field for Lipid Membrane Simulations with Implicit Solvent," *Journal of Chemical Theory and Computation*, vol. 11, no. 1, pp. 260–275, 2014, ISSN: 1549-9618. DOI: 10.1021/ct500477k.

[23] L. Darré, M. R. Machado, A. F. Brandner, H. C. González, S. Ferreira, and S. Pantano, "SIRAH: A Structurally Unbiased Coarse-Grained Force Field for Proteins with Aqueous Solvation and Long-Range Electrostatics," *Journal of Chemical Theory and Computation*, vol. 11, no. 2, pp. 723–739, 2015, ISSN: 1549-9618. DOI: 10.1021/ct5007746.

[24] J. A. McCammon, B. R. Gelin, and M. Karplus, "Dynamics of folded proteins," *Nature*, vol. 267, no. 5612, pp. 585–590, Jun. 1977. DOI: 10.1038/267585a0. [Online]. Available: https://doi.org/10.1038%2F267585a0.

[25] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw, "How Fast-Folding Proteins Fold," *Science*, vol. 334, no. 6055, pp. 517–520, 2011, ISSN: 0036-8075. DOI: 10.1126/science.1208351.

[26] M. W. Sung, Z. Yang, C. M. Driggers, B. L. Patton, B. Mostofian, J. D. Russo, D. M. Zuckerman, and S.-L. Shyng, "Vascular KATP channel structural dynamics reveal regulatory mechanism by Mg-nucleotides," *Proceedings of the National Academy of Sciences*, vol. 118, no. 44, e2109441118, 2021, ISSN: 0027-8424. DOI: 10.1073/pnas.2109441118.

[27] M. W. Sung, C. M. Driggers, B. Mostofian, J. D. Russo, B. L. Patton, D. M. Zuckerman, and S.-L. Shyng, "Ligand-mediated Structural Dynamics of a Mammalian Pancreatic KATP Channel," *Journal of Molecular Biology*, vol. 434, no. 19, p. 167 789, 2022, ISSN: 0022-2836. DOI: 10.1016/j.jmb.2022.167789.

[28] C. G. Nichols, "KATP channels as molecular sensors of cellular metabolism," *Nature*, vol. 440, no. 7083, pp. 470–476, 2006, ISSN: 0028-0836. DOI: 10.1038/nature04711.

[29] G. M. Martin, M. W. Sung, Z. Yang, L. M. Innes, B. Kandasamy, L. L. David, C. Yoshioka, and S.-L. Shyng, "Mechanism of pharmacochaperoning in a mammalian KATP channel revealed by cryo-EM," *eLife*, vol. 8, e46417, 2019. DOI: 10.7554/elife.46417.

[30] J. M. Sullivan and P. Shukla, "Time-Resolved Rhodopsin Activation Currents in a Unicellular Expression System," *Biophysical Journal*, vol. 77, no. 3, pp. 1333–1357, 1999, ISSN: 0006-3495. DOI: 10.1016/s0006-3495(99)76983-3.

[31] S. Lu, X. He, Z. Yang, Z. Chai, S. Zhou, J. Wang, A. U. Rehman, D. Ni, J. Pu, J. Sun, and J. Zhang, "Activation pathway of a G protein-coupled receptor uncovers conformational intermediates as targets for allosteric drug design," *Nature Communications*, vol. 12, no. 1, p. 4721, 2021. DOI: 10.1038/s41467-021-25020-9.

[32] M. C. Zwier and L. T. Chong, "Reaching biological timescales with all-atom molecular dynamics simulations," *Current Opinion in Pharmacology*, vol. 10, no. 6, pp. 745–752, 2010, ISSN: 1471-4892. DOI: 10.1016/j.coph.2010.09.008.

[33] L. Casalino, Z. Gaieb, J. A. Goldsmith, C. K. Hjorth, A. C. Dommer, A. M. Harbison, C. A. Fogarty, E. P. Barros, B. C. Taylor, J. S. McLellan, E. Fadda, and R. E. Amaro, "Beyond shielding: The roles of glycans in the SARS-CoV-2 spike protein," *ACS Cent. Sci. Central Science*, vol. 6, no. 10, pp. 1722–1734, Sep. 2020. DOI: 10.1021/acscentsci.0c01056. [Online]. Available: https://doi.org/10.1021%2Facscentsci.0c01056.

[34] D. E. Shaw, J. Grossman, J. A. Bank, B. Batson, J. A. Butts, J. C. Chao, M. M. Deneroff, R. O. Dror, A. Even, C. H. Fenton, A. Forte, J. Gagliardo, G. Gill, B. Greskamp, C. R. Ho, D. J. Ierardi, L. Iserovich, J. S. Kuskin, R. H. Larson, T. Layman, L.-S. Lee, A. K. Lerer, C. Li, D. Killebrew, K. M. Mackenzie, S. Y.-H. Mok, M. A. Moraes, R. Mueller, L. J. Nociolo, J. L. Peticolas, T. Quan, D. Ramot, J. K. Salmon, D. P. Scarpazza, U. B. Schafer, N. Siddique, C. W. Snyder, J. Spengler, P. T. P. Tang, M. Theobald, H. Toma, B. Towles, B. Vitale, S. C. Wang, and C. Young, "Anton 2: Raising the Bar for Performance and Programmability in a Special-Purpose Molecular Dynamics Supercomputer," *SC14: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 41–53, 2014. DOI: 10.1109/sc.2014.9.

[35] A. Onufriev, "Chapter 7 Implicit Solvent Models in Molecular Dynamics Simulations: A Brief Overview," *Annual Reports in Computational Chemistry*, vol. 4, pp. 125–137, 2008, ISSN: 1574-1400. DOI: 10.1016/s1574-1400(08)00007-8.

[36] N. Singh and W. Li, "Recent Advances in Coarse-Grained Models for Biomolecules and Their Applications," *International Journal of Molecular Sciences*, vol. 20, no. 15, p. 3774, 2019. DOI: 10.3390/ijms20153774.

[37] S. J. Marrink, H. J. Risselada, S. Yefimov, D. P. Tieleman, and A. H. d. Vries, "The MARTINI Force Field: Coarse Grained Model for Biomolecular Simulations," *The Journal of Physical Chemistry B*, vol. 111, no. 27, pp. 7812–7824, 2007, ISSN: 1520-6106. DOI: 10.1021/jp071097f.

[38] D. H. d. Jong, G. Singh, W. F. D. Bennett, C. Arnarez, T. A. Wassenaar, L. V. Schafer, X. Periole, D. P. Tieleman, and S. J. Marrink, "Improved Parameters for the Martini Coarse-Grained Protein Force Field," *Journal of Chemical Theory and Computation*, vol. 9, no. 1, pp. 687–697, 2013, ISSN: 1549-9618. DOI: 10.1021/ct300646g.

[39] L. Monticelli, S. K. Kandasamy, X. Periole, R. G. Larson, D. P. Tieleman, and S.-J. Marrink, "The MARTINI Coarse-Grained Force Field: Extension to Proteins," *Journal of Chemical Theory and Computation*, vol. 4, no. 5, pp. 819–834, 2008, ISSN: 1549-9618. DOI: 10.1021/ct700324x.

[40] A. Liwo, M. Khalili, C. Czaplewski, S. Kalinowski, S. Ołdziej, K. Wachucik, and H. A. Scheraga, "Modification and Optimization of the United-Residue (UNRES) Potential Energy Function for Canonical Simulations. I. Temperature Dependence of the Effective Energy Function and Tests of the Optimization Method with Single Training Proteins," *The Journal of Physical Chemistry B*, vol. 111, no. 1, pp. 260–285, 2007, ISSN: 1520-6106. DOI: 10.1021/jp065380a.

[41] Y. C. Kim and G. Hummer, "Coarse-grained Models for Simulations of Multiprotein Complexes: Application to Ubiquitin Binding," *Journal of Molecular Biology*, vol. 375, no. 5, pp. 1416–1433, 2008, ISSN: 0022-2836. DOI: 10.1016/j.jmb.2007.11.063.

[42] S. J. Marrink and D. P. Tieleman, *Perspective on the Martini model*, 2013. [Online]. Available: https://pubs.rsc.org/en/content/articlepdf/2013/cs/c3cs60093a (visited on 04/21/2023).

[43] G. Huber and S. Kim, "Weighted-ensemble Brownian dynamics simulations for protein association reactions," *Biophysical Journal*, vol. 70, no. 1, pp. 97–110, 1996, ISSN: 0006-3495. DOI: 10.1016/s0006-3495(96)79552-8.

[44] D. M. Zuckerman and L. T. Chong, "Weighted Ensemble Simulation: Review of Methodology, Applications, and Software," *Annual Review of Biophysics*, vol. 46, no. 1, pp. 1–15, 2016, ISSN: 1936-122X. DOI: 10.1146/annurev-biophys-070816-033834.

[45] J. D. Russo, S. Zhang, J. M. G. Leung, A. T. Bogetti, J. P. Thompson, A. J. DeGrave, P. A. Torrillo, A. J. Pratt, K. F. Wong, J. Xia, J. Copperman, J. L. Adelman, M. C. Zwier, D. N. LeBard, D. M. Zuckerman, and L. T. Chong, "WESTPA 2.0: High-Performance Upgrades for Weighted Ensemble Simulations and Analysis of Longer-Timescale Applications," *Journal of Chemical Theory and Computation*, 2021, ISSN: 1549-9618. DOI: 10.1021/acs.jctc.1c01154.

[46] Y. Sugita and Y. Okamoto, "Replica-exchange molecular dynamics method for protein folding," *Chemical Physics Letters*, vol. 314, no. 1-2, pp. 141–151, 1999, ISSN: 0009-2614. DOI: 10.1016/s0009-2614(99)01123-9.

[47] A. Barducci, M. Bonomi, and M. Parrinello, "Metadynamics," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 1, no. 5, pp. 826–843, 2011, ISSN: 1759-0876. DOI: 10.1002/wcms.31.

[48] E. Vanden-Eijnden and M. Venturoli, "Markovian milestoning with Voronoi tessellations," *The Journal of Chemical Physics*, vol. 130, no. 19, p. 194 101, 2009, ISSN: 0021-9606. DOI: 10.1063/1.3129843.

[49] E. Hruska, J. R. Abella, F. Nüske, L. E. Kavraki, and C. Clementi, "Quantitative comparison of adaptive sampling methods for protein dynamics," *The Journal of Chemical Physics*, vol. 149, no. 24, p. 244 119, 2018, ISSN: 0021-9606. DOI: 10.1063/1.5053582.

[50] E. Hruska, V. Balasubramanian, H. Lee, S. Jha, and C. Clementi, "Extensible and Scalable Adaptive Sampling on Supercomputers," *Journal of Chemical Theory and Computation*, vol. 16, no. 12, pp. 7915–7925, 2020, ISSN: 1549-9618. DOI: 10.1021/acs.jctc.0c00991.

[51] C. Abrams and G. Bussi, "Enhanced Sampling in Molecular Dynamics Using Metadynamics, Replica-Exchange, and Temperature-Acceleration," *Entropy*, vol. 16, no. 1, pp. 163–199, 2013. DOI: 10.3390/e16010163. eprint: 1401.0387.

[52] J. Hénin, T. Lelièvre, M. R. Shirts, O. Valsson, and L. Delemotte, "Enhanced Sampling Methods for Molecular Dynamics Simulations [Article v1.0]," *Living Journal of Computational Molecular Science*, vol. 4, no. 1, 2022. DOI: 10.33011/livecoms.4.1.1583. eprint: 2202.04164.

[53] Y. Miao, V. A. Feher, and J. A. McCammon, "Gaussian Accelerated Molecular Dynamics: Unconstrained Enhanced Sampling and Free Energy Calculation," *Journal of Chemical Theory and Computation*, vol. 11, no. 8, pp. 3584–3595, 2015, ISSN: 1549-9618. DOI: 10.1021/acs.jctc.5b00436.

[54] R. Elber, J. M. Bello-Rivas, P. Ma, A. E. Cardenas, and A. Fathizadeh, "Calculating Iso-Committor Surfaces as Optimal Reaction Coordinates with Milestoning," *Entropy*, vol. 19, no. 5, p. 219, 2017, ISSN: 1099-4300. DOI: 10.3390/e19050219.

[55] A. K. Faradjian and R. Elber, "Computing time scales from reaction coordinates by milestoning," *The Journal of Chemical Physics*, vol. 120, no. 23, pp. 10 880–10 889, 2004, ISSN: 0021-9606. DOI: 10.1063/1.1738640.

[56] J. M. Bello-Rivas and R. Elber, "Exact milestoning," *The Journal of Chemical Physics*, vol. 142, no. 9, 03B602_1, 2015.

[57] C. Schütte, F. Noé, J. Lu, M. Sarich, and E. Vanden-Eijnden, "Markov state models based on milestoning," *The Journal of Chemical Physics*, vol. 134, no. 20, p. 204 105, 2011, ISSN: 0021-9606. DOI: 10.1063/1.3590108.

[58] D. Ray, S. E. Stone, and I. Andricioaei, "Markovian weighted ensemble milestoning (m-WEM): Long-time kinetics from short trajectories," *J. Chem. Theory Comput.*, vol. 18, no. 1, pp. 79–95, Dec. 2021. DOI: 10.1021/acs.jctc.1c00803. [Online]. Available: https://doi.org/10.1021%2Facs.jctc.1c00803.

[59] D. Moroni, T. S. v. Erp, and P. G. Bolhuis, "Investigating rare events by transition interface sampling," *Physica A: Statistical Mechanics and its Applications*, vol. 340, no. 1-3, pp. 395–401, 2004, ISSN: 0378-4371. DOI: 10.1016/j.physa.2004.04.033. eprint: cond-mat/0311571.

[60] T. S. v. Erp and P. G. Bolhuis, "Elaborating transition interface sampling methods," *Journal of Computational Physics*, vol. 205, no. 1, pp. 157–181, 2005, ISSN: 0021-9991. DOI: 10.1016/j.jcp.2004.11.003. eprint: cond-mat/0405116.

[61] D. W. H. Swenson and P. G. Bolhuis, "A replica exchange transition interface sampling method with multiple interface sets for investigating networks of rare events," *The Journal of Chemical Physics*, vol. 141, no. 4, p. 044 101, Jul. 2014. DOI: 10.1063/1.4890037. [Online]. Available: https://doi.org/10.1063%2F1.4890037.

[62] A. Warmflash, P. Bhimalapuram, and A. R. Dinner, "Umbrella sampling for nonequilibrium processes," *The Journal of chemical physics*, vol. 127, no. 15, p. 114 109, 2007.

[63] A. Dickson, A. Warmflash, and A. R. Dinner, "Nonequilibrium umbrella sampling in spaces of many order parameters," *The Journal of chemical physics*, vol. 130, no. 7, 02B605, 2009.

[64] G. Torrie and J. Valleau, "Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling," *Journal of Computational Physics*, vol. 23, no. 2, pp. 187–199, 1977, ISSN: 0021-9991. DOI: 10.1016/0021-9991(77)90121-8.

[65] P. G. Bolhuis, C. Dellago, and D. Chandler, "Sampling ensembles of deterministic transition pathways," *Faraday Discussions*, vol. 110, no. 0, pp. 421–436, 1998, ISSN: 1359-6640. DOI: 10.1039/a801266k.

[66] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, "TRANSITION PATH SAMPLING: Throwing Ropes Over Rough Mountain Passes, in the Dark," *Annual Review of Physical Chemistry*, vol. 53, no. 1, pp. 291–318, 2002, ISSN: 0066-426X. DOI: 10.1146/annurev.physchem.53.082301.113146.

[67] C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler, "Transition path sampling and the calculation of rate constants," *The Journal of Chemical Physics*, vol. 108, no. 5, pp. 1964–1977, 1998, ISSN: 0021-9606. DOI: 10.1063/1.475562.

[68] L. R. Pratt, "A statistical method for identifying transition states in high dimensional problems," *The Journal of Chemical Physics*, vol. 85, no. 9, pp. 5045–5048, 1986, ISSN: 0021-9606. DOI: 10.1063/1.451695.

[69] R. J. Allen, D. Frenkel, and P. R. t. Wolde, "Simulating rare events in equilibrium or nonequilibrium stochastic systems," *The Journal of Chemical Physics*, vol. 124, no. 2, p. 024 102, 2006, ISSN: 0021-9606. DOI: 10.1063/1.2140273. eprint: cond-mat/0509499.

[70] V. Thapar and F. A. Escobedo, "Simultaneous estimation of free energies and rates using forward flux sampling and mean first passage times," *The Journal of Chemical Physics*, vol. 143, no. 24, p. 244 113, 2015, ISSN: 0021-9606. DOI: 10.1063/1.4938248.

[71] C. Dellago and P. G. Bolhuis, "Advanced Computer Simulation Approaches for Soft Matter Sciences III," pp. 167–233, 2009. DOI: 10.1007/978-3-540-87706-6_3.

[72] L. T. Chong, A. S. Saglam, and D. M. Zuckerman, "Path-sampling strategies for simulating rare events in biomolecular systems," *Current Opinion in Structural Biology*, vol. 43, pp. 88–94, 2017, ISSN: 0959-440X. DOI: 10.1016/j.sbi.2016.11.019.

[73] D. Aristoff, J. Copperman, G. Simpson, R. J. Webber, and D. M. Zuckerman, "Weighted ensemble: Recent mathematical developments," *arXiv*, 2022. DOI: 10.48550/arxiv.2206.14943. eprint: 2206.14943.

[74] U. Adhikari, B. Mostofian, J. Copperman, S. R. Subramanian, A. A. Petersen, and D. M. Zuckerman, "Computational Estimation of Microsecond to Second Atomistic Folding Times," *Journal of the American Chemical Society*, vol. 141, no. 16, pp. 6519–6526, 2019, ISSN: 0002-7863. DOI: 10.1021/jacs.8b10735.

[75] A. Nunes-Alves, D. M. Zuckerman, and G. M. Arantes, "Escape of a Small Molecule from Inside T4 Lysozyme by Multiple Pathways," *Biophysical Journal*, vol. 114, no. 5, pp. 1058–1066, 2018, ISSN: 0006-3495. DOI: 10.1016/j.bpj.2018.01.014.

[76] S. Zhang, J. P. Thompson, J. Xia, A. T. Bogetti, F. York, A. G. Skillman, L. T. Chong, and D. N. LeBard, "Mechanistic Insights into Passive Membrane Permeability of Drug-like Molecules from a Weighted Ensemble of Trajectories," *Journal of Chemical Information and Modeling*, vol. 62, no. 8, pp. 1891–1904, 2022, ISSN: 1549-9596. DOI: 10.1021/acs.jcim.1c01540.

[77] M. C. Zwier, A. J. Pratt, J. L. Adelman, J. W. Kaus, D. M. Zuckerman, and L. T. Chong, "Efficient atomistic simulation of pathways and calculation of rate constants for a protein–peptide binding process: Application to the MDM2 protein and an intrinsically disordered p53 peptide," *J. Phys. Chem. Lett.*, vol. 7, no. 17, pp. 3440–3445, Aug. 2016. DOI: 10.1021/acs.jpclett.6b01502. [Online]. Available: https://doi.org/10.1021%2Facs.jpclett.6b01502.

[78] C. M. S. OpenEye, *Openeye Orion*, http://www.eyesopen.com, Santa Fe, NM, 2022.

[79] J. D. Russo, *Jdrusso/msm_we*, version 0.1.27, Mar. 2023. DOI: 10.5281/zenodo.7786837. [Online]. Available: https://doi.org/10.5281/zenodo.7786837.

[80] A. T. Bogetti, B. Mostofian, A. Dickson, A. Pratt, A. S. Saglam, P. O. Harrison, J. L. Adelman, M. Dudek, P. A. Torrillo, A. J. DeGrave, U. Adhikari, M. C. Zwier, D. M. Zuckerman, and L. T. Chong, "A Suite of Tutorials for the WESTPA Rare-Events Sampling Software [Article v1.0]," *Living Journal of Computational Molecular Science*, vol. 1, no. 2, 2019. DOI: 10.33011/livecoms.1.2.10607.

[81] A. T. Bogetti, J. M. G. Leung, J. D. Russo, S. Zhang, J. P. Thompson, A. S. Saglam, D. Ray, R. C. Abraham, J. R. Faeder, I. Andricioaei, J. L. Adelman, M. C. Zwier, D. N. LeBard, D. M. Zuckerman, and L. T. Chong, "A Suite of Advanced Tutorials for the WESTPA 2.0 Rare-Events Sampling Software [Article v2.0]," *Living Journal of Computational Molecular Science*, vol. 5, no. 1, 2022. DOI: 10.33011/livecoms.5.1.1655.

[82] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé, "Markov models of molecular kinetics: Generation and validation," *The Journal of Chemical Physics*, vol. 134, no. 17, p. 174 105, 2011, ISSN: 0021-9606. DOI: 10.1063/1.3565032.

[83] J. D. Chodera and F. Noé, "Markov state models of biomolecular conformational dynamics," *Current Opinion in Structural Biology*, vol. 25, pp. 135–144, 2014, ISSN: 0959-440X. DOI: 10.1016/j.sbi.2014.04.002.

[84] G. R. Bowman, X. Huang, and V. S. Pande, "Using generalized ensemble simulations and Markov state models to identify conformational states," *Methods*, vol. 49, no. 2, pp. 197–201, 2009, ISSN: 1046-2023. DOI: 10.1016/j.ymeth.2009.04.013.

[85] G. R. Bowman, V. S. Pande, and F. Noé, "An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation," *Advances in Experimental Medicine and Biology*, vol. 797, pp. 1–6, 2014, ISSN: 0065-2598. DOI: 10.1007/978-94-007-7606-7_1.

[86] B. E. Husic and V. S. Pande, "Markov State Models: From an Art to a Science," *Journal of the American Chemical Society*, vol. 140, no. 7, pp. 2386–2396, 2018, ISSN: 0002-7863. DOI: 10.1021/jacs.7b12191.

[87] F. Noé and E. Rosta, "Markov Models of Molecular Kinetics," *The Journal of Chemical Physics*, vol. 151, no. 19, p. 190 401, 2019, ISSN: 0021-9606. DOI: 10.1063/1.5134029. eprint: 1911.00774.

[88] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T. R. Weikl, "Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 19 011–19 016, 2009, ISSN: 0027-8424. DOI: 10.1073/pnas.0905466106.

[89] J. Strahan, A. Antoszewski, C. Lorpaiboon, B. P. Vani, J. Weare, and A. R. Dinner, "Long-Time-Scale Predictions from Short-Trajectory Data: A Benchmark Analysis of the Trp-Cage Miniprotein," *Journal of Chemical Theory and Computation*, vol. 17, no. 5, pp. 2948–2963, 2021, ISSN: 1549-9618. DOI: 10.1021/acs.jctc.0c00933.

[90] J. D. Chodera, W. C. Swope, J. W. Pitera, and K. A. Dill, "Long-Time Protein Folding Dynamics from Short-Time Molecular Dynamics Simulations," *Multiscale Modeling & Simulation*, vol. 5, no. 4, pp. 1214–1226, 2006, ISSN: 1540-3459. DOI: 10.1137/06065146x.

[91] F. Nüske, H. Wu, J.-H. Prinz, C. Wehmeyer, C. Clementi, and F. Noé, "Markov state models from short non-equilibrium simulations—Analysis and correction of estimation bias," *The Journal of Chemical Physics*, vol. 146, no. 9, p. 094 104, 2017, ISSN: 0021-9606. DOI: 10.1063/1.4976518. eprint: 1701.01665.

[92] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20 150 202, 2016, ISSN: 1364-503X. DOI: 10.1098/rsta.2015.0202.

[93] L. Molgedey and H. G. Schuster, "Separation of a mixture of independent signals using time delayed correlations," *Physical Review Letters*, vol. 72, no. 23, pp. 3634–3637, 1994, ISSN: 0031-9007. DOI: 10.1103/physrevlett.72.3634.

[94] Y. Naritomi and S. Fuchigami, "Slow dynamics in protein fluctuations revealed by time-structure based independent component analysis: The case of domain motions," *The Journal of Chemical Physics*, vol. 134, no. 6, p. 065 101, 2011, ISSN: 0021-9606. DOI: 10.1063/1.3554380.

[95] C. R. Schwantes and V. S. Pande, "Improvements in Markov State Model Construction Reveal Many Non-Native Interactions in the Folding of NTL9," *Journal of Chemical Theory and Computation*, vol. 9, no. 4, pp. 2000–2009, 2013, ISSN: 1549-9618. DOI: 10.1021/ct300878a.

[96] G. Pérez-Hernández, F. Paul, T. Giorgino, G. D. Fabritiis, and F. Noé, "Identification of slow molecular order parameters for Markov model construction," *The Journal of Chemical Physics*, vol. 139, no. 1, p. 015 102, 2013, ISSN: 0021-9606. DOI: 10.1063/1.4811489. eprint: 1302.6614.

[97] H. Wu and F. Noe, "Variational Approach for Learning Markov Processes from Time Series Data," *Journal of Nonlinear Science*, vol. 30, no. 1, pp. 23–66, 2019, ISSN: 0938-8974. DOI: 10.1007/s00332-019-09567-y.

[98] B. Trendelkamp-Schroer and F. Noé, "Efficient Bayesian estimation of Markov model transition matrices with given stationary distribution," *The Journal of Chemical Physics*, vol. 138, no. 16, p. 164 113, 2013, ISSN: 0021-9606. DOI: 10.1063/1.4801325. eprint: 1301.2078.

[99] E. Suárez, R. P. Wiewiora, C. Wehmeyer, F. Noé, J. D. Chodera, and D. M. Zuckerman, "What Markov State Models Can and Cannot Do: Correlation versus Path-Based Observables in Protein-Folding Models," *Journal of Chemical Theory and Computation*, 2021, ISSN: 1549-9618. DOI: 10.1021/acs.jctc.0c01154.

[100] T. L. Hill, *Free Energy Transduction and Biochemical Cycle Kinetics*. Dover, 2004, ISBN: 978-1-4612-3558-3.

[101] D. Bhatt and D. M. Zuckerman, "Beyond Microscopic Reversibility: Are Observable Nonequilibrium Processes Precisely Reversible?" *Journal of Chemical Theory and Computation*, vol. 7, no. 8, pp. 2520–2527, 2011, ISSN: 1549-9618. DOI: 10.1021/ct200086k.

[102] J. Copperman and D. M. Zuckerman, "Accelerated Estimation of Long-Timescale Kinetics from Weighted Ensemble Simulation via Non-Markovian "Microbin" Analysis," *Journal of Chemical Theory and Computation*, vol. 16, no. 11, pp. 6763–6775, 2020, ISSN: 1549-9618. DOI: 10.1021/acs.jctc.0c00273.

[103] E. Suárez, A. J. Pratt, L. T. Chong, and D. M. Zuckerman, "Estimating first-passage time distributions from weighted ensemble simulations and non-Markovian analyses," *Protein Science*, vol. 25, no. 1, pp. 67–78, 2016, ISSN: 1469-896X. DOI: 10.1002/pro.2738.

[104] M. K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Pérez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J.-H. Prinz, and F. Noé, "PyEMMA 2: A software package for estimation, validation, and analysis of markov models," *J. Chem. Theory Comput.*, vol. 11, no. 11, pp. 5525–5542, Oct. 2015. DOI: 10.1021/acs.jctc.5b00743. [Online]. Available: https://doi.org/10.1021%2Facs.jctc.5b00743.

[105] M. Rappa, P. Jones, J. Freire, S. Chakrabarti, and D. Sculley, "Web-scale k-means clustering," *Proceedings of the 19th international conference on World wide web*, pp. 1177–1178, 2010. DOI: 10.1145/1772690.1772862.

[106] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[107] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html.

[108] J. D. Russo, *Jdrusso/mr_toolkit*, Mar. 2023. DOI: 10.5281/zenodo.7786843. [Online]. Available: https://doi.org/10.5281/zenodo.7786843.

[109] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan, and I. Stoica, "Ray: A Distributed Framework for Emerging AI Applications," *arXiv*, 2017. DOI: 10.48550/arxiv.1712.05889. eprint: 1712.05889.

[110] J. D. Russo, *Jdrusso/synd*, Dec. 2022. DOI: 10.5281/zenodo.7416145. [Online]. Available: https://doi.org/10.5281/zenodo.7416145.

[111] ——, *Jdrusso/synd-examples*, version 1.0, Mar. 2023. DOI: 10.5281/zenodo.7786847. [Online]. Available: https://doi.org/10.5281/zenodo.7786847.

[112] G. Ashraf, N. Greig, T. Khan, I. Hassan, S. Tabrez, S. Shakil, I. Sheikh, S. Zaidi, M. Akram, N. Jabir, C. Firoz, A. Naeem, I. Alhazza, G. Damanhouri, and M. Kamal, "Protein Misfolding and Aggregation in Alzheimer's Disease and Type 2 Diabetes Mellitus," *CNS & Neurological Disorders - Drug Targets*, vol. 13, no. 7, pp. 1280–1293, 2014, ISSN: 1871-5273. DOI: 10.2174/18715273136666140917095514.

[113] J. M. Tan, E. S. Wong, and K.-L. Lim, "Protein Misfolding and Aggregation in Parkinson's Disease," *Antioxidants & Redox Signaling*, vol. 11, no. 9, pp. 2119–2134, 2009, ISSN: 1523-0864. DOI: 10.1089/ars.2009.2490.

[114] C. Scheckel and A. Aguzzi, "Prions, prionoids and protein misfolding disorders," *Nature Reviews Genetics*, vol. 19, no. 7, pp. 405–418, 2018, ISSN: 1471-0056. DOI: 10.1038/s41576-018-0011-4.

[115] E. Deuerling, M. Gamerdinger, and S. G. Kreft, "Chaperone Interactions at the Ribosome," *Cold Spring Harbor Perspectives in Biology*, vol. 11, no. 11, a033977, 2019, ISSN: 1943-0264. DOI: 10.1101/cshperspect.a033977.

[116] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021, ISSN: 0028-0836. DOI: 10.1038/s41586-021-03819-2.

[117] L. A. Defelipe, J. P. Arcon, C. P. Modenutti, M. A. Marti, A. G. Turjanski, and X. Barril, "Solvents to Fragments to Drugs: MD Applications in Drug Design," *Molecules*, vol. 23, no. 12, p. 3269, 2018. DOI: 10.3390/molecules23123269.

[118] M. D. Vivo, M. Masetti, G. Bottegoni, and A. Cavalli, "Role of Molecular Dynamics and Related Methods in Drug Discovery," *Journal of Medicinal Chemistry*, vol. 59, no. 9, pp. 4035–4061, 2016, ISSN: 0022-2623. DOI: 10.1021/acs.jmedchem.5b01684.

[119] J. D. Durrant and J. A. McCammon, "Molecular dynamics simulations and drug discovery," *BMC Biology*, vol. 9, no. 1, p. 71, 2011. DOI: 10.1186/1741-7007-9-71.

[120] T. I. Adelusi, A.-Q. K. Oyedele, I. D. Boyenle, A. T. Ogunlana, R. O. Adeyemi, C. D. Ukachi, M. O. Idris, O. T. Olaoba, I. O. Adedotun, O. E. Kolawole, Y. Xiaoxing, and M. Abdul-Hammed, "Molecular modeling in drug discovery," *Informatics in Medicine Unlocked*, vol. 29, p. 100 880, 2022, ISSN: 2352-9148. DOI: 10.1016/j.imu.2022.100880.

[121] A. L. Perryman, S. Forli, G. M. Morris, C. Burt, Y. Cheng, M. J. Palmer, K. Whitby, J. A. McCammon, C. Phillips, and A. J. Olson, "A Dynamic Model of HIV Integrase Inhibition and Drug Resistance," *Journal of Molecular Biology*, vol. 397, no. 2, pp. 600–615, 2010, ISSN: 0022-2836. DOI: 10.1016/j.jmb.2010.01.033.

[122] X. Fu, T. Xie, N. J. Rebello, B. Olsen, and T. S. Jaakkola, "Simulate Time-integrated Coarse-grained Molecular Dynamics with Geometric Machine Learning," in *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022.

[123] M. J. Eslamibidgoli, M. Mokhtari, and M. H. Eikerling, "Recurrent Neural Network-based Model for Accelerated Trajectory Analysis in AIMD Simulations," *arXiv*, 2019. eprint: 1909.10124.

[124] H. Wu, A. Mardt, L. Pasquali, and F. Noe, "Deep generative markov state models," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/deb54ffb41e085fd7f69a75b6359c989-Paper.pdf.

[125] P. R. Vlachas, J. Zavadlav, M. Praprotnik, and P. Koumoutsakos, "Accelerated Simulations of Molecular Systems through Learning of Effective Dynamics," *Journal of Chemical Theory and Computation*, vol. 18, no. 1, pp. 538–549, 2022, ISSN: 1549-9618. DOI: 10.1021/acs.jctc.1c00809.

[126] H. Sidky, W. Chen, and A. L. Ferguson, "Molecular latent space simulators," *Chemical Science*, vol. 11, no. 35, pp. 9459–9467, 2020, ISSN: 2041-6520. DOI: 10.1039/d0sc03635h.

[127] S.-T. Tsai, E.-J. Kuo, and P. Tiwary, "Learning molecular dynamics with simple language model built upon long short-term memory neural network," *Nature Communications*, vol. 11, no. 1, p. 5115, 2020. DOI: 10.1038/s41467-020-18959-8. eprint: 2004.12360.

[128] S.-T. Tsai, E. Fields, Y. Xu, E.-J. Kuo, and P. Tiwary, "Path sampling of recurrent neural networks by incorporating known physics," *arXiv*, 2022. eprint: 2203.00597.

[129] A. Davtyan, J. F. Dama, G. A. Voth, and H. C. Andersen, "Dynamic force matching: A method for constructing dynamical coarse-grained models with realistic time dependence," *The Journal of Chemical Physics*, vol. 142, no. 15, p. 154 104, 2015, ISSN: 0021-9606. DOI: 10.1063/1.4917454.

[130] J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N. E. Charron, G. d. Fabritiis, F. Noé, and C. Clementi, "Machine Learning of Coarse-Grained Molecular Dynamics Force Fields," *ACS Central Science*, vol. 5, no. 5, pp. 755–767, 2019, ISSN: 2374-7943. DOI: 10.1021/acscentsci.8b00913.

[131] F. Noé, A. Tkatchenko, K.-R. Müller, and C. Clementi, "Machine Learning for Molecular Simulation," *Annual Review of Physical Chemistry*, vol. 71, no. 1, pp. 1–30, 2020, ISSN: 0066-426X. DOI: 10.1146/annurev-physchem-042018-052331. eprint: 1911.02792.

[132] S. Kumar, J. M. Rosenberg, D. Bouzida, R. H. Swendsen, and P. A. Kollman, "The weighted histogram analysis method for free-energy calculations on biomolecules," *Journal of Computational Chemistry*, vol. 13, no. 8, pp. 1011–1021, 1992, ISSN: 1096-987X. DOI: 10.1002/jcc.540130812.

[133] R. J. Allen, C. Valeriani, and P. R. t. Wolde, "Forward flux sampling for rare event simulations," *Journal of Physics: Condensed Matter*, vol. 21, no. 46, p. 463 102, 2009, ISSN: 0953-8984. DOI: 10.1088/0953-8984/21/46/463102. eprint: 0906.4758.

[134] A. Dickson and A. R. Dinner, "Enhanced Sampling of Nonequilibrium Steady States," *Annual Review of Physical Chemistry*, vol. 61, no. 1, pp. 441–459, 2010, ISSN: 0066-426X. DOI: 10.1146/annurev.physchem.012809.103433.

[135] J. D. Chodera, W. C. Swope, J. W. Pitera, and K. A. Dill, "Long-Time Protein Folding Dynamics from Short-Time Molecular Dynamics Simulations," *Multiscale Modeling & Simulation*, vol. 5, no. 4, pp. 1214–1226, 2006, ISSN: 1540-3459. DOI: 10.1137/06065146x.

[136] T. Hempel, M. J. d. Razo, C. T. Lee, B. C. Taylor, R. E. Amaro, and F. Noé, "Independent Markov decomposition: Toward modeling kinetics of biomolecular complexes," *Proceedings of the National Academy of Sciences*, vol. 118, no. 31, e2105230118, 2021, ISSN: 0027-8424. DOI: 10.1073/pnas.2105230118.

[137] E. Suárez, J. L. Adelman, and D. M. Zuckerman, "Accurate Estimation of Protein Folding and Unfolding Times: Beyond Markov State Models," *Journal of Chemical Theory and Computation*, vol. 12, no. 8, pp. 3473–3481, 2016, ISSN: 1549-9618. DOI: 10.1021/acs.jctc.6b00339.

[138] J. D. Russo, J. Copperman, D. Aristoff, G. Simpson, and D. M. Zuckerman, "Unbiased estimation of equilibrium, rates, and committors from Markov state model analysis," *arXiv*, 2021. eprint: 2105.13402.

[139] S. Cao, A. Montoya-Castillo, W. Wang, T. E. Markland, and X. Huang, "On the advantages of exploiting memory in Markov state models for biomolecular dynamics," *The Journal of Chemical Physics*, vol. 153, no. 1, p. 014 105, 2020, ISSN: 0021-9606. DOI: 10.1063/5.0010787.

[140] A. Kells, V. Koskin, E. Rosta, and A. Annibale, "Correlation functions, mean first passage times, and the Kemeny constant," *The Journal of Chemical Physics*, vol. 152, no. 10, p. 104 108, 2020, ISSN: 0021-9606. DOI: 10.1063/1.5143504. eprint: 1911.01729.

[141] A. Agarwal, S. Gnanakaran, N. Hengartner, A. F. Voter, and D. Perez, "Arbitrarily accurate representation of atomistic dynamics via Markov Renewal Processes," *arXiv*, 2020. eprint: 2008.11623.

[142] H. Wu, J.-H. Prinz, and F. Noé, "Projected metastable Markov processes and their estimation with observable operator models," *The Journal of Chemical Physics*, vol. 143, no. 14, p. 144 101, 2015, ISSN: 0021-9606. DOI: 10.1063/1.4932406.

[143] M. K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Pérez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J.-H. Prinz, and F. Noé, "PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models," *Journal of Chemical Theory and Computation*, vol. 11, pp. 5525–5542, Oct. 2015, ISSN: 1549-9618. DOI: 10.1021/acs.jctc.5b00743. [Online]. Available: http://dx.doi.org/10.1021/acs.jctc.5b00743 (visited on 10/19/2015).

[144] S. V. Krivov, "Nonparametric Analysis of Nonequilibrium Simulations," *Journal of Chemical Theory and Computation*, vol. 17, no. 9, pp. 5466–5481, 2021, ISSN: 1549-9618. DOI: 10.1021/acs.jctc.1c00218.

[145] Y. Matsunaga and Y. Sugita, "Refining markov state models for conformational dynamics using ensemble-averaged data and time-series trajectories," *The Journal of Chemical Physics*, vol. 148, no. 24, p. 241 731, 2018.

[146] V. S. Pande, K. Beauchamp, and G. R. Bowman, "Everything you wanted to know about Markov State Models but were afraid to ask," *Methods*, vol. 52, no. 1, pp. 99–105, 2010, ISSN: 1046-2023. DOI: 10.1016/j.ymeth.2010.06.002.

[147] G. R. Bowman, D. L. Ensign, and V. S. Pande, "Enhanced Modeling via Network Theory: Adaptive Sampling of Markov State Models," *Journal of Chemical Theory and Computation*, vol. 6, no. 3, pp. 787–794, 2010, ISSN: 1549-9618. DOI: 10.1021/ct900620b.

[148] R. Hall, T. Dixon, and A. Dickson, "On Calculating Free Energy Differences Using Ensembles of Transition Paths," *Frontiers in Molecular Biosciences*, vol. 7, p. 106, 2020, ISSN: 2296-889X. DOI: 10.3389/fmolb.2020.00106.

[149] F. Noé, H. Wu, J.-H. Prinz, and N. Plattner, "Projected and hidden Markov models for calculating kinetics and metastable states of complex molecules," *The Journal of Chemical Physics*, vol. 139, no. 18, p. 184 114, 2013, ISSN: 0021-9606. DOI: 10.1063/1.4828816. eprint: 1309.3220.

[150] J. D. Chodera, P. Elms, F. Noé, B. Keller, C. M. Kaiser, A. Ewall-Wice, S. Marqusee, C. Bustamante, and N. S. Hinrichs, "Bayesian hidden Markov model analysis of single-molecule force spectroscopy: Characterizing kinetics under measurement uncertainty," *arXiv*, 2011. eprint: 1108.1430.

[151] B. Trendelkamp-Schroer, H. Wu, F. Paul, and F. Noé, "Estimation and uncertainty of reversible Markov models," *The Journal of Chemical Physics*, vol. 143, no. 17, p. 174 101, 2015, ISSN: 0021-9606. DOI: 10.1063/1.4934536.

[152] S. Olsson, H. Wu, F. Paul, C. Clementi, and F. Noé, "Combining experimental and simulation data of molecular processes via augmented Markov models," *Proceedings of the National Academy of Sciences*, vol. 114, no. 31, pp. 8265–8270, 2017, ISSN: 0027-8424. DOI: 10.1073/pnas.1704803114.

[153] H. Wan and V. A. Voelz, "Adaptive Markov state model estimation using short reseeding trajectories," *The Journal of Chemical Physics*, vol. 152, no. 2, p. 024 103, 2020, ISSN: 0021-9606. DOI: 10.1063/1.5142457. eprint: 1912.05724.

[154] D. Aristoff, J. M. Bello-Rivas, and R. Elber, "A mathematical framework for exact milestoning," *Multiscale Modeling & Simulation*, vol. 14, no. 1, pp. 301–322, 2016.

[155] M. Bause, T. Wittenstein, K. Kremer, and T. Bereau, "Microscopic reweighting for nonequilibrium steady-state dynamics," *Physical Review E*, vol. 100, no. 6, p. 060 103, 2019, ISSN: 2470-0045. DOI: 10.1103/physreve.100.060103. eprint: 1907.08480.

[156] M. Bacci, A. Caflisch, and A. Vitalis, "On the removal of initial state bias from simulation data," *The Journal of Chemical Physics*, vol. 150, no. 10, p. 104 105, 2019, ISSN: 0021-9606. DOI: 10.1063/1.5063556.

[157] W. C. Swope, J. W. Pitera, and F. Suits, "Describing Protein Folding Kinetics by Molecular Dynamics Simulations. 1. Theory †," *The Journal of Physical Chemistry B*, vol. 108, no. 21, pp. 6571–6581, 2004, ISSN: 1520-6106. DOI: 10.1021/jp037421y.

[158] J. R. Norris, *Markov Chains*. Cambridge University Press, 1998.

[159] R. Costaouec, H. Feng, J. Izaguirre, and E. Darve, "Analysis of the accelerated weighted ensemble methodology," 2013.

[160] M. S. Apaydin, D. L. Brutlag, C. Guestrin, D. Hsu, J.-C. Latombe, and C. Varma, "Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion," *Journal of Computational Biology*, vol. 10, no. 3-4, pp. 257–281, 2003.

[161] M. Baudel, A. Guyader, and T. Lelièvre, "On the Hill relation and the mean reaction time for metastable processes," *Stochastic Processes and their Applications*, vol. 155, pp. 393–436, 2023, ISSN: 0304-4149. DOI: 10.1016/j.spa.2022.10.014.

[162] J. D. Russo and D. M. Zuckerman, "Synthetic molecular dynamics for efficient trajectory generation," *arXiv*, 2022. eprint: 2204.04343.

[163] M. Hoffmann, M. Scherer, T. Hempel, A. Mardt, B. d. Silva, B. E. Husic, S. Klus, H. Wu, N. Kutz, S. L. Brunton, and F. Noé, "Deeptime: a Python library for machine learning dynamical models from time series data," *Machine Learning: Science and Technology*, vol. 3, no. 1, p. 015 009, 2022. DOI: 10.1088/2632-2153/ac3de0. eprint: 2110.15013.

[164] R. Anandakrishnan, Z. Zhang, R. Donovan-Maiye, and D. M. Zuckerman, "Biophysical comparison of ATP synthesis mechanisms shows a kinetic advantage for the rotary process," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 113, no. 40, pp. 11 220–11 225, Sep. 2016. DOI: 10.1073/pnas.1608533113. [Online]. Available: https://doi.org/10.1073%2Fpnas.1608533113.

[165] G. Zhao, J. R. Perilla, E. L. Yufenyuy, X. Meng, B. Chen, J. Ning, J. Ahn, A. M. Gronenborn, K. Schulten, C. Aiken, and P. Zhang, "Mature HIV-1 capsid structure by cryo-electron microscopy and all-atom molecular dynamics," *Nature*, vol. 497, no. 7451, pp. 643–646, May 2013. DOI: 10.1038/nature12162. [Online]. Available: https://doi.org/10.1038%2Fnature12162.

[166] J. R. Perilla and K. Schulten, "Physical properties of the HIV-1 capsid from all-atom molecular dynamics simulations," *Nat Commun*, vol. 8, no. 1, Jul. 2017. DOI: 10.1038/ncomms15959. [Online]. Available: https://doi.org/10.1038%2Fncomms15959.

[167] L. Casalino, A. C. Dommer, Z. Gaieb, E. P. Barros, T. Sztain, S.-H. Ahn, A. Trifan, A. Brace, A. T. Bogetti, A. Clyde, H. Ma, H. Lee, M. Turilli, S. Khalid, L. T. Chong, C. Simmerling, D. J. Hardy, J. D. Maia, J. C. Phillips, T. Kurth, A. C. Stern, L. Huang, J. D. McCalpin, M. Tatineni, T. Gibbs, J. E. Stone, S. Jha, A. Ramanathan, and R. E. Amaro, "AI-driven multiscale simulations illuminate mechanisms of SARS-CoV-2 spike dynamics," *The International Journal of High Performance Computing Applications*, vol. 35, no. 5, pp. 432–451, Apr. 2021. DOI: 10.1177/10943420211006452. [Online]. Available: https://doi.org/10.1177%2F10943420211006452.

[168] J. Jung, W. Nishima, M. Daniels, G. Bascom, C. Kobayashi, A. Adedoyin, M. Wall, A. Lappala, D. Phillips, W. Fischer, C.-S. Tung, T. Schlick, Y. Sugita, and K. Y. Sanbonmatsu, "Scaling molecular dynamics beyond 100,000 processor cores for large-scale biophysical simulations," *J Comput Chem*, vol. 40, no. 21, pp. 1919–1930, Apr. 2019. DOI: 10.1002/jcc.25840. [Online]. Available: https://doi.org/10.1002%2Fjcc.25840.

[169] D. E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R. O. Dror, M. P. Eastwood, J. A. Bank, J. M. Jumper, J. K. Salmon, Y. Shan, and W. Wriggers, "Atomic-level characterization of the structural dynamics of proteins," *Science*, vol. 330, no. 6002, pp. 341–346, Oct. 2010. DOI: 10.1126/science.1187409. [Online]. Available: https://doi.org/10.1126%2Fscience.1187409.

[170] D. M. Zuckerman and T. B. Woolf, "Efficient dynamic importance sampling of rare events in one dimension," *Phys. Rev. E*, vol. 63, no. 1, Dec. 2000. DOI: 10.1103/physreve.63.016702. [Online]. Available: https://doi.org/10.1103%2Fphysreve.63.016702.

[171] A. M. A. West, R. Elber, and D. Shalloway, "Extending molecular dynamics time scales with milestoning: Example of complex kinetics in a solvated peptide," *The Journal of Chemical Physics*, vol. 126, no. 14, p. 145 104, Apr. 2007. DOI: 10.1063/1.2716389. [Online]. Available: https://doi.org/10.1063%2F1.2716389.

[172] T. S. van Erp, D. Moroni, and P. G. Bolhuis, "A novel path sampling method for the calculation of rate constants," *The Journal of Chemical Physics*, vol. 118, no. 17, pp. 7762–7774, May 2003. DOI: 10.1063/1.1562614. [Online]. Available: https://doi.org/10.1063%2F1.1562614.

[173] R. S. DeFever and S. Sarupria, "Contour forward flux sampling: Sampling rare events along multiple collective variables," *J. Chem. Phys.*, vol. 150, no. 2, p. 024 103, Jan. 2019. DOI: 10.1063/1.5063358. [Online]. Available: https://doi.org/10.1063%2F1.5063358.

[174] A. S. Saglam and L. T. Chong, "Protein–protein binding pathways and calculations of rate constants using fully-continuous, explicit-solvent simulations," *Chem. Sci.*, vol. 10, no. 8, pp. 2360–2372, 2019. DOI: 10.1039/c8sc04811h. [Online]. Available: https://doi.org/10.1039%2Fc8sc04811h.

[175] S. D. Lotz and A. Dickson, "Unbiased molecular dynamics of 11 min timescale drug unbinding reveals transition state stabilizing interactions," *J. Am. Chem. Soc.*, vol. 140, no. 2, pp. 618–628, Jan. 2018. DOI: 10.1021/jacs.7b08572. [Online]. Available: https://doi.org/10.1021%2Fjacs.7b08572.

[176] M. C. Zwier, J. L. Adelman, J. W. Kaus, A. J. Pratt, K. F. Wong, N. B. Rego, E. Suárez, S. Lettieri, D. W. Wang, M. Grabe, D. M. Zuckerman, and L. T. Chong, "WESTPA: An interoperable, highly scalable software package for weighted ensemble simulation and analysis," *J. Chem. Theory Comput.*, vol. 11, no. 2, pp. 800–809, Jan. 2015. DOI: 10.1021/ct5010615. [Online]. Available: https://doi.org/10.1021%2Fct5010615.

[177] B. W. Zhang, D. Jasnow, and D. M. Zuckerman, "The "weighted ensemble" path sampling method is statistically exact for a broad class of stochastic processes and binning procedures," *The Journal of Chemical Physics*, vol. 132, no. 5, p. 054 107, Feb. 2010. DOI: 10.1063/1.3306345. [Online]. Available: https://doi.org/10.1063%2F1.3306345.

[178] B. Abdul-Wahid, H. Feng, D. Rajan, R. Costaouec, E. Darve, D. Thain, and J. A. Izaguirre, "AWE-WQ: Fast-forwarding molecular dynamics using the accelerated weighted ensemble," *J. Chem. Inf. Model.*, vol. 54,

no. 10, pp. 3033–3043, Sep. 2014. DOI: 10.1021/ci500321g. [Online]. Available: https://doi.org/10.1021%2Fci500321g.

[179] S. D. Lotz and A. Dickson, "Wepy: A flexible software framework for simulating rare events with weighted ensemble resampling," *ACS Omega Omega*, vol. 5, no. 49, pp. 31 608–31 623, Dec. 2020. DOI: 10.1021/acsomega.0c03892. [Online]. Available: https://doi.org/10.1021%2Facsomega.0c03892.

[180] L. T. C. Ali S. Saglam, "Highly efficient computation of the basal K_on using direct simulation of protein–protein association with flexible molecular models," *J. Phys. Chem. B*, vol. 120, no. 1, pp. 117–122, Dec. 2015. DOI: 10.1021/acs.jpcb.5b10747. [Online]. Available: https://doi.org/10.1021%2Facs.jpcb.5b10747.

[181] R. M. Donovan, J.-J. Tapia, D. P. Sullivan, J. R. Faeder, R. F. Murphy, M. Dittrich, and D. M. Zuckerman, "Unbiased rare event sampling in spatial stochastic systems biology models using a weighted ensemble of trajectories," *PLoS Comput Biol Computational Biology*, vol. 12, no. 2, M. Meier-Schellersheim, Ed., e1004611, Feb. 2016. DOI: 10.1371/journal.pcbi.1004611. [Online]. Available: https://doi.org/10.1371%2Fjournal.pcbi.1004611.

[182] J.-J. Tapia, A. S. Saglam, J. Czech, R. Kuczewski, T. M. Bartol, T. J. Sejnowski, and J. R. Faeder, "MCell-r: A particle-resolution network-free spatial modeling framework," in *Modeling Biomolecular Site Dynamics*, Springer New York, 2019, pp. 203–229. DOI: 10.1007/978-1-4939-9102-0_9. [Online]. Available: https://doi.org/10.1007%2F978-1-4939-9102-0_9.

[183] M. E. Johnson, A. Chen, J. R. Faeder, P. Henning, I. I. Moraru, M. Meier-Schellersheim, R. F. Murphy, T. Prüstel, J. A. Theriot, and A. M. Uhrmacher, "Quantifying the roles of space and stochasticity in computer simulations for cell biology and cellular biochemistry," *MBoC*, vol. 32, no. 2, A. Mogilner, Ed., pp. 186–210, Jan. 2021. DOI: 10.1091/mbc.e20-08-0530. [Online]. Available: https://doi.org/10.1091%2Fmbc.e20-08-0530.

[184] N. Donyapour, N. M. Roussey, and A. Dickson, "REVO: Resampling of ensembles by variation optimization," *J. Chem. Phys.*, vol. 150, no. 24, p. 244 112, Jun. 2019. DOI: 10.1063/1.5100521. [Online]. Available: https://doi.org/10.1063%2F1.5100521.

[185] D. Bhatt, B. W. Zhang, and D. M. Zuckerman, "Steady-state simulations using weighted ensemble path sampling," *The Journal of Chemical Physics*, vol. 133, no. 1, p. 014 110, Jul. 2010. DOI: 10.1063/1.3456985. [Online]. Available: https://doi.org/10.1063%2F1.3456985.

[186] E. Suárez, S. Lettieri, M. C. Zwier, C. A. Stringer, S. R. Subramanian, L. T. Chong, and D. M. Zuckerman, "Simultaneous computation of dynamical and equilibrium information using a weighted ensemble of trajectories," *J. Chem. Theory Comput.*, vol. 10, no. 7, pp. 2658–2667, Mar. 2014. DOI: 10.1021/ct401065r. [Online]. Available: https://doi.org/10.1021%2Fct401065r.

[187] A. T. Bogetti, B. Mostofian, A. Dickson, A. Pratt, A. S. Saglam, P. O. Harrison, J. L. Adelman, M. Dudek, P. A. Torrillo, A. J. DeGrave, U. Adhikari, M. C. Zwier, D. M. Zuckerman, and L. T. Chong, "A suite of tutorials for the WESTPA rare-events sampling software [article v1.0]," *LiveCoMS*, vol. 1, no. 2, 2019. DOI: 10.33011/livecoms.1.2.10607. [Online]. Available: https://doi.org/10.33011%2Flivecoms.1.2.10607.

[188] *WESTPA tutorials*, https://github.com/westpa/westpa_tutorials, 2021.

[189] P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, R. P. Wiewiora, B. R. Brooks, and V. S. Pande, "OpenMM 7: Rapid development of high performance algorithms for molecular dynamics," *PLoS Comput Biol Computational Biology*, vol. 13, no. 7, R. Gentleman, Ed., e1005659, Jul. 2017. DOI: 10.1371/journal.pcbi.1005659. [Online]. Available: https://doi.org/10.1371%2Fjournal.pcbi.1005659.

[190] C. Grebner, E. Malmerberg, A. Shewmaker, J. Batista, A. Nicholls, and J. Sadowski, "Virtual screening in the cloud: How big is big enough?" *J. Chem. Inf. Model.*, vol. 60, no. 9, pp. 4274–4282, Nov. 2019. DOI: 10.1021/acs.jcim.9b00779. [Online]. Available: https://doi.org/10.1021%2Facs.jcim.9b00779.

[191] *WESTPA 2.0 tutorials*, https://github.com/westpa/westpa2_tutorials, 2021.

[192] P. A. Torrillo, A. T. Bogetti, and L. T. Chong, "A minimal, adaptive binning scheme for weighted ensemble simulations," *J. Phys. Chem. A*, vol. 125, no. 7, pp. 1642–1649, Feb. 2021. DOI: 10.1021/acs.jpca.0c10724. [Online]. Available: https://doi.org/10.1021%2Facs.jpca.0c10724.

[193] J. L. Adelman and M. Grabe, "Simulating rare events using a weighted ensemble-based string method," *The Journal of Chemical Physics*, vol. 138, no. 4, p. 044 105, Jan. 2013. DOI: 10.1063/1.4773892. [Online]. Available: https://doi.org/10.1063%2F1.4773892.

[194] A. Dickson and C. L. Brooks, "WExplore: Hierarchical exploration of high-dimensional spaces using the weighted ensemble algorithm," *J. Phys. Chem. B*, vol. 118, no. 13, pp. 3532–3542, Feb. 2014. DOI: 10.1021/jp411479c. [Online]. Available: https://doi.org/10.1021%2Fjp411479c.

[195] R. T. McGibbon, K. A. Beauchamp, M. P. Harrigan, C. Klein, J. M. Swails, C. X. Hernández, C. R. Schwantes, L.-P. Wang, T. J. Lane, and V. S. Pande, "MDTraj: A modern open library for the analysis of molecular dynamics trajectories," *Biophysical Journal*, vol. 109, no. 8, pp. 1528–1532, Oct. 2015. DOI: 10.1016/j.bpj.2015.08.015. [Online]. Available: https://doi.org/10.1016%2Fj.bpj.2015.08.015.

[196] H. Nguyen, D. A. Case, and A. S. Rose, "NGLview–interactive molecular graphics for jupyter notebooks," *Bioinformatics*, vol. 34, no. 7, A. Valencia, Ed., pp. 1241–1242, Dec. 2017. DOI: 10.1093/bioinformatics/btx789. [Online]. Available: https://doi.org/10.1093%2Fbioinformatics%2Fbtx789.

[197] B. Hess, C. Kutzner, D. van der Spoel, and E. Lindahl, "GROMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation," *J. Chem. Theory Comput.*, vol. 4, no. 3, pp. 435–447, Feb. 2008. DOI: 10.1021/ct700301q. [Online]. Available: https://doi.org/10.1021%2Fct700301q.

[198] A. J. DeGrave, A. T. Bogetti, and L. T. Chong, "The RED scheme: Rate-constant estimation from pre-steady state weighted ensemble simulations," *J. Chem. Phys.*, vol. 154, no. 11, p. 114 111, Mar. 2021. DOI: 10.1063/5.0041278. [Online]. Available: https://doi.org/10.1063%2F5.0041278.

[199] B. Mostofian and D. M. Zuckerman, "Statistical uncertainty analysis for small-sample, high log-variance data: Cautions for bootstrapping and bayesian bootstrapping," *J. Chem. Theory Comput.*, vol. 15, no. 6, pp. 3499–3509, Apr. 2019. DOI: 10.1021/acs.jctc.9b00015. [Online]. Available: https://doi.org/10.1021%2Facs.jctc.9b00015.

[200] *Created with biorender.com*, http://biorender.com, 2021.

[201] R. Salomon-Ferrer, D. A. Case, and R. C. Walker, "An overview of the Amber biomolecular simulation package," *WIREs Comput Mol Sci*, vol. 3, no. 2, pp. 198–210, Sep. 2012. DOI: 10.1002/wcms.1121. [Online]. Available: https://doi.org/10.1002%2Fwcms.1121.

[202] *NMPathAnalysis*, https://github.com/ZuckermanLab/NMpathAnalysis, 2021.

[203] D. Bhowmik, S. Gao, M. T. Young, and A. Ramanathan, "Deep clustering of protein folding simulations," *BMC Bioinformatics Bioinformatics*, vol. 19, no. S18, Dec. 2018. DOI: 10.1186/s12859-018-2507-5. [Online]. Available: https://doi.org/10.1186%2Fs12859-018-2507-5.

[204] J. M. L. Ribeiro, P. Bravo, Y. Wang, and P. Tiwary, "Reweighted autoencoded variational bayes for enhanced sampling (RAVE)," *The Journal of Chemical Physics*, vol. 149, no. 7, p. 072 301, Aug. 2018. DOI: 10.1063/1.5025487. [Online]. Available: https://doi.org/10.1063%2F1.5025487.

[205] T. Romo and A. Grossfield, "LOOS: An extensible platform for the structural analysis of simulations," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, Sep. 2009. DOI: 10.1109/iembs.2009.5335065. [Online]. Available: https://doi.org/10.1109%2Fiembs.2009.5335065.

[206] T. D. Romo, N. Leioatts, and A. Grossfield, "Lightweight object oriented structure analysis: Tools for building tools to analyze molecular dynamics simulations," *J. Comput. Chem.*, vol. 35, no. 32, pp. 2305–2318, Oct. 2014. DOI: 10.1002/jcc.23753. [Online]. Available: https://doi.org/10.1002%2Fjcc.23753.

[207] N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. Beckstein, "MDAnalysis: A toolkit for the analysis of molecular dynamics simulations," *J. Comput. Chem.*, vol. 32, no. 10, pp. 2319–2327, Apr. 2011. DOI: 10.1002/jcc.21787. [Online]. Available: https://doi.org/10.1002%2Fjcc.21787.

[208] R. Gowers, M. Linke, J. Barnoud, T. Reddy, M. Melo, S. Seyler, J. Domański, D. Dotson, S. Buchoux, I. Kenney, and O. Beckstein, "MDAnalysis: A python package for the rapid analysis of molecular dynamics simulations," in *Proceedings of the Python in Science Conference*, SciPy, 2016. DOI: 10.25080/majora-629e541a-00e. [Online]. Available: https://doi.org/10.25080%2Fmajora-629e541a-00e.

[209] D. A. Case, H. M. Aktulga, K. Belfon, I. Y. Ben-Shalom, J. T. Berryman, S. R. Brozell, D. S. Cerutti, I. Cheatham T. E., G. A. Cisneros, V. W. D. Cruzeiro, T. A. Darden, R. E. Duke, G. Giambasu, M. K. Gilson, H. Gohlke, A. W. Goetz, R. Harris, S. Izadi, S. A. Izmailov, K. Kasavajhala, M. C. Kaymak, E. King, A. Kovalenko, T. Kurtzman, T. S. Lee, S. LeGrand, P. Li, C. Lin, J. Liu, T. Luchko, R. Luo, M. Machado, V. Man, M. Manathunga, K. M. Merz, Y. Miao, O. Mikhailovskii, G. Monard, H. Nguyen, K. A. O'Hearn, A. Onufriev, F. Pan, S. Pantano, R. Qi, A. Rahnamoun, D. R. Roe, A. Roitberg, C. Sagui, S. Schott-Verdugo, A. Shajan, J. Shen, C. L. Simmerling, N. R. Skrynnikov, J. Smith, J. Swails, R. C. Walker, J. Wang, J. Wang, H. Wei, R. M. Wolf, X. Wu, Y. Xiong, Y. Xue, D. M. York, S. Zhao, and P. A. Kollman, *Amber 2022*, San Francisco, 2022.

[210] D. A. Case, T. E. Cheatham, T. Darden, H. Gohlke, R. Luo, K. M. Merz, A. Onufriev, C. Simmerling, B. Wang, and R. J. Woods, "The Amber biomolecular simulation programs," *Journal of Computational Chemistry*, vol. 26, no. 16, pp. 1668–1688, 2005, ISSN: 0192-8651. DOI: 10.1002/jcc.20290.

Appendices

A Microscopic transition matrix

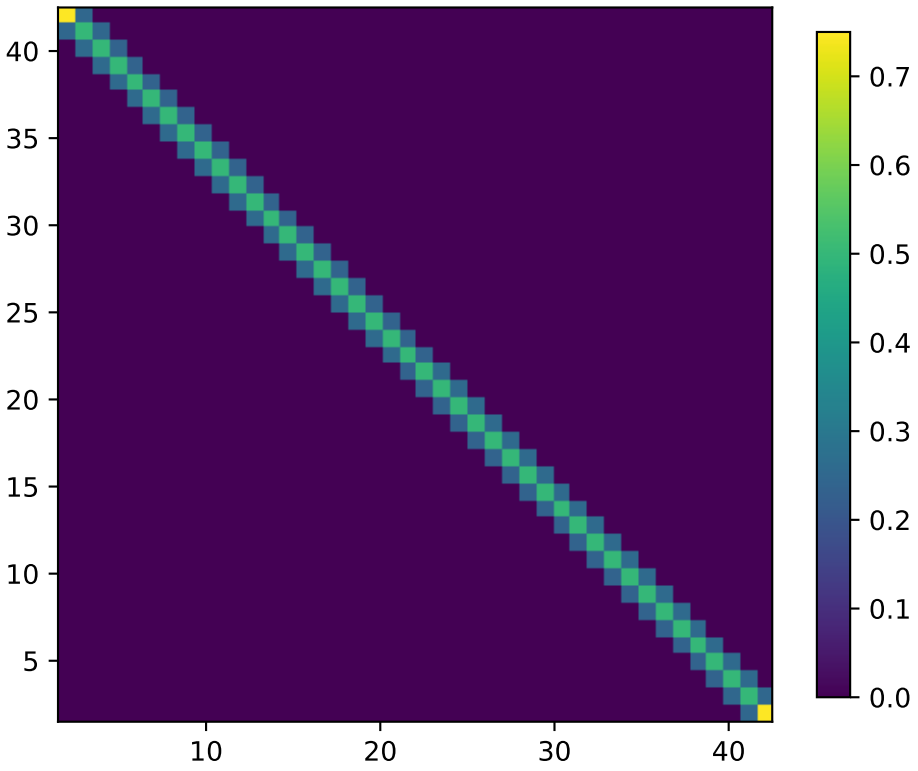


Figure A.1: Heatmap of the microscopic transition matrix P. Microstates at the left and right boundary (i.e. microstates 1 and 42) have a 0.75 self-transition probability, and 0.25 transition probability to the neighbor. Microstates 2 and 41 have a 0.5 self-transition probability, and 0.25 transition probability to each neighbor. A barrier is introduced by giving all other microstates a 0.5 self-transition probability, a 0.24 transition probability to the adjacent microstate closer to the middle of the system, and 0.26 transition probability to the adjacent microstate away from the middle.

B Iterative trajectory reweighting for estimation of equilibrium and non-equilibrium observables

Iterative trajectory reweighting for estimation of equilibrium and non-equilibrium observables

John D. Russo, Jeremy Copperman,* and Daniel M. Zuckerman†
Department of Biomedical Engineering,
Oregon Health and Science University, Portland, OR

(Dated: June 18, 2020)

We present two algorithms by which a set of short, unbiased trajectories can be iteratively reweighted to obtain various observables. The first algorithm estimates the stationary (steady state) distribution of a system by iteratively reweighting the trajectories based on the average probability in each state. The algorithm applies to equilibrium or non-equilibrium steady states, exploiting the 'left' stationarity of the distribution under dynamics – i.e., in a discrete setting, when the column vector of probabilities is multiplied by the transition matrix expressed as a left stochastic matrix. The second procedure relies on the 'right' stationarity of the committor (splitting probability) expressed as a row vector. The algorithms are unbiased, do not rely on computing transition matrices, and make no Markov assumption about discretized states. Here, we apply the procedures to a one-dimensional double-well potential, and to a 208μs atomistic Trp-cage folding trajectory from D.E. Shaw Research.

I. INTRODUCTION

The inability of molecular dynamics (MD) simulation to reach timescales pertinent to complex phenomena in biology and other fields [1–5] has motivated the development of numerous methods to enhance sampling in both equilibrium [6–8] and non-equilibrium [9–13] contexts. Markov state models (MSMs) effectively "stitch together" shorter trajectories dispersed in configuration space [14, 15] from which both equilibrium and non-equilibrium observables can be computed – e.g., state populations or kinetic properties. MSMs can be applied to transition phenomena even when no full, continuous trajectory of a particular transition is present in the original set of trajectories.

This article presents a simple, alternative method for reweighting MSM-like trajectory sets that provides both equilibrium and non-equilibrium information without bias. The essence of the strategy is to exploit the stationarity of a distribution or property to enable the calculation of that observable in a self-consistent way via iteration. *The key ingredient is the use of continuous trajectories as the sole basis for analysis, intrinsically accounting for all properties of the underlying dynamics.* Iteration is employed to reach a fully self-consistent stationary solution. Trajectory reweighting has previously been applied to biased trajectories (e.g., [16]) as well as to unbiased trajectories in MSM construction, albeit without self-consistent iteration and under a Markov assumption [17].

Observables that can be computed through the iterative approach, without any Markov assumption or lag-time limitation, include the equilibrium distribution, the distribution in a non-equilibrium steady state (NESS), the committor or splitting probability and the mean first-passage time (MFPT) associated with arbitrary macrostates. The only error in the procedures described below, besides statistical noise, arises from the discretization of phase space into bins. We emphasize that no Markov assumption is made.

The approach can be understood in the context of estimating the equilibrium distribution based on a set of *unbiased* trajectories initiated from an arbitrary set of initial configurations, presumably out of equilibrium. For example, the trajectories may be initiated from an approximately uniform distribution in the space of some coordinate of interest. We assume that a classification of the space has already been performed into bins whose populations are a proxy for the distribution. Given that the equilibrium distribution does not change in time, if we can assign suitable weights (probabilities) to each of a set of trajectories – such that the weighted distribution is in equilibrium based on *only* the initial points of each trajectory – that distribution must remain in equilibrium thereafter. Although the weights are unknown in advance, they can be set to arbitrary initial values and refined by iteration.

Continuing the equilibrium example, imagine that each trajectory is initially assigned an equal weight, with all weights summing to one. Now each bin can be monitored over *time*, and the average weight in each bin is recorded. This average weight is the first non-trivial estimate of the equilibrium probability in the bin. Physically, bins that attract more trajectories will be assigned larger weights as expected. In each iteration, the time-averaged probability from the prior iteration is divided among the trajectories which *start* in that bin. Time-averaged bin probabilities are recomputed and trajectory weights reassigned at each iteration until convergence to steady values. This procedure is described in Algorithm 1.

The same procedure can be applied for non-equilibrium

* copperma@ohsu.edu
† zuckermd@ohsu.edu

reweighting. To obtain the NESS distribution, external and/or boundary conditions must be properly accounted for in preparing trajectories for analysis (Algorithm 0), but this is not a significant complication. With the NESS distribution, the MFPT can be obtained from the Hill relation [18, 19].

The committor, also known as the splitting probability [18, 20–22], exhibits a different type of stationarity [23] and is estimated by a different but equally simple type of iterative procedure that averages over trajectories instead of bins (Algorithm 2). Defined as the probability to proceed from a designated initial phase point to a "target" macrostate prior to reaching a different "off-target" macrostate, the committor can be naively estimated by the fraction of trajectories from the initial point that first reach the target. In an iterative approach operating in the space of bins, we can exploit the committor's stationarity: at any fixed time, the average committor of all downstream trajectories emanating from a given bin must match that bin's committor value. Procedurally, each bin not in the target state is assigned a trivial initial committor estimate of, say, zero. A bin's estimate is updated at each iteration as the average over every time step of every trajectory after visiting the bin, with the "boundary conditions" that all time points after entering the target macrostate are evaluated as one or, after entering the off-target state, as zero.

We emphasize that these trajectory averaging and reweighting processes make no Markov assumption and are unbiased at the shortest available time discretization. As with any method, however, the approach is limited by the amount of data which in turn will dictate the sizes of bins which can be used. More data enables smaller bins and higher phase-space resolution. Because the dynamics of individual trajectories continually update observable estimates, the discretization error may be less would naively be expected from spatial discretization.

II. ALGORITHMS

A. Trajectory preparation

In order to demonstrate our algorithms, we extracted a set of trajectory fragments from one or more long trajectories according to Algorithm 0. Trajectory fragments may be of fixed length, or variable length if strict absorbing boundary conditions are used. Source-sink boundary conditions will use spliced fixed-length trajectories.

The analyses performed below are most easily understood based on trajectory fragments sorted by the starting configuration (phase-space point) of each fragment. These fragments are "copied" from the original long trajectory and hence may have overlapping sequences. For example, fragment 1 may consist of time steps 2 - 101 of the original trajectory, and fragment 2 might be steps 7 - 106. Correlations are thus introduced, but we estimate statistical uncertainty using fully independent datasets.

Algorithm 0 Trajectory fragment selection

Begin with one or more trajectories, discretized according to a set of bins i (or "microstates" in MSM terminology). For simplicity, we will assume a single long trajectory is used with t denoting the discrete time index.

For each bin i, generate a list of possible start points t_s which are the time indices of every configuration or phase point within that bin. The set of trajectory starts in bin i – denoted $\{t_s\}_i$ – is not indexed to avoid complex notation. That is, each of K_i start points indexed by $k = 1 \cdots K_i$ in bin i is fully denoted as $t_s(i,k)$

if no absorbing ('open') boundaries **then**

    The fragments associated with each bin i consist of time points $t_s, t_s+1, \ldots, t_s+M-1$ for each start point in the set $\{t_s\}_i$. These fixed-length fragments each have M steps.

else if strict absorbing boundary conditions **then**

    Two macrostates consisting of sets of bins should be defined, such that no bin is in more than one macrostate and some bins are "intermediate" – i.e., not in either macrostate.

    The fragments will start *only from intermediate bins* and consist of time points $t_s, t_s+1, t_s+2, \ldots$ for each start point t_s. Each fragment is terminated upon reaching *either* macrostate or at the end of the original trajectory, whichever comes first.

else if source-sink boundary conditions **then**

    Two macrostates consisting of sets of bins should be defined, such that no bin is in more than one macrostate and some bins are "intermediate" – i.e., not in either macrostate. *One macrostate will be the sink (a.k.a. target) and the other is the source state.*

    Define a time-independent source distribution γ over source bins such that $\sum_j \gamma_j = 1$ with $\gamma_j \geq 0$.

    The fragments initially consist of time points $t_s, t_s+1, t_s+2, \ldots t_s+M-1$ for each start point t_s. If the target is reached prior to the final point, let t_t be the time the target is first reached.

    Fragments reaching the target are spliced to fragments starting at the source. That is, to make a full segment of M steps, the initial list $t_s, t_s+1, t_s+2, \ldots t_t-1$ is concatenated with a trajectory segment from a source starting point $t_s(j,k)$ with $j \in$ source; this segment is re-indexed to start at t_t. The particular segment is chosen uniformly among the t_s for bin j after j is selected according to γ.

end if

B. Equilibrium distribution

Trajectories can be reweighted into the equilibrium distribution. Our procedure can be seen as a non-Markovian, fully self-consistent extension of the single-iteration trajectory reweighting recently proposed in a Markov context [17]. Reweighting is an old idea [24] which is limited by the well-known overlap problem [4]. Overlap remains a concern in any reweighting, but the present strategy uses additional information ignored in many other methods, namely, the dynamical information intrinsic to trajectories. Algorithm 1 infers a conformational distribution consistent with the underlying

continuous dynamics without any Markov assumption. Discretization necessarily introduces some error but because continuous trajectories evolve irrespective of bin boundaries, this error may be reduced. That is, trajectory dynamics automatically account for intra-bin landscape features.

Algorithm 1 uses stationarity of the equilibrium distribution to re-assign weights of trajectory fragments in a self-consistent manner. In every iteration, the weight of the fragments starting in a given bin is replaced by the time-averaged weight in the bin. Stationarity is enforced in a self-consistent way because the initial bin probability must match the time average.

Algorithm 1 Stationary distribution calculation

1: Prepare a set of fixed-length trajectory fragments with open boundary conditions (for equilibrium) or with source-sink conditions (for NESS) following Algorithm 0. Bins not visited by any fragment will be assigned zero probability. Note that sink/target bins have zero probability by definition.
2: Assign each trajectory fragment an initial weight. Initial weights are arbitrary, so long as total weight (probability) sums to 1, a condition which is preserved at every time step in every iteration. Here we assign initial weights so that each bin has equal total initial weight, which is evenly divided among fragments starting in the bin.
3: **repeat**
4: **for all** bins **do**
5: Sum the weights of all fragments in the bin at each time
6: The averaged-over-time bin weight is divided equally among trajectory fragments *starting* in that bin for the next iteration.
7: **end for**
8: **until** A user-defined convergence threshold is met
9: The entire iterative procedure can be repeated for trajectory sets generated by progressively trimming the first time-point from each trajectory (to decrease initial state bias), creating a basis for a final estimate averaged over trimmed trajectory sets. This protocol was not used to generate the data shown.
10: For NESS, the entire iterative procedure can be repeated for trajectory sets generated by progressively trimming the first time-point from each trajectory (to decrease initial state bias), creating a basis for a final estimate averaged over trimmed trajectory sets. Additionally the source-sink splicing of Algorithm

C. Non-equilibrium steady-state

The probability distribution of a non-equilibrium steady state (NESS) in the same way (Algorithm 1) except that suitable boundary conditions must be enforced. We focus here on a source-sink NESS because that is most pertinent to rate-constant estimation. Such a NESS requires defining (i) the absorbing source and sink macrostates, which shall consist strictly of non-overlapping sets of bins and (ii) the source, or feedback, *distribution* γ which describes how probability reaching the sink macrostate is redistributed at the source [25]. In a discrete picture, we let γ_i be the fractional probability to be initiated (or fed back) to bin i, such that $\sum_i \gamma_i = 1$. No bin with $\gamma_i > 0$ can be part of the sink. See Algorithm 0.

As a technical aside, we note that, somewhat confusingly, bins with positive γ values do not in themselves necessarily define the source macrostate. For example, in the important special case of the source-sink NESS which maintains an equilibrium distribution within the source macrostate (only), bins not on the *surface* of the macrostate strictly require $\gamma = 0$ [26]. In any case, our approach applies to arbitrary choices of the source distribution γ.

D. Committor calculation

The committor is not a probability distribution per se and exhibits a different kind of stationarity that has been noted previously [20–23]. The committor $\Pi(\mathbf{x})$ for a phase-space point $\mathbf{x}$ is defined to be the probability of trajectories initiated from $\mathbf{x}$ reaching a 'target' macrostate before reaching a different 'initial' macrostate, both of which can be arbitrarily defined if non-overlapping. We assume dynamics are stochastic and Markovian in the continuous phase space. Discrete bins used for calculation in the algorithm are not assumed to behave as Markov states.

The iterative algorithm can be understood by first considering 'brute force' committor estimation by initiating a large number, N, of trajectories from $\mathbf{x}$ and computing the fraction which reach the target first. However, instead of waiting for all such trajectories to be absorbed at one state or the other, we can imagine examining the distribution of phase points $p(\mathbf{x}^t)$ at finite time t which evolved from $\mathbf{x}$ – that is, from trajectories initiated at $t = 0$ from $\mathbf{x}$ with absorbing boundary conditions at initial and target states. If t is sufficiently short, such that no trajectories have yet been absorbed by either state, the expected fraction that eventually will be absorbed to the target *by definition* is given by the average committor of *current* phase points $\mathbf{x}^t$ [23]. That is, with trajectories indexed by i, the committor can be estimated by

$$\Pi(\mathbf{x}) \doteq (1/N) \sum_i \Pi(\mathbf{x}_i^t) \; . \tag{1}$$

This same expression can be used at longer t when some trajectories have been absorbed, if we introduce the 'overloaded' definitions $\Pi(\mathbf{x}_i^t) \equiv 1$ if trajectory i was absorbed to the target and zero if absorbed to the initial state. With this adjustment, the estimator (1) is applicable at any time t.

Algorithm 2 implements the preceding formulation using an iterative process for self-consistency. Because the committor average is stationary, we can use (1) at any time or by averaging over all times. Here, committor

values are updated based on following trajectories passing through a given phase point, approximated as a discrete bin, and calculating time averages of all the visited 'downstream' bins. By contrast, distribution estimation in Algorithm 1 averages over time for each bin separately, and do not follow trajectories. For convenience, trajectories which reach a macrostate are 'padded' with committor values of zero or one depending on the macrostate.

Once again, we expect a slight discretization error but using trajectories leverages the maximum possible information about intra-bin dynamics. Bins are not assumed to exhibit Markovian behavior.

Algorithm 2 Committor calculation

1: Begin with a set of absorbing boundary condition trajectory fragments, as described in Algorithm 0
2: Assign each bin within the target macrostate a committor of 1. All other bins are initialized to 0, including in the initial macrostate.
3: **for all** trajectory fragments **do**
4: **if** fragment reaches target or initial macrostate **then**
5: Pad the trajectory: Assign *fixed* committor values of 1 or 0, respectively, to all time points starting from the absorbing event and ending at the chosen fixed length M.
6: **end if**
7: **end for**
8: **repeat**
9: **for all** bins **do**
10: **if** bin is within a macrostate **then**
11: Do not change committor - it remains 0 or 1
12: **else**
13: The next estimated committor value is the average committor over all bins subsequently visited by all trajectories starting in this bin
14: **end if**
15: **end for**
16: **until** Change between iterations is below user-defined convergence threshold

III. SYSTEMS AND RESULTS

A. Systems

The iterative equilibrium distribution estimation technique is first applied to a set of simulated trajectories in a one-dimensional (1D) double-well potential with a 5 k_BT barrier, shown in Fig. 1 and simulated using overdamped Langevin dynamics.

Motion under overdamped Langevin dynamics obeys

$$x_{i+1} = x_i + -\frac{\Delta t}{m\gamma}\left.\frac{dV}{dx}\right|_{x_i} + \Delta x_{\text{rand}} \tag{2}$$

where $\gamma = 0.01\text{s}^{-1}$ is the friction coefficient, m is set to 1, Δx_{rand} is a stochastic displacement with its magnitude drawn from a Gaussian distribution centered at 0 with $\sigma = \sqrt{2k_BT\Delta t/m\gamma}$ where k_BT is set to 1 and $\Delta t = 5\times10^{-4}$s is the timestep. The double-well potential used is given by

$$V(x) = k_BT\left[\left(0.1\frac{x}{x_0}\right)^{10} - \left(0.7\frac{x}{x_0}\right)^{2}\right]. \tag{3}$$

where x_0 is an arbitrary reference length.

The full dataset consisted of 32 trajectories, each run for 2×10^6 steps. We used 130 equal-width states, of which 80 were in the intermediate region and 25 were in each of states A and B, shown in Fig. 1.

The other system analyzed is a 208 μs atomistic molecular dynamics simulation of Trp-cage folding saved with 200 ps resolution [27]. This trajectory is notable for being very long and well-sampled.

B. Equilibrium distribution

Fig. 2 illustrates the convergence of the iteratively estimated equilibrium distribution and Fig. 3 demonstrates the final result of the iterative calculation in the 1D double-well system. In general, the final converged iteration reproduces the Boltzmann distribution precisely and without bias.

Applying the iterative equilibrium distribution estimator to the Trp-cage folding trajectory (Fig. 4) fragments similarly shows reasonable agreement with simple counts. The right-most bin is a notable exception and warrants further investigation.

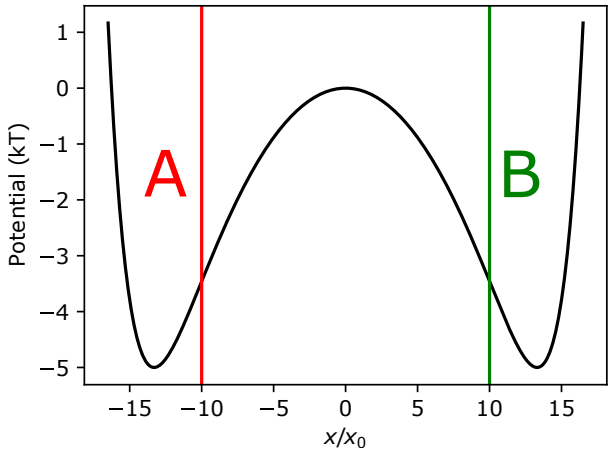


FIG. 1. Double-well potential used for overdamped Langevin dynamics simulations. Macrostate A is comprised of states at $x/x_0 < -10$, and B of states at $x/x_0 > 10$.

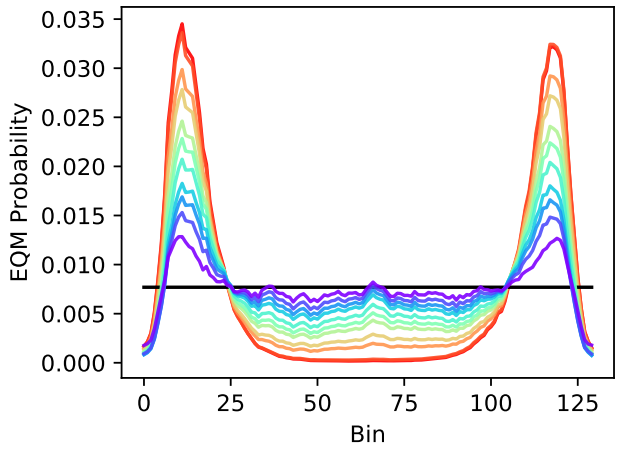


FIG. 2. Plot of the iterative equilibrium distribution estimator's convergence. Some intermediate iterations have been omitted for clarity. Warmer colors show later iterations, and the black line is the initial weight in each bin.

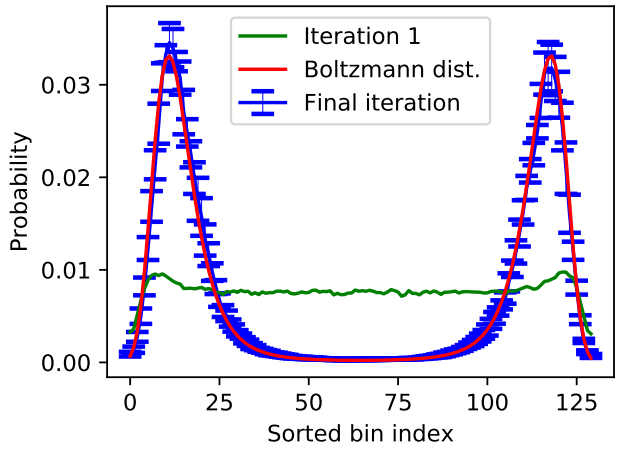


FIG. 3. Equilibrium distributions for the double-well potential system. Since the exact form of the potential is known, the Boltzmann distribution (red) provides reference equilibrium probabilities. Shown are the distribution after one iteration (green) and the distribution after the convergence criterion was met (blue). Error bars indicate one standard deviation across 5 independent trials.

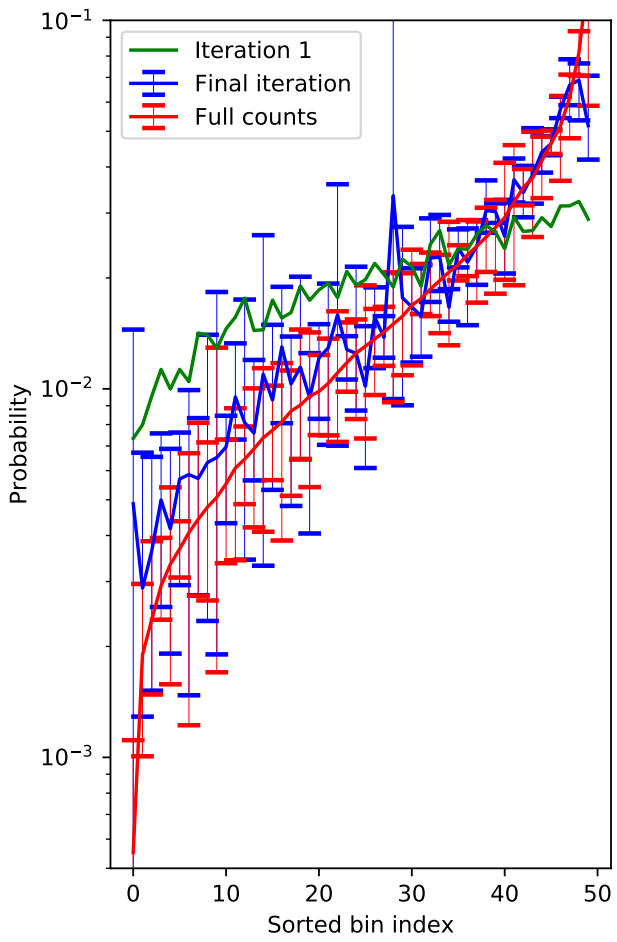


FIG. 4. Equilibrium distributions for the Trp-cage folding trajectory fragments, shown on a log and a linear scale. Shown are the distribution after one iteration (green), the distribution after the convergence criterion was met (blue), and counts in each bin from the original full trajectory (red), averaged across independent trials based on sub-dividing the full Shaw trajectory into five segments. Bins have been coarse-grained from 1000 initial bins for visualization. Error bars represent minima and maxima among five independent trials.

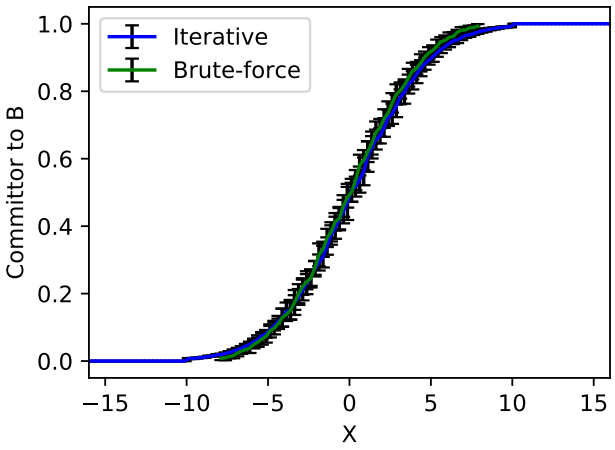


FIG. 5. Validation of iterative committor estimation in a one-dimensional model. Committor estimates are shown for the brute-force/naive calculation (green line) as well as the iterative approach (blue line) vs brute-force result, for the one-dimensional model of the potential in Eq. (3). Error bars indicate one standard deviation across 5 independent trials.

C. Committor calculation

As before, we first apply the committor estimator to the 1D double-well potential. With this simple 1D system we are able to directly compute the committor through a "brute-force" technique, where a number of trajectories are initialized from each point, and stopped when they reach a macrostate. Although the computational cost of this would be prohibitive for a more complex system, this is an unbiased reference.

Fig. 5 shows the result of the iterative committor estimator along with the brute-force reference for the 1D system. The committor profile follows the expected sigmoid shape between the two wells, with a value of 0.5 at the peak of the barrier. The iterative approach is thus validated as unbiased, by comparison to brute-force computation.

We also applied the iterative scheme to estimating the committor in for the Trp-cage system. Once again, brute-force reference committor values were obtained by following trajectory fragments originating in each bin until they reached a macrostate; the fraction that reaching state B before state A determined the committor. As seen in Fig. 6, the iterative committor estimation algorithm yields results for the Trp-cage data that track these brute-force estimates well, especially near the macrostates.

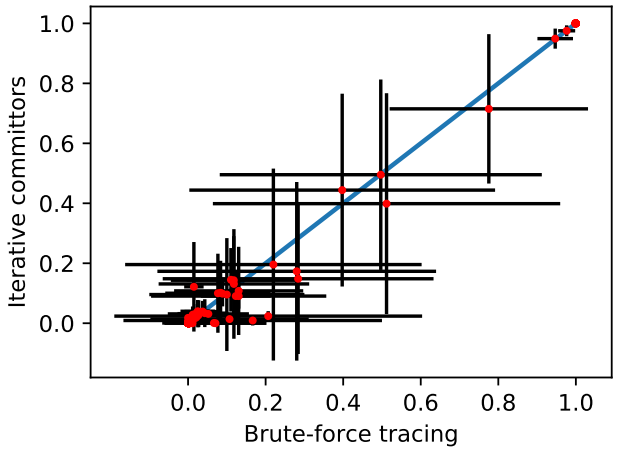


FIG. 6. Scatter plot of brute force committor values vs iterative committor values for Trp-cage. A line of slope 1 is shown in blue. Error bars represent a single standard deviation among independent trials based on sub-dividing the full Shaw trajectory into five segments.

IV. CONCLUSIONS

We have introduced algorithms that employ two well-known principles, iteration and stationarity, to estimate key observables from a trajectory or set of trajectories. In principle, the input trajectories need not follow any prescribed distribution. The procedures described do not rely on a Markov assumption. Although discrete bins are used for "accounting," the continuous trajectories embody all details of the landscape and dynamics which, in turn, are included implicitly in the analyses.

Subsequent work will show that the procedures described here are formally equivalent to 'power method' [28] evaluation of the stationary distribution of a non-standard transition matrix that accounts for trajectory dynamics over all available timescales, as pointed out to us by David Aristoff and Gideon Simpson.

ACKNOWLEDGMENTS

We appreciate helpful discussions with David Aristoff and Gideon Simpson. We thank DE Shaw Research for sharing the protein folding trajectory with us and the NIH for support through Grant GM115805.

[1] S. A. Hollingsworth and R. O. Dror, Molecular dynamics simulation for all, Neuron **99**, 1129 (2018).
[2] A. Grossfield and D. M. Zuckerman, Quantifying uncertainty and sampling quality in biomolecular simulations, Annual reports in computational chemistry **5**, 23 (2009).
[3] A. Grossfield, P. N. Patrone, D. R. Roe, A. J. Schultz, D. W. Siderius, and D. M. Zuckerman, Best practices for quantification of uncertainty and sampling quality in

molecular simulations [article v1. 0], Living journal of computational molecular science **1** (2018).
[4] D. M. Zuckerman, Equilibrium sampling in biomolecular simulations, Annual review of biophysics **40**, 41 (2011).
[5] L. Weng, S. L. Stott, and M. Toner, Exploring dynamics and structure of biomolecules, cryoprotectants, and water using molecular dynamics simulations: implications for biostabilization and biopreservation, Annual review of biomedical engineering **21**, 1 (2019).
[6] S. Kumar, J. M. Rosenberg, D. Bouzida, R. H. Swendsen, and P. A. Kollman, THE weighted histogram analysis method for free-energy calculations on biomolecules. I. The method, Journal of Computational Chemistry **13**, 1011 (1992).
[7] E. Darve, D. Rodríguez-Gómez, and A. Pohorille, Adaptive biasing force method for scalar and vector free energy calculations, The Journal of Chemical Physics **128**, 144120 (2008).
[8] Y. Sugita and Y. Okamoto, Replica-exchange molecular dynamics method for protein folding, Chemical Physics Letters **314**, 141 (1999).
[9] G. A. Huber and S. Kim, Weighted-ensemble Brownian dynamics simulations for protein association reactions, Biophysical Journal **70**, 97 (1996).
[10] C. Dellago, P. Bolhuis, and P. L. Geissler, Transition path sampling, Advances in chemical physics **123**, 1 (2002).
[11] T. S. van Erp, D. Moroni, and P. G. Bolhuis, A novel path sampling method for the calculation of rate constants, The Journal of Chemical Physics **118**, 7762 (2003).
[12] A. K. Faradjian and R. Elber, Computing time scales from reaction coordinates by milestoning, The Journal of Chemical Physics **120**, 10880 (2004).
[13] R. J. Allen, D. Frenkel, and P. R. ten Wolde, Simulating rare events in equilibrium or nonequilibrium stochastic systems, The Journal of Chemical Physics **124**, 24102 (2006).
[14] J. D. Chodera and F. Noé, Markov state models of biomolecular conformational dynamics, Current opinion in structural biology **25**, 135 (2014).
[15] G. R. Bowman, V. S. Pande, and F. Noé, *An introduction to Markov state models and their application to long timescale molecular simulation*, Vol. 797 (Springer Science & Business Media, 2013).
[16] D. M. Zuckerman and T. B. Woolf, Efficient dynamic importance sampling of rare events in one dimension, Phys. Rev. E **63**, 016702 (2000).
[17] H. Wan and V. A. Voelz, Adaptive Markov state model estimation using short reseeding trajectories, The Journal of Chemical Physics **152**, 24103 (2020).
[18] T. Hill, *Free Energy Transduction and Biochemical Cycle Kinetics*, Dover Books on Chemistry (Dover Publications, 2005).
[19] D. Bhatt, B. W. Zhang, and D. M. Zuckerman, Steady-state simulations using weighted ensemble path sampling, The Journal of chemical physics **133**, 014110 (2010).
[20] C. W. Gardiner, *Handbook of stochastic methods for physics, chemistry, and the natural sciences* (Springer-Verlag, Berlin New York, 1985).
[21] N. G. Van Kampen, *Stochastic processes in physics and chemistry* (Elsevier, Amsterdam Boston London, 2007).
[22] W. E. and E. Vanden-Eijnden, Towards a Theory of Transition Paths, Journal of Statistical Physics **123**, 503 (2006).
[23] J.-H. Prinz, M. Held, J. C. Smith, and F. No, Efficient computation, sensitivity, and error analysis of committor probabilities for complex dynamical processes, Multiscale Modeling & Simulation **9**, 545 (2011), https://doi.org/10.1137/100789191.
[24] A. M. Ferrenberg and R. H. Swendsen, New monte carlo technique for studying phase transitions, Physical review letters **61**, 2635 (1988).
[25] J. Copperman, D. Aristoff, D. E. Makarov, G. Simpson, and D. M. Zuckerman, Transient probability currents provide upper and lower bounds on non-equilibrium steady-state currents in the smoluchowski picture, The Journal of chemical physics **151**, 174108 (2019).
[26] D. Bhatt and D. M. Zuckerman, Beyond microscopic reversibility: Are observable nonequilibrium processes precisely reversible?, Journal of chemical theory and computation **7**, 2520 (2011).
[27] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw, How Fast-Folding Proteins Fold, Science **334**, 517 (2011).
[28] Power iteration, `https://en.wikipedia.org/wiki/Power_iteration`, accessed: 2020-06-14.

C Analyses of KATP Ion Channel MD Simulations

Vascular K_{ATP} channel structural dynamics reveal regulatory mechanism by Mg-nucleotides

Min Woo Sung[a], Zhongying Yang[a], Camden M. Driggers[a], Bruce L. Patton[a], Barmak Mostofian[b], John D. Russo[b], Daniel M. Zuckerman[b], and Show-Ling Shyng[a,1]

[a]Department of Chemical Physiology and Biochemistry, School of Medicine, Oregon Health and Science University, Portland, OR 97239; and [b]Department of Biomedical Engineering, School of Medicine, Oregon Health and Science University, Portland, OR 97239

Edited by Nieng Yan, Princeton University, Princeton, NJ, and approved August 30, 2021 (received for review May 21, 2021)

Vascular tone is dependent on smooth muscle K_{ATP} channels comprising pore-forming Kir6.1 and regulatory SUR2B subunits, in which mutations cause Cantú syndrome. Unique among K_{ATP} isoforms, they lack spontaneous activity and require Mg-nucleotides for activation. Structural mechanisms underlying these properties are unknown. Here, we determined cryogenic electron microscopy structures of vascular K_{ATP} channels bound to inhibitory ATP and glibenclamide, which differ informatively from similarly determined pancreatic K_{ATP} channel isoform (Kir6.2/SUR1). Unlike SUR1, SUR2B subunits adopt distinct rotational "propeller" and "quatrefoil" geometries surrounding their Kir6.1 core. The glutamate/aspartate-rich linker connecting the two halves of the SUR-ABC core is observed in a quatrefoil-like conformation. Molecular dynamics simulations reveal MgADP-dependent dynamic tripartite interactions between this linker, SUR2B, and Kir6.1. The structures captured implicate a progression of intermediate states between MgADP-free inactivated, and MgADP-bound activated conformations wherein the glutamate/aspartate-rich linker participates as mobile autoinhibitory domain, suggesting a conformational pathway toward K_{ATP} channel activation.

ATP-sensitive potassium channel | sulfonylurea receptor 2B | Kir6.1 | Cantú syndrome | ABC transporter

Dynamic regulation of K^+ channel gating is a primary point of control for processes governed by electrical excitability. ATP-sensitive potassium (K_{ATP}) channels, regulated by intracellular ATP to ADP ratios, transduce metabolic changes into electrical signals to govern many physiological processes (1). They are uniquely evolved hetero-octameric complexes comprising four pore-forming inwardly rectifying potassium channel subunits, Kir6.x, and four regulatory sulfonylurea receptors, SURx, nontransporting members of the ABCC subfamily of ABC transporters (2). Various Kir6.x/SURx combinations generate channel isoforms with distinct tissue distribution and function (3, 4). Kir6.2/SUR1 channels are expressed in pancreatic β cells and control glucose-stimulated insulin secretion. Kir6.2/SUR2A channels are the predominant isoform in myocardium, while Kir6.1/SUR2B channels are the major isoform found in vascular smooth muscle. SUR2A and 2B are two splice variants of *ABCC9* that differ in their C-terminal 42 amino acids (aa). In vascular smooth muscle, K_{ATP} activation leads to membrane hyperpolarization and vasodilation (5), while inhibition or deletion causes membrane depolarization, vasoconstriction, and hypertension (5–8). Mutations in the vascular K_{ATP} channel genes (*KCNJ8* and *ABCC9*) cause Cantú syndrome (9–11), a severe pleiotropic systemic hypotension disorder including hypertrichosis, osteochondrodysplasia, and cardiomegaly (12).

K_{ATP} channel gating by intracellular ATP and ADP involves allosteric sites on both subunits. ATP binding to Kir6.x inhibits the channel. SURx, through induced dimerization of the paired nucleotide binding domains (NBDs), requiring MgADP bound to NBD2 and MgATP bound to noncatalytic NBD1, activates the channel (1, 4, 13). Like all Kir channels, opening further requires PIP_2 bound to Kir6.x (14–16). Despite these commonalities, vascular Kir6.1/SUR2B K_{ATP} channels have distinct biophysical properties, nucleotide sensitivities, and pharmacology that differentiate them from other isoforms (17–19). First, vascular K_{ATP} channel unitary conductance is half that of Kir6.2-containing channels. Second, vascular channels lack spontaneous activity, only opening in the presence of NBD-dimerizing Mg-dinucleotides/trinucleotides; in contrast, pancreatic or cardiac channels containing Kir6.2 open spontaneously in the absence of ATP. Third, once activated, vascular K_{ATP} channels are relatively insensitive to ATP inhibition, requiring mM concentrations to observe an effect, while their pancreatic or cardiac counterparts are blocked by ATP at μM concentrations. Lastly, the antidiabetic sulfonylurea drug glibenclamide (Glib), which inhibits SUR1-containing pancreatic channels with high affinity, is ∼10-fold less potent toward the vascular and cardiac channels containing SUR2. Glib has been shown to reverse defects from gain-of-function Cantú mutations in mice (20). However, clinical application in Cantú patients is hindered by hypoglycemia from inhibition of pancreatic channels (21). Structural mechanisms underlying unique biophysical, physiological, and pharmacological properties among K_{ATP} channels are unknown.

Here, we report cryogenic electron microscopy (cryoEM) structures for the vascular K_{ATP} channel, Kir6.1/SUR2B, in the presence of ATP and Glib. The structures show conformations not previously seen in pancreatic K_{ATP} channels prepared under the

Significance

Vascular K_{ATP} channels formed by the potassium channel Kir6.1 and its regulatory protein SUR2B maintain blood pressure in the physiological range. Overactivity of the channel due to genetic mutations in either Kir6.1 or SUR2B causes severe cardiovascular pathologies known as Cantú syndrome. The cryogenic electron microscopy structures of the vascular K_{ATP} channel reported here show multiple, dynamically related conformations of the regulatory subunit SUR2B. Molecular dynamics simulations reveal the negatively charged ED-domain in SUR2B, a stretch of 15 glutamate (E) and aspartate (D) residues not previously resolved, play a key MgADP-dependent role in mediating interactions at the interface between the SUR2B and Kir6.1 subunits. Our findings provide a mechanistic understanding of how channel activity is regulated by intracellular MgADP.

Author contributions: M.W.S., D.M.Z., and S.-L.S. designed research; M.W.S., Z.Y., C.M.D., B.M., J.D.R., and S.-L.S. performed research; M.W.S., C.M.D., B.M., J.D.R., and S.-L.S. analyzed data; and M.W.S., B.L.P., D.M.Z., and S.-L.S. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1]To whom correspondence may be addressed. Email: shyngs@ohsu.edu.

This article contains supporting information online at http://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2109441118/-/DCSupplemental.

Published October 28, 2021.

same condition (22–24). First, unlike in Kir6.2, Kir6.1 cytoplasmic domains (CDs) were displaced from the membrane too far to interact with PIP_2 for channel opening. Second, unlike pancreatic channels, which have a predominant propeller-shaped conformation when bound to ATP and Glib (22, 24), vascular K_{ATP} channels held four distinct conformations, two resembling propellers and two quatrefoils, marked by varying degrees of rotation of SUR2B toward the core Kir6.1 tetramer. Importantly, a long segment of SUR not previously resolved in any K_{ATP} structures, linking NBD1 and transmembrane domain 2 (TMD2), was revealed within vascular K_{ATP} structures to mediate the cytosolic interface between SUR2B and Kir6.1. In particular, the linker's unique 15 glutamate/aspartate residues termed the ED-domain (25) established a nexus of interactions engaging SUR2B-NBD2 with Kir6.1 C-terminal domain (CTD). Molecular dynamics (MD) simulations showed MgADP binding to NBD2 was accompanied by substantial reconfiguration at this nexus, revealing the ED-domain provides a mobile autoinhibitory interaction that guards the transition of SUR2B from MgADP-free inactivated state to MgADP-bound activated state. Together, our findings point to a structural pathway through which SUR regulates Kir6 channel gating.

Results and Discussion

Structure Determination of Kir6.1/SUR2B K_{ATP} Channels with ATP and Glib. Vascular K_{ATP} channels were purified from COSm6 cells coexpressing rat Kir6.1 and SUR2B (97.6 and 97.2% sequence identity to human Kir6.1 and SUR2B, respectively). COSm6 cells lack endogenous K_{ATP} channels and have been used extensively as a heterologous expression system for K_{ATP} channel structure–function studies (16, 26). Channels were solubilized in digitonin, purified via an SUR2B epitope tag, and imaged in the presence of 1 mM ATP (no Mg^{2+}) and 10 μM Glib on graphene oxide–coated grids, as described in *Materials and Methods*.

In vascular K_{ATP} channel structures as in pancreatic channels, we found SUR2B anchored to Kir6.1 via interactions mediated by transmembrane helix 1 (TM1) of SUR2B-TMD0 and Kir6.1-TM1 (Fig. 1). However, conformational deviations from fourfold symmetry of the SUR2B were noted in two-dimensional class averages (*SI Appendix*, Fig. S1). To obtain clear SUR2B maps, we implemented symmetry expansion and extensive focused three-dimensional (3D) classification of Kir6.1 tetramer with individual SUR2B (*Materials and Methods*) (*SI Appendix*, Fig. S2), which isolated four 3D classes having identical Kir6.1 tetramer structures but different SUR2B orientations (Fig. 1 and *SI Appendix*, Fig. S2). When symmetrized, two of the 3D classes, designated P1 and P2, resembled the pancreatic channel propeller conformations previously reported (22, 24). The other two, designated Q1 and Q2, resembled the "quatrefoil conformation" reported for human pancreatic K_{ATP} in which the SUR1 NBDs are dimerized (27). Further refinement yielded cryoEM maps with overall resolutions of 3.4, 4.2, 4.0, and 4.2 Å for the P1, P2, Q1, and Q2 conformations, respectively (*SI Appendix*, Fig. S3). The maps were sufficient to build a full atomic model for all of Kir6.1 minus the disordered C terminus (365 to 424) for all

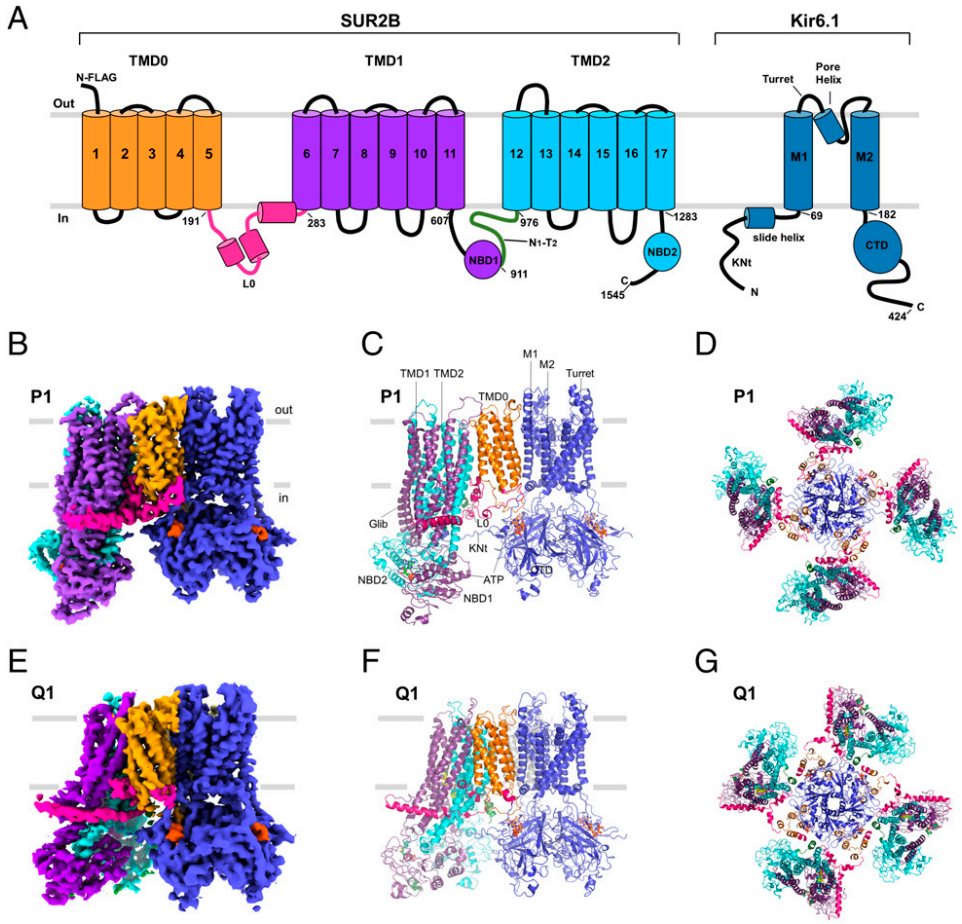


Fig. 1. Structures of the vascular K_{ATP} channel in the presence of ATP and Glib. (*A*) Schematics of SUR2B and Kir6.1 domain organization. (*B*) CryoEM density map of $(Kir6.1)_4SUR2B$ P1, side view. (*C*) Structural model of $(Kir6.1)_4SUR2B$ P1, side view. (*D*) Fourfold symmetrized structure model of P1 viewed from the top (i.e., extracellular side). (*E*) CryoEM density map of $(Kir6.1)_4SUR2B$ Q1, side view. (*F*) Structural model of $(Kir6.1)_4SUR2B$ Q1, side view. (*G*) Fourfold symmetrized structure model of Q1 viewed from the top.

conformations, with clear sidechain densities for most residues (*SI Appendix*, Fig. S3*d*) and also models for most of SUR2B (see *Materials and Methods* for details). Densities for ATP, Glib, and some lipids were well resolved (*SI Appendix*, Fig. S3*d*). Significantly, the Q1 conformation included definitive densities in SUR2B for L0, which is the linker connecting TMD0 and the ABC core, and also the N1-T2 linker, which connects NBD1 to TMD2; neither is resolved in the human pancreatic K_{ATP} quatrefoil structure previously determined (27). The P- and Q-like conformations differ by a major rotation of the SUR2B-ABC core toward the Kir6.1 tetramer, clockwise when viewed from the extracellular side (Fig. 1 *D* and *G*). P1 and Q1 were the dominant particle populations within the P- and Q-like forms, respectively, differing from P2 and Q2 by degree of rotation and specific features. We first focus on structural differences between P1 and Q1, which provided the highest resolutions.

Structural Correlates of Kir6.1 Functional Divergence. Although the Kir6.1 tetramer was similarly configured in all P and Q conformations for SUR2B, it included several features distinct from Kir6.2 in our published pancreatic channel structure determined under similar conditions with ATP and Glib (Protein Data Bank [PDB]: 6BAA). The Kir6.1 channel CD was extended intracellularly away from the membrane by ~5.8 Å, and simultaneously counterclockwise rotated (viewed from the extracellular face) by 12.4° (Fig. 2*A*). The Kir6.x CD is thus corkscrewed away from the membrane in Kir6.1/SUR2B, compared to Kir6.2/SUR1. Constrictions in the two cytoplasmic gates, namely the helix bundle crossing (F178) and the G-loop (G304, I305), indicate a closed Kir6.1 channel pore, similar to Kir6.2 under the same condition (*SI Appendix*, Fig. S4). However, the distance between the helix bundle crossing gate and the G-loop gate is significantly larger in Kir6.1 due to the untethered CD.

In K^+ channels, variations in the turret region surrounding the pore entryway have been shown to affect selectivity filter stability and ion conduction (28). Compared to Kir6.2, the turret of Kir6.1 contains an extra 11 aa (^{102}YAYMEKGITEK^{112}). We found this sequence formed a helix and loop structure that extends the turret further out into the extracellular space (Fig. 2 *B* and *C*). Functional studies using Kir6.1–Kir6.2 chimeras previously identified residues in Kir6.1 thought to impart its smaller unitary conductance, specifically M148 and N123-V124-R125 (29). M148 in Kir6.1 (replacing Kir6.2-V138) was proposed to reduce pore entrance diameter, while N123 in Kir6.1 (replacing Kir6.2-S113) was hypothesized to impact an intersubunit salt bridge between R146 and E150, which in other Kir channels is formed by corresponding residues and critical for channel conduction (29, 30). However, our structure found M148 facing the pore helix (Fig. 2*C*) rather than the entrance and that no significant difference exists in the adjacent pore diameters between Kir6.1 and Kir6.2, nor in their intersubunit salt bridges. Interestingly, N123-V124-R125 of Kir6.1 is located

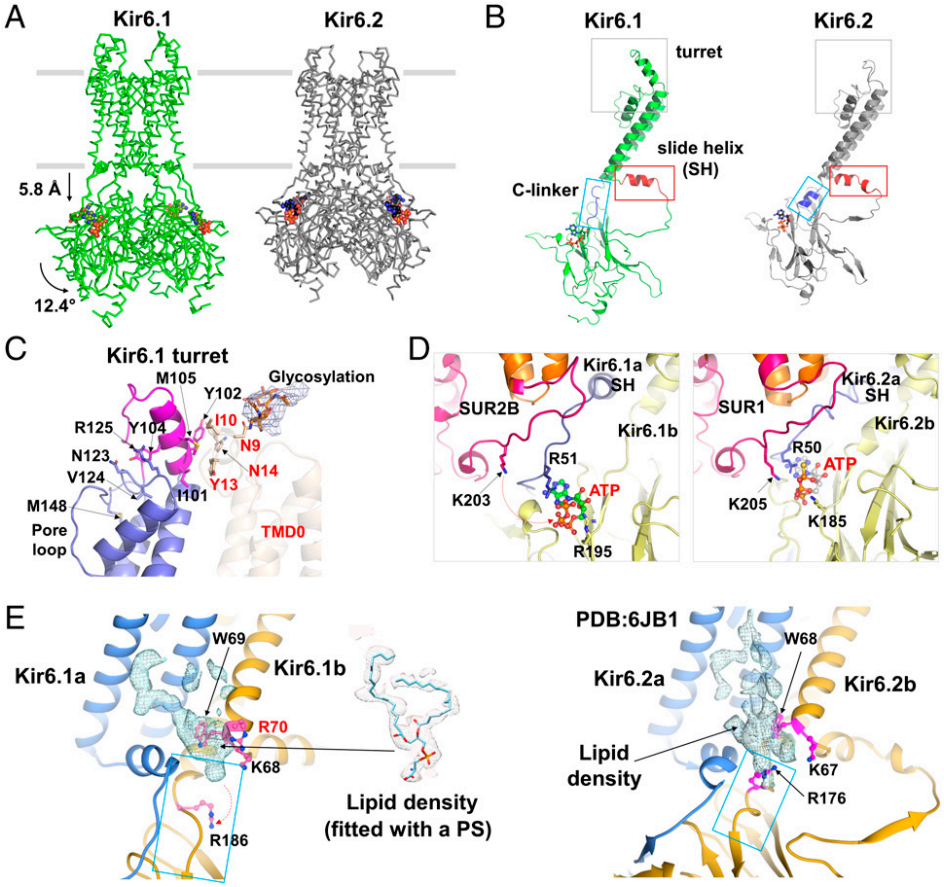


Fig. 2. Structural comparison between Kir6.1 and Kir6.2. (*A*) Comparison of Kir6.1 and Kir6.2 showing translational and rotational differences in the CD. (*B*) Major structural differences in the turret (gray box), SH (red box), and C-linker (cyan box) between Kir6.1 and Kir6.2. (*C*) Close-up view of the turret showing insertion of an additional 11 aa (magenta) in Kir6.1, which appears to be in position to interact with TMD0 of SUR2B (residues labeled in red). The density corresponding to glycosylation of N9 is fitted with two N-acetylglucosamines. (*D*) Close-up view of the Kir6.1 ATP binding site in comparison to Kir6.2 ATP binding site. (*E*) Close-up view of the PIP_2 binding site in Kir6.1 in comparison to that in Kir6.2. R70 (P69 in Kir6.2), which could interact with negatively charged phospholipid, is highlighted in red label.

between the turret extension and the pore loop and interacts with Y104 in the turret extension (Fig. 2*C*). Future mutagenesis studies will determine the contributions of these interactions to Kir6.1 channel conductance.

We next assessed structural differences between Kir6.1 and Kir6.2 in two elements intimately associated with activity at ATP and PIP_2 binding sites: the N-terminal amphipathic helix known as the slide helix (SH), and the connecting strand between TM2 and the CTD called the C-linker (Fig. 2 *B*, *D*, and *E*). In our Kir6.2 structure (22), SH is bent halfway at the D58 position resembling a 3_{10} helix (31). In contrast, SH in Kir6.1 formed a continuous helix extending toward the neighboring Kir6.1, thus compressing the PIP_2 binding pocket. In Kir6.2, the C-linker forms a helix that tethers the CTD close to the membrane, which positions critical PIP_2-binding residues such as R176 for PIP_2 interaction. Rather different, the C-linker in Kir6.1 unwound into an unstructured loop stretching toward the cytoplasm, which deflected R186 (corresponding to Kir6.2-R176) away from the PIP_2 binding site (Fig. 2 *B* and *E*). A previous study by Quinne et al. has shown that Kir6.1 binds and is modulated by PIP_2 (32). In our structure, we observed a strong nonprotein cryoEM density in the PIP_2 binding pocket. However, this density is better fitted with a phosphatidylserine (PS) than a PIP_2 (Fig. 2*E*). Recent MD simulations of Kir2.2 in membranes containing mixed phospholipids showed that PS can also occupy the PIP_2 pocket (33). Because no exogenous PIP_2 was added to our protein sample, the simplest interpretation of the structural data is that the more abundant PS resides in the Kir6.1 PIP_2 binding pocket. However, a possibility remains that the density includes endogenous PIP_2 or other phospholipids copurified with the channel. CryoEM densities matching ATP were clearly resolved in Kir6.1 tetramers, at sites located between the N-terminal domains and CTDs of adjacent Kir6.1 subunits, matching sites in Kir6.2/SUR1 channels. However, unlike for Kir6.2, ATP had fewer close residue interactions in Kir6.1 due to displacement of the Kir6.1-CD. In particular in pancreatic channels, SUR1-K205 (in L0) directly participates in binding ATP at its inhibitory site (23, 27, 34), while the corresponding vascular channel residue SUR2B-K203 was displaced from potential ATP binding (Fig. 2*D*). Thus, the constellation of ATP interactions was sparser and hence likely weaker when Kir6.1-CD was displaced from the membrane.

Taken together, the translocation of the Kir6.1-CD away from the membrane compromised binding of both ATP and PIP_2. This correlates well with the basal inactivity and reduced ATP sensitivity of the vascular K_{ATP} channel compared to Kir6.2 channels (35, 36). Rotation and downward movement of the Kir6.2-CD have been detected in minor subclasses of ATP- and Glib-bound pancreatic Kir6.2/SUR1 structures (23, 24), indicating similar dynamics occur but less stably persist. Moreover, translation and/or rotation of the CD is observed in Kir2, Kir3, and bacterial Kir channels (37–40), and recent cryoEM studies of Kir3 channels found that increased PIP_2 concentrations shift particle distributions toward those having CD tethered close to the PIP_2 membrane sites (41). Thus, a common model of K_{ATP} channel activity involves channel opening dependent on PIP_2 binding, which in turn depends on engagement by the Kir6.x-CD modulated by its vertical translocation/rotation. Accordingly, in vascular Kir6.1 channels, a greater energy barrier is involved in rotating the CD upward to interact with PIP_2 than in pancreatic channels whose Kir6.2-CD is more stably tethered to the membrane. This explains why Kir6.2-containing pancreatic channels are spontaneously active, while Kir6.1-containing vascular channels are not. By extension, vascular channel activation by Mg-nucleotides likely involves SUR2B-controlled upward movement of the Kir6.1-CD. It has been shown that Kir6.1 binds PIP_2 with higher affinity than Kir6.2 in biochemical assays and that once activated by Mg-nucleotides, vascular K_{ATP} channels are highly stable and more resistant to PIP_2 depletion by polylysine than pancreatic channels (32). In our Kir6.1 structure, the Kir6.1-R70 sidechain is directed toward the lipid density in the PIP_2 binding pocket (Fig. 2*E*). Interestingly, in Kir6.2 the corresponding residue is a proline (P69). It is possible that this sequence variation may contribute to the higher PIP_2 affinity and stability of open vascular channels, but future studies are necessary to investigate this. Higher-PIP_2 affinity also accounts for long-standing results showing activated vascular K_{ATP} channels are much less sensitive to ATP inhibition, as increased PIP_2 interaction reduces ATP inhibition in K_{ATP} channels (16).

SUR2B Dynamics. Focused 3D classification resolved four distinct conformations, P1, P2, Q1, and Q2, showing variable SUR2B orientations (*SI Appendix*, Fig. S5 and Movie S1). P conformations differed from Q conformations by a large rotation of the ABC core of SUR2B relative to the Kir6.1 tetramer (~41° between P1 and Q1, about the axis defined by N447 in TMD1 and N69 in TMD0, respectively, compared to 63° rotational difference between the propeller and quatrefoil conformations in human pancreatic NBD-dimerized channels measured from the equivalent residues). Within P and Q, P1 and Q1 particles predominated over P2 and Q2. Transitions from P1 to P2 and Q1 to Q2 involved alternative rotation stops: P1's ABC core was 10° further away from Kir6.1 than P2's, while Q1's ABC core was 8° closer to Kir6.1 than in Q2. In short, Q1 was the tightest quatrefoil and P1 the most extended propeller. 3D variability analysis in CryoSPARC (*SI Appendix*, Fig. S6*a*) indicated SUR2B subunits moved independently between P- and Q-like positions (Movie S2). Further multibody refinement in RELION3 revealed greater heterogeneity within Q1 conformations than in P1, indicating wider dynamic range (*SI Appendix*, Fig. S6*b* and Movie S3).

Accompanying rotation, the SUR2B-ABC core also tilts away from Kir6.1. Tilting elevated the ABC core TMD in the Q conformations relative to P conformations (by 2.6 Å from P1 to Q1, measured at SUR2B-Y370; *SI Appendix*, Fig. S5*b*). Between the pancreatic K_{ATP} propeller and quatrefoil forms (NBDs dimerized), the entire ABC-TMDs elevate ~3 Å without tilting (27). Tilt in our Q conformations may represent a partial transposition to be completed upon NBD dimerization. In the NBDs-dimerized pancreatic K_{ATP} quatrefoil is the dominant class. Here, Q conformations were less common than P conformations among vascular K_{ATP} channel structures in which the NBDs remain separated (*SI Appendix*, Fig. S2 and Table S1). Probabilities of SUR adopting P- or Q-like conformations therefore correlate with NBD dimerization state, although both occur regardless.

Rotation of SUR2B between P to Q conformations incorporated significant local structural changes. Extracellular contacts between transmembrane bundle 1 (TMB1) and TMD0 restructured both protein–protein and protein–lipid interactions (*SI Appendix*, Fig. S7). Hydrophobic and electrostatic interactions in P1 are lost in Q1, including T338, L339, and F344 in the TM6-TM7 loop of TMB1, with L165 and R166 in TM5 of TMD0. Moreover, a phosphatidylethanolamine molecule moved from between TM2 and TM7 in P1 to between TM3 and TM16 in Q1, likely stabilizing TMD0 and TMB1 interactions. Also noteworthy, in the pancreatic channel structure, SUR1 has an additional hydrophobic sequence (^{340}FLGVYFV^{346}), which anchors the TM6-TM7 loop to TMD0 (*SI Appendix*, Fig. S7*d* (23, 34). Absence of this sequence in SUR2B may impart flexibility that enables SUR2B to swing into Q conformations not observed in SUR1 when ATP and Glib are bound.

The SUR2B-L0 Linker and the Glib Binding Pocket. Transition between P and Q conformations remodeled cytoplasmic structural elements

including L0, the N1-T2 linker, and Kir6.1 N-terminus (Kir6.1Nt), unexpectedly affecting interactions between SUR2B and Kir6.1. In SURx, L0 connects TMD0 to the ABC core and is crucial to K_{ATP} gating (42–45). In SUR2B, we obtained two distinct L0 conformations, corresponding to P and Q conformers. In SUR2B-P1, we observed continuous cryoEM density of L0 (Fig. 3*A*). The well-defined N-terminal portion lacked secondary structure. The central portion formed an amphipathic helix, inserted between TMD0 and TMB1. A C-terminal helix then extended along the periphery of TMB1, paralleling the membrane. In contrast, L0 of SUR2B-Q1 comprised a destabilized N-terminal portion in which aa 197 to 213 were unresolved; a central amphipathic helix shifted into the cytoplasm, and a C-terminal helix pulled away from the Kir6.1 core (Fig. 3 and Movie S4). In addition, lipids around the amphipathic L0 helix in P1 conformation were replaced by the descended amphipathic helix in Q1 (Fig. 3*A*). Together, restructuring resulted in a marked decrease in contact area between SUR2B's TMD0 (M1-R256, including lipids) and the adjacent ABC core (A257-V1541 in P1; A257-A1543 in Q1), from 2,106.2 Å^2 in P1 to 1,208.0 Å^2 in Q1, which lowered the estimated free energy of formation at this interface from −43.0 kcal/mol in P1 to −18.4 kcal/mol in Q1 [calculated using Protein Data Bank in Europe (Proteins, Interfaces, Structures and Assemblies), abbreviated as PDBePISA (46)]. As noted, the P1 conformer predominates among the ATP- and Glib-bound vascular K_{ATP} particles we observed. The simplest interpretation is that this interface is a principal determinant in maintaining SUR2B in P1 to a greater extent than Q1. The Q conformers would thus represent a divergent state in which a principal interface stabilizing a closed channel is compromised. It is worth noting that L0 in SUR1 (aa 192 to 262) is unresolved in the quatrefoil structure for the human pancreatic K_{ATP} channel, in which NBDs are dimerized (PDB: 6C3O) (27). Therefore, the Q-like structures presented here may represent intermediary states that offer a glimpse into the conformational transitions of L0 that anticipate NBDs dimerization. The striking rearrangement of L0 likely results from the torque generated by rotation of the ABC core and further permits the channel to undergo the conformational changes for gating.

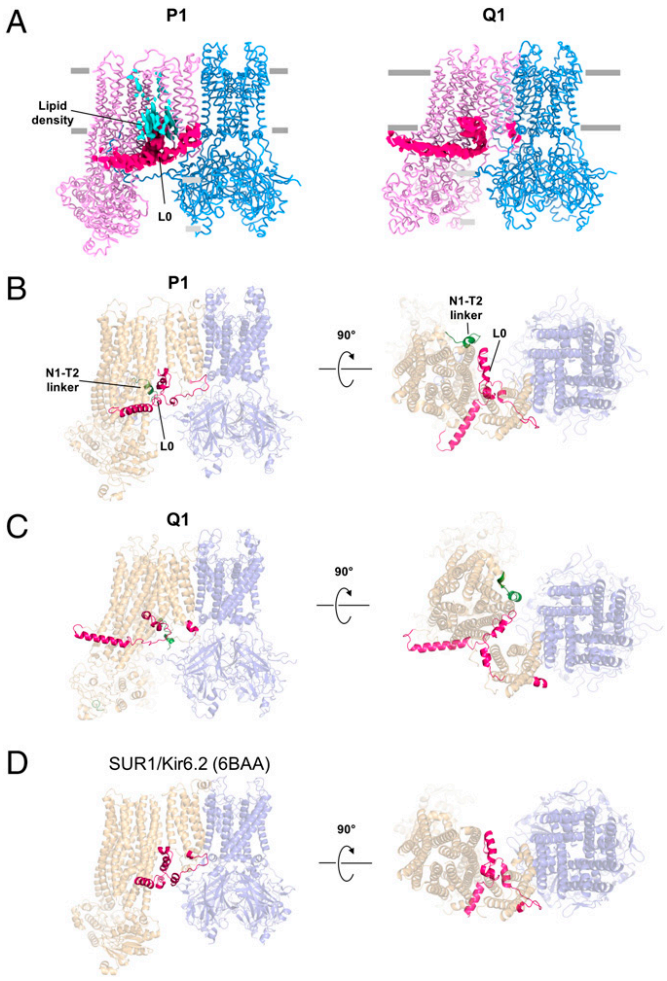


Fig. 3. SUR2B-L0 undergoes structural remodeling from P1 to Q1 conformations. (*A*) Comparison of the L0 cryoEM density (hot pink) in P1 and Q1 conformations. Lipid density seen in P1 but absent in Q1 is shown in cyan. (*B*) Structure of $(Kir6.1)_4$SUR2B in P1 conformation showing L0 (red) viewed from the side (*Left*) and from the cytoplasmic side near the membrane (*Right*). The N1-T2 linker visible in these views is shown in green. (*C*) Structure of $(Kir6.1)_4$SUR2B in Q1 conformation viewed from the side and the cytoplasm. (*D*) Structure of $(Kir6.2)_4$SUR1 (PDB: 6BAA) bound to Glib and ATP for comparison.

Vascular K_{ATP} channels are inhibited by Glib but are ∼10- to 50-fold less sensitive than pancreatic K_{ATP} channels (17, 47). Glib cryoEM density was well resolved in both P1 and Q1 conformations of the vascular K_{ATP} structure, where it bound within the same pocket of SUR2B (Fig. 4) as in SUR1 (24, 26). Also similar to pancreatic Kir6.2/SUR1 channels, cryoEM density of the distal KNt of Kir6.1 lay within the cleft between the two halves of the ABC core and immediately adjacent the Glib binding pocket (26). The structure model of the Glib binding site in P1 shows key interactions are largely conserved between SUR1 and SUR2B (Fig. 4*C*). However, the binding pose of Glib in the Q1 conformation is compressed compared to that in P1, with the Y1205 sidechain moved upward, which requires the 1-chloro-4-methoxy-benzene group to also move to avoid W423 in a neighboring helix (Fig. 4*C*). Also, an electrostatic interaction between chloride in Glib and nitrogen of R304 is eliminated in Q1. Worth noting, the SUR2B-Y1205 equivalent residue in SUR1 is S1238, and substitution of serine by tyrosine at this position has been shown to partly underlie Glib's lower-affinity inhibition of SUR2 channels (48). Of particular interest, substitution of S1238 to Y in SUR1 converts the Glib inhibition of pancreatic Kir6.2/SUR1 channels from nearly irreversible to readily reversible similar to SUR2-containing channels (49, 50), suggesting the S1238Y mutation may affect Glib off rate. This may arise through steric hindrance from the flexible tyrosine sidechain, as observed in the Q1 conformation.

The P1 to Q1 translocation was accompanied by more substantial change to the opposite side of the Glib binding pocket at F215, T227, and Y228 of the L0 linker (Fig. 4*C*). Previous studies of L0 of SUR1 have shown that Glib binding indirectly involves Y230 and W232 (Y228 and W230 in SUR2B), which stabilize the TM helices lining the Glib binding pocket (22); mutation of these residues to alanine reduces sensitivity to Glib (49, 50). In SUR2B P1 conformation, we found the hydrophobic Y228 and W230 sidechains, as well as F215 in the lower part of L0, similarly stabilized the TM helices along the Glib binding pocket (Fig. 4*C*), as occurs in SUR1. Specifically, F215 lay buried in a hydrophobic cavity formed by W230 from L0 and Y371, F1207, and L1206 from TMB1. However, in Q1, L0 was significantly remodeled at the interface with TMB1. In particular, a loop segment including P218-Y228 seen in P1 is raised and transformed into a helix in Q1. This helical element newly filled the hydrophobic cavity between TMD0 and TMB1, otherwise occupied by lipids in P1 (Fig. 3*A*). As further consequence in Q1, Y228 and F215 in L0 are displaced from the cavity, and Y371 and T227 occupy the space vacated by the sidechain of Y228. The movement of Y228 out of the cavity eliminates hydrophobic packing between L0 and the TM helices lining the Glib binding pocket, thus disrupting the integrity of the pocket

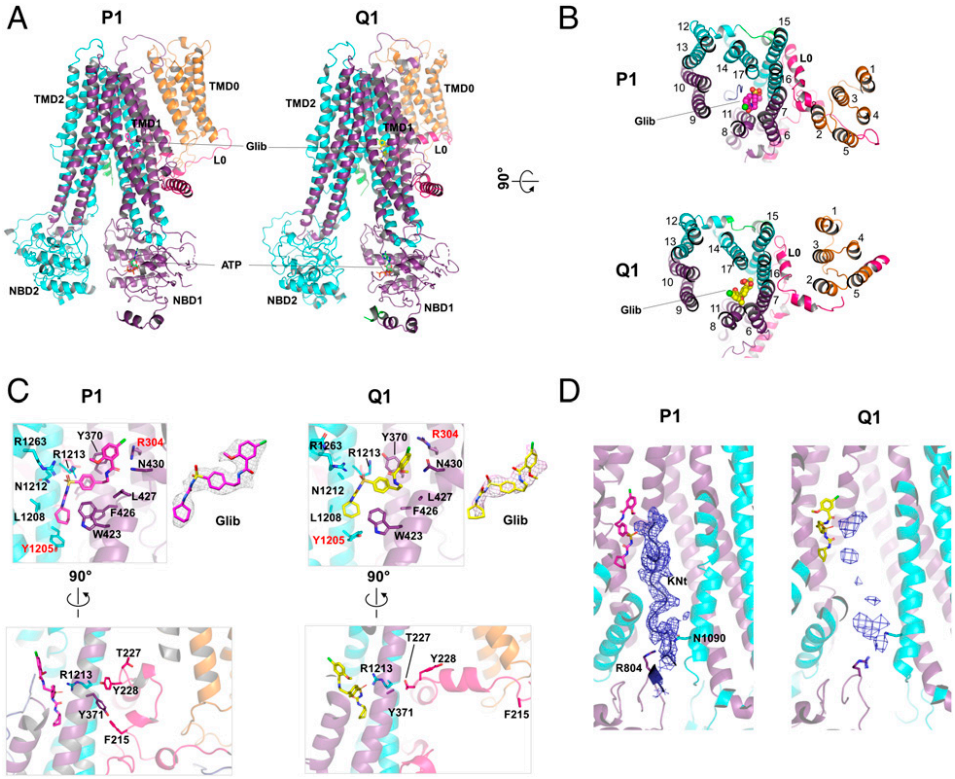


Fig. 4. Comparison of the SUR2B Glib binding pocket in P1 and Q1 conformations. (*A* and *B*) Overview from the side and the top, respectively. (*C*) Close-up view of the Glib binding site in P1 and Q1 conformations. Note the slightly different pose of Glib. Two key residues different in SUR2B and SUR1 are highlighted in red (R304 and Y1205). CryoEM density with the Glib structure model fitted into it is shown to the right of the binding site figure. *Bottom*: a different view of the Glib binding site highlighting the changes in L0 residues that impact the Glib binding site. (*D*) CryoEM density of the KNt in P1 and Q1 conformations. The KNt cryoEM density is stronger in P1 and allows modeling with a polyalanine chain. Note two residues in the NBD1 (R804) and TMD2 (N1090) sandwich the KNt to stabilize it in the central cavity between the two TMBs of SUR2B.

in similar fashion to the Y230A mutational effect in SUR1 (51). Lastly, the density of Kir6.1Nt in the ABC core central cleft also differed between P1 and Q1 (Fig. 4*D*). In P1, a strong continuous density of KNt was present, braced by R804 and N1090 of SUR2B, a pair of residues guarding entry to the cleft. The KNt density in Q1 was considerably weaker and discontinuous, indicating a more-labile conformation that may contribute to weak Glib binding at its adjacent pocket (26, 52). In summary, as the SUR2B-ABC core changes from P conformation to Q, L0 and Kir6.1Nt undergo remodeling that affects the Glib binding pocket.

The N1-T2 Linker. In all published pancreatic K_{ATP} channel structures, the critical N1-T2 linker of SUR1 has remained unresolved (22–24, 26, 27, 34, 53), suggesting dynamic instability. In the density map of our vascular Kir6.1/SUR2B channel from the P1 particle set, the C-terminal end of the N1-T2 linker was sufficiently resolved (Fig. 5*A*), and we were able to build a polyalanine helical structure into the density map (residues 961 to 976). Density for the rest of N1-T2 (residues 911 to 960) remained largely unresolved in P1. However, the density for the entire linker was apparent in the map for our vascular K_{ATP} channel Q1 structure (Fig. 5*B*), although resolution of residues 911 to 960 was insufficient for modeling. Specifically, the linker extended from NBD1 through the space between the two NBDs, then continued through the gap between the outer surface of NBD2 and the adjacent CTD of Kir6.1, before connecting to TMD2 (Fig. 5*B* and *SI Appendix*, Fig. S8). The location of the SUR2B N1-T2 linker contrasts sharply with locations of corresponding linkers in other ABCC proteins, including the Cl^- channel CFTR and the yeast cadmium transporter Ycf1p. In CFTR, the N1-T2 linker equivalent is known as the R domain, which is phosphorylated by PKA to allow CFTR gating by Mg-nucleotides. In the unphosphorylated CFTR structure, the R domain is wedged in the cleft between the two halves of the ABC core, preventing NBD dimerization (54). In the phosphorylated CFTR structure, the R domain relocates to the outer surface of NBD1 (*SI Appendix*, Fig. S8), which allows NBD dimerization, hence CFTR gating by Mg-nucleotides (55, 56). In the Ycf1p structure, the N1-T2 linker is found at the outer surface of NBD1 similar to phosphorylated CFTR (57) even though the NBDs are separate. The peculiar location of the SUR2B N1-T2 linker suggests the linker has adopted a separate role in regulating functional coupling between SUR2B and Kir6.1.

In SUR2, the N1-T2 linker includes at its C-terminal end a stretch of 15 aa consisting exclusively of negative-charged glutamate and aspartate designated the ED-domain (947 to 961) (*SI Appendix*, Fig. S8), which is unique among all ABCC proteins. Previous mutational studies have implicated the ED-domain in transducing MgADP binding in SUR2A to opening of Kir6.2 (25). Disruption of the ED-domain prevented the normal activation response to MgADP and to pinacidil, a potassium channel opener. In the Q1 structure, the density corresponding to the ED-domain is sandwiched between NBD2 and Kir6.1-CTD (Fig. 5) and surrounded by positively charged residues from Kir6.1Nt, Kir6.1-CTD, and NBD2 of SUR2B (Fig. 5*B*), an array of partners for electrostatic interactions. To understand the potential molecular interactions and their functional relevance, we employed MD simulations of the $(Kir6.1)_4SUR2B$ Q1 structure.

MD Simulations Reveal MgADP-Dependent Dynamic Interactions between the ED-Domain, NBD2, and Kir6.1-CTD. To assess conformational dynamics of the ED-domain and its interacting

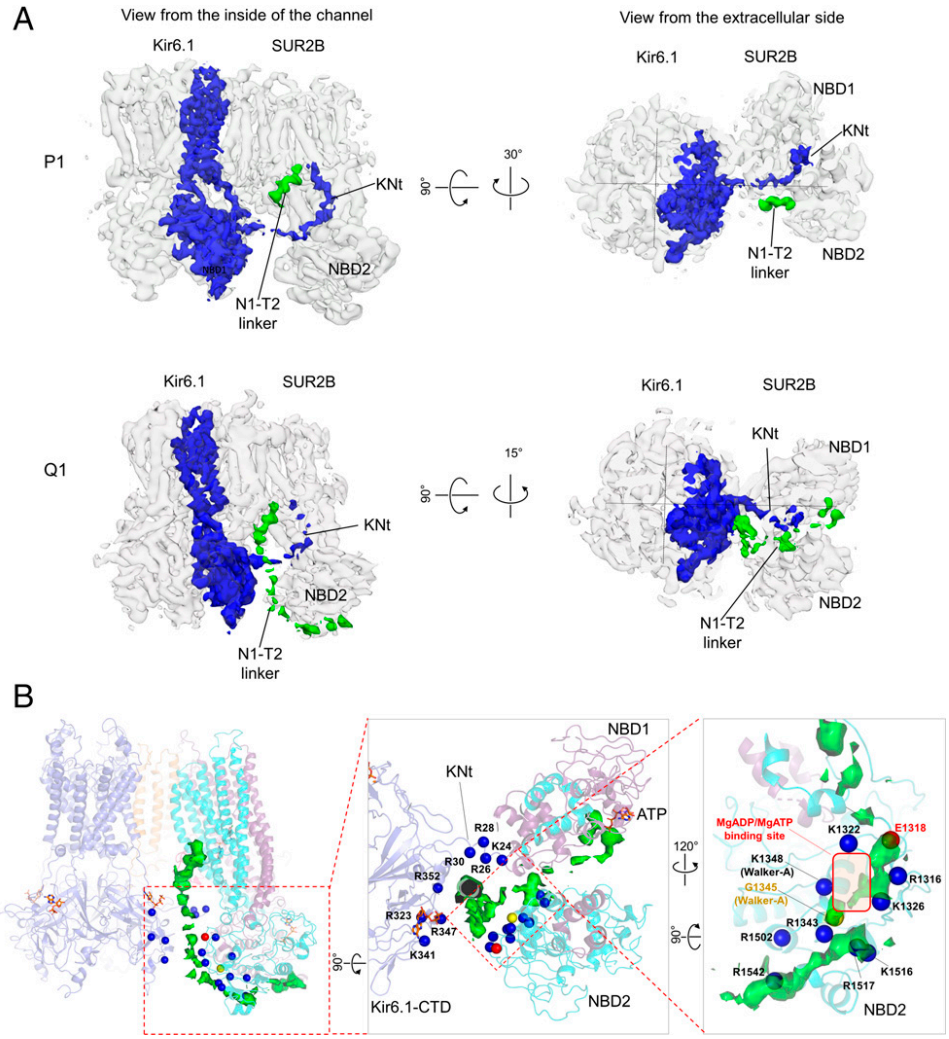


Fig. 5. Comparison of cryoEM densities of Kir6.1 N terminus and SUR2B N1-T2 linker in P1 and Q1 conformations. (*A*) Overall cryoEM density of $(Kir6.1)_4$-SUR2B in gray with density of one Kir6.1 and its N terminus (KNt) highlighted in blue and density of the SUR2B N1-T2 linker highlighted in green. (*B*) Close-up view of the N1-T2 linker density in $(Kir6.1)_4$SUR2B structure. Blue spheres are positively charged residues near the ED-domain. G1345 in the NBD2 Walker A motif and E1318 in the A-loop of NBD2 (^{1315}VRYEN^{1319}) are shown as reference points.

partners and how they may be dependent on the nucleotide binding status at the two NBDs, we performed simulations under two conditions. In one, ATP is bound to Kir6.1 and NBD1 of SUR2B, as present in our cryoEM structure. In the second condition, Mg^{2+} is included with ATP bound at NBD1, and MgADP is docked into NBD2 (Fig. 6*A*). In both conditions, Glib was omitted from the structure to allow the SUR2B TMDs to be free of constraint during simulations. To assess reliability, three independent 1-μs simulations for each condition were carried out (*SI Appendix*, Fig. S9*a*). As with many biomolecular simulations, ours do not exhibit true equilibrium-like repeated fluctuations about mean values (58), although the combined 6 μs permitted structural inferences (Fig. 6 *C* and *E*). The root-mean-square fluctuation (RMSF) analyses showed high degrees of fluctuations of NBD1, the N1-T2 linker, and NBD2 (*SI Appendix*, Fig. S9*b*), consistent with overall lower resolutions of these domains in cryoEM maps (*SI Appendix*, Fig. S3*b*). However, particular interactions between the ED-domain and NBD2 depended on whether NBD2 was occupied by MgADP, and in turn those ED-domain–NBD2 interactions controlled direct interaction of NBD2 with Kir6.1-CTD.

During simulations, the ED-domain exchanged interactions between surrounding positively charged residues from Kir6.1Nt, Kir6.1-CTD, and NBD2 (Fig. 5*B* and Movies S5 and S6). When MgADP was absent at NBD2, the first one-half of the ED-domain (947 to 953) was most frequently in contact with SUR2B-NBD2 Walker A K1348; this was infrequent with MgADP bound at NBD2. To quantify MgADP dependence of the ED-domain–Walker A interaction, we measured the minimum distances between the ED-domain residues 947 to 953 and K1348 throughout simulations, comparing results with and without MgADP at NBD2 (Fig. 6 *B* and *C*). In the absence of MgADP, sidechain oxygens from ED residues were frequently within 4 Å of the sidechain nitrogen of K1348, supporting a salt bridge or strong electrostatic interaction (59). In contrast, in the presence of MgADP, ED residues remained too distant from K1348 for direct bonding. Moreover, in one MgADP simulation run in which MgADP dissociated (Fig. 6*C* red trace, ~300 ns), the ED-domain subsequently moved to within 4 Å of K1348, the distance frequently observed in simulations lacking MgADP (Fig. 6*C*). The difference in ED–K1348 interactions between simulations is similarly evidenced by tracking the center of mass for C-α of ED residues 947 to 953 and the C-α of K1348 (Fig. 6*D*).

NBD2 also frequently formed close contacts with Kir6.1-CTD in the absence of MgADP but not when NBD2 included MgADP (Movies S5 and S6). With no MgADP, a loop

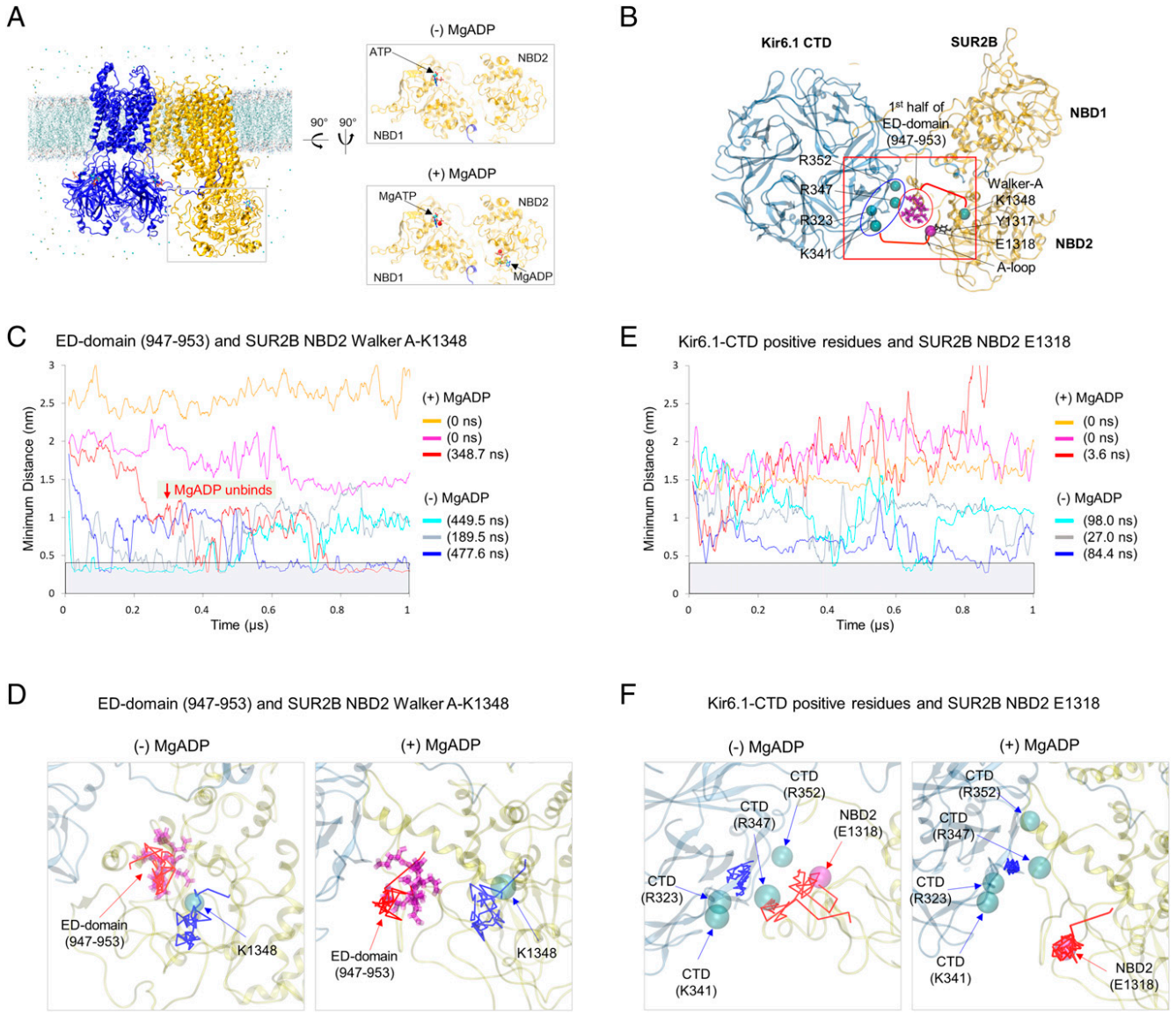


Fig. 6. MD simulations of the ED-domain dynamics in relation to SUR2B-NBD2 and Kir6.1-CTD. (*A*) MD simulation starting model (Q1) and conditions. In (−)MgADP condition, only ATP is present in NBD1. In (+)MgADP condition, MgADP is bound in NBD2 and MgATP is bound in NBD1. (*B*) Structural model marking residues of interest for distance analysis during MD simulations. These include the ED-domain residues (magenta sticks in red oval) and the Walker A K1348 (cyan sphere) in SUR2B NBD2 and R323, K341, R347, and R369 (cyan spheres in blue oval) in Kir6.1 CTD and E1318 in SUR2B NBD2. The A-loop containing Y1317, which coordinates adenine ring binding of MgADP is also labeled. (*C*) Measurement of minimum distance between the sidechain oxygen of any of the ED-domain 947 to 953 glutamate/aspartate residues and the sidechain nitrogen of K1348 in the three individual runs under both conditions. Note in one of the (+)MgADP runs (red), MgADP unbinds from NBD2 (marked by the red downward arrow). The gray bar marks the area where the distance is ≤4 Å. The total dwell time in distance ≤4 Å for each run is shown on the *Right*. Note the plot was window averaged with 10-ns scale, and the dwell time was calculated with raw data which has 100-ps scale (the same applies to *E*). (*D*) Movement of the center of mass of the C-α of the ED-domain residues 947 to 953 during simulation (red trace) relative to that of K1348 (blue trace). (*E*) Same as *C*, except the distance measured is between sidechain nitrogen of Kir6.1-CTD residues R323, K341, R347, R352, and the sidechain oxygen of E1318. (*F*) Same as *D*, except the blue trace represents the center of mass of the C-α of Kir6.1 CTD residues R323, K341, R347, R352, and the red trace is the C-α of E1318.

upstream of the Walker A motif in NBD2 (^{1315}VRYEN^{1319}, named A-loop for aromatic residue interacting with the adenine ring of ATP) (60) frequently extended across the intersubunit gap to interact with a cluster of positively charged residues in Kir6.1-CTD, including R323, K341, R347, and R352 (Movie S5). In direct contrast, when MgADP was bound to SUR2B-NBD2, the A-loop instead consistently interacted with MgADP at NBD2, far from the Kir6.1-CTD. The A-loop in SUR2B includes Y1317, which interacts with the adenine ring of bound MgADP at NBD2. Simultaneously, the dissociation of the ED-domain from Walker A K1348 that occurred with MgADP binding at NBD2 freed the ED-domain to move in between NBD2 and Kir6.1-CTD. There, the ED-domain interacted with positive-charged residues in Kir6.1-CTD that in the absence of MgADP interacted with NBD2 A-loop E1328 (Movies S5 and S6). Effectively, the Kir6.1-CTD exchanged the A-loop for the ED-domain and stabilized each conformation. Quantitatively, minimum distances measured between E1318 in A-loop and the four positive residues in Kir6.1-CTD documented the closer relation of A-loop and Kir6.1-CTD throughout the simulations in the absence of MgADP than when MgADP was bound (Fig. 6*E*). Minimum distance below 4 Å sufficient for E1318 salt bridge formation was seen in all three runs lacking MgADP but only transiently (3.6 ns) in a single of

the three runs with MgADP (Fig. 6*E*). The nucleotide-dependent dynamics between the NBD2 A-loop and Kir6.1-CTD was also shown by tracking distance between the C-α of E1318 and center of the mass of the C-α for the Kir6.1-CTD–positive residues (an example run for each condition shown in Fig. 6*F*).

The dynamic, tripartite interactions between the ED-domain, NBD2, and Kir6.1-CTD, and the dependence of these interactions on MgADP found in MD simulations significantly advances our understanding of the mechanism of SUR-mediated channel stimulation by Mg-nucleotides. In the absence of MgADP, the ED-domain has preferred interactions with NBD2 Walker A K1348, while the A-loop E1318 is engaged with Kir6.1-CTD. This hinders NBD2 from undergoing further conformational transition toward that of the NBDs-dimerized human pancreatic channel quatrefoil structure (27), which shows SUR1-NBD2 further rotated toward NBD1 and also away from the positively charged residues in Kir6-CTD (PDB: 6C3O) (27). Upon MgADP binding to NBD2, the ED-domain is dissociated from K1348, while the NBD2 A-loop becomes stabilized by the bound MgADP, unable to extend toward Kir6.1-CTD. As a sequence of results, the ED-domain is free to move toward other surrounding positively charged residues including those in Kir6.1-CTD, which further prevents the interactions between NBD2 and Kir6.1-CTD, thus allowing NBD2 to undergo further rotation toward dimerization with NBD1. Supporting this understanding, an ion pair formed by R347 in the Kir6.1-CTD, with E1318 in the A-loop of SUR2B-NBD2, has previously been reported to play a role in channel activation by MgADP and the potassium channel opener pinacidil (61). Disruption of this ion pair by charge neutralization enhances MgADP/pinacidil gating, while charge swap restored wild-type–like sensitivity to MgADP/pinacidil (61). Our findings support the hypothesis that in order for NBDs to dimerize, interactions between SUR-NBD2 and Kir6-CTD must dissolve. Accordingly, disruption of the Kir6.1 R347-SUR2B E1318 salt bridge facilitates MgADP/pinacidil stimulation, as breaking the salt bridge promotes nucleotide binding at NBD2 and allows the further NBD2 movement needed for NBD dimerization and channel activation. The ED-domain in particular, by interacting with Walker A K1348, acts essentially as a mobile autoinhibitory motif, akin to autoinhibition mechanisms in many kinases (62), that occludes NBDs dimerization in the absence of MgADP and is deflected to permit dimerization

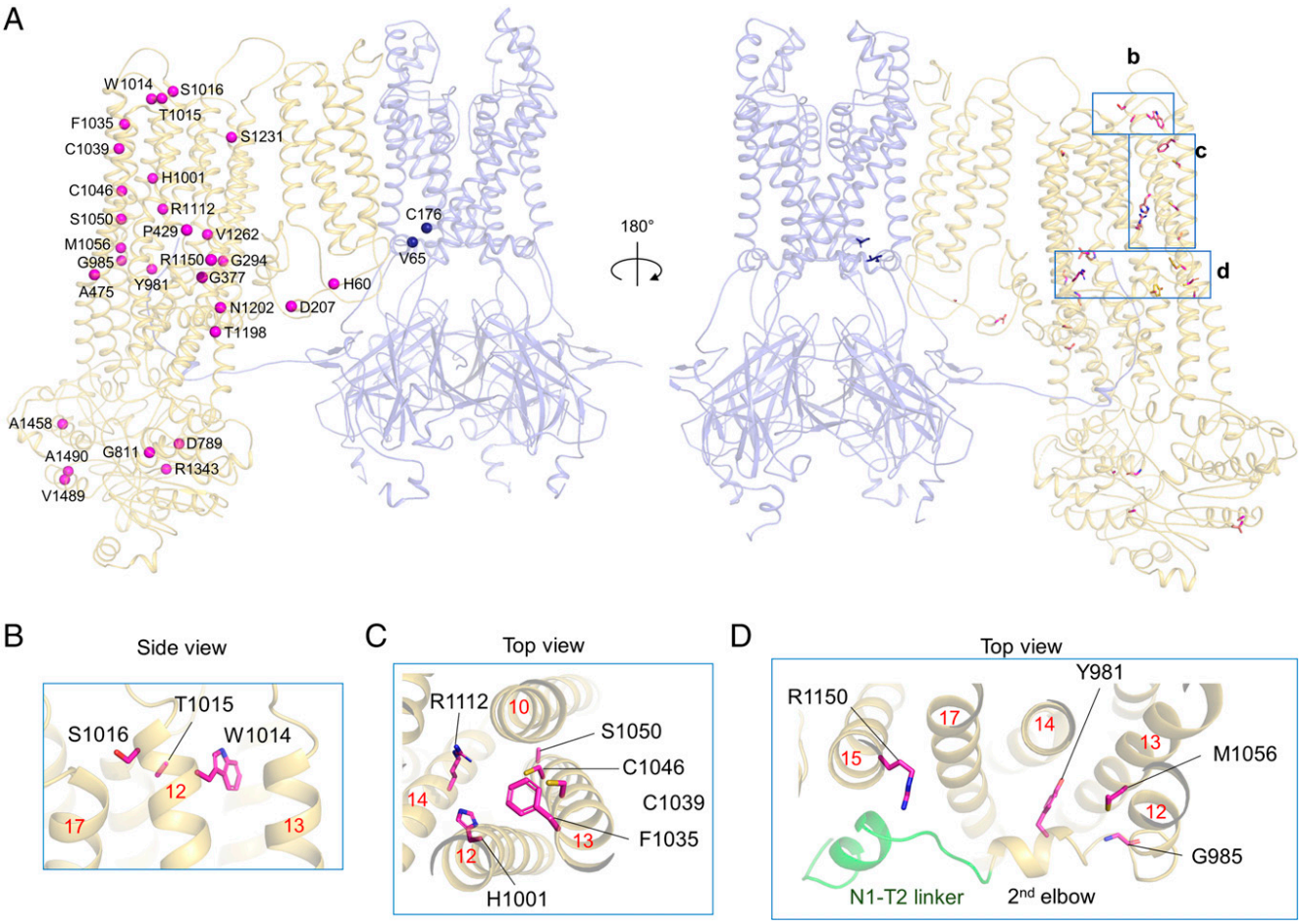


Fig. 7. Residues mutated in Cantú patients mapped onto the Kir6.1/SUR2B channel structure. (*A*) Residues mutated are shown as blue (Kir6.1) or magenta (SUR2B) in P1 conformation as spheres (*Left*) or in stick model (*Right*). Rat SUR2B numbering is used. Corresponding human mutations with rat residue in parentheses are as follows: H60Y (H60), D207E (D207), G294E (G294), G380C (G377), P432L (P429), A478V (A475), D793V (D789), G815A (G811), Y985S (Y981), G989E (G985), H1005L (H1001), W1018G (W1014), T1019E/K (T1015), S1020P (S1016), F1039S (F1035), S1054Y (S1050), C1043Y (C1039), C1050F (C1046), M1060I (M1056), R1116H/C/G (R1112), R1154G/Q/W (R1150), T1202M (T1198), N1206K (N1202), S1235F (S1231), V1266M (V1262), R1347C (R1343), A1462G (A1458), V1490E (V1489), and A1494T (A1490). (*B–D*) Close-up side or top views of boxed regions labeled in the overall structure in *A* (*Right*). In *D*, the N1-T2 linker is colored green and labeled together with the second elbow helix leading to TM12 of TMD2 in SUR2B. Red numbers mark the TM helices shown.

when MgADP has bound to NBD2. In this way, the ED-domain functions as a gatekeeper to prevent unregulated channel activation in the absence of MgADP.

Implications for Cantú Mutations. Taken together, our structures and MD simulations capture conformations that appear intermediate between the NBD-separated inactive and NBD-dimerized active states. The structural knowledge sheds light on how Cantú mutations (Fig. 7*A*) may cause gain of function in vascular K_{ATP} channels. In Kir6.1, V65M in the SH and C176S in the pore-lining helix likely enhance function by increasing channel P_o, which has been demonstrated in equivalent mutations in Kir6.2 (63). Most Cantú mutations identified to date are in *ABCC9*. Significantly, many of them affect residues in TM12, including Y981 and G985 in the second elbow helix, and W1014, T1015, and S1016 at the top (Fig. 7). TM12 is connected to the N1-T2 linker (Fig. 7*D*). Many other Cantú mutations are in domains interacting with TM12, including a series throughout TM13 (F1035, C1039, C1046, S1050, and M1056), as well as H1001 in TM12 and R1112 in TM14, which interface TM13 (Fig. 7*C*). One most frequently mutated residue R1150 of TM15 is adjacent to the structured helix portion of the N1-T2 linker, C-terminal to the ED-domain (Fig. 7*D*). The interconnectivity of these residues and their association with the N1-T2 linker suggest they may in common govern a critical conformational change during channel gating by Mg-nucleotides at the NBDs. Consistent with this notion, Y981S, G985E, and M1056I have been shown to enhance channel response to MgADP stimulation (42). Of note, C1039Y in TM13, has been shown to increase channel P_o similar to D207E in L0, rather than enhance MgADP response (42). It is possible that C1039Y alters interactions of SUR-TMDs with Kir6.1Nt and/or Kir6.1-TMs to affect channel P_o. Finally, two other mutations that also enhance MgADP response but are not directly connected to the N1-T2 linker are P429L in TM8 and A475V in TM9 of TMD1 (42). TM8 and TM9 are part of the TM bundles above NBD1 and NBD2 respectively, and P429L and A475V may affect the dynamics of the NBDs to alter MgADP response. Future studies correlating the effects of Cantú mutations on channel conformations and gating will further illuminate the structural basis of channel gating and in turn mechanisms of disease mutations.

Summary. Insights into how a particular complex operates is often gained by comparing related complexes, anticipating that similarities and differences in structure and function will correlate. In this study, we sought to determine the cryoEM structure of vascular K_{ATP} channels, composed of Kir6.1 and SUR2B, in the presence of ATP and Glib, for comparison to pancreatic K_{ATP} channel (Kir6.2/SUR1) structures determined with the same conditions. The structures we obtained reveal multiple elements showing distinct configurations that may account for channel-specific conductance, ATP inhibition, and drug sensitivities. In contrast, the serendipitous appearance of quatrefoil-like conformations, and SURx linkers which have been missing in previous K_{ATP} structures and are now seen at critical domain interfaces, affords insights into the long-sought structural dynamics shared by K_{ATP} channels in regulating their activity. The Q conformations adopted by SUR2B are most simply interpreted as transitional states between the inactive NBD-separated and the active NBD-dimerized SUR conformations.

The several conformations isolated from the cryoEM dataset, together with the dynamics revealed by 3D variability analyses and captured by MD simulations, suggest a model hypothesis for how Mg-nucleotide interactions with SUR2B activates Kir6.1 (Fig. 8). In this model, individual SUR2B subunits transition between P and Q conformations. In the Q conformations and without Mg-nucleotides at NBD2, the ED-domain in the N1-T2 linker acts as an autoinhibitory motif that prevents unregulated activation. Specifically, ED-domain interaction with Walker A K1348 at NBD2 promotes electrostatic interaction between NBD2 A-loop and Kir6.1-CTD, which further corrupts the Mg-nucleotide binding site and

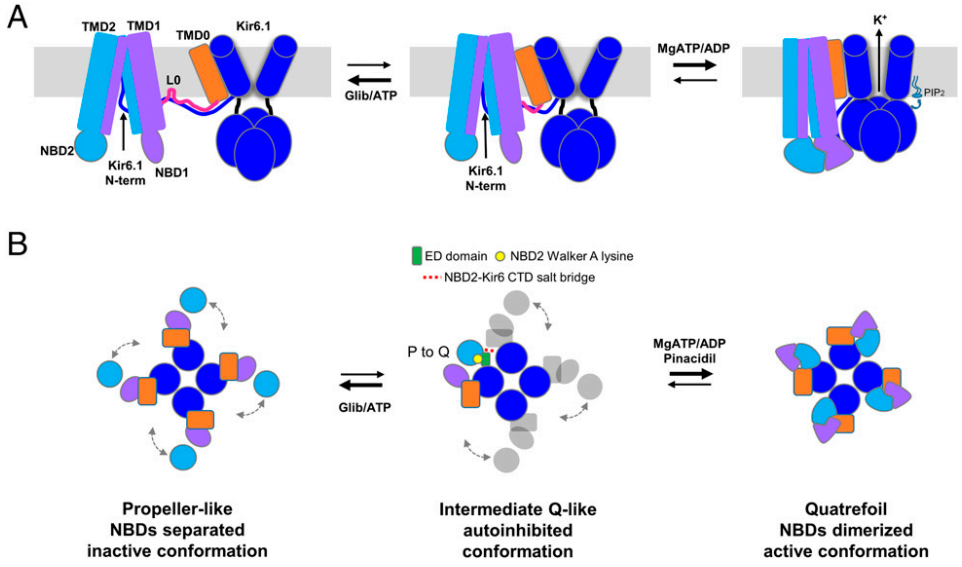


Fig. 8. Proposed model of vascular K_{ATP} channel conformational dynamics. Cartoon representation of channel side view (*A*) and top/down view (*B*) in inactive P conformation, Q-like intermediate conformation (only one SUR2B is colored to highlight structural interactions), and active, NBD-dimerized quatrefoil conformation. In the presence of Glib and ATP, the P conformation dominates. Addition of MgATP/ADP promotes NBD dimerization, which is postulated to cause Kir6.1-CTD to move close to the membrane to interact with PIP_2 for channel opening. In *B*, individual SUR subunits undergo P-Q conformation transitions independently. In the absence of MgADP at NBD2, the ED-domain interacts with NBD2-Walker A lysine (1348). The A-loop E1318 in NBD2 forms salt bridges with positively charged residues in Kir6.1-CTD, preventing further rotation of NBD2 needed for NBDs dimerization, thus arresting SUR in an autoinhibited intermediate conformation. Increasing MgATP/ADP concentrations increases the probability of MgATP/ADP binding to all SUR2B subunits to release autoinhibition and promotes conformational change to the NBD-dimerized quatrefoil state for channel activation.

also withholds NBD2 from dimerization with NBD1. Addition of Mg-nucleotides relieves autoinhibition imposed by the ED-domain, coupling organization of the Mg-nucleotide binding site to liberation of NBD2 to rotate toward NBD1 for dimerization. Yet-to-be-determined mechanisms are required to explain how dimerization of NBDs in SUR2B leads Kir6.1-CTD to move up to the membrane to interact with PIP_2 for channel opening. The model would predict that inhibitory ligands such as Glib or stimulatory ligands such as Mg-nucleotides or the potassium channel opener pinacidil, are able to shift the equilibrium of SUR2B toward P or Q conformations to drive channel closure or opening, respectively. It is important to note that dimerization of the NBDs was not observed during the 1-μs simulation in the presence of MgADP/MgATP; moreover, only one SUR2B is present in the simulations, which prevents consideration of potential structural impact of neighboring SUR2B subunits. Future structures with NBDs dimerized and MD simulations of the full channel are required to confirm and extend our understanding of K_{ATP} channel activation. This notwithstanding, we speculate the general scheme of the model applies to other K_{ATP} channels with variations to explain isoform-specific sensitivities for Mg-nucleotides and drugs. The structures presented here serve as a framework for understanding channel regulation and dysregulation and will aid development of isoform-specific pharmacological modulators to correct channel defects in Cantú and other diseases involving vascular K_{ATP} dysfunction.

Materials and Methods

Expression and purification of Kir6.1/SUR2B channels, cryoEM imaging, data processing, and modeling were performed using published protocols (22, 23, 26, 64) and are described in detail in *SI Appendix*. Briefly, recombinant adenoviruses containing the coding sequences of Kir6.1 and FLAG-tagged SUR2B were used to infect COSm6 cells and expressed channels purified via the FLAG tag. Purified channel complexes were spotted on grids coated with graphene-oxide, vitrified, and imaged on a Titan Krios 300 kV cryoelectron microscope. Image processing and analysis were carried out in RELION-3.0 and CryoSPARC. Models were built by fitting previously published Kir6.2/SUR1 structures and in SWISS-MODEL and refined in Coot and Phenix.

MD simulations were performed at all-atom resolution using AMBER 16 (65) with graphics processing unit (GPU) acceleration. The starting structure was developed from the Q1 model (four Kir6.1 and one SUR2B) with flexible linkers built in SWISS-MODEL. Glib was removed to allow the TMDs to relax during simulations. The structures were protonated at pH 7 and inserted in a bilayer membrane composed of 1-palmitoyl-2-oleoyl-phosphatidylcholine lipids and surrounded by an aqueous solution of 0.15 M KCl. Pairwise distances were analyzed from the simulated trajectories using the gmx pairdist tool in Gromacs 2019.4 (66). Detailed methods for MD simulations and data analysis are provided in *SI Appendix*.

Data Availability. CryoEM density maps have been deposited to the Electron Microscopy Data Bank (P1: EMD-23864, P2: EMD-23881, Q1: EMD-23880, and Q2: EMD-23882). Coordinates for $(Kir6.1)_4SUR2B$ atomic models have been deposited to the Protein Data Bank (P1: 7MIT, P2: 7MJP, Q1: 7MJO, and Q2: 7MJQ). MD simulation data have been deposited to the open-access repository Zenodo (5546127). All other study data are included in the article and/or supporting information.

ACKNOWLEDGMENTS. A portion of this research was supported by NIH Grant No. U24GM129547 and performed at the Pacific Northwest Cryo-EM Center (PNCC) at Oregon Health and Science University (OHSU) and accessed through the Environmental Molecular Sciences Laboratory (EMSL) (grid.436923.9), a Department of Energy Office (DOE) of Science User Facility sponsored by the Office of Biological and Environmental Research. We also thank Dr. Nancy Meyer at the PNCC and staff at the Multiscale Microscopy Core of OHSU for technical support. We acknowledge the support by the NIH Grant No. R01DK066485 (to S.-L.S.) and by the NSF Grant No. MCB 17158233 (to D.M.Z.).

1. C. G. Nichols, KATP channels as molecular sensors of cellular metabolism. *Nature* **440**, 470–476 (2006).
2. L. Aguilar-Bryan *et al.*, Toward understanding the assembly and structure of KATP channels. *Physiol. Rev.* **78**, 227–245 (1998).
3. M. N. Foster, W. A. Coetzee, KATP channels in the cardiovascular system. *Physiol. Rev.* **96**, 177–252 (2016).
4. A. P. Babenko, L. Aguilar-Bryan, J. Bryan, A view of sur/KIR6.X, KATP channels. *Annu. Rev. Physiol.* **60**, 667–687 (1998).
5. M. T. Nelson, J. M. Quayle, Physiological roles and properties of potassium channels in arterial smooth muscle. *Am. J. Physiol.* **268**, C799–C822 (1995).
6. M. T. Nelson, J. B. Patlak, J. F. Worley, N. B. Standen, Calcium channels, potassium channels, and voltage dependence of arterial smooth muscle tone. *Am. J. Physiol.* **259**, C3–C18 (1990).
7. M. T. Nelson, N. B. Standen, J. E. Brayden, J. F. Worley, 3rd, Noradrenaline contracts arteries by activating voltage-dependent calcium channels. *Nature* **336**, 382–385 (1988).
8. Q. Aziz *et al.*, The ATP-sensitive potassium channel subunit, Kir6.1, in vascular smooth muscle plays a major role in blood pressure control. *Hypertension* **64**, 523–529 (2014).
9. Y. Huang, D. Hu, C. Huang, C. G. Nichols, Genetic discovery of ATP-sensitive K^+ channels in cardiovascular diseases. *Circ. Arrhythm. Electrophysiol.* **12**, e007322 (2019).
10. B. W. van Bon *et al.*, Cantú syndrome is caused by mutations in ABCC9. *Am. J. Hum. Genet.* **90**, 1094–1101 (2012).
11. M. Harakalova *et al.*, Dominant missense mutations in ABCC9 cause Cantú syndrome. *Nat. Genet.* **44**, 793–796 (2012).
12. D. K. Grange *et al.*, Cantú syndrome: Findings from 74 patients in the International Cantú Syndrome Registry. *Am. J. Med. Genet. C. Semin. Med. Genet.* **181**, 658–681 (2019).
13. F. M. Ashcroft, F. M. Gribble, Correlating structure and function in ATP-sensitive K+ channels. *Trends Neurosci.* **21**, 288–294 (1998).
14. T. Baukrowitz *et al.*, PIP2 and PIP as determinants for ATP inhibition of KATP channels. *Science* **282**, 1141–1144 (1998).
15. O. Fürst, B. Mondou, N. D'Avanzo, Phosphoinositide regulation of inward rectifier potassium (Kir) channels. *Front. Physiol.* **4**, 404 (2014).
16. S. L. Shyng, C. G. Nichols, Membrane phospholipid control of nucleotide sensitivity of KATP channels. *Science* **282**, 1138–1141 (1998).
17. A. Fujita, Y. Kurachi, Molecular aspects of ATP-sensitive K+ channels in the cardiovascular system and K+ channel openers. *Pharmacol. Ther.* **85**, 39–53 (2000).
18. W. W. Shi, Y. Yang, Y. Shi, C. Jiang, K(ATP) channel action in vascular tone regulation: From genetics to diseases. *Sheng Li Xue Bao* **64**, 1–13 (2012).
19. A. Tinker, Q. Aziz, Y. Li, M. Specterman, ATP-sensitive potassium channels and their physiological and pathophysiological roles. *Compr. Physiol.* **8**, 1463–1511 (2018).
20. C. McClenaghan *et al.*, Glibenclamide reverses cardiovascular abnormalities of Cantu syndrome driven by KATP channel overactivity. *J. Clin. Invest.* **130**, 1116–1121 (2020).
21. A. Ma *et al.*, Glibenclamide treatment in a Cantú syndrome patient with a pathogenic ABCC9 gain-of-function variant: Initial experience. *Am. J. Med. Genet. A.* **179**, 1585–1590 (2019).
22. G. M. Martin, B. Kandasamy, F. DiMaio, C. Yoshioka, S. L. Shyng, Anti-diabetic drug binding site in a mammalian K_{ATP} channel revealed by Cryo-EM. *eLife* **6**, e31054 (2017).
23. G. M. Martin *et al.*, Cryo-EM structure of the ATP-sensitive potassium channel illuminates mechanisms of assembly and gating. *eLife* **6**, e24149 (2017).
24. J. X. Wu *et al.*, Ligand binding and conformational changes of SUR1 subunit in pancreatic ATP-sensitive potassium channels. *Protein Cell* **9**, 553–567 (2018).
25. A. B. Karger *et al.*, Role for SUR2A ED domain in allosteric coupling within the K(ATP) channel complex. *J. Gen. Physiol.* **131**, 185–196 (2008).
26. G. M. Martin *et al.*, Mechanism of pharmacochaperoning in a mammalian K_{ATP} channel revealed by cryo-EM. *eLife* **8**, e46417 (2019).
27. K. P. K. Lee, J. Chen, R. MacKinnon, Molecular structure of human KATP in complex with ATP and ADP. *eLife* **6**, e32481 (2017).
28. Y. Zhao, Z. Chen, Z. Cao, W. Li, Y. Wu, Diverse structural features of potassium channels characterized by scorpion toxins as molecular probes. *Molecules* **24**, 2045 (2019).
29. V. P. Repunte *et al.*, Extracellular links in Kir subunits control the unitary conductance of SUR/Kir6.0 ion channels. *EMBO J.* **18**, 3317–3324 (1999).
30. J. Yang, M. Yu, Y. N. Jan, L. Y. Jan, Stabilization of ion selectivity filter by pore loop ion pairs in an inwardly rectifying potassium channel. *Proc. Natl. Acad. Sci. U.S.A.* **94**, 1568–1572 (1997).
31. R. S. Vieira-Pires, J. H. Morais-Cabral, 3(10) helices in channels and other membrane proteins. *J. Gen. Physiol.* **136**, 585–592 (2010).
32. K. V. Quinn, Y. Cui, J. P. Giblin, L. H. Clapp, A. Tinker, Do anionic phospholipids serve as cofactors or second messengers for the regulation of activity of cloned ATP-sensitive K+ channels? *Circ. Res.* **93**, 646–655 (2003).
33. A. L. Duncan, R. A. Corey, M. S. P. Sansom, Defining how multiple lipid species interact with inward rectifier potassium (Kir2) channels. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 7803–7813 (2020).
34. D. Ding, M. Wang, J. X. Wu, Y. Kang, L. Chen, The structural basis for the binding of repaglinide to the pancreatic KATP channel. *Cell Rep.* **27**, 1848–1857.e4 (2019).
35. M. Yamada *et al.*, Sulphonylurea receptor 2B and Kir6.1 form a sulphonylurea-sensitive but ATP-insensitive K+ channel. *J. Physiol.* **499**, 715–720 (1997).

36. E. Satoh *et al.*, Intracellular nucleotide-mediated gating of SUR/Kir6.0 complex potassium channels expressed in a mammalian cell line and its modification by pinacidil. *J. Physiol.* **511**, 663–674 (1998).
37. S. B. Hansen, X. Tao, R. MacKinnon, Structural basis of PIP2 activation of the classical inward rectifier K+ channel Kir2.2. *Nature* **477**, 495–498 (2011).
38. M. R. Whorton, R. MacKinnon, Crystal structure of the mammalian GIRK2 K+ channel and gating regulation by G proteins, PIP2, and sodium. *Cell* **147**, 199–208 (2011).
39. V. N. Bavro *et al.*, Structure of a KirBac potassium channel with an open bundle crossing indicates a mechanism of channel gating. *Nat. Struct. Mol. Biol.* **19**, 158–163 (2012).
40. O. B. Clarke *et al.*, Domain reorientation and rotation of an intracellular assembly regulate conduction in Kir potassium channels. *Cell* **141**, 1018–1029 (2010).
41. Y. Niu, X. Tao, K. K. Touhara, R. MacKinnon, Cryo-EM analysis of PIP_2 regulation in mammalian GIRK channels. *eLife* **9**, e60552 (2020).
42. C. McClenaghan *et al.*, Cantu syndrome-associated SUR2 (ABCC9) mutations in distinct structural domains result in K_{ATP} channel gain-of-function by differential mechanisms. *J. Biol. Chem.* **293**, 2041–2052 (2018).
43. T. Pipatpolkai, S. Usher, P. J. Stansfeld, F. M. Ashcroft, New insights into K_{ATP} channel gene mutations and neonatal diabetes mellitus. *Nat. Rev. Endocrinol.* **16**, 378–393 (2020).
44. A. P. Babenko, J. Bryan, Sur domains that associate with and gate KATP pores define a novel gatekeeper. *J. Biol. Chem.* **278**, 41577–41580 (2003).
45. R. Masia *et al.*, A mutation in the TMD0-L0 region of sulfonylurea receptor-1 (L225P) causes permanent neonatal diabetes mellitus (PNDM). *Diabetes* **56**, 1357–1362 (2007).
46. E. Krissinel, K. Henrick, Inference of macromolecular assemblies from crystalline state. *J. Mol. Biol.* **372**, 774–797 (2007).
47. U. Russ, U. Lange, C. Löffler-Walz, A. Hambrock, U. Quast, Interaction of the sulfonylthiourea HMR 1833 with sulfonylurea receptors and recombinant ATP-sensitive K(+) channels: Comparison with glibenclamide. *J. Pharmacol. Exp. Ther.* **299**, 1049–1055 (2001).
48. R. Ashfield, F. M. Gribble, S. J. Ashcroft, F. M. Ashcroft, Identification of the high-affinity tolbutamide site on the SUR1 subunit of the K(ATP) channel. *Diabetes* **48**, 1341–1347 (1999).
49. F. F. Yan, J. Casey, S. L. Shyng, Sulfonylureas correct trafficking defects of disease-causing ATP-sensitive potassium channels by binding to the channel complex. *J. Biol. Chem.* **281**, 33403–33413 (2006).
50. P. K. Devaraneni, G. M. Martin, E. M. Olson, Q. Zhou, S. L. Shyng, Structurally distinct ligands rescue biogenesis defects of the KATP channel complex via a converging mechanism. *J. Biol. Chem.* **290**, 7980–7991 (2015).
51. W. H. Vila-Carriles, G. Zhao, J. Bryan, Defining a binding pocket for sulfonylureas in ATP-sensitive potassium channels. *FASEB J.* **21**, 18–25 (2007).
52. P. Kühner *et al.*, Importance of the Kir6.2 N-terminus for the interaction of glibenclamide and repaglinide with the pancreatic K(ATP) channel. *Naunyn Schmiedebergs Arch. Pharmacol.* **385**, 299–311 (2012).
53. N. Li *et al.*, Structure of a pancreatic ATP-sensitive potassium channel. *Cell* **168**, 101–110.e10 (2017).
54. F. Liu, Z. Zhang, L. Csanády, D. C. Gadsby, J. Chen, Molecular structure of the human CFTR ion channel. *Cell* **169**, 85–95.e8 (2017).
55. Z. Zhang, F. Liu, J. Chen, Conformational changes of CFTR upon phosphorylation and ATP binding. *Cell* **170**, 483–491.e8 (2017).
56. L. Csanády, P. Vergani, D. C. Gadsby, Structure, gating, and regulation of the Cftr anion channel. *Physiol. Rev.* **99**, 707–738 (2019).
57. S.C. Bickers, S. Benlekbir, J. L. Rubinstein, V. Kanelis, Structure of Ycf1p reveals the transmembrane domain TMD0 and the regulatory R region of ABCC transporters. *Proc. Natl. Acad. Sci. U.S.A.* **118**, e2025853118 (2021).
58. A. Grossfield *et al.*, Best practices for quantification of uncertainty and sampling quality in molecular simulations [Article v1.0]. *Living J. Comput. Mol. Sci.* **1**, 5067 (2018).
59. S. Kumar, R. Nussinov, Relationship between ion pair geometries and electrostatic strengths in proteins. *Biophys. J.* **83**, 1595–1612 (2002).
60. S. V. Ambudkar, I. W. Kim, D. Xia, Z. E. Sauna, The A-loop, a novel conserved aromatic acid subdomain upstream of the Walker A motif in ABC transporters, is critical for ATP binding. *FEBS Lett.* **580**, 1049–1055 (2006).
61. D. Lodwick *et al.*, Sulfonylurea receptors regulate the channel pore in ATP-sensitive potassium channels via an intersubunit salt bridge. *Biochem. J.* **464**, 343–354 (2014).
62. G. M. Cheetham, Novel protein kinases and molecular mechanisms of autoinhibition. *Curr. Opin. Struct. Biol.* **14**, 700–705 (2004).
63. P. E. Cooper, C. McClenaghan, X. Chen, A. Stary-Weinzinger, C. G. Nichols, Conserved functional consequences of disease-associated mutations in the slide helix of Kir6.1 and Kir6.2 subunits of the ATP-sensitive potassium channel. *J. Biol. Chem.* **292**, 17387–17398 (2017).
64. C. M. Driggers, S. L. Shyng, Production and purification of ATP-sensitive potassium channel particles for cryo-electron microscopy. *Methods Enzymol.* **653**, 121–150 (2021).
65. D. A. Case *et al.*, AMBER 2016. University of California, San Francisco. (2016). https://ambermd.org/doc12/Amber16.pdf. Accessed 11 October 2021.
66. M. J. Abraham *et al.*, GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **1–2**, 19–25 (2015).

Research Article

Ligand-mediated Structural Dynamics of a Mammalian Pancreatic K_{ATP} Channel

Min Woo Sung [1†], Camden M. Driggers [1†], Barmak Mostofian [2], John D. Russo [2], Bruce L. Patton [1], Daniel M. Zuckerman [2*] and Show-Ling Shyng [1*]

*1 - **Department of Chemical Physiology and Biochemistry,** School of Medicine, Oregon Health & Science University, Portland, OR, USA*

*2 - **Department of Biomedical Engineering,** School of Medicine, Oregon Health & Science University, Portland, OR, USA*

Correspondence to Daniel M. Zuckerman and Show-Ling Shyng: Department of Biomedical Engineering, Schoolof Medicine, Oregon Health & Science University, 3181 S.W. Sam Jackson Park Road, Portland, OR 97239, USA (D. M. Zuckerman). Department of Chemical Physiology and Biochemistry, School of Medicine, Oregon Health & Science University, 3181 S.W. Sam Jackson Park Road, Portland, OR 97239, USA (S.-L. Shyng).
zuckermd@ohsu.edu (D.M. Zuckerman), shyngs@ohsu.edu (S.-L. Shyng)@MinWooSung5 (M.W. Sung), @DMZuckermanLab (D.M. Zuckerman)
https://doi.org/10.1016/j.jmb.2022.167789
Edited by Daniel L. Minor

Abstract

Regulation of pancreatic K_{ATP} channels involves orchestrated interactions of their subunits, Kir6.2 and SUR1, and ligands. Previously we reported K_{ATP} channel cryo-EM structures in the presence and absence of pharmacological inhibitors and ATP, focusing on the mechanisms by which inhibitors act as pharmacological chaperones of K_{ATP} channels (Martin et al., 2019). Here we analyzed the same cryo-EM datasets with a focus on channel conformational dynamics to elucidate structural correlates pertinent to ligand interactions and channel gating. We found pharmacological inhibitors and ATP enrich a channel conformation in which the Kir6.2 cytoplasmic domain is closely associated with the transmembrane domain, while depleting one where the Kir6.2 cytoplasmic domain is extended away into the cytoplasm. This conformational change remodels a network of intra- and inter-subunit interactions as well as the ATP and PIP_2 binding pockets. The structures resolved key contacts between the distal N-terminus of Kir6.2 and SUR1′s ABC module involving residues implicated in channel function and showed a SUR1 residue, K134, participates in PIP_2 binding. Molecular dynamics simulations revealed two Kir6.2 residues, K39 and R54, that mediate both ATP and PIP_2 binding, suggesting a mechanism for competitive gating by ATP and PIP_2.

Introduction

Pancreatic ATP-sensitive potassium (K_{ATP}) channels functionally couple glucose metabolism to insulin release and are crucial for glucose homeostasis.[4,51] Structurally, the pancreatic K_{ATP} channel is an octameric complex composed of two distinct integral membrane proteins.[15,31,38,41,48,66] A tetrameric core of Kir6.2 subunits form the central transmembrane pore of the channel. A coronal array of four sulfonylurea receptor 1 (SUR1) subunits surrounds the channel core, each SUR1 companioned with one Kir6.2 subunit. Genetic mutations of these subunits that dysregulate K_{ATP} channel activity are causes of neonatal diabetes (gain of function) and congenital hyperinsulinism

(loss of function).[4] K_{ATP} channels harbor multiple distinct and antagonistic binding sites for their primary physiological regulators, intracellular ATP and ADP, which close the ion channel through a binding site in Kir6.2, but open the channel through Mg-dependent binding on SUR1.[52,62] In addition, channel activity is operationally governed by binding sites for specific membrane phospholipids, particularly PIP_2, which directly promote opening as well as antagonize the ATP inhibition at the Kir6.2 binding sites.[52] Finally, the pancreatic K_{ATP} channel is the drug binding target for sulfonylurea and glinide anti-diabetic medications, which inhibit channel activity and thus stimulate insulin secretion.[26] The long held principal objective of K_{ATP} channel research has been to understand the protein dynamics by which these several ligand interactions, separately and in concert, ultimately determine levels of K_{ATP} channel activity, and hence control insulin release.

CryoEM structures of K_{ATP} channels have provided direct insights into the structural mechanisms of ligand recognition and gating regulation. In a previous study, we reported comparative cryoEM structures for pancreatic K_{ATP} channels in the absence of ligands (apo); in the presence of ATP; and in the combined presence of ATP with alternative pharmacological inhibitors: glibenclamide, repaglinide, or carbamazepine.[47] The study found all pharmacological inhibitors occupy a common binding pocket located within SUR1 and that this binding pocket lies adjacent to the deep binding site for the Kir6.2 N-terminal tail, which courses through the prominent cleft between the two halves of the ABC (ATP Binding Cassette) module of SUR1. The findings offered mechanistic insight into how distinct pharmacological inhibitors inhibit channel activity and also facilitate channel assembly by stabilizing the interaction between Kir6.2 N-terminus and SUR1. However, it was noted during image analysis that each dataset in the study possessed considerable conformational heterogeneity, suggesting classification analyses within datasets may further illuminate channel structural dynamics relevant to ligand binding and gating.

Here, we show results from reprocessing of cyroEM datasets previously reported, focusing on conformational analysis and augmented by molecular dynamics (MD) simulations. Most notably, we found that the cytoplasmic domain (CTD) of Kir6.2 adopted two distinct conformations. In one, the CTD is tethered close to the membrane (Kir6.2-CTD-up). In the other, the CTD is counterclockwise corkscrewed away from the membrane, towards the cytoplasm (Kir6.2-CTD-down). Across structure datasets, the ratio of CTD-up versus CTD-down conformations strongly correlated with the occupation of inhibitory ligand binding sites. Importantly, drug binding and CTD conformation were associated with significant structural reorganization at the ATP and PIP_2 binding sites, and at domain interfaces within and between subunits, suggesting ligands act as molecular glues to stabilize the channel in the Kir6.2-CTD-up conformation. Of further importance, improved cryoEM maps and functional analysis revealed that binding of the activating ligand PIP_2 involves a direct interaction with SUR1 lysine-134 in TMD0, implicating a mechanism by which SUR1 enhances Kir6.2 PIP_2 sensitivity. Moreover, MD simulations uncovered Kir6.2 residues that participate in both ATP and PIP_2 binding, providing an explanation for how ATP and PIP_2 compete to control K_{ATP} channel gating. Together, our findings provide a framework for understanding how ligands shift channel conformation to regulate channel activity and how mutations, now observed to affect key protein–protein and protein–ligand interfaces, disrupt channel function and cause disease.

Results

K_{ATP} channel conformation analysis

Focused 3D classification of the Kir6.2 tetramer core plus one SUR1 subunit (denoted K_4S hereinafter) following symmetry expansion and signal subtraction[65] was performed on our previously published five datasets: apo, ATP only, carbamazepine and ATP (CBZ/ATP), glibenclamide and ATP (GBC/ATP), repaglinide and ATP (RPG/ATP)[47] (see Methods). This strategy was employed to circumvent alignment difficulty due to flexible SUR1 (Figure S1, S2). The analysis revealed two major K_4S conformations: Kir6.2-CTD-up and Kir6.2-CTD-down, wherein the Kir6.2-CTD was alternatively located closer to, or further from, the Kir6.2 membrane spanning channel domains, respectively. Particularly, translation of the CTD between up and down conformations further involved a rotation, together comprising a corkscrew movement wherein the CTD (from an extracellular point of view) was rotated clockwise in Kir6.2-CTD-up conformation relative to Kir6.2-CTD-down (Figure 1). The CTD-up and CTD-down conformations appear qualitatively similar to the T(tense)-state and R(relaxed)-state previously reported by others using a fusion SUR1-Kir6.2 protein under three different ligand conditions, GBC + ATPγS, ATPγS, or MgADP, wherein the T-state exhibits 10.6–12.5° CW rotation viewed from the extracellular side and 3–4.2 Å translation towards the membrane relative to the R-state.[78] Within both the CTD-up and CTD-down conformations, rocking and rotation of the Kir6.2-CTD was discernable using Relion 3 multibody refinement principal component analysis.[50] The heterogeneity was greater in the CTD-down than the CTD-up population of particles (Figure S3), consistent with an increase in CTD mobility when detached from the

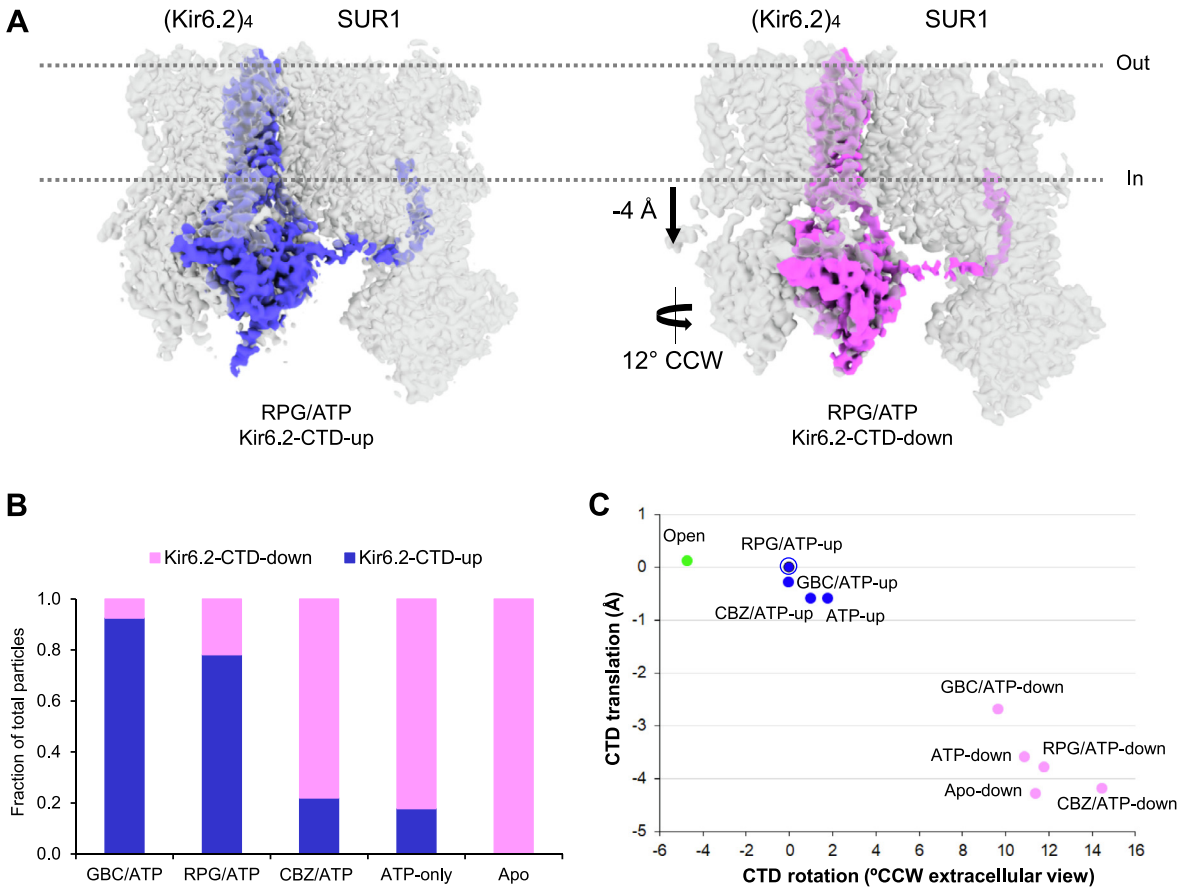


Figure 1. Two distinct conformations of the Kir6.2-CTD in RPG/ATP bound K_{ATP} channels. (A) CryoEM maps for the Kir6.2 tetramer core plus one SUR1 subunit are shown (semi-transparent grey, 1.0 σ contour), with the Kir6.2 subunit including KNtp in the Kir6.2-CTD-up (blue, 0.7 σ contour) and the Kir6.2-CTD-down (magenta, 0.7 σ contour) conformations. Compared to the CTD-up conformation, the Kir6.2-CTD in the CTD-down conformation is translocated from near the lipid bilayer towards the cytoplasm by ~4 Å (distance measured from the center of mass of G295- Cα in the G-loop gate of all four Kir6.2 subunits to the center of mass of F168-Cα in the helix bundle crossing gate of all four Kir6.2 subunits), and rotated counterclockwise by 12° (viewed from the extracellular side), measured by aligning structures onto the TM domain (residues 55–175) of the RPG CTD-up reference model, and calculating dihedral angles between K338-Cα of Chain A of Kir6.2 and the centers of mass of G245-Cα. **(B)** Fraction of particles in Kir6.2-CTD-up (blue) and Kir6.2-CTD-down (magenta) in K_{ATP} channels bound to different ligands or in apo state. **(C)** Variations in the extent of CTD translation and rotation observed for Kir6.2-CTD-up (blue dots) and Kir6.2-CTD-down (magenta dots) are shown for all datasets using RPG/ATP-Kir6.2-CTD-up as reference (circled blue dot). CTD translation away from the membrane is shown in negative value. There is a linear correlation between Kir6.2-CTD translation and Kir6.2-CTD-rotation (R^2 = 0.9682, y = -0.3031x − 0.1755). Translation and rotation of the CTD from a human open K_{ATP} structure,[81] PDB: 7S5T) relative to the RPG/ATP Kir6.2-CTD-up reference structure is included for comparison (green dot).

membrane. Similar Kir6.2-CTD dynamics were observed using cryoSPARC 3D variability analysis.[63]

Both the Kir6.2-CTD-up and Kir6.2-CTD-down conformations were observed in the GBC/ATP, RPG/ATP, CBZ/ATP and ATP-only datasets; however, relative abundance of the two conformations varied in the different liganded states (Figure 1(B), Table S1). The GBC/ATP and RPG/ATP datasets had the highest percentages of particles in the CTD-up conformation, with 92.5% and 71.2%, respectively. The CBZ/ATP dataset, which only had CBZ density in SUR1 but no clear ATP density in Kir6.2 likely due to lower concentrations of ATP used during sample preparation (see Methods) had a significantly lower percentage (22%) of particles in the CTD-up conformation, comparable to the ATP only dataset of 17.8%. In the absence of added ligands, i.e. the apo state, only Kir6.2-CTD-down conformation was observed. These findings show Kir6.2-CTD exists largely in two discrete conformations. That the distribution of the two states correlated with the binding of inhibitory ligands implicates the

switching between these conformations as a crucial mechanistic event controlling channel opening and closing. The RPG/ATP dataset gave the highest resolution maps for both the Kir6.2-CTD-up and CTD-down conformations (3.4 Å and 3.6 Å, respectively; Figure S1, S2, Table S1). The improved map quality compared to our previously published structure[47] allowed us to reevaluate ligand and protein densities that were previously ambiguous. We have therefore focused on the RPG/ATP dataset for structural analyses hereinafter.

In the RPG/ATP state, the predominant SUR1 conformation is arranged like a propeller when symmetrized, as described in our previous study.[47] In addition, we identified a minor conformation (~27% of all particles; Figure S1, Table S1) showing a large clockwise rotation of SUR1 towards the Kir6.2 tetramer (viewed from the extracellular side). This conformation is qualitatively similar to the quatrefoil conformation previously reported in the MgATP/MgADP-bound, NBDs-dimerized Kir6.2-SUR1 fusion channel structure,[38] and likewise our recently reported quatrefoil-like Kir6.1-SUR2B vascular K_{ATP} channel structure bound to GBC and ATP with separate NBDs.[69] The overall map resolution of this minor class is ~7 Å (Figure S1), which precluded detailed structural analysis. Nonetheless, it reveals that even in the presence of RPG and ATP, a large rotation of SUR1 resembling that seen in NBDs dimerized SUR1 quatrefoil conformation occurs, albeit much less frequently. Heterogeneity of SUR1 within the dominant propeller conformation with more subtle rotations of SUR1′s ABC core around the Kir6.2 tetramer was also observed regardless of whether the Kir6.2-CTD is up or down. We explored the details of SUR1 dynamics using the Kir6.2-CTD-up class of particles, classifying the dynamic motion as discrete eignevectors using Relion 3 multibody refinement (see Methods). Particles with amplitudes between 5 and 20, and between −5 and −20, along eigenvector 1 were refined separately to generate two maps, referred to as SUR1-in and SUR1-out conformations at 3.9 Å and 3.8 Å (Figure S4), respectively, for model building (Table S2) and structural analysis.

Comparison of different K_{ATP} conformations

The changes in conformation of Kir6.2-CTD and SUR1 were accompanied by remodeling of subunit and domain interfaces as well as protein–ligand interactions pertinent to gating. Comparing CTD-down to the CTD-up conformation, the Kir6.2-CTD is translated down into the cytoplasm by ~4 Å along an axis perpendicular to the embedding membrane, and simultaneously counterclockwise (CCW) rotated by 12° about that axis viewed from the extracellular side (Figure 1, movie 1). The descended location of Kir6.2-CTD reconfigured the interfacial (IF) helix (also called the slide helix, herein taken to include G53-D65) in the Kir6.2 N-terminus, and also the C-linker (herein taken to include H175-L181) by which inner helix M2 interacts with the CTD (Figure 2). Specifically, in the CTD-up conformation, the IF helix adopted a 3_{10} helix characteristic[75] wherein a directional kink at D58 demarcated the helix into N-terminal and C-terminal halves, with the N-terminal half pivoted towards SUR1 instead of along the adjacent Kir6.2 subunit. In contrast, the IF helix in the CTD-down conformation formed a continuous helix that extended towards the neighboring Kir6.2 subunit. Respecting the C-linker, in the CTD-up conformation, the C-linker formed a helical structure that participated in membrane PIP_2 binding; while in the CTD-down conformation, the C-linker was fully unraveled into a loop, in which a key PIP_2 interacting residue R176[8,68] was distant from the membrane and incapable of direct PIP_2 interaction. In comparisons between the SUR1 propeller in and out structures (Figure S1, S4), the ABC module in the SUR1-in structure is rotated clockwise closer to a neighboring Kir6.2 (viewed from the extracellular side). In this rotated position, the SUR1-L0 loop was pulled away from its interaction with SUR1′s direct Kir6.2 subunit partner. Specific molecular changes at the subunit and domain interfaces and ligand binding sites in different conformations and their relevance to gating are described below.

Intra-Kir6.2 and inter Kir6.2-Kir6.2 interactions-- Inspection of the Kir6.2 CTD-up and CTD-down structures revealed changes in intra- and inter-Kir6.2 interactions involving structural elements which occupy the transitional space between the membrane spanning and cytoplasmic domains of Kir6.2. These include the IF helix, the C-linker, the DE loop (the loop that connects βD and βE), the G-loop (the lower cytoplasmic gate) as well as the N-terminus and ATP. Specifically, in the CTD-up conformation, D204 at the start of the DE loop forms an intrasubunit salt bridge with R177 in the C-linker, which connects to Kir6.2 TM helix 2; and R206, also in the DE loop forms a salt bridge with D58 in the IF helix of the adjacent Kir6.2 subunit (Figure 2(C)). Two hydrogen bonds: one between D65 in the IF helix and T293 in the G-loop from adjacent subunits, and one between K39 at the N-terminus and ATP at a neighboring Kir6.2 subunit were also observed (Figure 2(C)). The salt bridges and hydrogen bonds bind the different structural elements together and hold the CTD in the up conformation. In contrast, the aforementioned salt bridges and hydrogen bonds were eliminated in the CTD-down structures (Figure 2(D)). In particular, separation of D204 from R177 in CTD-down was accompanied by uncoiling and extension at the end of the C-linker helix including R177, while separation of R206 from D58 was accompanied by a significant straightening of the kink in the IF helix and reorientation of the continuing chain, which connects to the base of the KNtp (Figure 2).

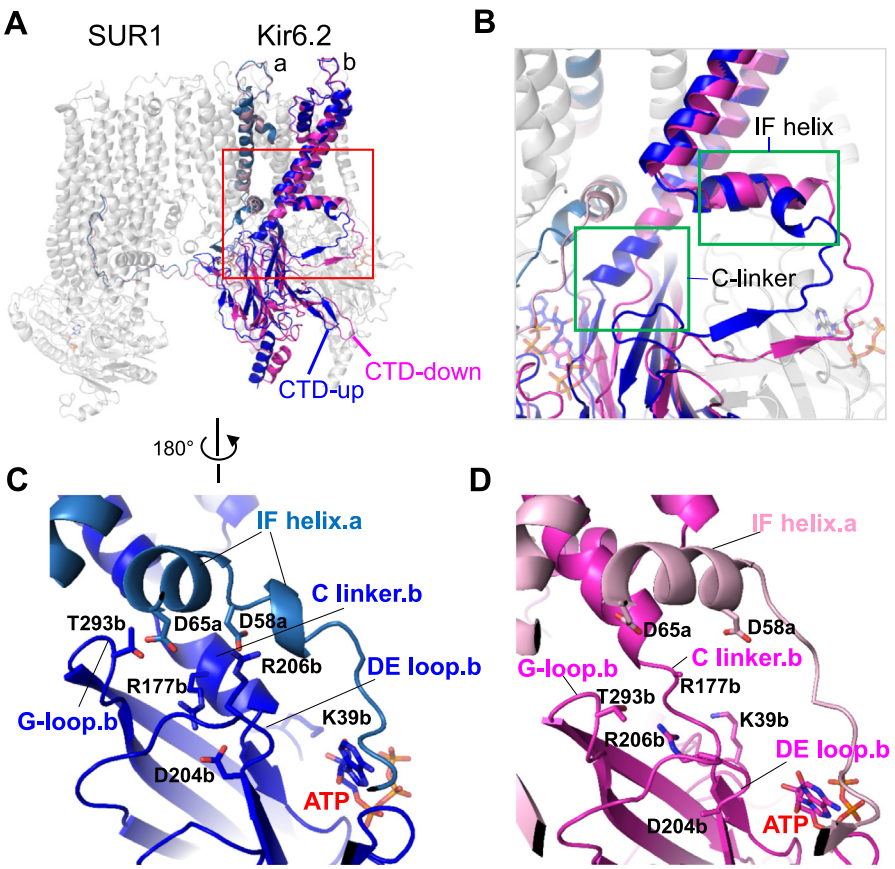


Figure 2. Structural comparison between RPG/ATP CTD-up and CTD-down conformations. (A) Superposition of the Kir6.2-CTD-up (blue) and CTD-down (magenta) structures. The red boxed region shows significant secondary structural difference at the IF helix and the C-linker. **(B)** Close-up view with structural differences highlighted (green boxes) at the IF helix and the C-linker in the two conformations. **(C)** Close-up view of the inter-subunit H-bond interactions between D65 and T293, and an inter-subunit salt-bridge between D58 and R206, as well as an intra-subunit salt-bridge between R177 and D204, and an inter-subunit ATP-binding interaction from K39 in the Kir6.2-CTD-up conformation. **(D)** Same close-up view as in (C) but of the Kir6.2-CTD-down conformation showing disruption of those interactions seen in (C). Structural elements and residues from chain a or chain b are labeled with a or b at the end.

Consistent with the notion that these labile salt bridges have critical roles in channel gating, previously published functional studies have implicated the participating Kir6.2 residues in channel regulation. D58 was previously suggested to be involved in anchoring Kir6.2 CTD to the TM domain through results of targeted mutagenesis.[40] Our findings resolve the salt bridge partnership with R206 in the neighboring Kir6.2 polypeptide, and further reveal such tethering is dynamically incorporated into the conformational changes of ion channel activation. Consolidating this view, R206 has been separately implicated in channel activation by PIP_2, through scanning mutagenesis investigations of positive residues involved in effecting bound PIP_2, wherein mutation R206A was found to abolish PIP_2 response and thus diminish channel activity.[40,42,67] Similar to R206A, mutation R177A also abolishes or greatly attenuates channel activity by diminishing PIP_2 response.[40,67] Thus, in corroborating earlier functional studies, our structural findings here elucidate key molecular interactions that place the Kir6.2-CTD close to the membrane in position to interact with membrane-bound phospholipids for channel opening. These insights are directly relevant to human health. Mutations of each of the four residues in the above salt bridges have been identified in congenital hyperinsulinism, a disease caused by loss of function of K_{ATP} channels. These include D58V,[18] R177W,[3] D204E,[59] and R206H.[9] Our structures here provide a mechanistic illustration of how perturbation of residues involved in conformational dynamics cause loss of channel function and hyperinsulinism.

Kir6.2 and SUR1 interactions-- Our previous study of pancreatic K_{ATP} channel structure suggested that a key regulatory interface through which SUR1 controls Kir6.2 channel activity is formed by the extended N-terminus of Kir6.2 (residues 1–30; referred to as KNtp) that is

inserted within its SUR1 subunit partner, wherein the KNtp is located within the SUR1 ABC transporter module.[47] We showed in particular that the KNtp is located between the two transmembrane helix bundles (TMBs) of SUR1 ABC module, and adjacent to the drug binding pocket of GBC, RPG, and CBZ. More detailed structural analysis was hindered by insufficient resolution for the density of KNtp in those published cryoEM maps. The additional analysis methods applied in our current study yielded clearer, contiguous densities and produced significantly improved maps revealing specific interactions between residues in KNtp and SUR1 (Figure 3). In the distal part of KNtp, which lies deep in the SUR1 ABC core cavity, Kir6.2-R4 is in bonding position with SUR1-T1139 and N1301. In the middle section of KNtp, which is near the entrance to the SUR1 ABC core cavity, Kir6.2-L17 interacts with R826 of NBD1 and G1119 and N1123 of TMB2. In the proximal end of KNtp, cryoEM density corresponding to residues P24, Y26 and R27 comes into close contact with cryoEM density corresponding to SUR1's NBD1-TMD2 linker around residue S988. In the Kir6.2-CTD down structure, interactions at the distal and middle segments of KNtp with SUR1 remain largely unchanged; however, the proximal section of KNtp is significantly further away from the NBD1-TMD2 of SUR1 (Figure 3(D)). In a recent Kir6.1-SUR2B cryoEM structure we showed that the NBD1-TMD2 linker has a role in regulating MgADP-dependent interactions between SUR2B-NBD2 and Kir6.1-CTD[69]. Our structures presented here reveal an additional contact between NBD1-TMD2 linker and KNtp. Whether this contact and changes at this interface in different conformations have functional significance warrants future investigation. As reported previously,[47] the cryoEM density of KNtp is the strongest in the GBC/ATP, RPG/ATP, and CBZ/ATP datasets, followed by ATP only, and is the

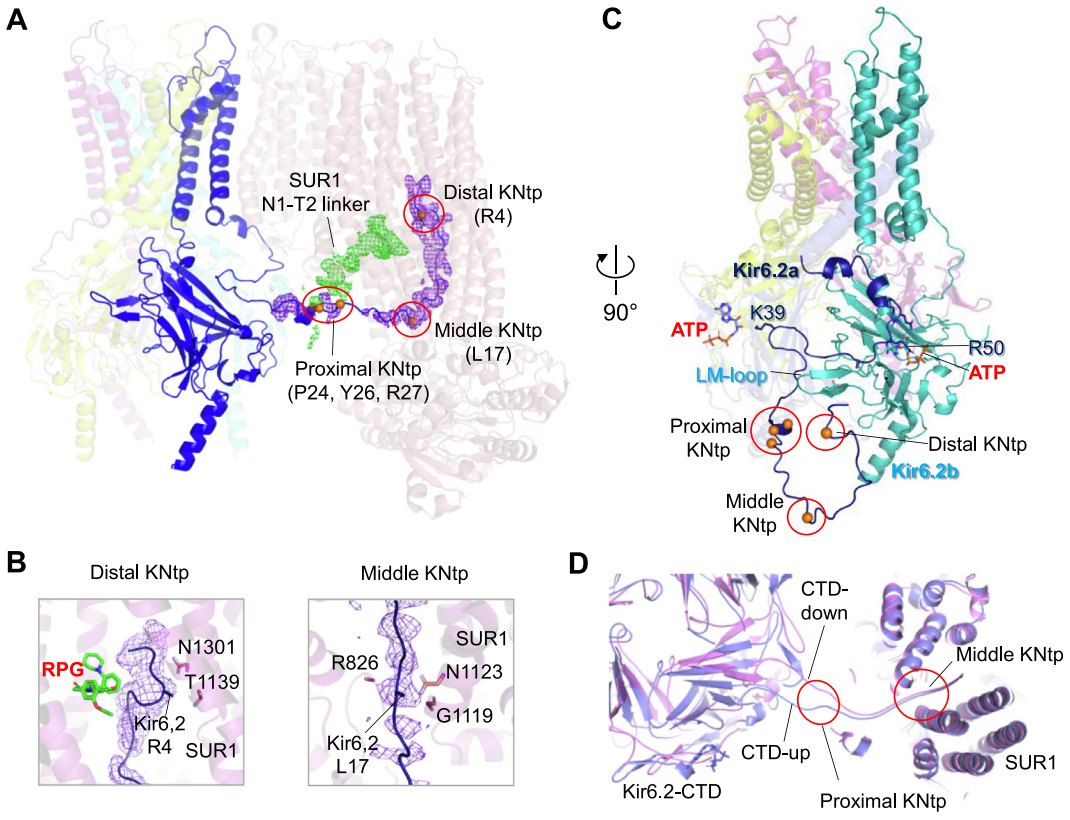


Figure 3. KNtp and SUR1 interface. (A) KNtp (Kir6.2 aa 1–30) cryoEM density (purple mesh, 1.0 σ contour) in RPG-Kir6.2-CTD-up structure, with key residues that interact with SUR1 shown as orange spheres within red circles. The distal portion of KNtp is located deep in the cavity of the SUR1-ABC core module. The middle section of KNtp lies near the entrance of the cavity. The proximal part (i.e. C-terminal part) of KNtp including P24, Y26 and R27 contacts the SUR1 N1-T2 linker density (shown as green mesh; 1.0 σ contour) near S988. **(B)** *Left*: Close view of distal KNtp cryoEM density (purple mesh, 1.0 σ contour) showing interaction of R4 with SUR1 T1139 and N1301 in TMD2 and proximity to bound RPG. *Right*: Close-up view of middle KNtp in cryoEM density (purple mesh, 1.0 σ contour) showing interaction of L17 with SUR1 R826, G1119 and N1123. **(C)** KNtp viewed without SUR1, showing its interconnectivity with two ATP binding sites and the LM-loop in the CTD of a neighboring Kir6.2. **(D)** Superposition of Kir6.2-CTD-up (blue) and CTD-down (pink) structures viewed from the extracellular side showing divergent positions of proximal KNtp.

weakest in the apo state, indicating inhibitory ligands stabilize the KNtp-SUR1 interface. Deletion of KNtp is known to increase channel open probability,[6,36,64] while immobilizing KNtp in the SUR1 ABC core cavity via engineered crosslinking between Kir6.2-L2C and SUR1-C1142 inhibits channel activity.[47] Worth noting, mutations L2P,[2] R4C/H, L17P, R24C, R27C/H in Kir6.2[18] as well as R826W[19] and N1123D[70] of SUR1 have all been reported in neonatal diabetes or congenital hyperinsulinism, which underscores the importance of the KNtp-SUR1 interface in channel gating.

In addition to KNtp forming interactions with the SUR1-ABC module, regions C-terminal to KNtp in the Kir6.2-N terminal domain were also observed to be intimately involved in protein–protein and protein–ligand interactions. First, in the CTD-up structure, Kir6.2 R31-R34 is close to the short loop that connects βL and βM (LM loop)[48] of the neighboring subunit (Figure 3(C)). A mutation D323K in the LM loop has been shown to disrupt ATP inhibition.[11] Second, further downstream K39 has its sidechain oriented towards ATP bound to the neighboring Kir6.2 on the other side. Thus, we found the Kir6.2 N-terminus is connected simultaneously to two ATP binding pockets. Dynamic movement of the KNtp between the CTD-up and CTD-down conformations may therefore impact the interactions of downstream Kir6.2-N terminal domain with neighboring subunits on both sides and with ATP. Finally, the refined structures showed that a loop (K47-Q52) N-terminal to the IF helix of Kir6.2 has close interaction with SUR1′s TMD0-intracellular loop 1 (ICL1), ICL2 and ICL3 (i.e. L0). Here, a compact network of interactions stabilizes the Kir6.2-CTD close to the membrane and also reinforces ATP binding. In the CTD-down conformation, the Kir6.2 pre-IF helix loop becomes more distant from the SUR1-ICLs such that the Kir6.2-CTD is no longer tethered close to the membrane, which also impacts the ATP binding pocket (see below).

ATP binding pocket--Rapid and reversible closure upon non-hydrolytic binding of ATP at Kir6.2 is a cardinal feature of K_{ATP} channels.[52] The improved map quality in the current study allowed us to refine interpretation of the ATP cryoEM density and the interaction network that coordinates ATP binding and follow how the ATP binding pocket becomes reconfigured in different conformations.

In our improved maps, the ATP density could be modeled with ATP in two alternative poses. In the first, the γ-phosphate is oriented upward towards R50 of Kir6.2, which is consistent with functional studies indicating that R50 interacts with the γ-phosphate of ATP[71]. This pose was used to model ATP density in our previously published structure bound to GBC and ATP (PDB: 6BAA)[46] and also to model ATPγS bound to a rodent SUR1-39aa-Kir6.2 fusion K_{ATP} channel[78]. In the second pose, the ATP's γ-phosphate is oriented downward facing N335. This alternative orientation is also supported by functional data showing that N335Q decreases ATP sensitivity[22] and used to model ATP density bound to Kir6.2 in cryoEM structures of a human SUR1-6aa-Kir6.2 fusion K_{ATP} channel (PDB: 6C3O and 6C3P).[38] Of note, we also observed an unassigned protruding density in ATP in our initial GBC/ATP map (EMD-7073), which we speculated may be contaminating Mg^{2+}[46] but which can be well modeled by the alternative pose of the γ-phosphate. The simplest interpretation is that the cryoEM density of ATP we observed is likely an ensemble of the two possible γ-phosphate poses.

The improved map also showed clear cryoEM density for the side chain of K205 in the L0 of SUR1. We have previously proposed that K205 participates in ATP binding[48] based on an early finding that K205E reduces ATP inhibition.[61] However, our previously published cryoEM map (EMD-7073) does not resolve the side chain density of K205 sufficiently to permit definitive conclusion. In our current map, K205 side chain was clearly oriented to the bound ATP (Figure 4(B)), stabilizing interactions with the β- and γ-phosphates of ATP. Similar observations have been reported by Ding et al.[21]. The role of K205 in ATP binding is further substantiated by Usher et al.[74] in which binding affinity between a fluorescent ATP analogue and the channel was assessed by FRET measurements between the ATP analogue and a fluorescent unnatural amino acid ANAP (3-(6-acetylnaphthalen-2-ylamino) − 2-aminopropanoic acid) engineered at Kir6.2 amino acid position 311. The study found that SUR1-K205A and K205E mutations reduce ATP binding affinity by ∼5 and 10-fold.

The conformational dynamics we observed in Kir6.2-CTD and SUR1 had significant impact on the structure of the ATP binding site. In the Kir6.2-CTD-up conformation, the Kir6.2-CTD was packed tightly against SUR1′s ICL1, ICL2, and the initial segment of L0 (Figure 4(A); contact surface area ∼144 $Å^2$, calculated using PDBePISA). In this conformation, ATP was fitted snugly into the pocket formed by the N- and C-terminal domains of adjacent Kir6.2 subunits and L0 of SUR1, and had an ATP interface area ∼430 $Å^2$. In the CTD-down conformation, the Kir6.2-CTD became disengaged from the SUR1-ICLs (Figure 4(A); contact surface area ∼9 $Å^2$), which disrupted several interactions that had stabilized ATP binding, and reduced the ATP interface area to ∼380 $Å^2$. Specifically, R54, which was oriented towards the ATP in the CTD-up state became distant from the ATP binding pocket in the CTD-down structure. K39 in the N-terminus of neighboring Kir6.2, which also coordinated ATP binding in the CTD-up structure, was reoriented away from the ATP in the CTD-down structure (Figure 2(C), (D)). Moreover, the distance between K205 in the L0 of SUR1 and ATP increased in the CTD-down conformation. These

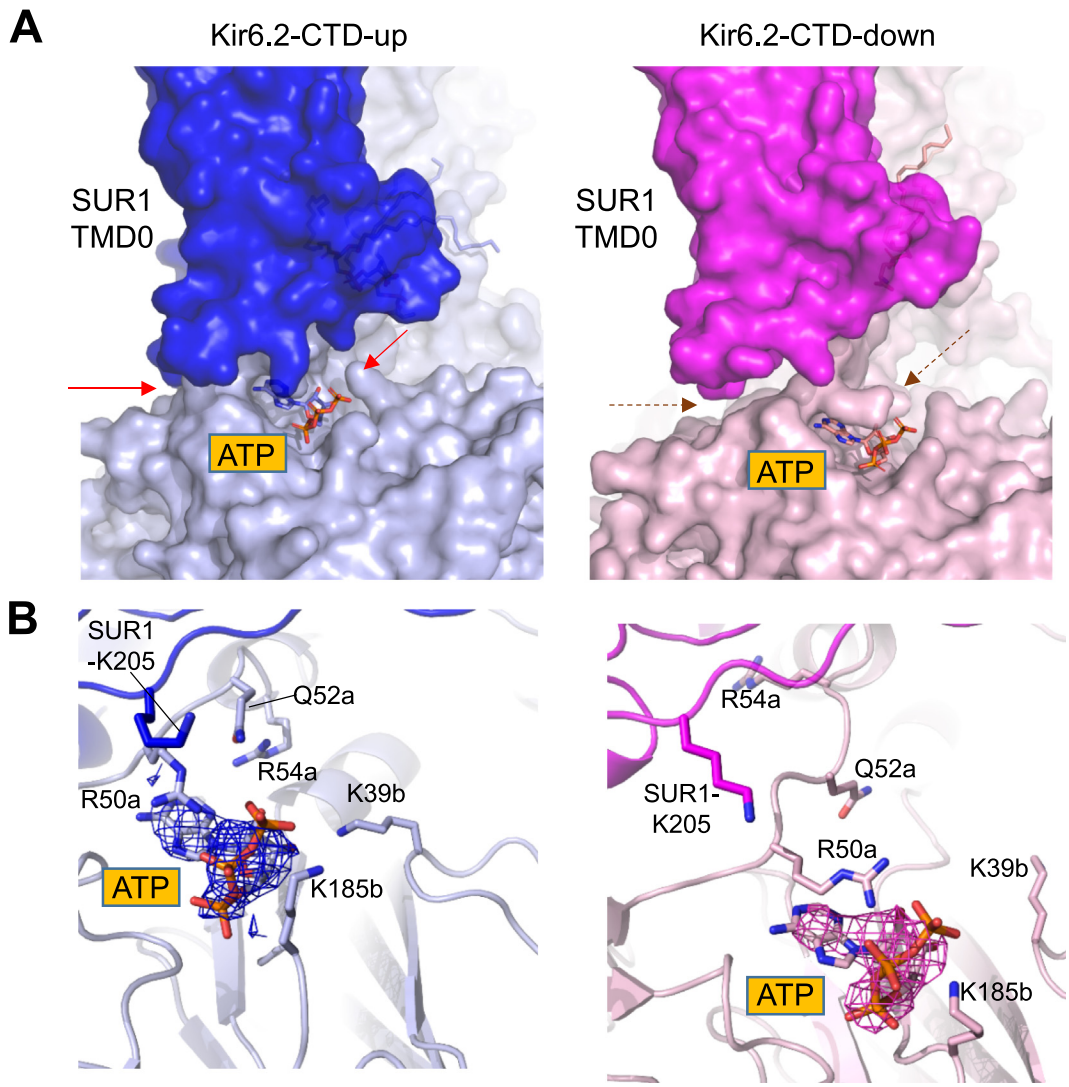


Figure 4. Comparison of ATP binding pocket and SUR1-TMD0/Kir6.2-CTD interface between Kir6.2-CTD-up and CTD-down conformations. (A) Surface representations of the SUR1-TMD0/Kir6.2-CTD interface and the ATP binding pocket in Kir6.2-CTD-up (left) and the CTD-down (right) conformations. SUR1 is shown in deep blue or pink hues, Kir6.2-CTD in pale blue or pink hues, and ATP as stick model. The arrows point to contacts between SUR1 and Kir6.2 in CTD-up panel which are lost (dashed arrows) in the CTD-down panel. Loss of the tight interaction between the SUR1-TMD0 and Kir6.2-CTD domains renders the ATP binding pocket less compact. **(B)** Detailed views of residues surrounding ATP in the Kir6.2 CTD-up conformation (blue, left) which become more distant from ATP and/or interacting partners in the CTD-down conformation (pink, right), including K205 of SUR1, Q52, R54 and K39 of Kir6.2. ATP cryoEM density (3.3 σ contour) is represented by a mesh.

changes together weakened the ATP binding pocket and exposed ATP to solvent. In addition to impacting Kir6.2-CTD dynamics, SUR1 rotation also affected ATP binding. When SUR1-ABC module rotated toward the Kir6.2 tetramer core (SUR1-in), L0 pulled away from the ATP binding pocket. As a result, SUR1-K205 lost interaction with ATP, destabilizing binding (Figure 4(B)).

PIP_2 binding site--At the binding pocket where PIP_2 is predicted to bind based on homology with Kir2 and Kir3 channels for which PIP_2 bound structures are available,[27,39,54] a lipid cryoEM density is seen in both Kir6.2-CTD-up and CTD-down conformations. Interestingly, the lipid density in the CTD-up conformation is significantly larger than that in the CTD-down conformation (Figure 5(A)). This was consistent for all datasets that include both conformations. We were able to fit, and tentatively model, the lipid density in the CTD-up structure with two phosphatidylserine (PS) molecules and that in the CTD-down structure with one PS molecule (Figure S5). Since no PIP_2 was added to our samples prior to imaging, we reasoned that the more abundant PS may have entered the binding pocket. In

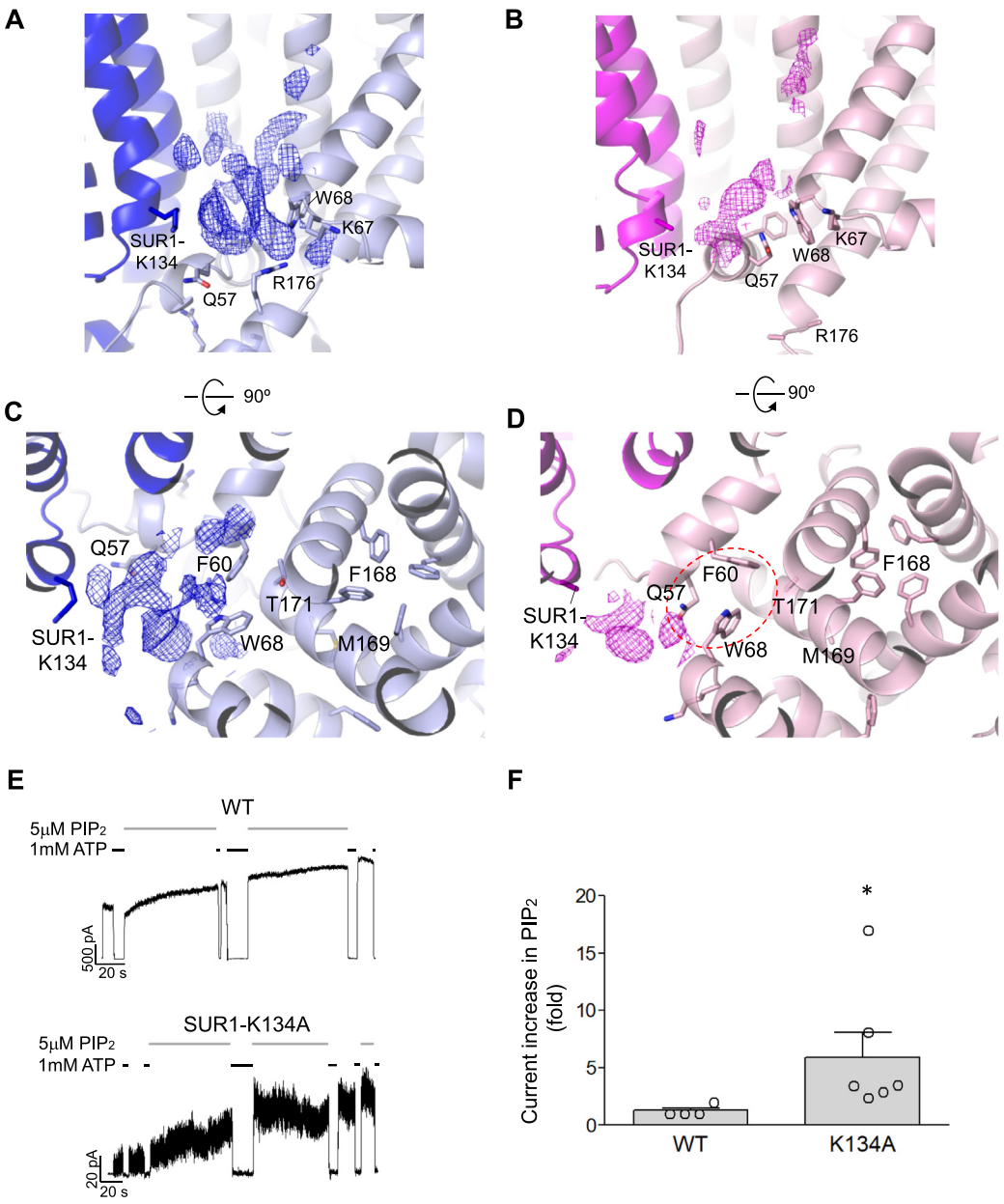


Figure 5. Comparison of the PIP_2 binding pocket in Kir6.2-CTD-up and CTD-down conformations. (A, B) PIP_2 binding pocket of Kir6.2-CTD up (A, blue) and Kir6.2-CTD down (B, pink) conformations viewed from the side. Lipid densities are shown as a mesh (1.0 σ contour). In (A), in addition to Kir6.2 residues previously implicated in phospholipid binding, SUR1-K134 side chain is pointed directly at the lipid density. SUR1 is shown in deep blue or pink, Kir6.2 in pale hues. **(C, D)** Same as (A, B) viewed from the extracellular side with the helical bundle crossing shown (F168). Note the PIP_2 binding pocket is more compressed in Kir6.2-CTD-down than CTD-up conformation due to secondary structural change at the IF helix that brings Q57 to interact closely with F60 and W68 (red dashed circle). **(E, F)** Inside-out patch-clamp recording (examples in E) show greater fold current increase in response to PIP_2 of the SUR1-K134A mutant channel than WT channel (left), with statistically significant difference (*$p < 0.05$, student's *t*-test).

a recent study by Zhao and MacKinnon,[81] it was shown that PIP_2 is not required for K_{ATP} channel activity, suggesting other phospholipids that occupy the PIP_2 binding site could potentially support channel activity. Whether the density in our structure represents PS, co-purified endogenous PIP_2, or other phospholipids requires further investigation.

In the Kir6.2-CTD-up structure, in addition to the set of Kir6.2 residues K67 and W68 in the outer helix, and R176 in the C-linker previously implicated in PIP_2 binding,[12,16,68] we found K134 in TMD0 of SUR1 was in close contact with the density corresponding to lipid headgroups (Figure 5 (A)). To test whether SUR1-K134 has a role in the

PIP_2 sensitivity of K_{ATP} channel opening, we functionally characterized a mutant K_{ATP} channel in which this residue is mutated to alanine (SUR1-K134A), using inside-out patch-clamp recording (see Methods). Compared to WT channels, the SUR1-K134A mutants exhibited substantially smaller initial currents in ATP-free solution. Upon PIP_2 addition, however, the mutant channel currents increased by 5.93 ± 2.17-fold, which is significantly higher than the 1.27 ± 0.23-fold current increase seen in WT channels (Figure 5(E), (F)), indicating the SUR1-K134A mutation reduced intrinsic P_o and PIP_2 interactions. It is well documented that channel P_o, determined by channel interaction with PIP_2, is primarily conferred by SUR1 association with Kir6.2. The Kir6.2 channel itself has low P_o, but co-expression with SUR1 or just the TMD0 domain of SUR1 increases channel P_o by more than 10-fold.[5,14,24,60] Our results show SUR1-TMD0 participates in PIP_2 interaction, at least in part through K134, which strengthens PIP_2 interactions with Kir6.2.

In the CTD-down structure, the IF helix was closer to W68 near the cytoplasmic end of the outer helix of the neighboring Kir6.2 than in the CTD-up structure, causing compression of the PIP_2 binding pocket (Figure 5(A)-(D)). This provides an explanation for why the lipid cryoEM density in the CTD-down structure was significantly weaker than that in the CTD-up structure and could be tentatively fit by only one PS molecule (Figure S5). Moreover, the simultaneous unwinding of the C-linker in the CTD-down structure withdrew the key PIP_2-interacting residue R176 to a position too distant for interaction. In this conformation, Kir6.2 is expected to be inactive.

Elucidating the relationship between ATP and PIP_2 binding by MD simulations

ATP and PIP_2 compete with each other to close and open the channel, respectively.[8,68] However, the structural mechanism underlying this functional competition has remained unresolved. Previous mutation-function correlation studies led to a proposal that ATP and PIP_2 have overlapping but non-identical binding residues.[16,67,72] To test this hypothesis, we conducted MD simulation studies using as a starting point the Kir6.2 (32–352) plus SUR1-TMD0 (1–193) tetramer part the RPG/ATP Kir6.2-CTD-up structure. Previous studies have shown that Kir6.2 and TMD0 of SUR1 form "mini K_{ATP} channels",[5,14] which like WT channels exhibit functional antagonism between ATP and PIP_2.[5,60] The mini K_{ATP} channel system is therefore suitable for simulating residues which may participate in binding of both ligands. Three independent 1 μs simulations were carried out for each of two ligand conditions, either without ATP or PIP_2 (apo), or with both ATP and PIP_2 in their respective binding pockets (ATP + PIP_2 (Figure S6(A), (B); for details see Methods).

Comparing the two different conditions, there was an overall increase in dynamics of the Kir6.2-CTD in the apo simulations versus the ATP + PIP_2 simulations (Figure 6, Figure S6, movie 2–5). First, significant secondary structural changes at the IF helix and the C-linker were observed in the apo simulations, resembling the changes from the CTD-up to the CTD-down conformation we observed in cryoEM structures. Second, the entire CTD relaxed towards the cytoplasm in the apo simulations. This was quantified by measuring the distance between the helix bundle crossing (HBC) gate at F168 and the G-loop gate at G295 (Figure 6(A)). In apo simulations, this distance increased over time in all three runs, whereas it remained relatively unchanged for the ATP + PIP_2 simulations (Figure 6(B)), except in run 2 during which the distance increased when ATP became partially dissociated at around 500 ns (Figure 6 (B)). These findings show that in the absence of ligands, the Kir6.2-CTD has a tendency to relax toward the CTD-down conformation.

Analysis of the minimum distance between each of the ligands and their surrounding residues within 4 Å over the entire simulation revealed that K39 and R54 of Kir6.2 engaged in both ATP and PIP_2 binding. Figure 6(D) shows the fraction of time over the entire simulation each residue in each subunit and each run came into contact with ATP or PIP_2. R54 and K39 each showed partial ATP and PIP_2 occupancy (Figure 6(C), (D), Figure S6(C), (D), movie 4, 5), which was in contrast to well established ATP binding residues, including R50 and K185, and PIP_2 binding residues K67 and R176, which showed nearly 100% ATP or PIP_2 occupancy. Of the two dual occupancy residues, R54 showed greater interactions compared to K39 with both ATP and especially with PIP_2. K39, while showing interaction with ATP in all three runs, only showed significant interaction with PIP_2 in one of the three runs (Figure S6(D)). The analysis also identified residues that had specific, although less stable, interactions with either ATP or PIP_2 as defined by distance between residue and ligand < 4 Å. In particular, Kir6.2-Q52 specifically interacted with ATP, and SUR1-K134 with PIP_2, contrasting with the dual ligand binding mode of R54 and K39. A previous mutagenesis study has shown that mutation of either K39 or R54 to alanine reduces channel open probability as well as sensitivity to ATP inhibition,[16] which early implicated a role for these residues in channel gating by PIP_2 and ATP. Confirming a role in physiological regulation, mutations R54C and R54H are linked to congenital hyperinsulinism[18] and mutation K39R to transient neonatal diabetes.[80] Our MD simulation results sug-

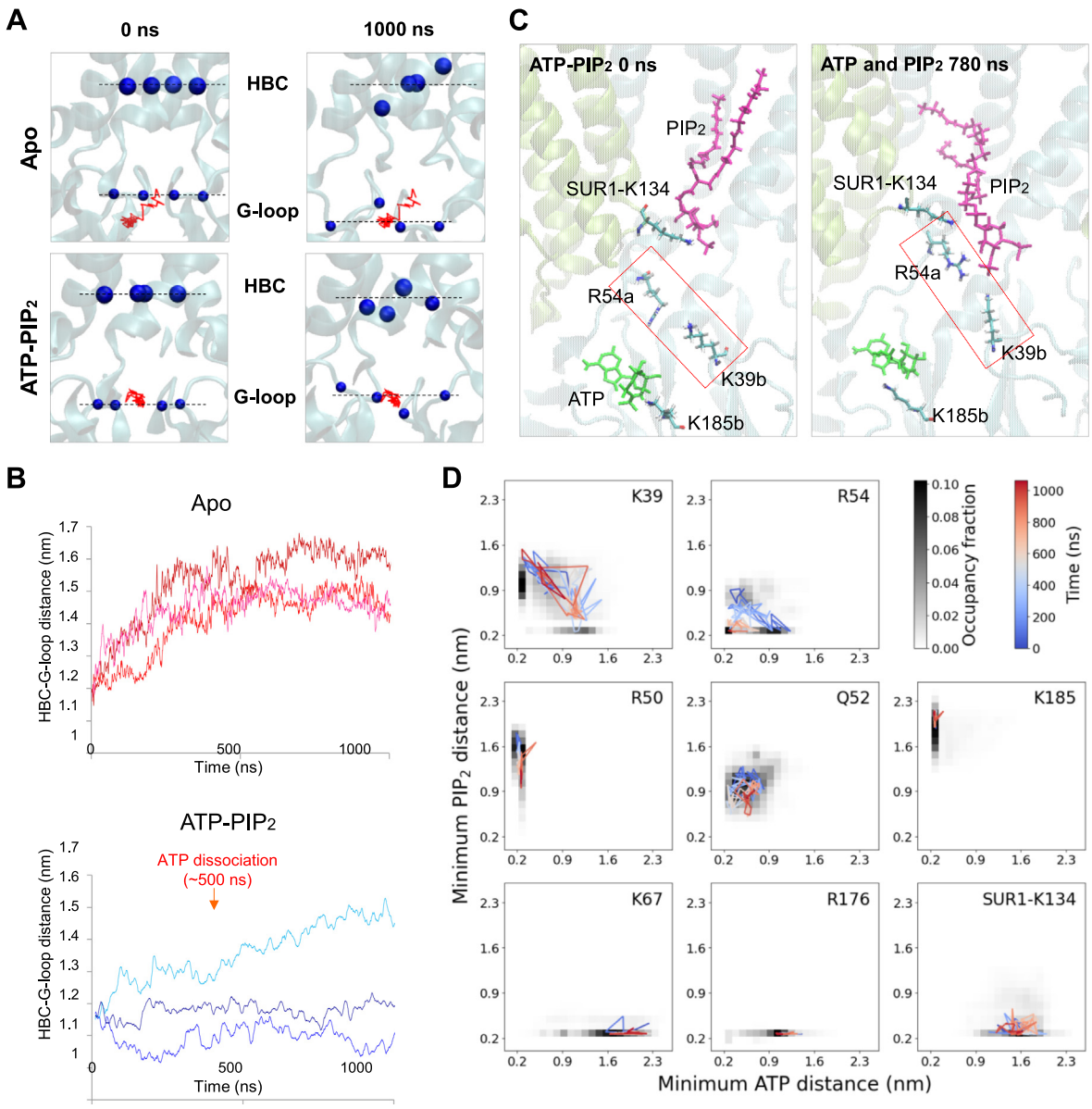


Figure 6. MD simulations of Apo versus ATP + PIP_2 state. (A) Positioning of the helix bundle crossing (HBC) and G-loop gate in a representative apo-simulation and ATP-PIP_2-simulation. The time-varying position of the geometric center (plotted in red) of G295 (G-loop gate) Cα carbons of all four chains is overlaid on beginning and end snapshots of G295 Cα carbons, after aligning trajectories to the Kir6.2 TM domain. **(B)** Plots of the distance between the geometric centers of all four Cα atoms of F168 and of G295 for the entire simulations. Greater distances between the two gates in the apo state compared to the ATP + PIP_2 state indicate relaxation of the CTD in the absence of ligands. In one apo simulation (light blue), ATP became partially dissociated at around 500 ns (red arrow). **(C)** Snapshots of simulations in the presence of ATP and PIP_2 showing interactions of R54 and K39 with either ATP or PIP_2. Kir6.2-K185 and SUR1-K134 which only interact with ATP or PIP_2 respectively are also shown. **(D)** Heatmaps showing fraction of time residues spend at increasing distances from ATP (horizontal axis) or PIP_2 (vertical axis). A representative trajectory corresponding to chain 1 in run 1 and colored to show time evolution is shown for each residue. K39 and R54 (top row) exhibit switching behavior and occasional simultaneous contact, while essentially exclusive ATP binding residues are shown in the middle row, and PIP_2 binding residues in the bottom row.

gest both residues participate directly in ATP and PIP_2 binding, providing mechanistic insight into how mutation of these residues affect PIP_2 and ATP sensitivities and cause disease.

Discussion

Cryo-preserved purified protein samples may contain multiple protein structures that represent

distinct functional or transitional states and can provide mechanistic insight.[55] In this study, analysis of five K_{ATP} channel cyroEM datasets collected in different ligand conditions revealed conformational heterogeneity of the Kir6.2-CTD and SUR1 ABC module. We observed the Kir6.2-CTD in either an "up" position tethered close to the plasma membrane, or a "down" position corkscrewed away from the membrane towards the cytoplasm. The ratio of the two conformations correlated with occupancy of inhibitory ligands at the SUR1 and Kir6.2 binding sites (Figure 1), suggesting inhibitory ligands help stabilize the Kir6.2-CTD close to the membrane. Furthermore, in both Kir6.2-CTD conformations the SUR1 ABC module was observed oriented with a range of rotation around the Kir6.2 tetramer central axis (Figure S4). We observed a restructuring of protein–protein and protein–ligand interfaces in different conformations that sheds light on how ligands shift channel conformational dynamics to regulate gating.

Correlation between Kir6.2-CTD conformation and channel function

The structures analyzed in this study all represent closed channels. Recently, an open human K_{ATP} channel structure containing Kir6.2 C166S and G334D mutations was reported,[81] which showed a Kir6.2-CTD that is further CW rotated (extracellular view) and slightly upward translated compared to our Kir6.2-CTD-up conformation (Figure 1(C)). Rearrangement of the molecular interactions between the IF helix, the C-linker, and TM residues as well as increased distance between the pre-IF helix loop and SUR1 L0 compared to the ATP-bound closed WT channel structure (equivalent to our Kir6.2-CTD-up structure) were observed (Figure 7). The restructuring widens the ATP-binding pocket, explaining the absence of ATP cryoEM density despite high concentrations of ATP in the sample, and stabilizes HBC gate opening via side chain interactions between F60 in the IF helix and the HBC gate residue F168. Another pre-open K_{ATP} channel structure using a rat SUR1-39aa-Kir6.2 H175K fusion construct published while this manuscript was under review[76] showed similar structural characteristics as the open channel structure. Taken together, a picture emerges wherein rotational and translational position of the Kir6.2-CTD determines the functional state of the channel. When the CTD is in the down position, the phospholipid binding pocket is compressed due to a change

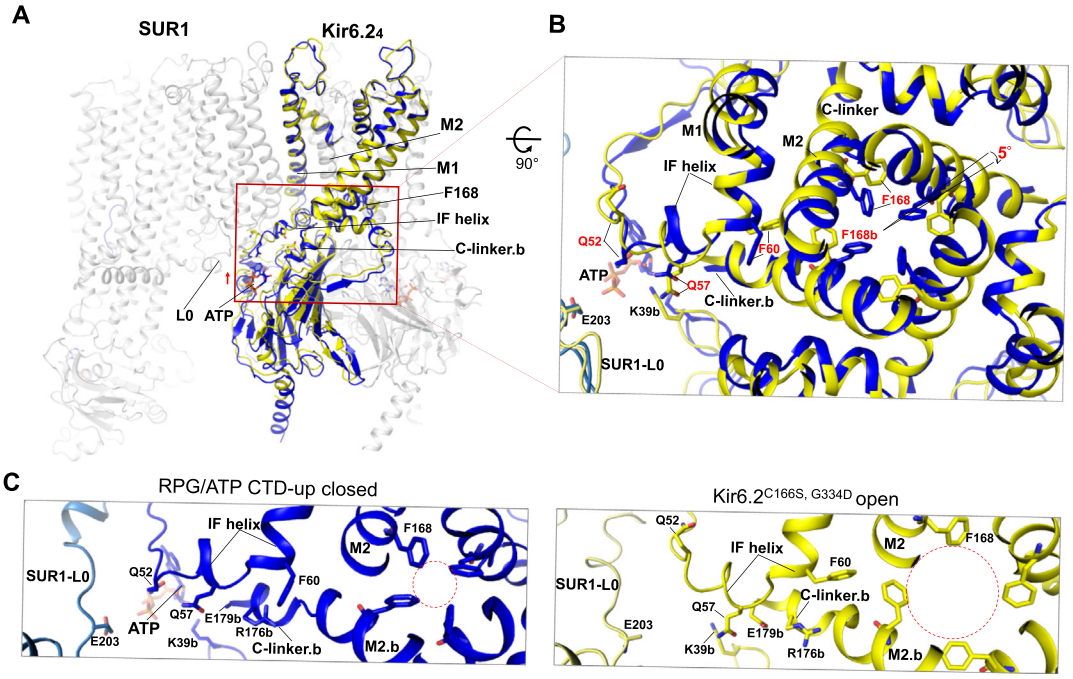


Figure 7. Comparison of an inhibitor-bound CTD-up closed structure and a mutant open structure. (A) Overlay of RPG + ATP CTD-up structure (blue) with a Kir6.2 double mutant (C166S, G334D) structure (PDB:7S61; yellow). Only two Kir6.2 subunits are colored. A small upward translation of the mutant open structure relative to the inhibitor-bound closed structure is indicated by the small red arrow next to the red box, which is shown in a 90^{O} rotated enlarged view in B. (B) Overlay of the Kir6.2 tetramer with part of the SUR1-L0 viewed from the extracellular side. A CW rotation (5^{O}) of the CTD from the inhibitor-bound closed structure to the mutant open structure is noted. Structural elements and residues from chain b are labeled with b at the end to distinguish from those from chain a. Residues which show significant differences in the two structures are labeled in red. (C) Same as B but with the two structures shown separately and residues in the C-linker.b visible. The red dashed circles illustrate the enlargement of the potassium ion path at the helix bundle crossing (F168) in the open structure.

in secondary structures of the IF helix (Figure 5) and the C-linker is unwound by the increased separation of the CTD from the membrane domain. We propose this conformation corresponds to an "inactivated" state in which the CTD is unable to engage with membrane phospholipids and thus open the channel. When the CTD is in the clockwise upscrewed position with ATP and/or drug bound, the channel is primed for opening but is arrested in an inhibited state due to an interaction network between SUR1-L0 and the Kir6.2-N terminal domain, an interaction stabilized by ATP that prevents further rotation of the CTD needed to open the HBC gate. Upon ATP dissociation, the CTD is released for further CW rotation, enabling the gate to widen and open the channel.

To explain our structural data and a wealth of electrophysiological data we propose the Kir6.2-CTD undergoes transitions between four principal conformation states in dynamic equilibrium: CTD-down inactivated, CTD-up unliganded and closed, CTD-up bound to inhibitory ligands, and CTD-up open, with the probability to occupy a given conformation driven by ligands (Figure 8), similar to the model previously proposed by Borschel et al..[10] In the absence of ligands the CTD-down conformation dominates, as seen in our apo state dataset, and the channel is inactivated. While not observed in our structural studies here, inactivated channels can transition into a short-lived unliganded CTD-up conformation at low probability. Binding of physiological inhibitor ATP at Kir6.2, and/or a pharmacological inhibitor at SUR1, shifts Kir6.2 towards a stable CTD-up but closed conformation. Binding of phospholipids such as PIP_2, when coupled with unbinding of inhibitory ligands, shifts the equilibrium towards the Kir6.2-CTD-up open position. Under physiological conditions with high intracellular ATP concentrations and ambient phospholipids, we expect the Kir6.2-CTD to be mostly in an ATP inhibited CTD-up conformation, with a small fraction in a CTD-up state having the phospholipids bound and a further-rotated-open conformation, and rarely in an unliganded CTD-up conformation; the CTD-down inactivated conformation would also be rare. However, in pathological conditions the CTD-down inac-

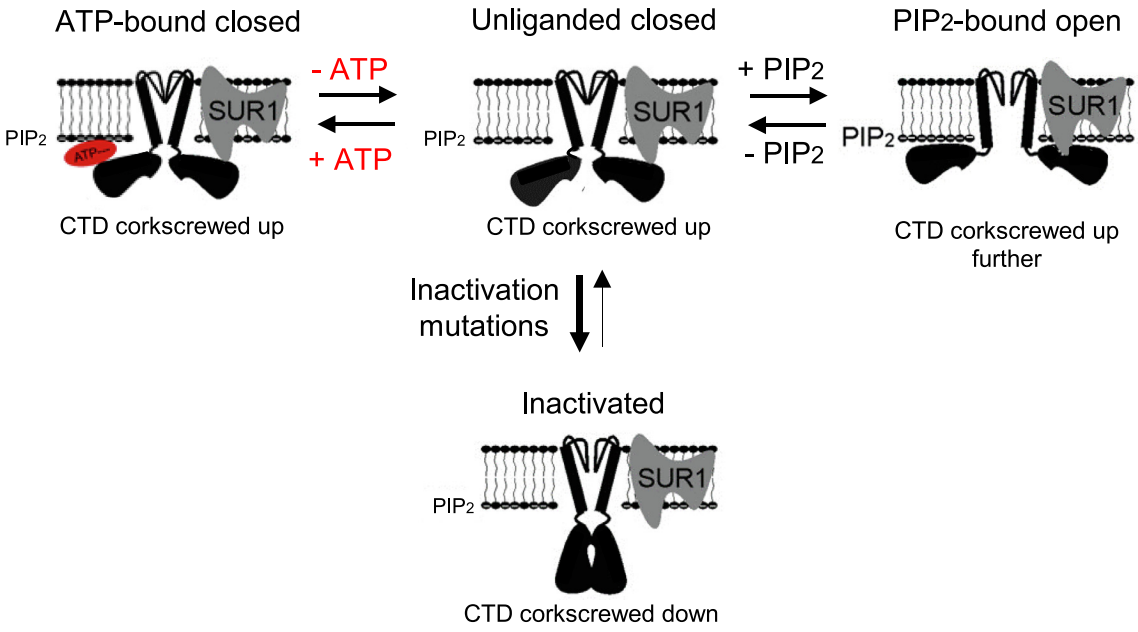


Figure 8. Correlating Kir6.2-CTD structures with functional states of pancreatic K_{ATP} channels. Cartoon representation of the structural and functional states of K_{ATP} channels. In the presence of high concentrations of intracellular ATP and ambient PIP_2, channels are mostly in ATP-bound closed state in which the Kir6.2-CTD is in the up-conformation and docked near the membrane and rotated CW from an extracellular perspective. Removal of ATP results in a transient unliganded closed state with the CTD in the up position conducive to binding PIP_2. Binding of PIP_2 opens the channel in which the Kir6.2-CTD is further CW rotated and moved up towards the membrane. Inactivation occurs when the CTD in the unliganded state transitions into the down conformation, a process that is enhanced by inactivation mutations including those known to cause hyperinsulinism. ATP facilitates channel recovery from inactivation by shifting the equilibrium towards the ATP-bound closed state in which the Kir6.2-CTD is in the up-conformation to allow the channel to transition to the unliganded CTD-up closed state upon subsequent removal of ATP, primed for PIP_2 binding and channel opening. Addition of PIP_2 also counters inactivation by shifting the equilibrium via the unliganded CTD-up closed state towards PIP_2 bound open state. Note the unliganded, CTD-up closed state shown to account for functional and kinetic modeling data in the literature is likely short-lived and its structure is yet to be captured by cryoEM.

tivated state could be prevalent. We and others have previously reported several mutations including congenital hyperinsulinism-causing mutations at the Kir6.2 subunit-subunit interface (such as R192A, E229A, R314A, R301A/C/H) that promote channel inactivation.[10,43–44,67] Channels containing such mutations briefly open upon patch excision into ATP-free solution but then quickly inactivate. Interestingly, inactivation can be overcome by transiently exposing channels to high concentrations of ATP, followed by washout of ATP. We propose that these mutations increase an energy barrier for Kir6.2-CTD to transition from the CTD-down conformation to CTD-up conformation, thus trapping the CTD in the down inactivated state. ATP thus effectively acts as a molecular glue at its Kir6.2 domain interfaces. Exposure to ATP shifts the Kir6.2-CTD back to the CTD-up position such that channels can open again when ATP is washed out. The model similarly explains the ability of PIP_2 to prevent and reverse inactivation mutants from inactivation[10,43–44,67] by stabilizing Kir6.2-CTD in the up and further rotated open position.

The Kir6.2-CTD-up and CTD-down conformations observed in our structures are similar to the T and R states observed in a SUR1-39aa-Kir6.2 fusion channel alternatively bound to GBC + ATPγS, ATPγS alone, or MgADP by Wu et al..[78] However, the percentage of particles in the T-state (corresponding to our Kir6.2-CTD-up state) in their ATPγS + GBC or ATPγS datasets is ~40% and 43% respectively, which differ significantly from the ~93% and 22% in our GBC + ATP and ATP alone datasets. Compared to our CTD-up state in the ATP condition, the higher percentage of T-state particles in the fusion channel's ATPγS condition could potentially derive from the 10-fold higher concentrations of ATPγS used to generate the Wu et al. structure. An explanation for the markedly higher percentage of Kir6.2-CTD-up state particles in our GBC + ATP condition, over T-state particles in their GBC + ATPγS condition is not obvious. In principle, however, the extra 39aa linker between SUR1 C-terminus and Kir6.2 N-terminus in the fusion construct could uncouple drug binding from Kir6.2-CTD conformation. Consistent with this, GBC was shown to be ineffective in inhibiting the fusion channel in contrast to the WT channel formed by separate SUR1 and Kir6.2 proteins.[78]

Previous K_{ATP} channel cryoEM studies have provided evidence that KNtp interacts with the central cavity of the SUR ABC core.[21,47,69,78] The structures presented here refines our view of the molecular interactions between the KNtp and different parts of SUR1. We have previously shown that engineered crosslinking between Kir6.2-L2C and SUR1-C1142 reduces channel activity.[47] A likely scenario is that stapling KNtp along SUR1 via the contact sites we observe stabilizes the inhibited Kir6.2-CTD-up conformation and prevents further rotation of the CTD needed to open the channel. This interpretation can explain why deletion of KNtp increases channel open probability,[6,36,64] while drugs such as GBC, RPG and CBZ, which stabilize KNtp in the transmembrane cavity of the SUR1 ABC module, mimic the physiological inhibitor ATP and block channel activity.[20]

Comparison to other Kir channels

Differential proximity of the CTD to the membrane has also been reported in Kir2[27,39,79] and Kir3,[53] suggesting there may be a common theme in Kir channel conformation and gating transitions. Supporting this, mutations which disrupt cytoplasmic domain subunit interface in Kir2.1 also reduce channel activity, akin to the inactivation mutations reported in Kir6.2.[10] Distinctively however, unlike Kir2 and Kir3 channels, K_{ATP} channels have an additional ATP-bound Kir6.2-CTD-up conformation stop between the CTD-down inactivated and CTD-up open conformations, which allows for a rapid and reversible inhibition of the channel in response to metabolic signals.

Another unique feature of Kir6.2 channels is the requirement of SUR1 co-assembly to achieve the high ATP sensitivity and open probability of native K_{ATP} channels.[30,73] Several studies have now provided functional, biochemical and structural evidence that SUR1 directly participates in ATP binding via K205 in the L0 linker.[21,61,74] SUR1 increases the open probability of Kir6.2 by more than 10-fold, an effect that is largely mediated by TMD0.[5,14] We show in our structure that K134 in SUR1-ICL2 is oriented towards the lipid headgroup density in the PIP_2 binding pocket, suggesting SUR1-TMD0 increases channel open probability by directly contributing to binding of PIP_2 or other phospholipids. Supporting this, MD simulations show K134-PIP_2 interactions (Figure 6(D), Figure S6(D)) and functional experiments show that mutation of SUR1 K134 to alanine reduces channel P_o (Figure 5(E), (F)).

Structural basis of ATP and PIP_2 antagonism

ATP and PIP_2 functionally compete to inhibit and activate K_{ATP} channels, respectively.[8,68] Molecular dynamics (MD) simulations reveal that two Kir6.2 residues, K39 and R54, interact with both ATP and PIP_2, providing evidence that competition for binding residues between the two ligands underlies, at least in part, functional competition between ATP and PIP_2. Interestingly, in one of the ATP + PIP_2 simulation runs, ATP dissociates from its binding pocket (Figure 6(B)). This dissociation event is likely captured because SUR1-L0, which contains the ATP stabilizing residue K205 is not included in the simulation structure. It offers a view of how ATP may dissociate such that binding residues shared between ATP and PIP_2 have greater freedom to interact with PIP_2, favoring channel opening. The rotational movement of SUR1 towards the Kir6.2-

tetramer observed in our multibody refinement analysis (Figure S4) increases the distance between SUR1-K205 and bound ATP, which may initiate ATP dissociation by weakening ATP binding and thus provide a pathway for channel transition from ATP bound inhibited state to PIP_2 bound open state.

In summary, the structural analysis, MD simulations, and functional studies presented here together with the recent open K_{ATP} channel structure reported by others[81] offer insight into several longstanding questions on K_{ATP} channel gating mechanisms. A principal unresolved question regards the full extent of the conformational dynamics of the SUR1 subunit and how it may relate to channel function. The large rotation of the SUR ABC module that leads to a quatrefoil channel conformation has previously been reported in a human SUR1-6aa-Kir6.2 fusion protein channel in which the NBDs are bound to MgATP/MgADP and dimerized.[38] Recently, a similar large rotation is reported in a SUR2B/Kir6.1 vascular K_{ATP} channel bound to GBC and ATP.[69] However, the quatrefoil conformation is not observed in the most recent human SUR1 NBDs dimerized-Kir6.2 mutant open channel.[81] Whether the variable findings stem from protein preparation methods or involve differences in data processing will be important to resolve in order to fully understand K_{ATP} channel structure and function relationship.

Methods

Image processing and particle classification

CryoEM images of pancreatic K_{ATP} channels (co-assembled from hamster SUR1 and rat Kir6.2) collected in different liganded conditions in our previous publication[47] were reprocessed from the initial 2D classification step that we described previously using RELION-3.1.[82] Classes displaying fully and partially assembled complexes with high signal/noise were selected. The particles were re-extracted at 1.045 Å/pix for RPG/ATP, 1.399 Å/pix for GBC/ATP, 1.72 Å/pix for CBZ/ATP and ATP only, and 1.826 Å/pix for apo state datasets, and then used as input for 3D classification in RELION-3.1. Figure S1 shows the data processing workflow for the RPG/ATP dataset. Channel particles refined in the final C4 reconstruction (150,707 total particles) were subjected to C4 symmetry expansion and yielded 4-fold more copies. Further refinement was performed without symmetry restraints or masking such that possible heterogeneous particles can be aligned without any restraints. 2D class averages from all data sets showed significant heterogeneity of the SUR1-ABC module, indicating dynamic SUR1 motions were captured during vitrification of the cryoEM samples. To probe potential novel conformations due to dynamics of SUR1-ABC module relative to the Kir6.2 tetramer, or for novel conformations that arise due to dynamic motions of other domains of the K_{ATP} channel, a soft mask that includes the Kir6.2 tetramer and one SUR1 in a propeller form was created in Chimera using our previously published model (PDB:6BAA),[46] and extensive focused 3D classification was performed without particle alignment. This revealed two major classes with different Kir6.2-CTD conformations that are either anchored up towards the plasma membrane (CTD-up) or extended down further towards the cytoplasm (CTD-down).

Focused refinement of SUR1 was carried out after partial signal subtraction that removed signals outside the masked region, followed by further 3D classification without alignment at higher regularization T values (ranged from 6 to 20) and local refinement of signal subtracted particles.[65] Extensive 3D classification sorted out remaining minor groups of particles that did not align well with the propeller conformation but no other conformations emerged. The dominant class then underwent three iterations of CTF refinement and 3D refinement. To test whether some of the SUR1 particles adopt quatrefoil-like conformation reported previously, a mask that includes the Kir6.2 tetramer and one SUR1 was also created using a quatrefoil-like model of our previously published Kir6.1/SUR2B structure (PDB:7MJO),[69] which was used for classification following the same scheme described for the propeller form mask. A minor quatrefoil form at 7.1 Å overall resolution was identified from the RPG/ATP dataset (Figure S1). Final maps were subjected to Map-modification implemented in *Phenix* with two independent half maps and corresponding mask and model as input. They were then sharpened with model-based auto sharpening with the corresponding model using *Phenix*, a step that was iterated during model building.

The same workflow was used to process the other four datasets, GBC/ATP, CBZ/ATP, ATP only and apo states. With the exception of the apo dataset which yielded only the CTD-down conformation, all other datasets showed both Kir6.2-CTD classes similar to those identified in the RPG/ATP dataset but with varying ratios of the two conformations. Particle distributions and final map resolutions for all datasets are summarized in Table S1. Upon carrying out this further analysis, we noted the CBZ/ATP dataset previously reported to be collected in the presence of 10 μM CBZ and 1 mM ATP did not yield a map with clear ATP density at the Kir6.2 ATP binding site. Upon inspection of the ATP used it was discovered that the concentration had been mistakenly reported as 1 mM rather than 0.1 mM, which likely explained the lack of clear ATP cryoEM density.

Multibody refinement

For Kir6.2 tetramer multibody refinement, particles pooled from both the RPG/ATP CTD-up

and CTD-down classes and also each class separately were used.[50] To interrogate the Kir6.2 CTD movements relative to its TM domain, we masked out the SUR1 density and assigned the Kir6.2 TM domain (58–173) and CTD (174–352) as two separate rigid bodies (Figure S3(A)). Principal component analysis showed that a dominant eigenvector accounted for 25.3% of the overall variance (Figure S3(B)). Histograms of the amplitudes along this eigenvector shows a bimodal distribution with two peaks, indicating two conformationally distinct populations differing in the distance between the CTD and the TM domain and rotation of the CTD as expected. Rotation and rocking motions of the CTD were also observed within the Kir6.2-CTD-up and Kir6.2-CTD-down class particles. Although similar in degrees of motion, greater heterogeneity was seen in the CTD-down population of particles than in the CTD-up (Figure S3(C), (D)), consistent with increased mobility of the CTD when extended away from the membrane.

For $(Kir6.2)_4$-SUR1 multibody refinement, the map was divided into three bodies: body 1, Kir6.2-tetramer (E30-D352); body 2, ABC-core (Q211-V1578) of SUR1 plus KNtp (M1-E19) of Kir6.2; and body 3, TMD0 (M1-L210) of SUR1 (Figure S4(A)). Multibody refinement was repeated with varying standard deviations for the degrees of rotation, and pixels of translation, to rule out artifacts. Principal component analysis in the *relion_flex_analyse* program revealed that approximately 17.5% of the variance is explained by the dominant eigenvector 1 (Figure S4(B), (C)) corresponding to horizontal swinging motion of SUR1.[50] We then used the program to generate two separate STAR files, each containing ~35,000 particles with amplitudes less than −5 or greater than +5 along eigenvector 1 (Figure S4(D)). These two sets of particles were further refined using a soft mask, yielding two maps which we refer to as SUR1-out and SUR1-in with an overall resolution of 3.8 and 3.9 Å, respectively.

Model building and refinement

The RPG/ATP dataset yielded the highest resolution maps and were used for modeling (Table S2). Initial models for the $Kir6.2_4$-SUR1 channel were obtained by docking Kir6.2-TMD (32–171) and Kir6.2-CTD (172–352) from our previously published model (PDB:6BAA),[48] and TMD0/L0 (1–284), TMD1 (285–614), NBD1 (615–928), NBD1-TMD2-linker (992–999), TMD2 (1000–1319) and NBD2 (1320–1582) of SUR1 (PDB:6PZA),[47] into either the RPG-CTD-up or the RPG-CTD-down cryoEM density map by rigid-body fitting using Chimera's 'Fit in' tool.[58] Then different domains were combined using Chimera to form a composite model for further refinement. We used Coot to manually build and edit residues 32–78 and residues 79–361 at the interface of two Kir6.2 subunits[23], we then copied those changes to each of the other four Kir6.2 subunits. Further edits and refinements were done independently to each chain without strict NCS restraints. The models were then iteratively built and refined in Coot[23] and Phenix,[1] with Ramachandran restraints, secondary structure restraints, and side-chain rotamer restraints. The N-terminus of Kir6.2 (residues 1–31, KNtp) had sufficient continuous density and the density was sufficiently clear to allow modeling of several key interactions with SUR1, although accurate modeling of side chains was not possible for the entire KNtp. Similarly, the NBDs of SUR1 and particularly NBD2 had weaker density than most of the reconstructed map, imposing reliance on restraints and prior models. In addition to modeling the protein, ATP was modeled at the inhibitory ATP-binding site on each of the four Kir6.2 subunits, and at the nucleotide binding site in NBD1 of SUR1 which had sufficient cryoEM density. Phosphatidylserine, phosphatidylcholine, and phosphatidylethanolamine were modeled liberally into plausible lipid density; 17 lipids were modeled for the RPG-CTD-up model and 13 lipids modeled for the RPG-CTD-down model.

MD simulations and analysis

All MD simulations were performed at all-atom resolution using AMBER 16[13] with GPU acceleration. Initial coordinates were developed from the RPG/ATP-CTD-up model including four Kir6.2 (32–352) and four SUR1-TMD0 (1–193) without the SUR1-ABC core. ATP in the cryoEM structure was removed for simulations in the apo condition. For simulations in the presence of ATP and PIP_2, the ATP from the cryoEM structure was kept, and a PIP_2 molecule (DMPI24, di-myristoyl-inositol-(4, 5)-bisphosphate) was docked in the PIP_2 binding pocket using Kir3.2-PIP_2 structure (PDB ID: 6 M84) as a template.

The simulation starting structures were protonated by the H++ webserver (https://biophysics.cs.vt.edu/H++) at pH 7 and inserted in a bilayer membrane composed of 1-palmitoyl-2-oleoyl-phosphatidylcholine (POPC) lipids and surrounded by an aqueous solution of 0.15 M KCl. The optimal protein orientations in the membrane were obtained from the OPM database.[45] All systems contain 650–680 POPC lipids and ~87,000 water molecules, resulting in a total of ~385,000 atoms. They were assembled using the CHARMM-GUI webserver,[33,37,77] which also generated all simulation input files.

The CHARMM36m protein[29] and CHARMM36 lipid[35,57] force field parameters were used with the TIP3P water model.[34] Langevin dynamics[56] were applied to control the temperature at 300 K with a damping coefficient of 1/ps. Van der Waals (vdW) interactions were truncated via a force-based switching function with a switching distance of 10 Å and a cutoff distance of 12 Å. Short-range Coulomb interactions were cut off at 12 Å, long-

range electrostatic interactions were calculated by the Particle-Mesh Ewald summation[17,25]. Bonds to hydrogen atoms were constrained using the SHAKE algorithm.[32]

The atomic coordinates were first minimized for 5000 steps using the steepest-descent and conjugate gradient algorithms, followed by a ~2 ns equilibration simulation phase, during which dihedral restraints on lipid and protein heavy atoms were gradually removed from 250 to 0 kcal/mol/Å^2, the simulation time step was increased from 1 fs to 2 fs, and the simulation ensemble was switched from NVT to NPT. To keep the pressure at 1 bar, a semi-isotropic pressure coupling was applied that allows the z-axis to expand and contract independently from the x-y plane.[49] The simulations were then run for over 1 μs with a time step of 4 fs enabled by hydrogen mass repartitioning.[7,28]

Analysis of ATP/PIP$_2$ occupancy

ATP and PIP$_2$ residue occupancies (Figure 6(D)) were computed using the histogram of minimum hydrogen bond lengths between each residue and PIP$_2$/ATP, to show the amount of time spent at different distances. Summarized residue occupancies (Figure S6(D)) were calculated as the fraction of time each residue spent in contact with ATP/PIP$_2$, where contact is defined as a minimum hydrogen bond length of below 4 Å. For both ligands, minimum bond lengths were used regardless of which pairs formed the bond. A 10 ns window average was used to smooth the minimum bond length time series data.

Functional studies

Point mutation SUR1-K134A was introduced into hamster SUR1 cDNA in pECE using the QuikChange site-directed mutagenesis kit (Stratagene). Mutation was confirmed by DNA sequencing. For electrophysiology, wild-type or mutant SUR1 cDNA and rat Kir6.2 in pcDNA1 along with cDNA for green fluorescent protein GFP (to facilitate identification of transfected cells) were co-transfected into COS cells using FuGENE®6, and plated onto glass coverslips 24 hours after transfection for recording, as described previously.[47] Recording pipettes were pulled from non-heparinized Kimble glass (Fisher Scientific) on a horizontal puller (Sutter Instrument, Co., Novato, CA, USA). Electrode resistance was typically 1–2 MΩ when filled with K-INT solution containing 140 mM KCl, 10 mM K-HEPES, 1 mM K-EGTA, pH 7.3. ATP was added as the potassium salt. PI4,5P$_2$ (Avanti Polar Lipids) was reconstituted in K-INT solution at 5 μM and bath sonicated in ice water for 20 min before use. Inside-out patches of cells bathed in K-INT were voltage-clamped with an Axopatch 1D amplifier (Axon Inc., Foster City, CA). Exposure of membrane patches to ATP- or PIP$_2$-containing K-INT bath solution was as specified in Figure 5 legend. All currents were measured at room temperature at a membrane potential of −50 mV (pipette voltage = +50 mV) and inward currents shown as upward deflections. Data were analyzed using pCLAMP10 software (Axon Instrument). Off-line analysis was performed using Microsoft Excel programs. Data were presented as mean ± standard error of the mean (S.E.M) and statistical analysis was performed using two-tailed student's *t*-test, with $p < 0.05$ considered statistically significant.

DATA AVAILABILITY

Coordinates and cryoEM density maps for K_{ATP} channel models of the Kir6.2 tetramer core plus one SUR1 subunit have been deposited to the Protein Data Bank and the Electron Microscopy Data Bank with accession numbers as follows: RPG/ATP Kir6.2-CTD-up (PDB code 7TYS, EMDB code EMD-26193); RPG/ATP Kir6.2-CTD-down (PDB code 7TYT, EMDB code EMD-26194); RPG/ATP Kir6.2-CTD-up SUR1-in (PDB code 7U1Q, EMDB code EMD-26303); RPG/ATP Kir6.2-CTD-up SUR1-out (PDB code 7U1S, EMDB code EMD-26304); GBC/ATP Kir6.2-CTD-up (PDB code 7U24, EMDB code EMD-26307); GBC/ATP Kir6.2-CTD-down (PDB code 7U6Y, EMDB code EMD-26308); CBZ/ATP Kir6.2-CTD-up (PDB code 7U7M, EMDB code EMD-26309); CBZ/ATP Kir6.2-CTD-down (PDB code 7U2X, EMDB code EMD-26321); ATP-only Kir6.2-CTD-up (PDB code 7UAA, EMDB EMD-26312); ATP-only Kir6.2-CTD-down (PDB 7U1E, EMDB code EMD-26299); Apo Kir6.2-CTD-down (PDB code 7UQR, EMDB code EMD-26320). Molecular dynamics trajectories are available at Zenodo, https://zenodo.org/record/7017342 with DOI 10.5281/zenodo.7017342. All other study data are included in the article and supplementary data.

Acknowledgements

We thank Zhongying Yang for technical support and Assmaa ElSheikh for helpful discussions. The originial cryoEM datasets used for analaysis in the current manuscript were collected at the Multi-Scale Microscopy Core at the Oregon Health and Science University. The project was supported by the National Institutes of Health grant R01DK066485 (to S.-L. S.) and the National Science Foundation grant MCB 2119837 (to D.M. Z.).

Author contributions

MWS performed image analysis, atomic modeling, MD simulation analysis, prepared figures and wrote the manuscript. CMD performed atomic modeling, analyzed data, prepared figures

and wrote the manuscript. BM performed MD simulations. JDR analyzed MD simulation data, prepared figures and contributed to manuscript preparation. BLP contributed to discussion of the project and edited the manuscript. DMZ provided guidance to the design and analysis of MD simulation studies and edited the manuscript. SLS conceived the project, performed electrophysiology experiments, analyzed data, prepared figures and wrote the manuscript.

Competing interests

The authors declare that they have no competing financial or non-financial interests with the contents of this article.

Appendix A. Supplementary Data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jmb.2022.167789.

Received 12 May 2022;
Accepted 9 August 2022;
Available online 11 August 2022

Keywords:
ATP-sensitive potassium channel;
inward rectifying potassium channel;
sulfonylurea receptor;
cryoEM structure;
congenital hyperinsulinism;
neonatal diabetes

† These authors contributed equally to the work.

References

1. Afonine, P.V., Poon, B.K., Read, R.J., Sobolev, O.V., Terwilliger, T.C., Urzhumtsev, A., Adams, P.D., (2018). Real-space refinement in PHENIX for cryo-EM and crystallography. *Acta Crystallogr D Struct Biol.* **74**, 531–544.
2. Alkorta-Aranburu, G., Carmody, D., Cheng, Y.W., Nelakuditi, V., Ma, L., Dickens, J.T., Das, S., Greeley, S. A.W., et al., (2014). Phenotypic heterogeneity in monogenic diabetes: the clinical and diagnostic utility of a gene panel-based next-generation sequencing approach. *Mol. Genet. Metab.* **113**, 315–320.
3. Arya, V.B., Guemes, M., Nessa, A., Alam, S., Shah, P., Gilbert, C., Senniappan, S., Flanagan, S.E., et al., (2014). Clinical and histological heterogeneity of congenital hyperinsulinism due to paternally inherited heterozygous ABCC8/KCNJ11 mutations. *Eur. J. Endocrinol.* **171**, 685–695.
4. Ashcroft, F.M., (2005). ATP-sensitive potassium channelopathies: focus on insulin secretion. *J. Clin. Investig.* **115**, 2047–2058.
5. Babenko, A.P., Bryan, J., (2003). Sur domains that associate with and gate KATP pores define a novel gatekeeper. *J. Biol. Chem.* **278**, 41577–41580.
6. Babenko, A.P., Gonzalez, G., Bryan, J., (1999). The N-terminus of KIR6.2 limits spontaneous bursting and modulates the ATP-inhibition of KATP channels. *Biochem. Biophys. Res. Commun.* **255**, 231–238.
7. Balusek, C., Hwang, H., Lau, C.H., Lundquist, K., Hazel, A., Pavlova, A., Lynch, D.L., Reggio, P.H., et al., (2019). Accelerating Membrane Simulations with Hydrogen Mass Repartitioning. *J. Chem. Theory Comput.* **15**, 4673–4686.
8. Baukrowitz, T., Schulte, U., Oliver, D., Herlitze, S., Krauter, T., Tucker, S.J., Ruppersberg, J.P., Fakler, B., (1998). PIP2 and PIP as determinants for ATP inhibition of KATP channels. *Science* **282**, 1141–1144.
9. Boodhansingh, K.E., Kandasamy, B., Mitteer, L., Givler, S., De Leon, D.D., Shyng, S.L., Ganguly, A., Stanley, C.A., (2019). Novel dominant KATP channel mutations in infants with congenital hyperinsulinism: Validation by in vitro expression studies and in vivo carrier phenotyping. *Am. J. Med. Genet. A.* **179**, 2214–2227.
10. Borschel, W.F., Wang, S., Lee, S., Nichols, C.G., (2017). Control of Kir channel gating by cytoplasmic domain interface interactions. *J. General Physiol.* **149**, 561–576.
11. Brennan, S., Rubaiy, H.N., Imanzadeh, S., Reid, R., Lodwick, D., Norman, R.I., Rainbow, R.D., (2020). Kir 6.2-D323 and SUR2A-Q1336: an intersubunit interaction pairing for allosteric information transfer in the KATP channel complex. *Biochem. J.* **477**, 671–689.
12. Brundl, M., Pellikan, S., Stary-Weinzinger, A., (2021). Simulating PIP2-Induced Gating Transitions in Kir6.2 Channels. *Front. Mol. Biosci.* **8** (711975)
13. Case, D.A., Betz, R.M., Cerutti, D.S., Cheatham, T.E., III, Darden, T.A., Duke, R.E., Giese, T.J., Gohlke, H., et al., 2016. AMBER 2016.
14. Chan, K.W., Zhang, H., Logothetis, D.E., (2003). N-terminal transmembrane domain of the SUR controls trafficking and gating of Kir6 channel subunits. *EMBO J.* **22**, 3833–3843.
15. Clement, J.P.T., Kunjilwar, K., Gonzalez, G., Schwanstecher, M., Panten, U., Aguilar-Bryan, L., Bryan, J., (1997). Association and stoichiometry of K(ATP) channel subunits. *Neuron* **18**, 827–838.
16. Cukras, C.A., Jeliazkova, I., Nichols, C.G., (2002). The role of NH2-terminal positive charges in the activity of inward rectifier KATP channels. *J. General Physiol.* **120**, 437–446.
17. Darden, T., York, D., Pedersen, L., (1993). Particle mesh Ewald: An Nṡlog(N) method for Ewald sums in large systems. *J. Chem. Phys.* **98**, 10089–10092.
18. De Franco, E., Saint-Martin, C., Brusgaard, K., Knight Johnson, A.E., Aguilar-Bryan, L., Bowman, P., Arnoux, J. B., Larsen, A.R., et al., (2020). Update of variants identified in the pancreatic beta-cell KATP channel genes KCNJ11 and ABCC8 in individuals with congenital hyperinsulinism and diabetes. *Hum. Mutat.* **41**, 884–905.
19. de Wet, H., Proks, P., Lafond, M., Aittoniemi, J., Sansom, M.S., Flanagan, S.E., Pearson, E.R., Hattersley, A.T., et al., (2008). A mutation (R826W) in nucleotide-binding domain 1 of ABCC8 reduces ATPase activity and causes transient neonatal diabetes. *EMBO Rep.* **9**, 648–654.
20. Devaraneni, P.K., Martin, G.M., Olson, E.M., Zhou, Q., Shyng, S.L., (2015). Structurally distinct ligands rescue biogenesis defects of the KATP channel complex via a converging mechanism. *J. Biol. Chem.* **290**, 7980–7991.
21. Ding, D., Wang, M., Wu, J.X., Kang, Y., Chen, L., (2019). The Structural Basis for the Binding of Repaglinide to the Pancreatic KATP Channel. *Cell Reports.* **27** 1848-1857 e1844.

22. Drain, P., Li, L., Wang, J., (1998). KATP channel inhibition by ATP requires distinct functional domains of the cytoplasmic C terminus of the pore-forming subunit. *PNAS* **95**, 13953–13958.
23. Emsley, P., Lohkamp, B., Scott, W.G., Cowtan, K., (2010). Features and development of Coot. *Acta Crystallogr. Section D, Biol. Crystallogr.* **66**, 486–501.
24. Enkvetchakul, D., Loussouarn, G., Makhina, E., Shyng, S. L., Nichols, C.G., (2000). The kinetic and physical basis of K(ATP) channel gating: toward a unified molecular understanding. *Biophys. J.* **78**, 2334–2348.
25. Essmann, U., Perera, L., Berkowitz, M.L., Darden, T., Lee, H., Pedersen, L.G., (1995). A smooth particle mesh Ewald method. *J. Chem. Phys.* **103**, 8577–8593.
26. Gribble, F.M., Reimann, F., (2003). Sulphonylurea action revisited: the post-cloning era. *Diabetologia* **46**, 875–891.
27. Hansen, S.B., Tao, X., MacKinnon, R., (2011). Structural basis of PIP2 activation of the classical inward rectifier K+ channel Kir2.2. *Nature* **477**, 495–498.
28. Hopkins, C.W., Le Grand, S., Walker, R.C., Roitberg, A.E., (2015). Long-Time-Step Molecular Dynamics through Hydrogen Mass Repartitioning. *J. Chem. Theory Comput.* **11**, 1864–1874.
29. Huang, J., Rauscher, S., Nawrocki, G., Ran, T., Feig, M., de Groot, B.L., Grubmuller, H., MacKerell Jr., A.D., (2017). CHARMM36m: an improved force field for folded and intrinsically disordered proteins. *Nat. Methods* **14**, 71–73.
30. Inagaki, N., Gonoi, T., Clement, J.P.T., Namba, N., Inazawa, J., Gonzalez, G., Aguilar-Bryan, L., Seino, S., Bryan, J., et al., (1995). Reconstitution of IKATP: an inward rectifier subunit plus the sulfonylurea receptor. *Science* **270**, 1166–1170.
31. Inagaki, N., Gonoi, T., Seino, S., (1997). Subunit stoichiometry of the pancreatic beta-cell ATP-sensitive K+ channel. *FEBS Lett.* **409**, 232–236.
32. Jean-Paul Ryckaert, G.C., Herman, J.C., Berendsen, (1977). Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *J. Comput. Phys.* **23**, 327–341.
33. Jo, S., Kim, T., Iyer, V.G., Im, W., (2008). CHARMM-GUI: a web-based graphical user interface for CHARMM. *J. Comput. Chem.* **29**, 1859–1865.
34. Jorgensen, W.L., Chandrasekhar, J., Madura, J.D., (1983). Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.* **79**, 926–935.
35. Klauda, J.B., Venable, R.M., Freites, J.A., O'Connor, J. W., Tobias, D.J., Mondragon-Ramirez, C., Vorobyov, I., MacKerell Jr., A.D., et al., (2010). Update of the CHARMM all-atom additive force field for lipids: validation on six lipid types. *J. Phys. Chem. B* **114**, 7830–7843.
36. Koster, J.C., Sha, Q., Shyng, S., Nichols, C.G., (1999). ATP inhibition of KATP channels: control of nucleotide sensitivity by the N-terminal domain of the Kir6.2 subunit. *J. Physiol.* **515** (Pt 1), 19–30.
37. Lee, J., Cheng, X., Swails, J.M., Yeom, M.S., Eastman, P. K., Lemkul, J.A., Wei, S., Buckner, J., et al., (2016). CHARMM-GUI Input Generator for NAMD, GROMACS, AMBER, OpenMM, and CHARMM/OpenMM Simulations Using the CHARMM36 Additive Force Field. *J. Chem. Theory Comput.* **12**, 405–413.
38. Lee, K.P.K., Chen, J., MacKinnon, R., (2017). Molecular structure of human KATP in complex with ATP and ADP. *eLife* **6**
39. Lee, S.J., Ren, F., Zangerl-Plessl, E.M., Heyman, S., Stary-Weinzinger, A., Yuan, P., Nichols, C.G., (2016). Structural basis of control of inward rectifier Kir2 channel gating by bulk anionic phospholipids. *J. General Physiol.* **148**, 227–237.
40. Li, J.B., Huang, X., Zhang, R.S., Kim, R.Y., Yang, R., Kurata, H.T., (2013). Decomposition of slide helix contributions to ATP-dependent inhibition of Kir6.2 channels. *J. Biol. Chem.* **288**, 23038–23049.
41. Li, N., Wu, J.X., Ding, D., Cheng, J., Gao, N., Chen, L., (2017). Structure of a Pancreatic ATP-Sensitive Potassium Channel. *Cell* **168** 101–110 e110.
42. Lin, C.W., Yan, F., Shimamura, S., Barg, S., Shyng, S.L., (2005). Membrane phosphoinositides control insulin secretion through their effects on ATP-sensitive K+ channel activity. *Diabetes* **54**, 2852–2858.
43. Lin, Y.W., Bushman, J.D., Yan, F.F., Haidar, S., MacMullen, C., Ganguly, A., Stanley, C.A., Shyng, S.L., (2008). Destabilization of ATP-sensitive potassium channel activity by novel KCNJ11 mutations identified in congenital hyperinsulinism. *J. Biol. Chem.*. **283**, 9146–9156.
44. Lin, Y.W., Jia, T., Weinsoft, A.M., Shyng, S.L., (2003). Stabilization of the activity of ATP-sensitive potassium channels by ion pairs formed between adjacent Kir6.2 subunits. *J. General Physiol.* **122**, 225–237.
45. Lomize, M.A., Pogozheva, I.D., Joo, H., Mosberg, H.I., Lomize, A.L., (2012). OPM database and PPM web server: resources for positioning of proteins in membranes. *Nucleic Acids Res.* **40**, D370–D376.
46. Martin, G.M., Kandasamy, B., DiMaio, F., Yoshioka, C., Shyng, S.L., (2017). Anti-diabetic drug binding site in a mammalian KATP channel revealed by Cryo-EM. *eLife* **6**
47. Martin, G.M., Sung, M.W., Yang, Z., Innes, L.M., Kandasamy, B., David, L.L., Yoshioka, C., Shyng, S.L., (2019). Mechanism of pharmacochaperoning in a mammalian KATP channel revealed by cryo-EM. *eLife* **8**
48. Martin, G.M., Yoshioka, C., Rex, E.A., Fay, J.F., Xie, Q., Whorton, M.R., Chen, J.Z., Shyng, S.L., (2017). Cryo-EM structure of the ATP-sensitive potassium channel illuminates mechanisms of assembly and gating. *eLife* **6**
49. Martyna, G.J., (1994). Constant pressure molecular dynamics algorithms. *J. Chem. Phys.* **101**, 4177–4189.
50. Nakane, T., Kimanius, D., Lindahl, E., Scheres, S.H., (2018). Characterisation of molecular motions in cryo-EM single-particle data by multi-body refinement in RELION. *eLife* **7**
51. Nichols, C.G., (2006). KATP channels as molecular sensors of cellular metabolism. *Nature* **440**, 470–476.
52. Nichols, C.G., Shyng, S.L., Nestorowicz, A., Glaser, B., Clement, J.P.T., Gonzalez, G., Aguilar-Bryan, L., Permutt, M.A., et al., (1996). Adenosine diphosphate as an intracellular regulator of insulin secretion. *Science* **272**, 1785–1787.
53. Niu, Y., Tao, X., Touhara, K.K., MacKinnon, R., (2020). Cryo-EM analysis of PIP2 regulation in mammalian GIRK channels. *eLife* **9**
54. Niu, Y., Tao, X., Vaisey, G., Olinares, P.D.B., Alwaseem, H., Chait, B.T., MacKinnon, R., (2021). Analysis of the mechanosensor channel functionality of TACAN. *Elife.* **10**
55. Nogales, E., Scheres, S.H., (2015). Cryo-EM: A Unique Tool for the Visualization of Macromolecular Complexity. *Mol. Cell* **58**, 677–689.
56. Pastor, R.W., Brooks, B.R., Szabo, A., (1988). An analysis of the accuracy of Langevin and molecular dynamics algorithms. *Mol. Phys.* **65**, 1409–1419.

57. Pastor, R.W., Mackerell Jr., A.D., (2011). Development of the CHARMM Force Field for Lipids. *J. Phys. Chem. Lett.* **2**, 1526–1532.
58. Pettersen, E.F., Goddard, T.D., Huang, C.C., Couch, G.S., Greenblatt, D.M., Meng, E.C., Ferrin, T.E., (2004). UCSF Chimera–a visualization system for exploratory research and analysis. *J. Comput. Chem.* **25**, 1605–1612.
59. Pinney, S.E., MacMullen, C., Becker, S., Lin, Y.W., Hanna, C., Thornton, P., Ganguly, A., Shyng, S.L., et al., (2008). Clinical characteristics and biochemical mechanisms of congenital hyperinsulinism associated with dominant KATP channel mutations. *J. Clin. Investig.* **118**, 2877–2886.
60. Pratt, E.B., Tewson, P., Bruederle, C.E., Skach, W.R., Shyng, S.L., (2011). N-terminal transmembrane domain of SUR1 controls gating of Kir6.2 by modulating channel sensitivity to PIP2. *J. General Physiol.* **137**, 299–314.
61. Pratt, E.B., Zhou, Q., Gay, J.W., Shyng, S.L., (2012). Engineered interaction between SUR1 and Kir6.2 that enhances ATP sensitivity in KATP channels. *J. General Physiol.* **140**, 175–187.
62. Puljung, M.C., (2018). Cryo-electron microscopy structures and progress toward a dynamic understanding of KATP channels. *J. General Physiol.* **150**, 653–669.
63. Punjani, A., Fleet, D.J., (2021). 3D variability analysis: Resolving continuous flexibility and discrete heterogeneity from single particle cryo-EM. *J. Struct. Biol.* **213**, 107702.
64. Reimann, F., Tucker, S.J., Proks, P., Ashcroft, F.M., (1999). Involvement of the n-terminus of Kir6.2 in coupling to the sulphonylurea receptor. *J. Physiol.* **518** (Pt 2), 325–336.
65. Scheres, S.H., (2016). Processing of Structurally Heterogeneous Cryo-EM Data in RELION. *Methods Enzymol.* **579**, 125–157.
66. Shyng, S., Nichols, C.G., (1997). Octameric stoichiometry of the KATP channel complex. *J. General Physiol.* **110**, 655–664.
67. Shyng, S.L., Cukras, C.A., Harwood, J., Nichols, C.G., (2000). Structural determinants of PIP(2) regulation of inward rectifier K(ATP) channels. *J. General Physiol.* **116**, 599–608.
68. Shyng, S.L., Nichols, C.G., (1998). Membrane phospholipid control of nucleotide sensitivity of KATP channels. *Science* **282**, 1138–1141.
69. Sung, M.W., Yang, Z., Driggers, C.M., Patton, B.L., Mostofian, B., Russo, J.D., Zuckerman, D.M., Shyng, S. L., (2021). Vascular KATP channel structural dynamics reveal regulatory mechanism by Mg-nucleotides. *Proc. Nat. Acad. Sci. USA*, 118.
70. Suzuki, S., Makita, Y., Mukai, T., Matsuo, K., Ueda, O., Fujieda, K., (2007). Molecular basis of neonatal diabetes in Japanese patients. *J. Clin. Endocrinol. Metabol.* **92**, 3979–3985.
71. Trapp, S., Haider, S., Jones, P., Sansom, M.S., Ashcroft, F.M., (2003). Identification of residues contributing to the ATP binding site of Kir6.2. *EMBO J.* **22**, 2903–2912.
72. Tucker, S.J., Gribble, F.M., Proks, P., Trapp, S., Ryder, T. J., Haug, T., Reimann, F., Ashcroft, F.M., (1998). Molecular determinants of KATP channel inhibition by ATP. *EMBO J.* **17**, 3290–3296.
73. Tucker, S.J., Gribble, F.M., Zhao, C., Trapp, S., Ashcroft, F.M., (1997). Truncation of Kir6.2 produces ATP-sensitive K+ channels in the absence of the sulphonylurea receptor. *Nature* **387**, 179–183.
74. Usher, S.G., Ashcroft, F.M., Puljung, M.C., (2020). Nucleotide inhibition of the pancreatic ATP-sensitive K+ channel explored with patch-clamp fluorometry. *eLife* **9**
75. Vieira-Pires, R.S., Morais-Cabral, J.H., (2010). 3(10) helices in channels and other membrane proteins. *J. General Physiol.* **136**, 585–592.
76. Wang, M., Wu, J.X., Ding, D., Chen, L., (2022). Structural insights into the mechanism of pancreatic KATP channel regulation by nucleotides. *Nat. Commun.* **13**, 2770.
77. Wu, E.L., Cheng, X., Jo, S., Rui, H., Song, K.C., Davila-Contreras, E.M., Qi, Y., Lee, J., et al., (2014). CHARMM-GUI Membrane Builder toward realistic biological membrane simulations. *J. Comput. Chem.* **35**, 1997–2004.
78. Wu, J.X., Ding, D., Wang, M., Kang, Y., Zeng, X., Chen, L., (2018). Ligand binding and conformational changes of SUR1 subunit in pancreatic ATP-sensitive potassium channels. *Protein Cell.* **9**, 553–567.
79. Zangerl-Plessl, E.M., Lee, S.J., Maksaev, G., Bernsteiner, H., Ren, F., Yuan, P., Stary-Weinzinger, A., Nichols, C.G., (2020). Atomistic basis of opening and conduction in mammalian inward rectifier potassium (Kir2.2) channels. *J. General Physiol.* **152**
80. Zhang, M., Chen, X., Shen, S., Li, T., Chen, L., Hu, M., Cao, L., Cheng, R., et al., (2015). Sulfonylurea in the treatment of neonatal diabetes mellitus children with heterogeneous genetic backgrounds. *J. Pediatric Endocrinol. Metabol. : JPEM.* **28**, 877–884.
81. Zhao, C., MacKinnon, R., (2021). Molecular structure of an open human KATP channel. *Proc. Nat. Acad. Sci. USA*, 118.
82. Zivanov, J., Nakane, T., Forsberg, B.O., Kimanius, D., Hagen, W.J., Lindahl, E., Scheres, S.H., (2018). New tools for automated high-resolution cryo-EM structure determination in RELION-3. *eLife* **7**

D haMSM Analyses of SARS-CoV-2 Spike Protein WE Simulations

International Journal of HIGH PERFORMANCE COMPUTING APPLICATIONS

Special Issue Paper

#COVIDisAirborne: AI-enabled multiscale computational microscopy of delta SARS-CoV-2 in a respiratory aerosol

The International Journal of High Performance Computing Applications
2023, Vol. 37(1) 28–44

DOI: 10.1177/10943420221128233
journals.sagepub.com/home/hpc
SAGE

Abigail Dommer[1,†], Lorenzo Casalino[1,†], Fiona Kearns[1,†], Mia Rosenfeld[1], Nicholas Wauer[1], Surl-Hee Ahn[1], John Russo[2], Sofia Oliveira[3], Clare Morris[1], Anthony Bogetti[4], Anda Trifan[5,6], Alexander Brace[5,7], Terra Sztain[1,8], Austin Clyde[5,7], Heng Ma[5], Chakra Chennubhotla[4], Hyungro Lee[9], Matteo Turilli[9], Syma Khalid[10], Teresa Tamayo-Mendoza[11], Matthew Welborn[11], Anders Christensen[11], Daniel GA Smith[11], Zhuoran Qiao[12], Sai K Sirumalla[11], Michael O'Connor[11], Frederick Manby[11], Anima Anandkumar[12,13], David Hardy[6], James Phillips[6], Abraham Stern[13], Josh Romero[13], David Clark[13], Mitchell Dorrell[14], Tom Maiden[14], Lei Huang[15], John McCalpin[15], Christopher Woods[3], Alan Gray[13], Matt Williams[3], Bryan Barker[16], Harinda Rajapaksha[16], Richard Pitts[16], Tom Gibbs[13], John Stone[6,13], Daniel M. Zuckerman[2], Adrian J. Mulholland[3], Thomas Miller III[11,12], Shantenu Jha[9], Arvind Ramanathan[5], Lillian Chong[4] and Rommie E Amaro[1]

Abstract

We seek to completely revise current models of airborne transmission of respiratory viruses by providing never-before-seen atomic-level views of the SARS-CoV-2 virus within a respiratory aerosol. Our work dramatically extends the capabilities of multiscale computational microscopy to address the significant gaps that exist in current experimental methods, which are limited in their ability to interrogate aerosols at the atomic/molecular level and thus obscure our understanding of airborne transmission. We demonstrate how our integrated data-driven platform provides a new way of exploring the composition, structure, and dynamics of aerosols and aerosolized viruses, while driving simulation method development along several important axes. We present a series of initial scientific discoveries for the SARS-CoV-2 Delta variant, noting that the full scientific impact of this work has yet to be realized.

[1]UC San Diego, La Jolla, CA, USA
[2]Oregon Health & Science University, Portland, OR, USA
[3]University of Bristol, UK
[4]University of Pittsburgh, Pittsburgh, PA, USA
[5]Argonne National Laboratory, Lemont, IL, USA
[6]University of Illinois at Urbana-Champaign, Urbana, IL, USA
[7]University of Chicago, Chicago, IL, USA
[8]Freie Universitat Berlin
[9]Brookhaven National Lab and Rutgers University
[10]University of Oxford, UK
[11]Entos, Inc., San Diego, CA, USA
[12]California Institute of Technology, Pasadena, CA, USA
[13]NVIDIA Corp, Santa Clara, CA, USA
[14]Pittsburgh Supercomputing Center, Pittsburgh, PA, USA
[15]Texas Advanced Computing Center, Austin, TX, USA
[16]Oracle for Research, Austin, TX, USA

[†]Joint first authors

Corresponding author:
Daniel Zuckerman, Adrian Mulholland, Thomas Miller III, Shantenu Jha, Arvind Ramanathan, Lillian Chong, Rommie E. Amaro.
Email: ramaro@ucsd.edu

Keywords
molecular dynamics, deep learning, multiscale simulation, weighted ensemble, computational virology, SARS-CoV-2, aerosols, COVID-19, HPC, AI, GPU, Delta

Justification

We develop a novel HPC-enabled multiscale research framework to study aerosolized viruses and the full complexity of species that comprise them. We present technological and methodological advances that bridge time and length scales from electronic structure through whole aerosol particle morphology and dynamics.

Performance attributes

Performance attribute	Our submission
Category of achievement	Scalability, Time-to-solution
Type of method used	Explicit, Deep Learning
Results reported on the basis of	Whole application including I/O
Precision reported	Mixed Precision
System scale	Measured on full system
Measurement mechanism	Hardware performance counters
	Application timers
	Performance Modeling

Overview of the problem. Respiratory pathogens, such as SARS-CoV-2 and influenza, are the cause of significant morbidity and mortality worldwide. These respiratory pathogens are spread by virus-laden aerosols and droplets that are produced in an infected person, exhaled, and transported through the environment (Wang et al., 2021) (Figure 1). Medical dogma has long focused on droplets as the main transmission route for respiratory viruses, where either a person has contact with an infected surface (fomites) or direct droplet transmission by close contact with an infected individual. However, as we continue to observe with SARS-CoV-2, airborne transmission also plays a significant role in spreading disease. We know this from various super spreader events, for example, during a choir rehearsal (Miller et al., 2021). Intervention and mitigation decisions, such as the relative importance of surface cleaning or whether and when to wear a mask, have unfortunately hinged on a weak understanding of aerosol transmission, to the detriment of public health.

A central challenge to understanding airborne transmission has been the inability of experimental science to reliably probe the structure and dynamics of viruses once they are inside respiratory aerosol particles. Single particle experimental methods have poor resolution for smaller particles (<1 micron) and are prone to sample destruction during collection. Airborne viruses are present in low concentrations in the air and are similarly prone to viral inactivation during sampling. In addition, studies of the initial infection event, for example, in the deep lung, are limited in their ability to provide a detailed understanding of the myriad of molecular interactions and dynamics taking place in situ. Altogether, these knowledge gaps hamper our collective ability to understand mechanisms of infection and develop novel effective antivirals, as well as prevent us from developing concrete, science-driven mitigation measures (e.g., masking and ventilation protocols).

Here, we aim to reconceptualize current models of airborne transmission of respiratory viruses by providing never-before-seen views of viruses within aerosols. Our approach relies on the use of all-atom molecular dynamics (MD) simulations as a multiscale "computational microscope." MD simulations can synthesize multiple types of biological data (e.g., multiresolution structural datasets, glycomics, lipidomics, etc.) into cohesive, biologically "accurate" structural models. Once created, we then approximate the model down to its many atoms, creating trajectories of its time dependent dynamics under cell-like (or in this case, aerosol-like) conditions. Critically, MD simulations are more than just "pretty movies." MD equations are solved in a theoretically rigorous manner, allowing us to compute experimentally testable macroscopic observables from time-averaged microscopic properties. What this means is that we can directly connect MD simulations with experiments, each validating and providing testable hypotheses to the other, which is the real power of the approach. An ongoing challenge to the successful application of such methods, however, is the need for technological and methodological advances that make it possible to access length scales relevant to the study of large, biologically complex systems (spanning nanometers to ∼one micron in size) and, correspondingly, longer timescales (microseconds to seconds).

Such challenges and opportunities manifest in the study of aerosolized viruses. Aerosols are generally defined as being less than 5 microns in diameter, able to float in the air for hours, travel significant distances (i.e., can fill a room, like cigarette smoke), and be inhaled. Fine aerosols < 1 micron in size can stay in the air for over 12 h and are enriched with viral particles (Fennelly 2020; Coleman et al., 2021). Our work focuses on these finer aerosols that travel deeper into the respiratory tract. Several studies provide the molecular recipes necessary to reconstitute respiratory aerosols according to their actual biologically relevant

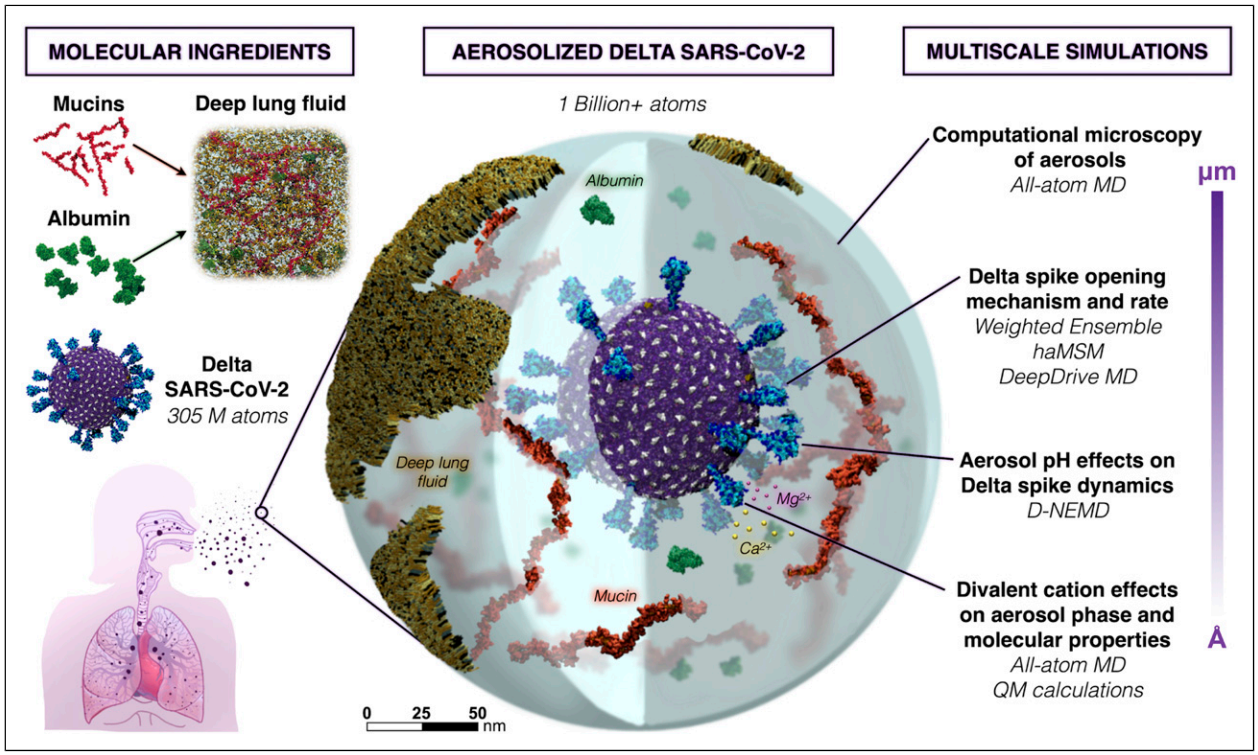


Figure 1. Overall schematic depicting the construction and multiscale simulations of Delta SARS-CoV-2 in a respiratory aerosol. (N.B.: The size of di-valent cations has been increased for visibility.)

composition (Vejerano and Marr 2018; Walker et al., 2021). These aerosols can contain lipids, cholesterol, albumin (protein), various mono- and di-valent salts, mucins, other surfactants, and water (Figure 1). Simulations of aerosolized viruses embody a novel framework for the study of aerosols: they will allow us and others to tune different species, relative humidity, ion concentrations, etc. to match experiments that can directly and indirectly connect to and inform our simulations, as well as test hypotheses. Some of the species under study here, for example, mucins, have not yet been structurally characterized or explored with simulations and thus the models we generate are expected to have impact beyond their roles in aerosols.

In addition to varying aerosol composition and size, the viruses themselves can be modified to reflect new variants of concern, where such mutations may affect interactions with particular species in the aerosol that might affect its structural dynamics and/or viability. The virion developed here is the Delta variant (B.1.617.2 lineage) of SARS-CoV-2 (Figure 2), which presents a careful integration of multiple biological datasets: (1) a complete viral envelope with realistic membrane composition, (2) fully glycosylated full-length spike proteins integrating 3D structural coordinates from multiple cryoelectron microscopy (cryoEM) studies (McCallum et al., 2021; Wrapp et al., 2020; Walls et al., 2020; Bangaru et al., 2020) (3) all biologically known features (post-translational modifications, palmitoylation, etc.), (4) any other known membrane proteins (e.g. the envelope (E) and membrane (M) proteins), and (5) virion size and patterning taken directly from cryoelectron tomography (cryoET). Each of the individual components of the virus are built up before being integrated into the composite virion, and thus represent useful molecular-scale scientific contributions in their own right (Casalino et al., 2020; Sztain et al., 2021).

Altogether in this work, we dramatically extend the capabilities of data-driven, multiscale computational microscopy to provide a new way of exploring the composition, structure, and dynamics of respiratory aerosols. While a seemingly limitless number of putative hypotheses could result from these investigations, the first set of questions we expect to answer are: *How does the virus exist within a droplet of the same order of magnitude in size, without being affected by the air-water interface, which is known to destroy molecular structure* (D'Imprima et al. 2019)*? How does the biochemical composition of the droplet, including pH, affect the structural dynamics of the virus? Are there species within the aerosols that "buffer" the viral structure from damage, and are there particular conditions under which the impact of those species changes?* Our simulations can also provide specific parameters that can be included in physical models of aerosols,

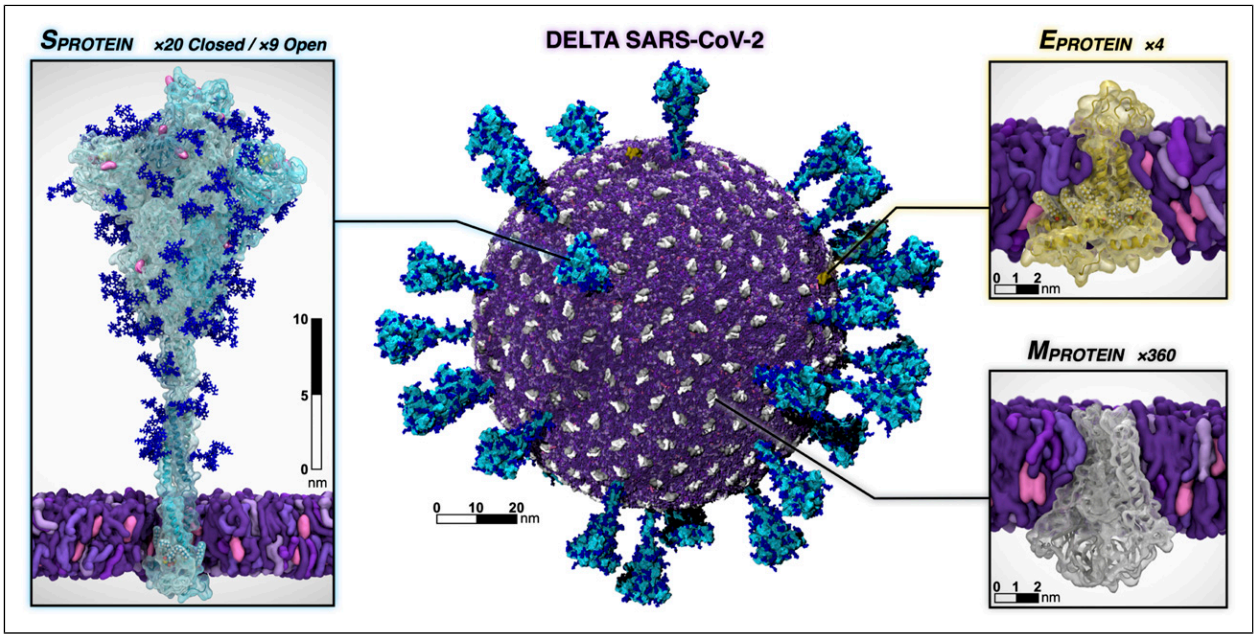


Figure 2. Individual protein components of the SARS-CoV-2 Delta virion. The spike is shown with the surface in cyan and with Delta's mutated residues and deletion sites highlighted in pink and yellow, respectively. Glycans attached to the spike are shown in blue. The E protein is shown in yellow and the M-protein is shown in silver and white. Visualized with VMD.

which still assume a simple water or water-salt composition even though it is well known that such models, for example, using kappa-Kohler theory, break down significantly as the molecular species diversify (Petters and Kreidenweis 2007).

Current state of the art

Current experimental methods are unable to directly interrogate the atomic-level structure and dynamics of viruses and other molecules within aerosols. Here we showcase computational microscopy as a powerful tool capable to overcome these significant experimental limitations. We present the major elements of our multiscale computational microscope and how they come together in an integrated manner to enable the study of aerosols across multiple scales of resolution. We demonstrate the impact such methods can bring to bear on scientific challenges that until now have been intractable, and present a series of new scientific discoveries for SARS-CoV-2.

Parallel molecular dynamics

All-atom molecular dynamics simulation has emerged as an increasingly powerful tool for understanding the molecular mechanisms underlying biophysical behaviors in complex systems. Leading simulation engines, NAMD (Phillips et al., 2020), AMBER (Case et al. [n. d.]), and GROMACS (Páll et al., 2020), are broadly useful, with each providing unique strengths in terms of specific methods or capabilities as required to address a particular biological question, and in terms of their support for particular HPC hardware platforms. Within the multiscale computational microscopy platform developed here, we show how each of these different codes contributes different elements to the overall framework, oftentimes utilizing different computing modalities/architectures, while simultaneously extending on state-of-the-art for each. Structure building, simulation preparation, visualization, and post hoc trajectory analysis are performed using VMD on both local workstations and remote HPC resources, enabling modeling of the molecular systems studied herein (Humphrey et al., 1996; Stone et al., 2013a,b, 2016b; Sener et al., 2021). We show how further development of each of these codes, considered together within the larger-scale collective framework, enables the study of SARS-CoV-2 in a wholly novel manner, with extension to numerous other complex systems and diseases.

AI-enhanced WE simulations

Because the virulence of the Delta variant of SARS-CoV-2 may be partly attributable to spike protein (S) opening, it is of pressing interest to characterize the mechanism and kinetics of the process. Although S-opening in principle can be studied via conventional MD simulations, in practice the system complexity and timescales make this wholly intractable. Splitting strategies that periodically replicate promising MD trajectories, among them the weighted

ensemble (WE) method (Huber and Kim 1996; Zuckerman and Chong 2017), have enabled simulations of the spike opening of WT SARS-CoV-2 (Sztain et al., 2021; Zimmerman et al., 2021). WE simulations can be orders of magnitude more efficient than conventional MD in generating pathways and rate constants for rare events (e.g. protein folding (Adhikari et al., 2019) and binding (Saglam and Chong 2019)). The WESTPA software for running WE (Zwier et al., 2015) is well-suited for high-performance computing with nearly perfect CPU/GPU scaling. The software is interoperable with any dynamics engine, including the GPU-accelerated AMBER dynamics engine (Salomon-Ferrer et al., 2013) that is used here. As shown below, major upgrades to WESTPA (v. 2.0) have enabled a dramatic demonstration of spike opening in the Delta variant (Figures 5 and 6) and exponentially improved analysis of spike-opening kinetics (Russo et al., 2022).

The integration of AI techniques with WE can further enhance the efficiency of sampling rare events (Noe 2020; Brace et al., 2021b; Casalino et al., 2021). One frontier area couples unsupervised linear and non-linear dimensionality reduction methods to identify collective variables/progress coordinates in high-dimensional molecular systems (Bhowmik et al., 2018; Clyde et al., 2021). Such methods may be well suited for analyzing the aerosolized virus. Integrating these approaches with WE simulations is advantageous in sampling the closed → open transitions in the Delta S landscape (Figure 5) as these unsupervised AI approaches automatically stratify progress coordinates (Figure 5(D)).

Dynamical non-equilibrium MD

Aerosols rapidly acidify during flight via reactive uptake of atmospheric gases, which is likely to impact the opening/closing of the S protein (Vejerano and Marr 2018; Warwicker 2021). Here, we describe the extension of dynamical non-equilibrium MD (D-NEMD) (Ciccotti and Ferrario 2016) to investigate pH effects on the Delta S. D-NEMD simulations (Ciccotti and Ferrario 2016) are emerging as a useful technique for identifying allosteric effects and communication pathways in proteins (Galdadas et al., 2021; Oliveira et al., 2019), including recently identifying effects of linoleic acid in the WT spike (Oliveira et al., 2021b). This approach complements equilibrium MD simulations, which provide a distribution of configurations as starting points for an ensemble of short non-equilibrium trajectories under the effect of the external perturbation. The response of the protein to the perturbation introduced can then be determined using the Kubo-Onsager relation (Oliveira et al., 2021a; Ciccotti and Ferrario 2016) by directly tracking the change in atomic positions between the equilibrium and non-equilibrium simulations at equivalent points in time (Oliveira et al., 2021a).

OrbNet

Ca^{2+} ions are known to play a key role in mucin aggregation in epithelial tissues (Hughes et al., 2019). Our RAV simulations would be an ideal case-study to probe such complex interactions between Ca^{2+}, mucins, and the SARS-CoV-2 virion in aerosols. However, Ca^{2+} binding energies can be difficult to capture accurately due to electronic dispersion and polarization, terms which are not typically modeled in classical mechanical force fields. Quantum mechanical (QM) methods are uniquely suited to capture these subtle interactions. Thus, we set out to estimate the correlation in Ca^{2+} binding energies between CHARMM36m and quantum mechanical estimates enabled via AI with OrbNet. Calculation of energies with sufficient accuracy in biological systems can, in many cases, be adequately described with density functional theory (DFT). However, its high cost limits the applicability of DFT in comparison to fixed charge force fields. To capture quantum quality energetics at a fraction of the computational expense, we employ a novel approach (OrbNet) based on the featurization of molecules in terms of symmetry-adapted atomic orbitals and the use of graph neural network methods for deep learning quantum-mechanical properties (Qiao et al., 2020). Our method outperforms existing methods in terms of its training efficiency and transferable accuracy across diverse molecular systems, opening a new pathway for replacing DFT in large-scale scientific applications such as those explored here. (Christensen et al., 2021).

Innovations realized

Construction and simulation of SARS-CoV-2 in a respiratory aerosol. Our approach to simulating the entire aerosol follows a composite framework wherein each of the individual molecular pieces is refined and simulated on its own before it is incorporated into the composite model. Simulations of each of the components are useful in their own right, and often serve as the basis for biochemical and biophysical validation and experiments (Casalino et al., 2020).

Throughout, we refer to the original circulating SARS-CoV-2 strain as "WT," whereas all SARS-CoV-2 proteins constructed in this work represent the Delta variant (Figure 2). All simulated membranes reflect mammalian ER-Golgi intermediate compartment (ERGIC) mimetic lipid compositions. VMD (Humphrey et al., 1996; Stone et al., 2016a), psfgen (Phillips et al., 2005), and CHARMM-GUI (Park et al., 2019) were used for construction and parameterization. Topologies and parameters for simulations were taken from CHARMM36m all-atom additive force fields (Guvench et al., 2009; Huang and Mackerell 2013; Huang et al., 2017; Klauda et al., 2010; Beglov and Roux 1994; Han et al., 2018; Venable et al., 2013). NAMD

was used to perform MD simulations (Phillips et al., 2020), adopting similar settings and protocols as in (Casalino et al., 2020). All systems underwent solvation, charge neutralization, minimization, heating, and equilibration prior to production runs. Refer to Table 1 for Abbreviations, PBC dimensions, total number of atoms, and total equilibration times for each system of interest.

Simulating the SARS-CoV-2 structural proteins. Fully glycosylated Delta spike (S) structures in open and closed conformations were built based on WT constructs from Casalino et al. (Casalino et al., 2020) with the following mutations: T19R, T95I, G142D, E156G, Δ157–158, L452R, T478K, D614G, P681R, and D950N (McCallum et al., 2021; Kannan et al., 2021). Higher resolved regions were grafted from PDB 7JJI (Bangaru et al., 2020). Additionally, coordinates of residues 128–167—accounting for a drastic conformational change seen in the Delta variant S—graciously made available to us by the Veesler Lab, were similarly grafted onto our constructs (McCallum et al., 2021). Finally, the S proteins were glycosylated following work by Casalino et al. (Casalino et al., 2020). By incorporating the Veesler Lab's bleeding-edge structure (McCallum et al., 2021) and highly resolved regions from 7JJI (Bangaru et al., 2020), our models represent the most complete and accurate structures of the Delta S to date. The S proteins were inserted into membrane patches and equilibrated for 3 × 110 ns. For non-equilibrium and weighted ensemble simulations, a closed S head (SH, residues 13–1140) was constructed by removing the stalk from the full-length closed S structure, then resolvated, neutralized, minimized, and subsequently passed to WE and D-NEMD teams. The M-protein was built from a structure graciously provided by the Feig Lab (paper in prep). The model was inserted into a membrane patch and equilibrated for 700 ns. RMSD-based clustering was used to select a stable starting M-protein conformation. From the equilibrated and clustered M structure, VMD's Mutator plugin (Humphrey et al., 1996) was used to incorporate the I82T mutation onto each M monomer to arrive at the Delta variant M. To construct the most complete E protein model to-date, the structure was patched together by resolving incomplete PDBs 5X29 (Surya et al., 2018), 7K3G (Mandala et al., 2020) and 7M4R (Chai et al., 2021). To do so, the transmembrane domain (residues 8–38) from 7K3G were aligned to the N-terminal domain (residues 1–7) and residues 39 to 68 of 5X29 and residues 69 to 75 of 7M4R by their C_α atoms. E was then inserted into a membrane patch and equilibrated for 40 ns.

Constructing the SARS-CoV-2 Delta virion. The SARS-CoV-2 Delta virion (V) model was constructed following Casalino et al. (Casalino et al., 2021) using CHARMM-GUI (Lee et al., 2016), LipidWrapper (Durrant and Amaro 2014), and Blender (Blender Online Community 2020), using a 350 Å lipid bilayer with an equilibrium area per lipid of 63 $Å^2$ and a 100 nm diameter Blender icospherical surface mesh (Turonova et al., 2020). The resulting lipid membrane was solvated in a 1100 $Å^3$ waterbox and subjected to four rounds of equilibration and patching (Casalino et al., 2021). 360 M dimers and 4 E pentamers were then tiled onto the surface, followed by random placement of 29 full-length S proteins

Table 1. Summary of all systems constructed in this work. See Figure 3 for illustration of aerosol construction.

[a]systems	[b]Abb	[c](Å × Å × Å)	[d]N_a	[e] (ns)
[f]M dimers	M	125 × 125 × 124	164,741	700
[f]E pentamers	E	123 × 125 × 102	136,775	41
Spikes				
[f] (Open)	S	206 × 200 × 410	1,692,444	330
[f] (Closed)	S	204 × 202 × 400	1,658,224	330
[g] (Closed, head)	SH	172 × 184 × 206	615,593	73μs
Mucins				
[f]short mucin 1	m_1	123 × 104 × 72	87,076	25
[f]short mucin 2	m_2	120 × 101 × 72	82,155	25
[f]long mucin 1	m_3	810 × 104 × 115	931,778	23
[f]long mucin 2	m_4	904 × 106 × 109	997,029	15
[f]long mucin 3	m_5	860 × 111 × 113	1,040,215	18
[f]S+m1/m2+ALB	SMA	227 × 229 × 433	2,156,689	840
[f]Virion	V	1460 × 1460 × 1460	305,326,834	41
[f]Resp.Aero.+Vir	RAV	2834 × 2820 × 2828	1,016,813,441	2.42
Total FLOPS			2.4 ZFLOPS	

[a]M, E, S, SH, and V models represent SARS-CoV-2 Delta strain. [b]Abbreviations used throughout document. [c]Periodic boundary dimensions. [d]Total number of atoms. [e]Total aggregate simulation time, including heating and equilibration runs. [f]Simulated with NAMD. [g]Simulated with NAMD, AMBER, and GROMACS.

(9 open, 20 closed) according to experimentally observed S protein density (Ke et al., 2020). M and E proteins were oriented with intravirion C-termini. After solvation in a 1460 Å waterbox, the complete V model tallied >305 million atoms (Table 1). V was equilibrated for 41 ns prior to placement in the respiratory aerosol (RA) model. The equilibrated membrane was 90 nm in diameter and remains in close structural agreement with the experimental studies (Ke et al., 2020).

Building and simulating the respiratory aerosol. Respiratory aerosols contain a complex mixture of chemical and biological species. We constructed a respiratory aerosol (RA) fluid based on a composition from artificial saliva and surrogate deep lung fluid recipes (Walker et al., 2021). This recipe includes 0.7 mM DPPG, 6.5 mM DPPC, 0.3 mM cholesterol, 1.4 mM Ca^{2+}, 0.8 mM Mg^{2+}, and 142 mM Na^{+} (Vejerano and Marr 2018; Walker et al., 2021), human serum albumin (ALB) protein, and a composition of mucins (Figure 3). Mucins are long polymer-like structures that are decorated by dense, heterogeneous, and complex regions of O-glycans. This work represents the first of its kind as, due to their complexity, the O-glycosylated regions of mucins have never before been constructed for molecular simulations. Two short (m_1, m_2, ~5 nm) and three long (m_3, m_4, m_5 ~55 nm) mucin models were constructed following known experimental compositions of protein and glycosylation sequences (Symmes et al., 2018; Hughes et al., 2019; Markovetz et al., 2019; Thomsson et al., 2005; Mariethoz et al., 2018) with ROSETTA (Raveh et al., 2010) and CHARMM-GUI Glycan Modeller (Jo et al., 2011). Mucin

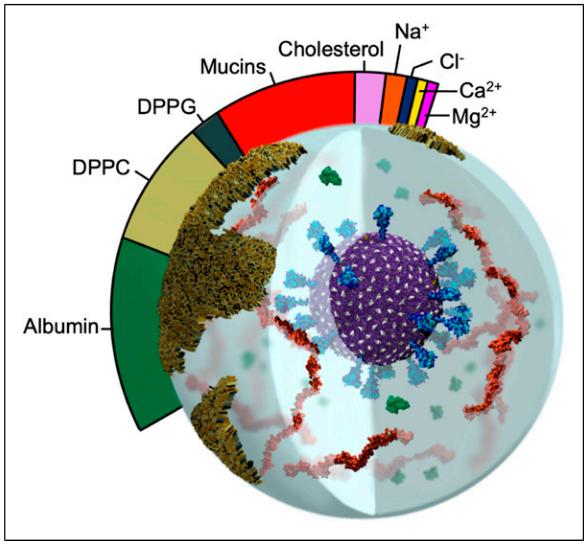


Figure 3. Image of RAV with relative mass ratios of RA molecular components represented in the colorbar. Water content is dependent on the relative humidity of the environment and is thus omitted from the molecular ratios.

models (short and long) were solvated, neutralized by charge matching with Ca^{2+} ions, minimized, and equilibrated for 15–25 ns each (Table 1). Human serum albumin (ALB), which is also found in respiratory aerosols, was constructed from PDB 1AO6 (Sugio et al., 1999). ALB was solvated, neutralized, minimized, and equilibrated for 7ns. Equilibrated structures of ALB and the three long mucins were used in construction of the RAV with m3+m4+m5 added at 6 g/mol and ALB at 4.4 g/mol.

Constructing the respiratory aerosolized virion model

A 100 nm cubic box with the RA fluid recipe specified above was built with PACKMOL (Martínez et al., 2009), minimized, equilibrated briefly on TACC Frontera, then replicated to form a 300 nm cube. The RA box was then carved into a 270 nm diameter sphere. To make space for the placement of V within the RA, a spherical selection with volume corresponding to that of the V membrane + S crown (radius 734 Å) was deleted from the center of the RA. The final equilibrated V model, including surrounding equilibrated waters and ions (733 Å radius), was translated into the RA. Atom clashes were resolved using a 1.2 Å cutoff. Hydrogen mass repartitioning (Hopkins et al., 2015) was applied to the structure to improve performance. The simulation box was increased to 2800 Å per side to provide a 100 Å vacuum atmospheric buffer. The RAV simulation was conducted in an NVT ensemble with a 4 fs timestep. After minimizing, the RAV was heated to 298 K with 0.1 kcal/mol $Å^2$ restraints on the viral lipid headgroups, then equilibrated for 1.5 ns. Finally, a cross-section of the RAV model—including and open S, m1/m2, and ALB (called the SMA system)—was constructed with PACKMOL to closely observe atomic scale interactions within the RAV model (Figure 4).

Parameter evaluation with OrbNet

Comparison to quantum methods reveals significant polarization effects, and shows that there is opportunity to improve the accuracy of fixed charge force fields. For the large system sizes associated with solvated Ca^{2+}-protein interaction motifs (over 1000 atoms, even in aggressively truncated systems), conventional quantum mechanics methods like density functional theory (DFT) are impractical for analyzing a statistically significant ensemble of distinct configurations (see discussion in Performance Results). In contrast, OrbNet allows for DFT accuracy with over 1000-fold speedup, providing a useful method for benchmarking and refining the force field simulation parameters with quantum accuracy (Christensen et al., 2021). To confirm the accuracy of OrbNet versus DFT (ωB97X-D/def2-TZVP), the inset of Figure 4(E) correlates the two methods for the Ca^{2+}-binding energy in a benchmark dataset

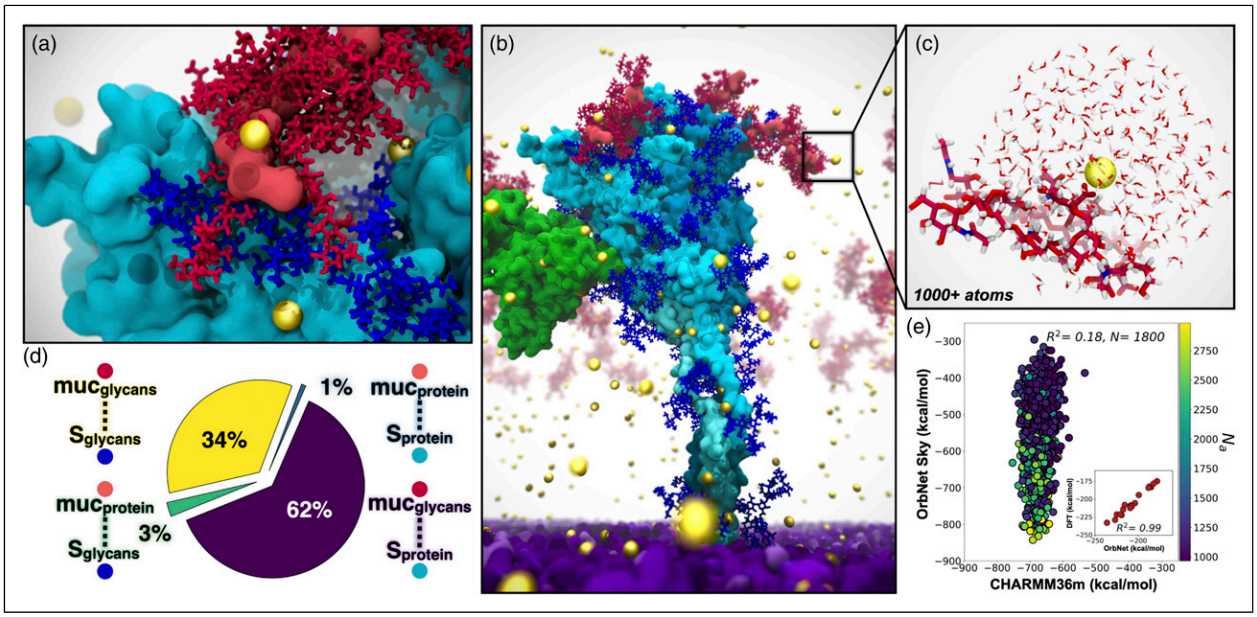


Figure 4. SMA system captured with multiscale modeling from classical MD to AI-enabled quantum mechanics. For all panels: S protein shown in cyan, S glycans in blue, m_1/m_2 shown in red, ALB in orange, Ca^{2+} in yellow spheres, viral membrane in purple. A) Interactions between mucins and S facilitated by glycans and Ca^{2+}. B) Snapshot from SMA simulations. C) Example Ca^{2+} binding site from SMA simulations (1800 sites, each 1000+ atoms) used for AI-enabled quantum mechanical estimates from OrbNet Sky. D) Quantification of contacts between S and mucin from SMA simulations. E) OrbNet Sky energies versus CHARMM36m energies for each sub-selected system, colored by total number of atoms. Performance of OrbNet Sky versus DFT in subplot (ωB97x-D3/def-TZVP, R^2=0.99, for 17 systems of peptides chelating Ca^{2+} (Hu et al., 2021)). Visualized with VMD.

of small Ca^{2+}-peptide complexes (Hu et al., 2021). The excellent correlation of OrbNet and DFT for the present use case is clear from the inset figure; six datapoints were removed from this plot on the basis of a diagnostic applied to the semi-empirical GFN-xTB solution used for feature generation of OrbNet (Christensen et al., 2021).

Figure 4 presents a comparison of the validated OrbNet method with the CHARMM36m force field for 1800 snapshots taken from the SMA MD simulations. At each snapshot, a subsystem containing a solvated Ca^{2+}-protein complex was extracted (Figure 4(E)), with protein bonds capped by hydrogens. For both OrbNet and the force field, the Ca^{2+}-binding energy was computed and shown in the correlation plot. Lack of correlation between OrbNet and the force field identifies important polarization effects, absent in a fixed charge description. Similarly, the steep slope of the best-fit line in Figure 4(E) reflects the fact that some of the configurations sampled using MD with the CHARMM36m force field are relatively high in energy according to the more accurate OrbNet potential. This approach allows us to test and quantify limitations of empirical force fields, such as lack of electronic polarization.

The practicality of OrbNet for these simulation snapshots with 1000+ atoms offers a straightforward multiscale strategy for refining the accuracy of the CHARMM36m force field. By optimizing the partial charges and other force field parameters, improved correlation with OrbNet for the subtle Ca^{2+}-protein interactions could be achieved, leading to near-quantum accuracy simulations with improved configurational sampling. The calculations presented here present a proof-of-concept of this iterative strategy.

AI-WE simulations of delta spike opening

While our previous WE simulations of the WT SARS-CoV-2 S-opening (Sztain et al., 2021) were notable in generating pathways for a seconds-timescale process of a massive system, we have made two critical technological advancements in the WESTPA software that greatly enhance the efficiency and analysis of WE simulations. These advances enabled striking observations of Delta variant S opening (Figures 5 and 6). First, in contrast to prior manual bins for controlling trajectory replication, we have developed automated and adaptive binning that enables more efficient surmounting of large barriers via early identification of "bottleneck" regions (Torrillo et al., 2021). Second, we have parallelized, memory-optimized, and implemented data streaming for the history-augmented Markov state model (haMSM) analysis scheme (Copperman and Zuckerman 2020) to enable application to the TB-scale S-opening datasets. The haMSM approach

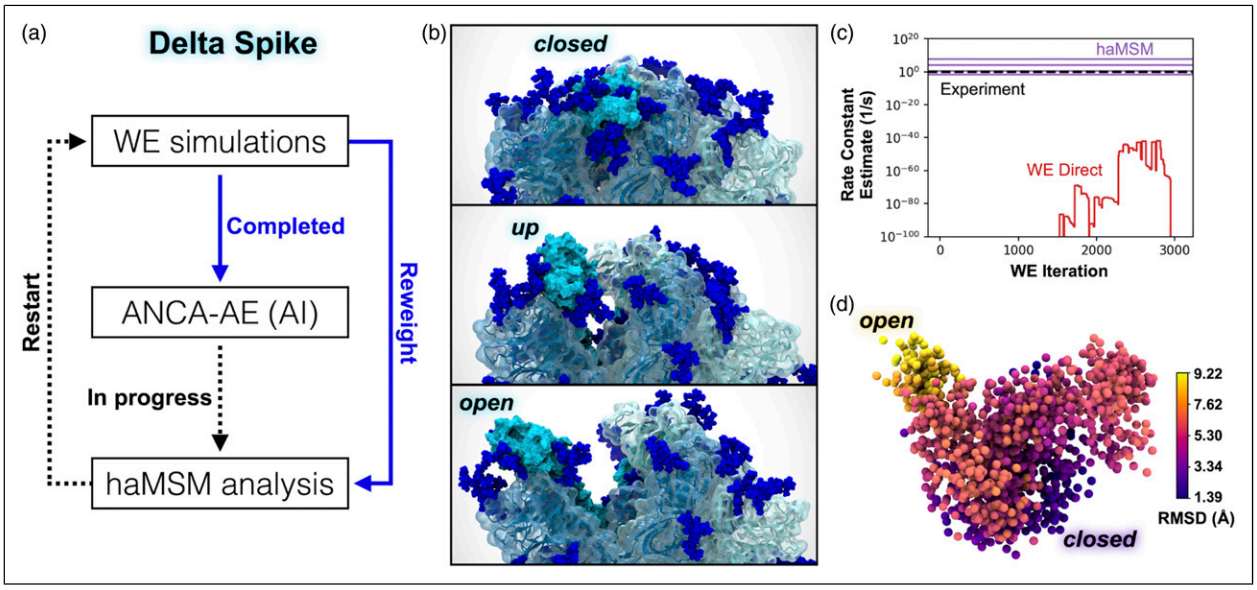


Figure 5. Delta variant spike opening from WE simulations, and AI/haMSM analysis. A) The integrated workflow. B) Snapshots of the "down," "up," and "open" states for Delta S-opening from a representative pathway generated by WE simulation, which represents ~ 10^5 speedup compared to conventional MD. C) Rate constant estimation with haMSM analysis of WE data (purple lines) significantly improves direct WE computation (red), by comparison to experimental measurement (black dashed). Varying haMSM estimates result from different featurizations which will be individually cross-validated. D) The first three dimensions of the ANCA-AE embeddings depict a clear separation between the closed (darker purple) and open (yellow) conformations of the Delta spike. A sub-sampled landscape is shown here where each sphere represents a conformation from the WE simulations and colored with the root-mean squared deviations (Å) with respect to the closed state. Visualized with VMD.

estimates rate constants from simulations that have not yet reached a steady state (Suarez et al., 2014).

Our WE simulations generated >800 atomically detailed, Delta variant S-opening pathways (Figures 5(B) and 6) of the receptor binding domain (RBD) switching from a glycan-shielded "down" to an exposed "up" state using 72 μs of total simulation time within 14 days using 192 NVIDIA V100 GPUs at a time on TACC's Longhorn supercomputer. Among these pathways, 83 reach an "open" state that aligns with the structure of the human ACE2-bound WT S protein (Benton et al., 2020) and 18 reach a dramatically open state (Figure 6). Our haMSM analysis of WT WE simulations successfully provided long-timescale (steady state) rate constants for S-opening based on highly transient information (Figure 5(C)).

We also leveraged a simple, yet powerful unsupervised deep learning method called Anharmonic Conformational Analysis enabled Autoencoders (ANCA-AE) Clyde et al. (2021) to extract conformational states from our long-timescale WE simulations of Delta spike opening (Figures 5(A) and (D)). ANCA-AE first minimizes the fourth order correlations in atomistic fluctuations from MD simulation datasets and projects the data onto a low dimensional space where one can visualize the anharmonic conformational fluctuations. These projections are then input to an autoencoder that further minimizes non-linear correlations in the atomistic fluctuations to learn an embedding where conformations are automatically clustered based on their structural and energetic similarity. A visualization of the first three dimensions from the latent space articulates the RBD opening motion from its closed state (Figure 5(D)). It is notable that while other deep learning techniques need special purpose hardware (such as GPUs), the ANCA-AE approach can be run with relatively modest CPU resources and can therefore scale to much larger systems (e.g., the virion within aerosol) when optimized.

D-NEMD explores pH effects on delta spike

We performed D-NEMD simulations of the SH system with GROMACS (Abraham et al., 2015) using a ΔpH=2.0 (from 7.0 to 5.0) as the external perturbation. We ran 3200-ns equilibrium MD simulations of SH to generate 87 configurations (29 configurations per replicate) that were used as the starting points for multiple short (10 ns) D-NEMD trajectories under the effect of the external perturbation (ΔpH=2.0). The effect of a ΔpH was modeled by changing the protonation state of histidines 66, 69, 146, 245, 625, 655, 1064, 1083, 1088, and 1101 (we note that other residues may also become protonated (Lobo and Warwicker 2021); the D-NEMD approach can also be applied to examine those). The structural response of the S to the pH

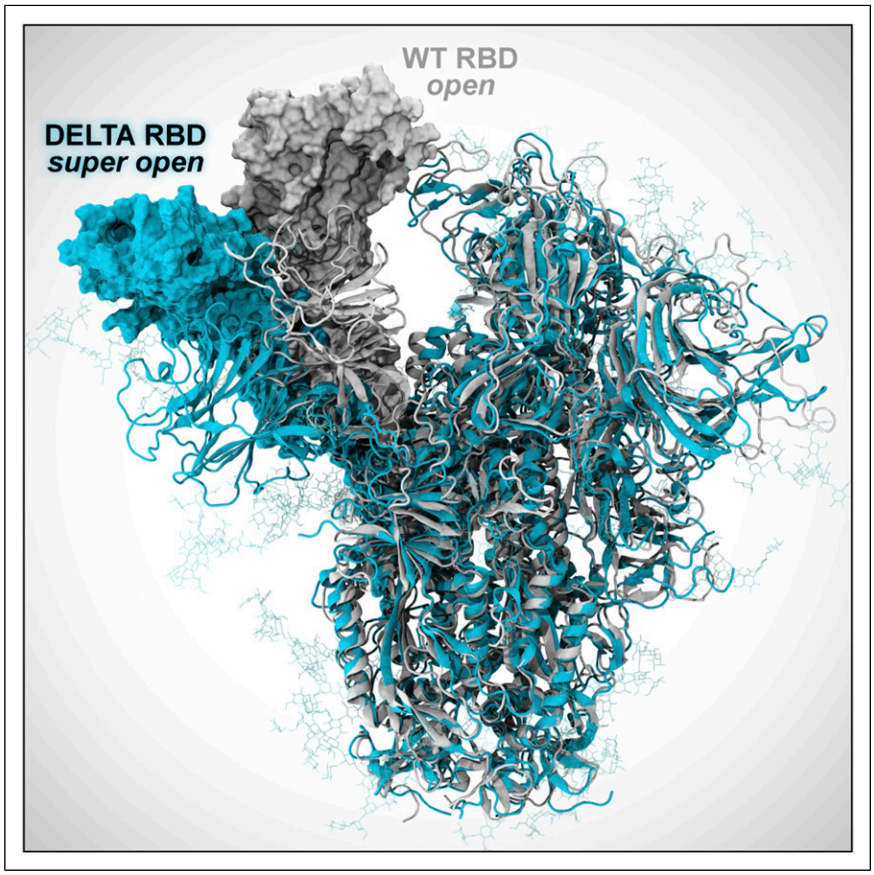


Figure 6. WE simulations reveal a dramatic opening of the Delta S (cyan), compared to WT S (white). While further investigation is needed, this super open state seen in the Delta S may indicate increased capacity for binding to human host-cell receptors.

decrease was investigated by measuring the difference in the position for each Cα atom between the equilibrium and corresponding D-NEMD simulation at equivalent points in time (Oliveira et al., 2021a), namely after 0, 0.1, 1, 5, and 10 ns of simulation. The D-NEMD simulations reveal that pH changes, of the type expected in aerosols, affect the dynamics of functionally important regions of the spike, with potential implications for viral behavior (Figure 7). As this approach involves multiple short independent non-equilibrium trajectories, it is well suited for cloud computing. All D-NEMD simulations were performed using Oracle Cloud.

How performance was measured

WESTPA. For the WE simulations of spike opening using WESTPA, we defined the time-to-solution as the total simulation time required to generate the first spike opening event. Spike opening is essentially impossible to observe via conventional MD. WESTPA simulations were run using the AMBER20 dynamics engine and 192 NVIDIA V100 GPUs at a time on TACC's Longhorn supercomputer.

NAMD. NAMD performance metrics were collected using hardware performance counters for FLOPs/step measurements, and application-internal timers for overall simulation rates achieved by production runs including all I/O for simulation trajectory and checkpoint output. NAMD FLOPs/step measurements were conducted on TACC Frontera, by querying hardware performance counters with the rdmsr utility from Intel msr-tools[1] and the "TACC stats" system programs.[2] For each simulation, FLOP counts were measured for NAMD simulation runs of two different step counts. The results of the two simulation lengths were subtracted to eliminate NAMD startup operations, yielding an accurate estimate of the marginal FLOPs per step for a continuing simulation (Phillips et al., 2002). Using the FLOPs/step values computed for each simulation, overall FLOP rates were computed by dividing the FLOPs/step value by seconds/step performance data reported by NAMD internal application timers during production runs.

GROMACS. GROMACS 2020.4 benchmarking was performed on Oracle Cloud Infrastructure (OCI)[3] compute shape BM.GPU4.8 consisting of 8×NVIDIA A100 tensor core GPUs, and 64 AMD Rome CPU cores. The simulation

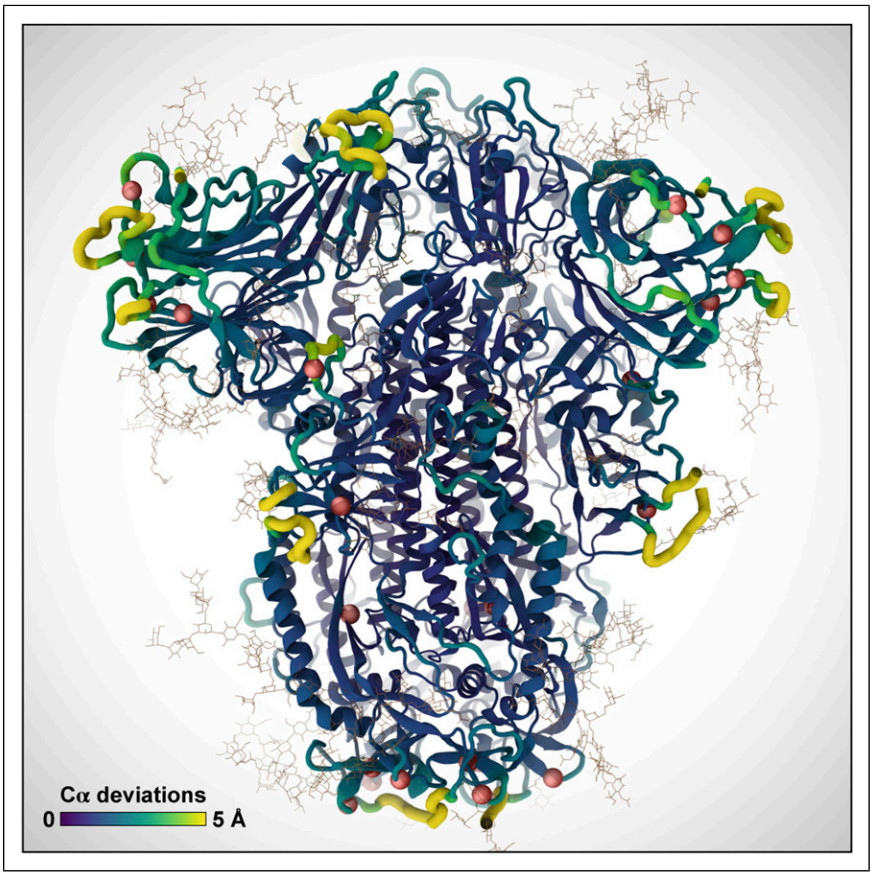


Figure 7. D-NEMD simulations reveal changes in key functional regions of the S protein, including the receptor binding domain, as the result of a pH decrease. Color scale and ribbon thickness indicate the degree of deviation of Cα atoms from their equilibrium position. Red spheres indicate the location of positively charged histidines.

Table 2. MD simulation floating point ops per timestep.

MD Simulation	Code	Atoms	[a]FLOPs/step
Spike, head	AMBER, GROMACS	0.6 M	62.14 GFLOPs/step
Spike	NAMD	1.7 M	43.05 GFLOPs/step
S+m_1/m_2+ALB	NAMD	2.1 M	54.86 GFLOPs/step
Resp. Aero.+Vir	NAMD	1B	25.81 TFLOPs/step

[a]FLOPs/step data were computed by direct FLOP measurements from hardware performance counters for NAMD simulations, or by using the application-reported FLOP rates and ns/day simulation performance in the case of GROMACS.

used for benchmarking contained 615,563 atoms and was run for 500,000 steps with 2 fs time steps. The simulations were run on increasing numbers of GPUs, from 1 to 8, using eight CPU cores per GPU, running for both the production (Nose-Hoover) and GPU-accelerated (velocity rescaling) thermostats. Particle–mesh Ewald (PME) calculations were pinned to a single GPU, with additional GPUs for multi-GPU jobs used for particle–particle calculations. Performance data (ns/day and average single-precision TFLOPS, calculated as total number of TFLOPs divided by total job walltime) were reported by GROMACS itself. Each simulation was repeated four times and average performance figures reported.

Performance results

Table 2.

NAMD performance. NAMD was used to perform all of the simulations listed in Table 1, except for the closed spike "SH" simulations described further below. With the exception of the aerosol and virion simulation, the other

NAMD simulations used conventional protocols and have performance and parallel scaling characteristics that closely match the results reported in our previous SARS-CoV-2 research Casalino et al. (2021). NAMD 2.14 scaling performance for the one billion-atom respiratory aerosol and virion simulation run on ORNL Summit is summarized in Tables 3 and 4. A significant performance challenge associated with the aerosol virion simulation relates to the roughly 50% reduction in particle density as compared with a more conventional simulation with a fully populated periodic cell. The reduced particle density results in large regions of empty space that nevertheless incur additional overheads associated with both force calculations and integration, and creates problems for the standard NAMD load balancing scheme that estimates the work associated with the cubic "patches" used for parallel domain decomposition. The PME electrostatics algorithm and associated 3-D FFT and transpose operations encompass the entire simulation unit cell and associated patches, requiring involvement in communication and reduction operations despite the inclusion of empty space. Enabling NAMD diagnostic output on a 512-node 1B-atom aerosol and virion simulation revealed that ranks assigned empty regions of the periodic cell had 66 times the number of fixed-size patches as ranks assigned dense regions. The initial load estimate for an empty patch was changed from a fixed 10 atoms to a runtime parameter with a default of 40 atoms, which reduced the patch ratio from 66 to 19 and doubled performance on 512 nodes.

WESTPA performance. Our time to solution for WE simulations of spike opening (to the "up" state) (Figure 5) using the WESTPA software and AMBER20 was 14 μs of total simulation time, which was completed in 4 days using 192 NVIDIA V100 GPUs at a time on TACC's Longhorn supercomputer. For reference, conventional MD would require an expected ~5 orders of magnitude more computing. The WESTPA software is highly scalable, with nearly perfect scaling out to >1000 NVIDIA V100 GPUs and this scaling is expected to continue until the filesystem is saturated. Thus, WESTPA makes optimal use of large supercomputers and is limited by filesystem I/O due to the periodic restarting of trajectories after short time intervals.

AI-enhanced WE simulations. DeepDriveMD is a framework to coordinate the concurrent execution of ensemble simulations and drive them using AI models Brace et al. (2021a); Lee et al. (2019). DeepDriveMD has been shown to improve the scientific performance of diverse problems: from-protein folding to conformation of protein-ligand complexes. We coupled WESTPA to DeepDriveMD, which is responsible for resource dynamism and concurrent heterogeneous task execution (ML and AMBER). The coupled workflow was executed on 1024 nodes on Summit (OLCF), and, in spite of the spatio-temporal heterogeneity of tasks involved, the resource utilization was in the high 90%. Consistent with earlier studies, the coupling of WESTPA to DeepDriveMD results in a 100x improvement in the exploration of phase space.

GROMACS performance. Figure 8 shows GROMACS parallelizes well across the eight NVIDIA A100 GPUs available on each BM.GPU4.8 instance used in the *Cluster in the Cloud*[4] running on OCI. There is a performance drop for two GPUs due to inefficient division of the PME and particle–particle tasks. Methods to address this exist for the two GPU case Páll et al. (2020), but were not adopted as we were targeting maximum raw performance across all eight

Table 3. NAMD performance: Respiratory Aerosol + Virion, 1B atoms, 4 fs timestep w/HMR, and PME every three steps.

Nodes	Summit CPU + GPU	Speedup	Efficiency
256	4.18 ns/day	~1.0×	~100%
512	7.68 ns/day	1.84×	92%
1024	13.64 ns/day	3.27×	81%
2048	23.10 ns/day	5.53×	69%
4096	34.21 ns/day	8.19×	51%

Table 4. Peak NAMD FLOP rates, ORNL Summit.

NAMD Simulation	Atoms, B	Nodes	Sim rate	Performance
Resp. Aero.+Vir	1	4096	34.21 ns/day	2.55 PFLOPS

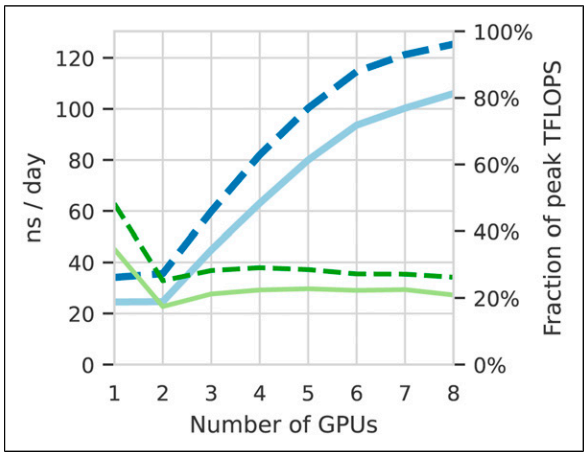


Figure 8. GROMACS performance across 1–8 A100 GPUs in ns/day (thicker, blue lines) and the fraction of maximum theoretical TFLOPS (thinner, green lines); production setup shown with solid line, and runs with the GPU-accelerated thermostat in dashed.

GPUs. Production simulations achieved 27% of the peak TFLOPS available from the GPUs. Multiple simulations were run across 10 such compute nodes, enabling the ensemble to run at an average combined speed of 425 TFLOPS and sampling up to 1μs/day. We note that the calculations will be able to run 20%–40% faster once the Nose-Hoover thermostat that is required for the simulation is ported to run on the GPU. Benchmarking using a velocity rescaling thermostat that has been ported to GPU shows that this would enable the simulation to extract 34% of the peak TFLOPS from the cards, enabling each node to achieve an average speed of 53.4 TFLOPS, and 125 ns/day. A cluster of 10 nodes would enable GROMACS to run at an average combined speed of over 0.5 PFLOPs, simulating over 1.2 μs/day.

A significant innovation is that this power is available on demand: Cluster in the Cloud with GPU-optimized GROMACS was provisioned and benchmarked within 1 day of inception of the project. This was handed to the researcher, who submitted the simulations. Automatically, up to 10 BM.GPU4.8 compute nodes were provisioned on-demand based on requests from the Slurm scheduler. These simulations were performed on OCI, using *Cluster in the Cloud* Williams (2021) to manage automatic scaling.

Cluster in the Cloud was configured to dynamically provision and terminate computing nodes based on the workload. Simulations were conducted using GROMACS 2020.4 compiled with CUDA support. Multiple simultaneous simulations were conducted, with each simulation utilizing a single BM.GPU4.8 node without multinode parallelism.

This allowed all production simulations to be completed within 2 days. The actual compute cost of the project was less than $6125 USD (on-demand OCI list price). The huge reduction in "time to science" that low-cost cloud enables changes the way that researchers can access and use HPC facilities. In our opinion, such a setup enables "exclusive on-demand" HPC capabilities for the scientific community for rapid advancement in science.

OrbNet performance. Prior benchmarking reveals that OrbNet provides over 1000-fold speedup compared to DFT (Christensen et al., 2021). For the calculations presented here, the cost of corresponding high quality range-separated DFT calculations (ωB97X-D/def2-TZVP) can be estimated. In Figure 4(E), we consider system sizes which would require 14,000–47,000 atomic orbitals for ωB97X-D/def2-TZVP, exceeding the range of typical DFT evaluations. Estimation of the DFT computational cost of the 1811 configurations studied in Figure 4(E) suggests a total of 115M core-hours on NERSC Cori Haswell nodes; in contrast, the OrbNet calculations for the current study require only 100k core-hours on the same nodes. DFT cost estimates were based on extrapolation from a dataset of over 1M ChEMBL molecules ranging in size from 40 to 107 atom systems considering only the cubic cost component of DFT (Christensen et al., 2021).

Implications

Our major scientific achievements are

1. We showcase an extensible AI-enabled multiscale computational framework that bridges time and length scales from electronic structure through whole aerosol particle morphology and dynamics.
2. We develop all-atom simulations of respiratory mucins, and use these to understand the structural basis of interaction with the SARS-CoV-2 spike protein. This has implications for viral binding in the deep lung, which is coated with mucins. We expect the impact of our mucin simulations to be far reaching, as malfunctions in mucin secretion and folding have been implicated in progression of severe diseases such as cancer and cystic fibrosis.
3. We present a significantly enhanced all-atom model and simulation of the SARS-CoV-2 Delta virion, which includes the hundreds of tiled M-protein dimers and the E-protein ion channels. This model can be used as a basis to understand why the Delta virus is so much more infectious than the WT or alpha variants.
4. We develop an ultra-large (1 billion+) all-atom simulation capturing massive chemical and biological complexity within a respiratory aerosol. This simulation provides the first atomic-level views of virus-laden aerosols and is already serving as a basis to develop an untold number of experimentally testable hypotheses. An immediate example suggests a mechanism through which mucins and other species, for example, lipids, which are present in the aerosol, arrange to protect the molecular structure of the virus, which otherwise would be exposed to the air-water interface. This work also opens the door for developing simulations of other aerosols, for example, sea spray aerosols, that are involved in regulating climate.
5. We evidence how changes in pH, which are expected in the aerosol environment, may alter dynamics and allosteric communication pathways in key functional regions of the Delta spike protein.
6. We characterize atomically detailed pathways for the spike-opening process of the Delta variant using WE simulations, revealing a dramatically open state that may facilitate binding to human host cells.
7. We demonstrate how parallelized haMSM analysis of WE data can provide physical rate estimates of spike opening, improving prior estimates by many

orders of magnitude. The pipeline can readily be applied to the any variant spike protein or other complex systems of interest.
8. We show how HPC and cloud resources can be used to significantly drive down time-to-solution for major scientific efforts as well as connect researchers and greatly enable complex collaborative interactions.
9. We demonstrate how AI coupled to HPC at multiple levels can result in significantly improved effective performance, for example, with AI-driven WESTPA, and extend the reach and domain of applicability of tools ordinarily restricted to smaller, less complex systems, for example, with OrbNet.
10. While our work provides a successful use case, it also exposes weaknesses in the HPC ecosystem in terms of support for key steps in large/complex computational science campaigns. We find lack of widespread support for high performance remote visualization and interactive graphical sessions for system preparation, debugging, and analysis with diverse science tools to be a limiting factor in such efforts.

Acknowledgements

We thank Prof. Kim Prather for inspiring and informative discussions about aerosols and for her commitment to convey the airborne nature of SARS-CoV-2. We thank D. Veesler for sharing the Delta spike NTD coordinates in advance of publication. We thank B. Messer, D. Maxwell, and the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory supported by the DOE under Contract DE-AC05-00OR22725. We thank the Texas Advanced Computing Center Frontera team, especially D. Stanzione and T. Cockerill, and for compute time made available through a Director's Discretionary Allocation (NSF OAC-1818253). We thank the Argonne Leadership Computing Facility supported by the DOE under DE-AC02-06CH11357. We thank the Pittsburgh Supercomputer Center for providing priority queues on Bridges-2 through the XSEDE allocation NSF TG-CHE060063. We thank N. Kern and J. Lee of the CHARMM-GUI support team for help converting topologies between NAMD and GROMACS. We thank J. Copperman, G. Simpson, D. Aristoff, and J. Leung for valuable discussions and support from NIH grant GM115805. NAMD and VMD are funded by NIH P41-GM104601. This work was supported by the NSF Center for Aerosol Impacts on Chemistry of the Environment (CAICE), National Science Foundation Center for Chemical Innovation (NSF CHE-1801971), as well as NIH GM132826, NSF RAPID MCB-2032054, an award from the RCSA Research Corp., a UC San Diego Moore's Cancer Center 2020 SARS-CoV-2 seed grant, to R.E.A. This work was also supported by Oracle Cloud credits and related resources provided by the Oracle for Research program. AJM and ASFO receive funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (PREDACTED Advanced Grant, Grant agreement No.: 101021207).

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by National Science Foundation (CHE-1801971); National Science Foundation (MCB- 2032054); National Science Foundation (OAC-1818253); National Science Foundation (TG-CHE060063); U.S. Department of Energy (DE-AC02-06CH11357); U.S. Department of Energy (DE-AC05-00OR22725); National Institutes of Health (P41-GM104601); National Institutes of Health (R01-GM132826).

ORCID iDs

Lorenzo Casalino https://orcid.org/0000-0003-3581-1148
Mia Rosenfeld https://orcid.org/0000-0002-8961-8231
Surl-Hee Ahn https://orcid.org/0000-0002-3422-805X
John Russo https://orcid.org/0000-0002-2813-6554
Sofia Oliveira https://orcid.org/0000-0001-8753-4950
Clare Morris https://orcid.org/0000-0002-4314-5387
Alexander Brace https://orcid.org/0000-0001-9873-9177
Hyungro Lee https://orcid.org/0000-0002-4221-7094
Zhuoran Qiao https://orcid.org/0000-0002-5704-7331
Anima Anandkumar https://orcid.org/0000-0002-6974-6797
James Phillips https://orcid.org/0000-0002-2296-3591
John McCalpin https://orcid.org/0000-0002-2535-1355
Christopher Woods https://orcid.org/0000-0001-6563-9903
Matt Williams https://orcid.org/0000-0003-2198-1058
Richard Pitts https://orcid.org/0000-0002-2037-3360
Daniel Zuckerman https://orcid.org/0000-0001-7662-2031
Adrian Mulholland https://orcid.org/0000-0003-1015-4567
Arvind Ramanathan https://orcid.org/0000-0002-1622-5488
Lillian Chong https://orcid.org/0000-0002-0590-483X
Rommie E Amaro https://orcid.org/0000-0002-9275-9553

Notes

1. https://github.com/intel/msr-tools
2. https://github.com/TACC/tacc_stats
3. https://www.oracle.com/cloud/
4. https://cluster-in-the-cloud.readthedocs.io/

References

Abraham MJ, Murtola T, Schulz R, et al. (2015) GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* 1-2: 19–25. DOI: 10.1016/j.softx.2015.06.001.

Adhikari U, Mostofian B, Copperman J, et al. (2019) Computational estimation of ms-sec atomistic folding times. *Journal of the American Chemical Society* 141: 6519–6526. DOI: 10.1101/427393.

Bangaru S, Gabriel O, Turner HL, et al. (2020) Structural analysis of full-length SARS-CoV-2 spike protein from an advanced vaccine candidate. *Science* 370: 65201089–65201094. DOI: 10.1126/science.abe1502.

Beglov D and Roux B (1994) Finite representation of an infinite bulk system: Solvent boundary potential for computer simulations. *The Journal of Chemical Physics* 100(12). DOI: 10.1063/1.466711.

Benton DJ, Wrobel AG, Xu P, et al. (2020) Receptor binding and priming of the spike protein of SARS-CoV-2 for membrane fusion. *Nature* 588: 7837327–7837330. DOI: 10.1038/s41586-020-2772-0.

Bhowmik D, Gao S, Young MT, et al. (2018) Deep clustering of protein folding simulations. *BMC Bioinformatics* 19(18): 484. DOI: 10.1186/s12859-018-2507-5.

Blender Online Community (2020) Blender - a 3D modelling and rendering package. http://www.blender.org.

Brace A, Lee H, Ma H, et al. (2021a) *Achieving 100X Faster Simulations of Complex Biological Phenomena by Coupling ML to HPC Ensembles*. arXiv: cs.DC/2104.04797.

Brace A, Michael S, Subbiah V, et al. (2021b) *Stream-AI-MD: Streaming AI-Driven Adaptive Molecular Simulations for Heterogeneous Computing Platforms*. New York, NY, USA: Association for Computing Machinery. DOI: 10.1145/3468267.3470578.

Casalino L, C Dommer A, Gaieb Z, et al. (2021) AI-driven multiscale simulations illuminate mechanisms of SARS-CoV-2 spike dynamics. *The International Journal of High Performance Computing Applications* 35(5): 432–451. DOI: 10.1177/10943420211006452.

Casalino L, Gaieb Z, Goldsmith JA, et al. (2020) Beyond Shielding: The Roles of Glycans in the SARS-CoV-2 Spike Protein. *ACS Central Science* 6(10): 1722–1734. DOI: 10.1021/acscentsci.0c01056.

Case DA, Cheatham TE III, Darden TA, et al. (n.d.). San Francisco: Publisher: University of California. Amber16. ([n. d.]).

Chai J, Cai Y, Pang C, et al. (2021) Structural basis for SARS-CoV-2 envelope protein recognition of human cell junction protein PALS1. *Nature Communications* 12(1): 3433. DOI: 10.1038/s41467-021-23533-x.

Christensen AS, Krishna Sirumalla S, Qiao Z, et al. (2021) *OrbNet Denali: A Machine Learning Potential for Biological and Organic Chemistry with Semi-empirical Cost and DFT Accuracy*. arXiv: physics.chem-ph/2107.00299.

Ciccotti G and Ferrario M (2016) Non-equilibrium by molecular dynamics: a dynamical approach. *Molecular Simulation* 42(16): 1385–1400. DOI: 10.1080/08927022.2015.1121543.

Clyde A, Galanie S, Kneller DW, et al. (2021) *High Throughput Virtual Screening and Validation of a SARS-CoV-2 Main Protease Non-covalent Inhibitor*. bioRxiv. arXiv: DOI: 10.1101/2021.03.27.437323. https://www.biorxiv.org/content/early/2021/04/02/2021.03.27.437323.full.pdf

Coleman KK, Wen Tay DJ, Tan KS, et al. (2021) *Viral Load of SARS-CoV-2 in Respiratory Aerosols Emitted by COVID-19 Patients while Breathing, Talking, and Singing*. Clinical Infectious Diseases. DOI: 10.1093/cid/ciab691.

Copperman J and Zuckerman DM (2020) Accelerated estimation of long-timescale kinetics from weighted ensemble simulation via non-Markovian "microbin" analysis. *Journal of Chemical Theory and Computation* 16(11): 6763–6775.

D'Imprima E, Floris D, Joppe M, et al. (2019) Protein denaturation at the air-water interface and how to prevent it. *eLife* 8: e42747. DOI: 10.7554/eLife.42747.

Durrant JD and Amaro RE (2014) LipidWrapper: An Algorithm for Generating Large-Scale Membrane Models of Arbitrary Geometry. *PLoS Computational Biology* 10: 7. DOI: 10.1371/journal.pcbi.1003720.

Fennelly KP (2020) Particle sizes of infectious aerosols: implications for infection control. *The Lancet Respiratory Medicine* 8(9): 914–924. DOI: 10.1016/S2213-2600(20)30323-4.

Galdadas I, Shen Q, F Oliveira AS, et al. (2021) Allosteric communication in class A β-lactamases occurs via cooperative coupling of loop dynamics. *eLife* 10: e66567. DOI: 10.7554/eLife.66567.

Guvench O, Hatcher E, Venable RM, et al. (2009) CHARMM additive all-atom force field for glycosidic linkages between hexopyranoses. *Journal of Chemical Theory and Computation* 5: 2353–2370. DOI: 10.1021/ct900242e.

Han K, Richard M, VenableBryant A-M, et al. (2018) Graph–Theoretic Analysis of Monomethyl Phosphate Clustering in Ionic Solutions. *The Journal of Physical Chemistry B* 122(4): 1484–1494. PMID: 29293344 DOI: 10.1021/acs.jpcb.7b10730.

Hopkins CW, ScottGrand Le, Walker RC, et al. (2015) Long-Time-Step Molecular Dynamics through Hydrogen Mass Repartitioning. *Journal of Chemical Theory and Computation* 11(1): 1864–1874. DOI: 10.1021/ct5010406.

Hu X, Lenz-Himmer M-O and Baldauf C (2021) *Better Force Fields Start with Better Data – A Data Set of Cation Dipeptide Interactions*. arXiv:q-bio.BM/2107.08855.

Huang J and Mackerell AD (2013) CHARMM36 all-atom additive protein force field: Validation based on comparison to NMR data. *Journal of Computational Chemistry* 34(25): 2135–2145. DOI: 10.1002/jcc.23354.

Huang J, Rauscher S, Nawrocki G, et al. (2017) CHARMM36m: An Improved Force Field for Folded and Intrinsically Disordered Proteins. *Nature Methods* 14(1): 71–73. DOI: 10.1038/nmeth.4067.

Huber GA and Kim S (1996) Weighted-ensemble Brownian dynamics simulations for protein association reactions. *Biophysical Journal* 70(1): 97–110. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1224912/

Hughes GW, Ridley C, Collins R, et al. (2019) The MUC5B mucin polymer is dominated by repeating structural motifs and its topology is regulated by calcium and pH. *Scientific Reports* 9(1): 17350. DOI: 10.1038/s41598-019-53768-0.

Humphrey W, Dalke A and Schulten K (1996) VMD – Visual Molecular Dynamics. *J. Mol. Graphics* 14(1): 33–38. DOI: 10.1016/0263-7855(96)00018-5.

Jo S, Song KC, Desaire H, et al. (2011) Glycan reader: Automated sugar identification and simulation preparation for carbohydrates and glycoproteins. *Journal of Computational Chemistry* 32(14): 3135–3141. DOI: 10.1002/jcc.21886.

Kannan SR, Spratt AN, Cohen AR, et al. (2021) Evolutionary analysis of the Delta and Delta Plus variants of the SARS-CoV-2 viruses. *Journal of Autoimmunity* 124(2021): 102715. DOI: 10.1016/j.jaut.2021.102715.

Ke Z, Oton J, Qu K, et al. (2020) Structures and distributions of SARS-CoV-2 spike proteins on intact virions. *Nature* 588(2020): 1–7. DOI: 10.1038/s41586-020-2665-2.

Klauda JB, Venable RM, Alfredo Freites J, et al. (2010) Update of the CHARMM All-Atom Additive Force Field for Lipids: Validation on Six Lipid Types. *The Journal of Physical Chemistry B* 114(23): 7830–7843. PMID: 20496934 DOI: 10.1021/jp101759q.

Lee H, Turilli M, Jha S, et al. (2019) DeepDriveMD: Deep-Learning Driven Adaptive Molecular Simulations for Protein Folding.In: 2019 IEEE/ACM Third Workshop on Deep Learning on Supercomputers (DLS), pp. 12–19.

Lee J, Cheng Xi, Jason M, et al. (2016) CHARMM-GUI Input Generator for NAMD, GROMACS, AMBER, OpenMM, and CHARMM/OpenMM Simulations Using the CHARMM36 Additive Force Field. *Journal of Chemical Theory and Computation* 12(1): 405–413. PMID: 26631602 DOI: 10.1021/acs.jctc.5b00935.

Lobo VR and Warwicker J (2021) Predicted pH-dependent stability of SARS-CoV-2 spike protein trimer from interfacial acidic groups. *Computational and Structural Biotechnology Journal* 19(2021): 5140–5148. DOI: 10.1016/j.csbj.2021.08.049.

Mandala VS, McKay MJ, Shcherbakov AA, et al. (2020) Structure and drug binding of the SARS-CoV-2 envelope protein transmembrane domain in lipid bilayers. *Nature Structural & Molecular Biology* 27(12): 1202–1208. DOI: 10.1038/s41594-020-00536-8.

Mariethoz J, Alocci D, Gastaldello A, et al. (2018) Glycomics@ExPASy: Bridging the Gap*. *Molecular and Cellular Proteomics* 17(11): 2164–2176. DOI: 10.1074/mcp.RA118.000799.

Markovetz MR, Subramani DB, Kissner WJ, et al. (2019) Endotracheal tube mucus as a source of airway mucus for rheological study. *American Journal of Physiology-Lung Cellular and Molecular Physiology* 317(4): L498–L509. PMID: 31389736 DOI: 10.1152/ajplung.00238.2019.

Martínez L, Andrade R, Birgin EG, et al. (2009) PACKMOL: A package for building initial configurations for molecular dynamics simulations. *Journal of Computational Chemistry* 30: 132157–132164. arXiv: DOI: 10.1002/jcc.21224.

McCallum M, Walls AC, Sprouse KR, et al. (2021) *Molecular Basis of Immune Evasion by the Delta and Kappa SARS-CoV-2 Variants*. bioRxiv. arXiv: DOI: 10.1101/2021.08.11.455956. https://www.biorxiv.org/content/early/2021/08/12/2021.08.11.455956.full.pdf

Miller SL, Nazaroff WW, Jimenez JL, et al. (2021) Transmission of SARS-CoV-2 by inhalation of respiratory aerosol in the Skagit Valley Chorale superspreading event. *Indoor Air* 31(2): 314–323. DOI: 10.1111/ina.12751.

Noe F (2020) *Machine Learning for Molecular Dynamics on Long Timescales*. Cham: Springer International Publishing, pp. 331–372. DOI: 10.1007/978-3-030-40245-7_16.

Oliveira ASF, Ciccotti G, Haider S, et al. (2021a) Dynamical nonequilibrium molecular dynamics reveals the structural basis for allostery and signal propagation in biomolecular systems. *The European Physical Journal B* 94(7): 144. DOI: 10.1140/epjb/s10051-021-00157-0.

Oliveira ASF, Edsall CJ, Woods CJ, et al. (2019) A General Mechanism for Signal Propagation in the Nicotinic Acetylcholine Receptor Family. *Journal of the American Chemical Society* 141(51): 19953–19958. PMID: 31805762 DOI: 10.1021/jacs.9b09055.

Oliveira ASF, Shoemark DK, Avila Ibarra A, et al. (2021b) The fatty acid site is coupled to functional motifs in the SARS-CoV-2 spike protein and modulates spike allosteric behaviour. *bioRxiv* 20arXiv: DOI: 10.1101/2021.06.07.447341. https://www.biorxiv.org/content/early/2021/06/09/2021.06.07.447341.full.pdf

Park SJ, Lee J, Qi Y, et al. (2019) CHARMM-GUI Glycan Modeler for modeling and simulation of carbohydrates and glycoconjugates. *Glycobiology* 29(4): 320–331. DOI: 10.1093/glycob/cwz003.

Petters MD and Kreidenweis SM (2007) A single parameter representation of hygroscopic growth and cloud condensation nucleus activity. *Atmospheric Chemistry and Physics* 7(8): 1961–1971. DOI: 10.5194/acp-7-1961-2007.

Phillips J, Zheng G, Kumar S, et al. (2002) NAMD: Biomolecular Simulation on Thousands of Processors. In:Proceedings of the IEEE/ACM SC2002 Conference. Baltimore, MD: IEEE Press, pp. 1–18. *Technical Paper 277*.

Phillips J., Braun R, Wang W, et al. (2005) *Scalable Molecular Dynamics with NAMD*. DOI: 10.1002/jcc.20289.

Phillips JC, Hardy DJ, Maia JDC, et al. (2020) Scalable molecular dynamics on CPU and GPU architectures with NAMD. *J. Chem. Phys* 153: 044130. DOI: 10.1063/5.0014475.

Páll S, Zhmurov A, Bauer P, et al. (2020) Heterogeneous parallelization and acceleration of molecular dynamics simulations in GROMACS. *The Journal of Chemical Physics* 153: 13134110. DOI: 10.1063/5.0018516.

Qiao Z, Welborn M, Anandkumar A, et al. (2020) OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features. *The Journal of Chemical Physics* 153: 12124111. DOI: 10.1063/5.0021955.

Raveh B, London N and Schueler-Furman O (2010) Sub-angstrom modeling of complexes between flexible peptides and globular proteins. *Proteins: Structure, Function, and Bioinformatics* 78(9): 2029–2040. arXiv: DOI: 10.1002/prot.22716. https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.22716

Russo JD, Zhang S and Leung JMG, et al. (2022) WESTPA 2.0: High-Performance Upgrades for Weighted Ensemble Simulations and Analysis of Longer-Timescale Applications. *Journal of Chemical Theory and Computation* 18: 638–649. DOI: 10.1021/acs.jctc.1c01154.

Saglam AS and Chong LT (2019) Protein–protein binding pathways and calculations of rate constants using fully-continuous, explicit-solvent simulations. *Chemical Science* 10(8): 2360–2372. DOI: 10.1039/c8sc04811h.

Salomon-Ferrer R, Götz AW, Duncan P, et al. (2013) Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 2. Explicit Solvent Particle Mesh Ewald. *Journal of Chemical Theory and Computation* 9(9): 3878–3888. DOI: 10.1021/ct400314y.

Sener Melih, Levy Stuart, Stone John E., et al (2021) *Multiscale Modeling and Cinematic Visualization of Photosynthetic Energy Conversion Processes from Electronic to Cell Scales*. Parallel Comput., p. 102698.

Stone JE, Hynninen A-P, Phillips JC, et al. (2016a) *Early Experiences Porting the NAMD and VMD Molecular Simulation and Analysis Software to GPU-Accelerated OpenPOWER Platforms*. International Workshop on OpenPOWER for HPC, pp. 188–206. (IWOPH'16).

Stone JE, Barry I and Schulten K (2013a) Early Experiences Scaling VMD Molecular Visualization and Analysis Jobs on Blue Waters. In: *Extreme Scaling Workshop (XSW)*, pp. 43–50. DOI: 10.1109/XSW.2013.10.

Stone JE, Sener M, Vandivort KL, et al. (2016b) Atomic Detail Visualization of Photosynthetic Membranes with GPU-Accelerated Ray Tracing. *Parallel Comput* 55: 17–27. DOI: 10.1016/j.parco.2015.10.015.

Stone JE, Vandivort KL and Schulten K (2013b) GPU-Accelerated Molecular Visualization on Petascale Supercomputing Platforms. In: Proceedings of the 8th International Workshop on Ultrascale Visualization (UltraVis '13). New York, NY, USA: ACM, p. 8. Article 6.

Suarez E, Lettieri S, Zwier MC, et al. (2014) Simultaneous computation of dynamical and equilibrium information using a weighted ensemble of trajectories. *Journal of Chemical Theory and Computation* 10(7): 2658–2667.

Sugio S, Kashima A, Mochizuki S, et al. (1999) Crystal structure of human serum albumin at 2.5 Å resolution. *Protein Engineering, Design and Selection* 12(6): 439–446. arXiv: DOI: 10.1093/protein/12.6.439. https://academic.oup.com/peds/article-pdf/12/6/439/18543407/120439.pdf

Surya W, Li Y and Torres J (2018) Structural model of the SARS coronavirus E channel in LMPG micelles. *Biochimica et Biophysica Acta (BBA) - Biomembranes* 1860(6): 1309–1317. DOI: 10.1016/j.bbamem.2018.02.017.

Symmes BA, Stefanski AL, Magin CM, et al. (2018) Role of mucins in lung homeostasis: regulated expression and biosynthesis in health and disease. *Biochemical Society Transactions* 46(3): 707–719. arXiv: DOI: 10.1042/BST20170455. https://portlandpress.com/biochemsoctrans/article-pdf/46/3/707/479418/bst-2017-0455c.pdf

Sztain T, Ahn S-H, Bogetti AT, et al. (2021) A glycan gate controls opening of the SARS-CoV-2 spike protein. *Nature Chemistry* 13(10): 963–968. DOI: 10.1038/s41557-021-00758-3.

Thomsson KA, Schulz BL, Packer NH, et al. (2005) MUC5B glycosylation in human saliva reflects blood group and secretor status. *Glycobiology* 15(8): 791–804. arXiv: DOI: 10.1093/glycob/cwi059. https://academic.oup.com/glycob/article-pdf/15/8/791/1787060/cwi059.pdf

Torrillo PA, Bogetti AT and Chong LT (2021) A Minimal, Adaptive Binning Scheme for Weighted Ensemble Simulations. *The Journal of Physical Chemistry A* 125(7): 1642–1649. DOI: 10.1021/acs.jpca.0c10724.

Turonova B, Sikora M, Schürmann C, et al. (2020) In situ structural analysis of SARS-CoV-2 spike reveals flexibility mediated by three hinges. *Science* 370(6513): 203–208. DOI: 10.1126/science.abd5223.

Vejerano EP and Marr LC (2018) Physico-chemical characteristics of evaporating respiratory fluid droplets. *Journal of the Royal Society Interface* 15(139): 1–10. DOI: 10.1098/rsif.2017.0939.

Venable RM, Luo Y, Gawrisch K, et al. (2013) Simulations of Anionic Lipid Membranes: Development of Interaction-Specific Ion Parameters and Validation Using NMR Data. *The Journal of Physical Chemistry B* 117(35): 10183–10192. PMID: 23924441 DOI: 10.1021/jp401512z.

Walker JS, Archer J, Florence K, et al. (2021) Accurate Representations of the Microphysical Processes Occurring during the Transport of Exhaled Aerosols and Droplets. *ACS Central Science* 7(1). DOI: 10.1021/acscentsci.0c01522.

Walls AC, YoungPark J, Alejandra Tortorici M, et al. (2020) Structure, Function, and Antigenicity of the SARS-CoV-2 Spike Glycoprotein. *Cell* 181(2): 281–292. DOI: 10.1016/j.cell.2020.02.058.

Wang CC, Prather KA, Sznitman J, et al. (2021) Airborne transmission of respiratory viruses. *Science* 373(6558): eabd9149. DOI: 10.1126/science.abd9149.

Warwicker J (2021) A model for pH coupling of the SARS-CoV-2 spike protein open/closed equilibrium. *Briefings in Bioinformatics* 22(2): 1499–1507. arXiv: https://doi.org/10.1093/bib/bbab056. https://academic.oup.com/bib/article-pdf/22/2/1499/36654668/bbab056.pdf

Williams M (2021) *Cluster in the Cloud*. https://cluster-in-the-cloud.readthedocs.io

Wrapp D, Wang N, Corbett KS, et al. (2020) Cryo-EM structure of the 2019-nCoV spike in the prefusion conformation. *Science* 367(6483): 1260–1263. DOI: 10.1126/science.abb2507.

Zimmerman MI, Porter JR, Ward MD, et al. (2021) SARS-CoV-2 simulations go exascale to predict dramatic spike opening and cryptic pockets across the proteome. *Nature Chemistry* 13(7): 651–659. DOI: 10.1038/s41557-021-00707-0.

Zuckerman DM and Chong LT (2017) Weighted Ensemble Simulation: Review of Methodology, Applications, and Software. *Annual Review of Biophysics* 46: 43–57. DOI: 10.1146/annurev-biophys-070816-033834.

Zwier MC, Adelman JL, Kaus JW, et al. (2015) WESTPA: An Interoperable, Highly Scalable Software Package for Weighted Ensemble Simulation and Analysis. *Journal of Chemical Theory and Computation* 11(2): 800–809. DOI: 10.1021/ct5010615.

E Pedagogical paper on trajectory analysis

A gentle introduction to the non-equilibrium physics of trajectories: Theory, algorithms, and biomolecular applications

Daniel M. Zuckerman and John D. Russo

Citation: American Journal of Physics **89**, 1048 (2021); doi: 10.1119/10.0005603
View online: https://doi.org/10.1119/10.0005603
View Table of Contents: https://aapt.scitation.org/toc/ajp/89/11
Published by the American Association of Physics Teachers

The Computational Physics Section publishes articles that help students and instructors learn about the computational tools used in contemporary research. Interested authors are encouraged to send a proposal to the editors of the Section, Jan Tobochnik (jant@kzoo.edu) or Harvey Gould (hgould@clarku.edu). Summarize the physics and the algorithm you wish to discuss and how the material would be accessible to advanced undergraduates or beginning graduate students.

A gentle introduction to the non-equilibrium physics of trajectories: Theory, algorithms, and biomolecular applications

Daniel M. Zuckerman[a] and John D. Russo
Department of Biomedical Engineering, Oregon Health & Science University, Portland, Oregon 97239

(Received 20 January 2021; accepted 25 June 2021)

Despite the importance of non-equilibrium statistical mechanics in modern physics and related fields, the topic is often omitted from undergraduate and core-graduate curricula. Key aspects of non-equilibrium physics, however, can be understood with a minimum of formalism based on a rigorous trajectory picture. The fundamental object is the ensemble of trajectories, a set of independent time-evolving systems, which easily can be visualized or simulated (e.g., for protein folding) and which can be analyzed rigorously in analogy to an ensemble of static system configurations. The trajectory picture provides a straightforward basis for understanding first-passage times, "mechanisms" in complex systems, and fundamental constraints on the apparent reversibility of complex processes. Trajectories make concrete the physics underlying the diffusion and Fokker–Planck partial differential equations. Last but not least, trajectory ensembles underpin some of the most important algorithms that have provided significant advances in biomolecular studies of protein conformational and binding processes.

https://doi.org/10.1119/10.0005603

I. INTRODUCTION

Most of the phenomena we encounter in daily life, from weather to cooking to biology, are fundamentally out of equilibrium and require physics typically not touched on in the undergraduate or even graduate physics curricula. Many physics students are alarmed at the complexity and abstraction of thermodynamics and "sadistical mechanics," and understandably would not seek out instruction in non-equilibrium statistical physics. Yet there is a surprising range of fundamental non-equilibrium material that can be made accessible in a straightforward way using *trajectories*, which are essentially movies of systems executing their natural dynamics. The trajectory picture first and foremost is fundamental;[1–3] for example, dynamics generate equilibrium, but not the other way around.[4] It can also lead, with a minimum of mathematics, to understanding key non-equilibrium phenomena (relaxation and steady states) and similarly to extremely powerful cutting-edge simulation methods (path sampling). Students deserve a taste of this material.

Why are trajectories fundamental? A trajectory is simply the sequence of phase-space points through which a system passes, recorded perhaps as a "movie" listing all atomic positions and velocities at evenly spaced time points—the "frames" of the movie. Such movies are fundamental because, as we learned from Newton, nature creates forces that lead to dynamics,[5] i.e., to trajectories. We may attempt to describe the dynamics in various average ways—e.g., using equilibrium ideas—but the trajectories are the basis of everything. Theories, such as equilibrium statistical mechanics, generally build in assumptions, if not approximations. In fact, the most fundamental definition of equilibrium itself derives from dynamics, via detailed balance,[1,4,6] whereby there must be an equal-and-opposite balance of flows between any two microstates.

Dynamical descriptions generally have more information in them than average or equilibrium theories.[4,7,8] As a simple example, perhaps you know that someone sleeps eight hours a day. However, that average hides the time at which sleep occurs as well as whether it includes an afternoon nap. In the case of diffusion, we know that particles observed in a localized region will tend to spread out over time. However, if we only observe the spatial density, we do not know which particles went where. Trajectories, which track particles over time, inherently capture this information.

A trajectory ensemble description, as described below, provides *the* key observables for transition processes: rate and mechanism. In a biomolecular context, these are essentially everything we want to know. Consider protein folding. We want to know how fast proteins fold and how folding rates change under specific mutations.[9,10] We also want to know the mechanism of folding: the conformations that are visited during the process which in turn can illuminate chemical-structure causes of rate changes due to mutation.[10,11] Other conformational processes in biomolecules arguably are of even greater interest, such as binding[12] and allostery,[10,13] due to their implications for drug design; here again, rate and mechanism are of utmost importance.[14,15]

This article will explain the theory of trajectory ensembles, starting with simple diffusion and moving to systems with complex energy landscapes. We will explore essential aspects of non-equilibrium statistical mechanics, focusing on timescale quantification via the mean first-passage time. The understanding of non-equilibrium trajectory ensembles leads directly to the "super parallel" weighted ensemble simulation methodology, widely used in computational biology,[16] which is explored in a one-dimensional pedagogical example. A number of exercises are given along with clearly demarcated more advanced material.

The statistical mechanics of trajectories has been addressed pedagogically, in different ways, in prior work. Clear, basic-level descriptions can be found in some textbooks[3,4] and path-sampling papers in the molecular-oriented literature.[17–19] Astumian and co-workers highlighted the importance of trajectories and their probabilistic description in multiple contexts[20,21] and provided important semi-microscopic, discrete-state descriptions of molecular motors,[22,23] building on the seminal work of Hill.[24,25] Ghosh and co-workers employed trajectory concepts in presenting Jaynes's maximum-caliber approach to inferring kinetics;[26] note the related work by Pressé co-workers[27] and Ghosh co-workers.[28] Swendsen's discussion of irreversibility is also of interest,[29] as is the classic treatment by Chandrasekhar.[30] The present discussion attempts to provide a more elementary discussion of trajectory physics, with a focus on computational applications not found in most prior work. Perhaps unexpectedly, the path sampling algorithms derivable from the present description are very much at the leading edge of molecular computation.[31]

II. BASICS: DYNAMICS AND TRAJECTORIES

In this section, we introduce the building blocks of our analysis, starting from one-dimensional Newtonian motion. We add fundamental stochastic elements and then develop the trajectory picture with an associated numerical recipe.

A. Stochastic dynamics

The starting point for our quantitative trajectory description is the simplest form of stochastic dynamics, often called Brownian dynamics, which we will justify starting from Newton's second law. Brownian dynamics are also known by more intimidating terminology, as overdamped Langevin dynamics, but their essence is simple to understand. As a familiar reference, we first write the one-dimensional (1D) law of classical motion,

$$m\frac{dx^2}{dt^2} = f, \tag{1}$$

where m is mass, x is position, and $f = -dU/dx$ is the force, with $U(x)$ the potential energy. Advancing one step in complexity, the 1D Langevin equation models motion in a viscous (frictional) medium by adding a damping force that always opposes the direction of motion (velocity), as well as a random force f_{rand} from collisions,[4,6] yielding

$$m\frac{d^2x}{dt^2} = f - \gamma m\frac{dx}{dt} + f_{\text{rand}}, \tag{2}$$

where $\gamma > 0$ is the friction constant, effectively a collision frequency, as can be seen by dimensional analysis. Details of the random force will be given later. Both forces are needed; otherwise, damping would eliminate all motion.

In the *overdamped* limit, inertia is ignored. This is akin to motion in a beaker filled with thick oil: there is minimal tendency for an object to continue in any given direction in the absence of force; with a force such as gravity, terminal (constant) velocity is reached quickly; i.e., no further acceleration occurs despite the force. At microscopic scales, however, there continues to be random thermal motion due to molecular collisions. Setting the inertial term $m\,d^2x/dt^2$ to zero in Eq. (2) and re-arranging terms, the overdamped Langevin equation is[4,6]

$$\frac{dx}{dt} = \frac{1}{m\gamma}(f + f_{\text{rand}}). \tag{3}$$

This simplified equation of motion may look unusual to those unfamiliar with it, but studying its application in a numerical context will make its physical basis and relation to diffusion more clear.

B. Time-discretized overdamped dynamics and computation

We will make most use of a discrete-time picture (fixed time steps) which not only greatly simplifies the mathematics but also translates directly into simple computer implementation. If we discretize the dynamics of Eq. (3) by writing the velocity as $\Delta x/\Delta t$ and multiplying through by Δt, we arrive at a very useful equation,

$$\Delta x = \frac{\Delta t}{m\gamma}(f + f_{\text{rand}}) = \Delta x_{\text{det}} + \Delta x_{\text{rand}}, \tag{4}$$

where $\Delta x_{\text{det}} = f\Delta t/m\gamma$ is the deterministic component of the spatial step due to an external force (e.g., molecular, gravitational, or electrostatic) and Δx_{rand} is the random part due to thermal molecular collisions. At finite temperature, microscopic motion must not cease; hence, in the Langevin picture, thermal fluctuations must balance "dissipation" due to damping of the γ term.[6,32] To accomplish this, Δx_{rand} is typically assumed to follow a zero-mean Gaussian distribution which must have its variance given by[4]

$$\sigma^2 = 2k_BT\Delta t/m\gamma. \tag{5}$$

The Gaussian assumption is justified on the basis of the central limit theorem[4] because a molecule in aqueous solution can experience upwards of 10^{13} collisions per second,[6,30] and hence, a large number of collisions occur in any $\Delta t > 1$ ns. The high collision frequency also justifies the implicit assumption here that sequential Δx_{rand} values are independent, i.e., not time-correlated.

With the distribution of Δx_{rand} specified, the discrete overdamped dynamics Eq. (4) is simultaneously a prescription for computer simulation of trajectories *and* directly implies a probabilistic description of trajectories. Let us start with computer simulation, which is simpler by far. Defining $x_j = x(t = j\Delta t)$, Eq. (4) is essentially a recipe for calculating the next position $x_{j+1} \equiv x_j + \Delta x$ in a time-sequence, given x_j. For a sufficiently small time step Δt, the force f will be approximately constant over the whole time interval, so we take

$$\Delta x_{\text{det}} = f(x_j)\Delta t/m\gamma, \tag{6}$$

and Δx_{rand} is chosen from a Gaussian (normal) distribution of variance σ^2 from Eq. (5). Looping over this process yields a discrete-time *trajectory*,

$$\text{traj} = \{x_0, x_1, x_2, \ldots\}, \tag{7}$$

which is just a list of positions at intervals of Δt. We can easily recast trajectory elements in terms of spatial increments,

$$\begin{aligned} x_0 &= x_0 \quad (\text{arbitrary}),\\ x_1 &= x_0 + \Delta x_1,\\ x_2 &= x_0 + \Delta x_1 + \Delta x_2 = x_1 + \Delta x_2,\\ &\ldots, \end{aligned} \tag{8}$$

which is useful for understanding simulation algorithms such as Eq. (4).

Trajectories of *simple diffusion* can be generated from Eq. (4) by setting $f = 0$ (hence $\Delta x_{\text{det}} = 0$). The recipe given above simplifies to choosing a Gaussian random step at each time point, i.e.,

$$\Delta x = \Delta x_{\text{rand}} \quad (\text{simple diffusion}), \tag{9}$$

as we would expect. Schematic examples of these simplest stochastic trajectories are shown in Fig. 1. There is no directionality in simple diffusion, but only a statistical tendency to diffuse away from the starting point, as will be quantified below.

III. SIMPLE DIFFUSION IN THE TRAJECTORY PICTURE

The basics of diffusion, such as Fick's law and the diffusion equation, are well known, so diffusion theory is a perfect context for introducing the trajectory formulation. Students may find that following the behavior of individual particles is a more concrete exercise than visualizing probability distributions. In this section, we show that the trajectory approach yields the familiar average description of simple diffusion in a force-free (constant-energy) landscape. In the bigger picture, we get an explicit sense of physical details of trajectories which are averaged (integrated) out to yield the distribution picture.

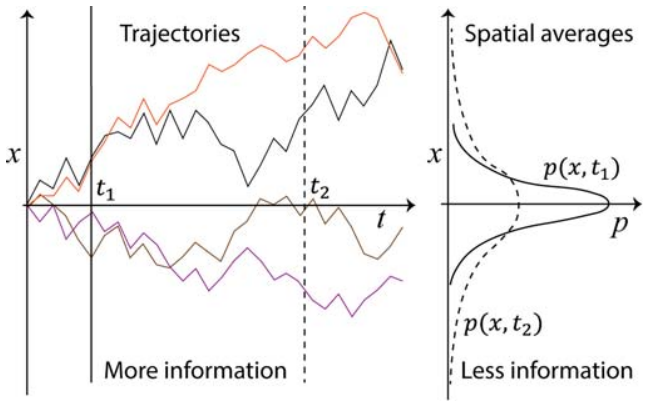


Fig. 1. Simple diffusion, two ways. At left are schematic time-discretized trajectories illustrating one-dimensional diffusion started from the initial point $x_0 = 0$. Averaging over the positions of many trajectories at specific time points t_1 and t_2 yields the distributions shown at right, with $p(x, t_i) = p(x_i|x_0)$. Averaging can aid interpretation but it also removes information, namely, the connectivity among the trajectories' sequences of points.

A. Probabilistic picture for trajectories

We start by analyzing diffusive trajectories based on random steps where the force f has been set to zero. The procedure (Eq. (9)) of repeatedly choosing a Gaussian step with variance from Eq. (5) implicitly but precisely defines a probability distribution for an entire trajectory (Eq. (7)), which will prove of fundamental importance. First, by construction, the probability of a *single* step Δx is given by

$$p_1(\Delta x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\Delta x^2/2\sigma^2}. \tag{10}$$

This is the meaning of choosing a Gaussian step. Note that Eq. (10) depends only on the magnitude and not on the starting point of the specific step, which is a characteristic of simple diffusion because no forces are present.

For the full trajectory, we use the simple rule that the probability of a sequence of independent steps is simply the product of the individual step probabilities: think of a sequence of fair coin flips characterized by 1/2 to the appropriate power. Hence, for an N-step trajectory defined by Eq. (8) starting from x_0, we have

$$p(\text{traj}) = p_1(\Delta x_1) \cdot p_1(\Delta x_2), \cdots, p_1(\Delta x_N) \tag{11}$$

$$= \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N \prod_{j=1}^{N} e^{-\Delta x_j^2/2\sigma^2}. \tag{12}$$

A multi-dimensional distribution such as Eq. (12) may not be trivial to understand for those not used to thinking in high dimensions. First, why is it a multi-dimensional distribution? Well, it describes the distribution of a *set* of points, the trajectory $\{x_0, x_1, x_2, \ldots, x_N\}$. Note that we immediately obtain the Δx values needed for Eq. (12) from the x values using Eq. (8): $\Delta x_1 = x_1 - x_0$ and so on. So if you are given a set of (trajectory) x values, you can convert them into Δx values and plug them into Eq. (12) to get the probability of that trajectory. You can do this for *any* set of x values, even ridiculously unphysical values with gigantic jumps, but of course the probability will be tiny for unphysical trajectories. For completeness, strictly speaking, Eq. (12) is a probability *density*[4] and absolute probabilities are only obtained by integrating over a finite region.

The distribution of trajectories encodes all the information we could possibly want about diffusive behavior, although some math is needed to get it. Alternatively, as a proxy for the distribution, multiple trajectories could be simulated to quantify their average behavior. In the case of simple diffusion, however, the math of the trajectory distribution is both tractable and illuminating.

As a fascinating technical aside, note that the product of exponentials in Eq. (12) can be re-written as the exponential of a sum $(-\sum_j \Delta x_j^2/2\sigma^2)$, which makes the probability look somewhat like a Boltzmann factor. Indeed, consulting the definition of σ^2 in Eq. (5), we find it is proportional to $k_B T$. Of course, the argument of our exponential is not a true energy, but can be considered an effective path energy, known as the "action."[4,33]

(In the non-diffusive case, $\Delta x_{\text{det}} \neq 0$ leads to an additional term in the exponent and the action; see below.) The action formulation, and the consideration of all possible paths, is the heart of the path-integral formulation of quantum mechanics.[34] The path-probability formulation is truly fundamental to physics.

B. Deriving the spatial distribution from trajectories

A key observable of interest is the distribution of x values at a fixed but arbitrary time point (Fig. 1). To build up to this, we will carefully derive the equation for the conditional probability distribution $p(x_2|x_0)$ of $x_2 = x(2\Delta t)$ values, i.e., the distribution for a fixed starting point x_0. The critical idea is that we can obtain the probability of any given x_2 value by summing (i.e., integrating) over all possible two-step trajectories that reach the particular value starting from x_0. Because both forward and backward motion are possible for the intermediate step, we must consider all possible x_1 values. Mathematically, this amounts to

$$\begin{aligned} p(x_2|x_0) &= \int_{-\infty}^{\infty} dx_1 \; p_1(\Delta x_1) \; p_1(\Delta x_2), \\ &= \int_{-\infty}^{\infty} d\Delta x_1 \; p_1(\Delta x_1) \cdot p_1(x_2 - (x_0 + \Delta x_1)), \end{aligned} \tag{13}$$

where we have used Eq. (11) to start and then Eq. (8) to substitute for Δx_2.

We can evaluate the integral in Eq. (13) exactly. Plugging in the expression for p_1 from Eq. (10) and setting $y = \Delta x_1$, we have

$$\begin{aligned} p(x_2|x_0) &= \frac{1}{2\pi\sigma^2} \int_{-\infty}^{\infty} dy \, e^{-y^2/2\sigma^2} e^{-(x_2 - x_0 - y)^2/2\sigma^2}, \\ &= \frac{1}{2\pi\sigma^2} e^{-(x_2 - x_0)^2/4\sigma^2} \int_{-\infty}^{\infty} dy \, e^{-[y - (x_2 - x_0)/2]^2/\sigma^2}, \\ &= \frac{1}{\sqrt{2\pi}\left(\sqrt{2}\sigma\right)} e^{-(x_2 - x_0)^2/2\cdot 2\sigma^2}, \end{aligned} \tag{14}$$

where the second line is derived by completing the square in the exponent and the third line is derived by performing the Gaussian integral shown.

The result Eq. (14) for the distribution of positions after $2\Delta t$ is very informative, especially by comparison to the single-step distribution Eq. (10). The distribution of possible outcomes is still a Gaussian of mean x_0, but the variance is doubled; equivalently, the standard deviation has increased by a factor of $\sqrt{2}$. See Fig. 1. It is important that *we derived this distribution by averaging (integrating) over the ensemble of two-step trajectories*. As promised, the information was indeed encoded in the original trajectory distribution Eq. (12).

From here, it is not hard to generalize to an arbitrary number of steps by repeating the integration process. The result is that the distribution of $x_n = n\Delta t$ values is also a Gaussian with mean x_0, but with variance $n\sigma^2$,

$$p(x_n|x_0) = \frac{1}{\sqrt{2n\pi}\sigma} e^{-(x_n - x_0)^2/2n\sigma^2}. \tag{15}$$

Equation (15) embodies the usual description of diffusion, as we will see in two ways, but it also contains *less* information than our initial trajectory description.

C. Confirming the probabilistic description of diffusion

Have we really recapitulated the usual description of diffusion? As a first check, we immediately recover the expected linear time dependence of the mean-squared displacement[4] based on Eq. (15). This is because the variance *is* the mean-squared displacement or deviation (MSD), and the number of steps $n = t/\Delta t$ is simply proportional to time. By the definition of a Gaussian distribution, the variance implicit in Eq. (15) is $n\sigma^2$, and we therefore have

$$\begin{aligned} \text{MSD} \equiv (x_n - x_0)^2 &= \int_{-\infty}^{\infty} dx_n (x_n - x_0)^2 p(x_n|x_0), \\ &= n\sigma^2 = (t/\Delta t)\sigma^2. \end{aligned} \tag{16}$$

If we define the diffusion constant via $\text{MSD} = 2Dt$ (in one dimension), then from Eqs. (5) and (16), we derive $D = k_B T/m\gamma$, which is a well-known result.[4]

Second, by renaming the variable $x_n \to x = x(t)$ in Eq. (15) and noting that time $t = n\Delta t$, we can see that Eq. (15) describes the time-evolving probability distribution of positions $p(x, t)$, which is the well-known solution to the 1D diffusion equation,

$$\frac{\partial p}{\partial t} = D \frac{\partial^2 p}{\partial x^2}. \tag{17}$$

This can be verified by direct differentiation, but see Sec. VIII for a hint. The agreement with the *continuous*-time diffusion equation implies that time discretization is irrelevant, but be warned that this is not always the case, as discussed in Sec. VIII.

D. What is missing from the standard description of diffusion?

Because the distribution of positions (Eq. (15)) is known for any time and provides the exact solution to the diffusion equation, it may seem there is nothing more to know. However, the key observables—the timescale (or rate) and mechanism of any particular process—either are not available at all from the positional distribution or not easily available.[35,36]

These shortcomings stem from the information missing from the spatial distribution. Even if we know the spatial distribution at two times, *we still do not know how any given diffusing particle went from one place to another*. That is, although we know the fraction of particles that will be located between any x and $x + dx$, we do not know which came from left or right and exactly from where. This information is encoded in the dynamics and recorded in the distribution of trajectories Eq. (11), which is essentially a distribution of paths taken through position space. It is fair to say, therefore, that the trajectory distribution *is* the mechanism, assuming that all trajectories considered conform to criteria of interest (e.g., starting at $x = 0$ and perhaps reaching a value $x > a$ after n steps.)

E. Beyond simple diffusion in one dimension

Before we move beyond a single dimension, a useful reference for developing intuition is the generalization of the single-step distribution Eq. (10) when a force is present. Reframing the procedure Eq. (4) probabilistically, the

distribution for overdamped dynamics of a 1D particle in the presence of a spatially varying potential $U(x)$ is a different Gaussian,

$$p_1(\Delta x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(\Delta x - \Delta x_{\text{det}})^2/2\sigma^2}. \tag{18}$$

In contrast to the simple-diffusion case Eq. (10), the distribution of possibilities is centered on the deterministic (force-driven or "drift") step Δx_{det} defined by Eq. (6). That is, the particle tends to move in the direction of the force, albeit stochastically.

Equation (18) should guide your intuition for single-step motion of a stochastic system: there is a distribution of possibilities centered on the deterministic step. The deterministic component generally could depend on inertia and/or force although in the overdamped case there is no inertia. Note that Δx_{det} in Eq. (18) implicitly depends on the starting position for the step: see Eq. (6). Below, we will make the position dependence more explicit.

IV. THE NON-EQUILIBRIUM STEADY STATE (NESS) AND THE HILL RELATION FOR RATES

Probably the most important observable in a dynamical process, at least in biomolecular studies, is the rate for a process. As we will see, the rate is closely related to a specific non-equilibrium steady state, which is essential to understand but also quite accessible.

Physicists often quantify a rate via the *mean first-passage time* (MFPT).[32,35–37] The first-passage time is simply the time required for a process from start to finish, e.g., the time required for a protein to fold, starting from when it is initialized in an unfolded state. In Fig. 2, this is the time from initiation in "source" state A to absorption in "sink" state B. (We are thus employing source-sink boundary conditions.) Chemists and biochemists quantify kinetics via the "rate constant" for a conformational process like protein folding, which has units of s^{-1} and can be defined as the reciprocal MFPT, although chemists prefer a definition based on directly measurable "relaxation times."[4,38,39] Our discussion will focus solely on the MFPT for simplicity.

The MFPT can be directly obtained from a steady-state trajectory ensemble, so we will start by defining a source-sink non-equilibrium steady state (NESS) as sketched in Fig. 2. Independent trajectories are initiated in the source macrostate A (e.g., the set of unfolded protein configurations) according to a specified distribution p_0 (e.g., a single configuration or the equilibrium distribution over A). A second, non-overlapping sink macrostate B is an absorbing state in that trajectories reaching B are terminated, although in our source-sink setup they are immediately restarted in A selected according to the p_0 distribution. If this process is allowed to run for long enough so that each trajectory has reached B and been recycled back to A many times, the system will reach a non-equilibrium steady state. Without a sink state or recycling, the system will relax to equilibrium, which is also a steady state. (See Sec. VIII to explore the difference between equilibrium and other steady states.)

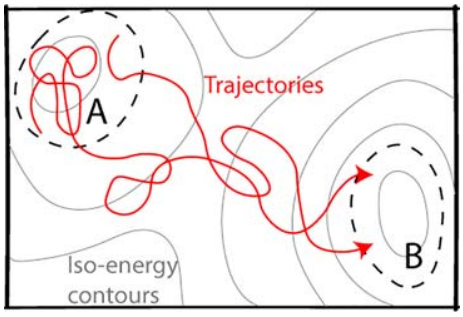


Fig. 2. Source-sink non-equilibrium steady state. Trajectories (red curves, color online) are initiated in state A and terminated upon reaching state B, with states bounded by dashed contours. Importantly, trajectories that reach B are then re-initiated from A. Such a system will reach a non-equilibrium steady state after a transient "relaxation" period. Gray solid lines show iso-energy contours of a schematic landscape.

The MPFT is derivable from a NESS trajectory ensemble in a direct way, which will seem obvious once we are aware of it. The derivation is simple, but requires some thought. Imagine we have a large number $M \gg 1$ of independent systems that together make up the source-sink NESS (Fig. 2). By construction, the NESS is characterized by a *constant flow* of trajectories into B. We can simply count the number of trajectories arriving during some time interval τ and call this count m. Thus, a fraction m/M of the total probability arrives in time τ.

To continue our derivation, we can estimate this same fraction of trajectories arriving based solely on the meaning of the MFPT. By definition, the average amount of time a trajectory requires to traverse from A to B *is* the MFPT, so the (average) probability for any given trajectory to arrive during an interval τ is precisely τ/MFPT, which in turn is the same as the fraction expected to arrive in τ. In other words, $m/M = \tau/\text{MFPT}$, and we have derived the *Hill relation*,[4,25]

$$\frac{1}{\text{MFPT}} = \frac{m/M}{\tau} = \text{Flux}(A \to B\,|\text{NESS}), \tag{19}$$

where the flux is the probability arriving to B per unit time in the NESS. Equation (19) is an exact relation with no hidden assumptions, although not surprisingly the MFPT is particular to the initiating distribution p_0 of the particular NESS in which the flux is measured. That is, the MFPT depends on where in state A trajectories are initiated.

The Hill relation hints at a remarkable possibility: estimation of a long timescale (the MFPT) based on an arbitrary short period of observation (τ). If this could be done routinely, it would represent a major accomplishment in computational physics.[31] In Sec. VI, we describe a simple algorithm that can leverage the Hill relation for practical computations in many systems. We also explain the challenges involved.

V. MORE ADVANCED DISCUSSION OF ENSEMBLES AND THERMODYNAMIC STATES

This section describes additional fundamental concepts in non-equilibrium physics, but the discussion necessarily becomes more technical. Readers can skip this section without compromising their ability to understand subsequent material.

A. Notation and nomenclature for multi-dimensional systems

We will frame our discussion a bit more generally in the context of multi-dimensional systems. Fortunately, this

extension adds only incremental conceptual and mathematical complexity. To keep notation as simple as possible, we will use $\vec{x}$ to represent *all* microscopic coordinates—the phase-space vector consisting of all positions and velocities of all atoms in our classical representation. In some cases, such as overdamped dynamics (Eq. (3)), velocities may be excluded from the description, but the $\vec{x}$ notation remains valid. A *macrostate* is defined to be a *set* of $\vec{x}$ points. These macrostates are not to be confused with thermodynamic states such as equilibrium at some constant temperature or a non-equilibrium steady state.

As with our discussion of simple diffusion above, we will strictly use discrete time: $t = 0,\ \Delta t,\ 2\Delta t, \ldots$. Discrete time greatly simplifies our description of trajectory probabilities without sacrificing any physical insights. In a trivial extension of Eq. (7), we therefore write a trajectory as

$$\text{traj} = \{\vec{x}_0, \vec{x}_1, \vec{x}_2, \ldots\}, \tag{20}$$

where $\vec{x}_j$ is the phase point at time $t = j\Delta t$.

B. The initialized trajectory ensemble in multiple dimensions

The probabilistic description of a multi-dimensional trajectory follows logic almost identical to the 1D diffusion formulation of Eq. (11), except for two details. First, we now include the possibility that the initial system phase point $\vec{x}_0$ itself is chosen from some distribution p_0. Second, in contrast to simple diffusion, where the distribution Eq. (10) of outcomes p_1 for any single step depends only on the magnitude of Δx, more generally the outcome depends on the starting point of the step because the force may vary in space. We therefore adopt a notation which makes this explicit: $p_1(\vec{x}_{j-1} \to \vec{x}_j) = p_1(\vec{x}_j|\vec{x}_{j-1})$ is the (conditional) probability distribution for $\vec{x}_j$ values, given the prior position $\vec{x}_{j-1}$. The probability of a full trajectory is then the product of the initial distribution and the sequence of stepwise distributions,

$$\begin{aligned} p(\text{traj}) = p_0(\vec{x}_0) \cdot p_1(\vec{x}_0 \to \vec{x}_1) \\ \cdot p_1(\vec{x}_1 \to \vec{x}_2) \cdots p_1(\vec{x}_{N-1} \to \vec{x}_N). \end{aligned} \tag{21}$$

The mathematical form of p_1 must now account for multi-dimensional aspects of the system, as well as any forces or inertia if present: see Eq. (18) and the discussion following it. Although specifying p_1 in generality is beyond the scope of our discussion, we should note that the form of Eq. (21) indicates we have assumed Markovian behavior: the distribution of outcomes p_1 at any time depends only on the immediately preceding time point.

We must be careful to specify our system without ambiguity. A given physical system, such as a particular protein molecule in a specified solvent at known temperature and pressure, can be considered in a variety of thermodynamic states, such as equilibrium or a non-equilibrium state. The system and the thermodynamic conditions both must be specified. Conveniently, the two aspects are described by different parts of the trajectory distribution equation Eq. (21): the intrinsic physical properties such as forces and dynamics are encoded in the single-step p_1 factors, while the thermodynamic state or ensemble is determined by the initial distribution p_0 along with boundary conditions. Some boundary conditions will be discussed below.

The distribution Eq. (21) describes the *initialized trajectory ensemble*, the set of trajectories originating from a specified phase-point distribution p_0 at time $t = 0$. For instance, p_0 could represent a single unfolded protein configuration (making p_0 a Dirac delta function), a set of unfolded configurations, a solid in a metastable state, or the set of initial positions of multiple dye molecules in a solvent. Figure 1 illustrates the one-dimensional trajectory ensemble initialized from $p_0(x) = \delta(x)$.

As with simple diffusion, we can revert to the simpler, averaged description of a spatial distribution that evolves in time due to the dynamics. That is, in principle, we can calculate the distribution of phase points at time $t = N\Delta t$ starting from p_0, denoted $p(\vec{x}_N|\,p_0)$. When forces are present, the diffusion (partial differential) equation Eq. (17) must be generalized to account for the tendency of a particle to move a certain direction, leading to the Fokker–Planck/Smoluchowski picture.[32,35,36] Appendix A describes the corresponding Smoluchowski equation that governs overdamped motion with forces. However, as with simple diffusion, the spatial distribution represents an average over the information-richer trajectories.

C. Connection to relaxation, state populations, and thermodynamics

It is important to note that, in general, an initialized system will "relax" away from its initial distribution p_0. For systems of interest here, the system's phase-point distribution $p(\vec{x}_N|\,p_0)$ will tend to relax toward a *steady state* dependent only on the boundary conditions. In a constant-temperature system with no particle exchange, for example, the distribution will approach equilibrium as embodied in the Boltzmann factor: $\lim_{N\to\infty} p(\vec{x}_N|\,p_0) \propto \exp\left(-H(\vec{x})/k_BT\right)$, where $H(\vec{x})$ is the total energy of point $\vec{x}$, k_B is Boltzmann's constant, and T is the absolute temperature. In general, whether equilibrium or not, the steady state that is reached typically will be independent of p_0 after sufficient time for a "well-behaved" system. In Sec. IV, we explored *non-equilibrium* steady states critical to understanding conformational transitions.

Whether the system is in the relaxation or steady regime, the phase-point distribution $p(\vec{x})$, obtainable from the trajectory picture, directly connects to observable and thermodynamic properties. Most simply, the time-dependent macrostate population can be obtained as the integral of $p(\vec{x})$ over a region of phase space: this is the fraction of probability in the state which evolves in time with p. At a system-wide level, both the entropy and average energy can be obtained from well-known integrals over p.[4,7] These also evolve with time, directly leading to the entropy production picture. Further detail on these topics is beyond the scope of the present discussion, and interested readers should consult suitable Refs. 7 and 27.

D. The ensemble of trajectories and the meaning of equilibrium

When we speak of an "ensemble" of trajectories, the word has the same meaning as in ordinary statistical mechanics,[4,6] namely, a set of *fully independent* trajectories generated under the conditions of interest (see below). That is, each member of the ensemble is a replica of the same physical system but is initiated from a phase point that typically will differ from others in the ensemble.

An ensemble in principle can be generated according to any process and under any conditions we care to specify. The dynamics of these trajectories could be governed by simple constant-temperature diffusion or there could be a temperature gradient, forces, or both. Trajectories could additionally be subject to certain boundary conditions: for example, they might be assumed to reflect off some boundary in phase space or be absorbed on reaching a certain "target" region as we considered in Sec. IV. The full set of rules governing a set of trajectories defines the ensemble by determining the weights of each trajectory as in Eq. (21), and we are often interested in ensemble or average behavior because this is what is usually observed experimentally although single-molecule studies are by now a well-established and important field of study.[21,40,41]

It is critical to appreciate that an individual trajectory generally cannot be considered to be of equilibrium or non-equilibrium character in an intrinsic sense. (A possible exception is an extremely long trajectory which itself fully embodies all defining criteria of the ensemble.[4]) Generally, it is the *distribution* of trajectories that determines whether a system is in equilibrium and, if not, what ensemble it represents. Two finite-length trajectories that have the same weight in the equilibrium ensemble might have different weights in a non-equilibrium ensemble. The trajectory distribution will be determined by the initial phase-point distribution p_0 in conjunction with the imposed boundary and thermodynamic conditions such as temperature.

Let us consider equilibrium in the trajectory ensemble picture. For simplicity, we will assume that our initial phase point distribution is already Boltzmann-distributed: $p_0(\vec{x}_0) \propto \exp\left(-H(\vec{x}_0)/k_BT\right)$. As trajectories evolve in time from their initial points, the system will remain in equilibrium if the thermodynamic and boundary conditions remain the same. Thus, dynamics underlie equilibrium. We can say dynamics *define* equilibrium through detailed balance: if we count transitions occurring between small volumes around phase points $\vec{x}_i$ and $\vec{x}_j$ over any interval of time, the counts $i \to j$ and $j \to i$ will be identical within noise; the same is true for any size volumes in equilibrium.[1,4] This detailed balance property not only keeps the distribution stationary in time, but it means there are no net flows anywhere in phase space. Detailed balance further implies there is no net flow along any trajectory-like path—i.e., the forward and exactly time-reversed trajectories will occur an equal number of times.[23,42]

Note that our discussion here applies to thermal (constant-temperature) equilibrium for systems whose full configurations or phase points may include real-space coordinates and/or chemical degrees of freedom. That is, the trajectory picture of equilibrium applies for conformational processes in molecules, such as isomerization or folding; for simple diffusion or diffusion with possibly space-varying "drift" forces; for molecular binding, which may include both translational and conformational processes; for chemical processes involving electronic degrees of freedom such as bond formation and breakage; and for any combination of these whether modeled in full detail or approximately, so long as there is no implicit addition or removal of energy or particles. The trajectory picture does not apply for mechanical equilibrium, the balance of forces.

VI. POWERFUL SIMULATION METHODOLOGY BASED ON TRAJECTORY ENSEMBLES

A. Goals and challenges of computation

To consider computational strategies, we should first understand the goals of computation. As we do so, keep in mind a concrete process like protein folding or another spontaneous transition from a metastable state to a more stable one, such as a conformational change in a protein, a change in crystal lattice form, or a re-arrangement of a molecular cluster. For any of these transitions, we might be interested in the following:

(i) The "kinetics"—the MFPT or some other measure of rate for the transition.
(ii) The "mechanism" or pathways of the process—the sequence(s) of states exhibited during the transition.
(iii) The "relaxation" process—the timescales and mechanisms of describing the transient way the system "settles in" to a steady state.

We will first consider a simple, though typically impractical, way to calculate any or all of the above. As sketched in Fig. 3, the naive "brute force" implementation would simply be to initiate a large number of trajectories using an initial distribution of interest p_0 and wait until all trajectories have made the transition of interest. From this set of trajectories, we could (i) average their durations to obtain the MFPT or (ii) analyze the states occurring during transitions to quantify the mechanism.[43] For (iii) relaxation, we could wait still longer until the spatial/configurational distribution becomes stationary (using "recycling" if studying a constant-temperature NESS) and quantify the relaxation time as well as the mechanism, perhaps via probability shifts that occur. However, the strategy of waiting for multiple spontaneous transitions will only work for the simplest systems, such as low-dimensional toy models. (See Sec. VIII.)

In general, the brute force approach will *not* be practical for complex systems, and if a system is complicated and directly pertinent to real-world problems, it is likely to be too expensive to permit thorough brute-force simulation. We can quantify the challenges with a back-of-the-envelope calculation. For the system of interest, say you can afford a total of M simulations of duration $t_{\max}$. This means, roughly, that

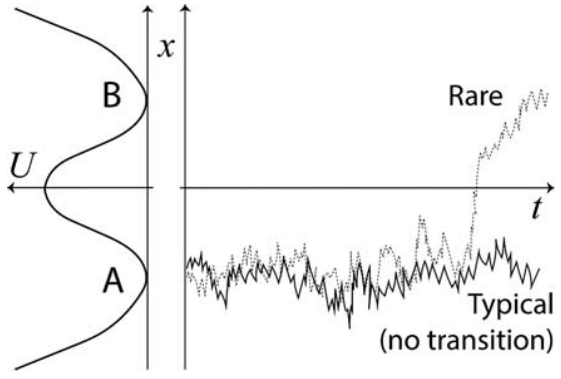


Fig. 3. The challenge of rare-event sampling in computation. Trajectories are initiated in state A, but in challenging systems most will remain in state A (solid trajectory). Transitions (dotted line) may be extremely unlikely or effectively unobservable in realistic, high-complexity systems such as protein conformational changes. Hence, typical "brute force" simulations can be both wasteful and expensive.

you can determine the distribution of phase points at any time $t < t_{max}$, denoted $p(\vec{x}, t)$, to a precision of $1/M$; typically, you will not have knowledge of behavior beyond t_{max}. As a point of reference in biomolecules, current hardware limits t_{max} to 1–10 μs in most systems (and to ms for small systems with extraordinary resources[9]), whereas most biological phenomena occur on a timescale of at least 100 μs and more typically on ms–s scales.

B. Efficient simulation via the weighted ensemble approach

Fortunately, there are now methods[31,44–47] that can sidestep the $1/M$ limitation just described, and we will focus on the most straightforward of these, known as the *weighted ensemble* (WE) strategy.[16,48,49] WE is a multi-trajectory "splitting method" based on a proposal credited to von Neumann[50] that can provide information on relaxation and steady-state behavior. WE can provide this information using less *overall* computing than naive simulation, i.e., the product Mt_{max} is smaller. It achieves this by re-allocating computing effort (trajectories) away from easy-to-sample regions of phase space toward rarer regions. WE is also an unbiased method: on average, it exactly recapitulates trajectory ensemble behavior and hence the time-evolution of the spatial distribution $p(\vec{x}, t)$;[49] the latter property reflects consistency with the Fokker–Planck equation,[35,36] which is briefly described in Appendix A.

WE simulation follows a fairly simple procedure, schematized in Fig. 4, which promotes the presence of trajectories in relatively rare regions of an energy landscape. In a basic implementation,[48] phase space is divided into non-overlapping bins of the user's construction, and a target number of trajectories per bin is set—say, 2, for concreteness. The bins should finely subdivide difficult-to-sample regions such as energy barriers to enable "statistical ratcheting" up hills if trajectories are examined frequently enough. That is, because short trajectories always have some probability to move uphill in energy, brief unbiased fluctuations can be "captured" for ratcheting and effectively concatenated to study otherwise rare events, sidestepping the $1/M$ limitation. Trajectories are started at the user's discretion; let us assume two trajectories are started in a bin of state A, with the goal of sampling transitions to B.

Trajectories in WE are run in parallel for brief intervals of time τ (with MFPT $\gg \tau \gg \Delta t$, where Δt is the simulation time step), then stopped and restarted according to simple probabilistic rules. In our example, each of the two trajectories is initially given a weight 1/2 at $t = 0$ and the essential idea is to ensure probability moves in an unbiased way, thus preserving the trajectory ensemble behavior and $p(\vec{x}, t)$. If a trajectory is found to occupy an otherwise empty bin after one of the τ intervals, *two* "child" trajectories are initiated from the final phase point of the "parent" trajectory, and each child inherits 1/2 of the parent's weight—a process called *splitting*. The two child trajectories in the previously unvisited bin create the ratcheting effect: there is twice the likelihood to explore that region, and to continue to still rarer regions, than if we did not replicate trajectories. Stochastic dynamics must be used; otherwise, child trajectories will evolve identically.

If more than two WE trajectories are found in a bin, pruning (or *merging*) is performed in a pairwise fashion: a random number is generated to select one of an arbitrary pair for continuation with probabilities proportional to their weights, and the selected trajectory absorbs the weight of the other trajectory, which is discontinued. In this fashion, energy minima do not collect large numbers of trajectories

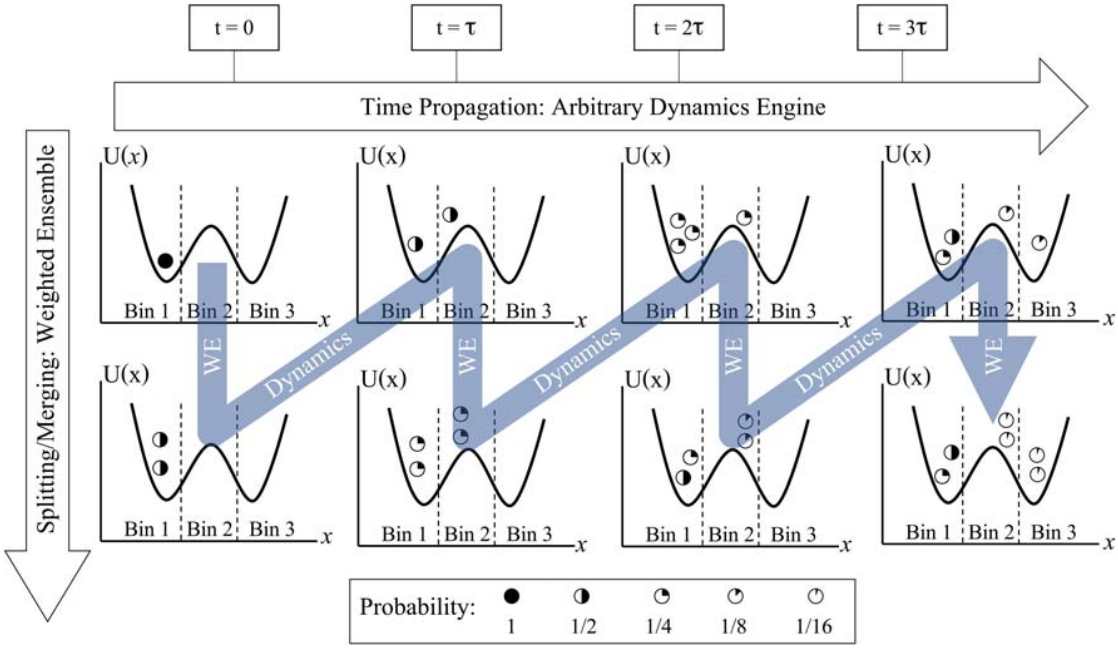


Fig. 4. Efficient simulation via the weighted ensemble (WE) method (Ref. 51). Phase space is divided into bins, and trajectories are started according to an initial distribution of interest (far left). Dynamics are run briefly, allowing trajectories to visit other bins, after which the WE steps of "splitting" (replication) and "merging" (pruning) are performed. Weights of parent trajectories are shared among children from splitting events, permitting the estimation of very low-probability events. In this example, a target of two trajectories per bin has been set. Reproduced with permission from Donovan *et al*, PLoS Comput. Biol. **12**(2), e1004611 (2016). Copyright 2016 Authors, licensed under a Creative Commons Attribution (CC BY) license.

which would add cost to the simulation but provide minimal statistical value. The processes described, in fact, constitute unbiased statistical *resampling*.[49] (See Sec. VIII.) In WE, the total trajectory cost is limited to the number of bins multiplied by the number of trajectories per bin and the trajectory length. This amounts to $M\,t_{\max}$ in our case, given $M/2$ bins.

Although the total simulation cost is bounded by $M\,t_{\max}$ (plus overhead for splitting/merging), events *much* rarer than $1/M$ can be seen because of the splitting procedure. Indeed, exponentially rare processes are elicited as WE produces an unbiased estimate of the trajectory ensemble and $p(\vec{x},t)$. A dramatic example is shown in Fig. 5 for diffusion and binding in a 3D box, where the distribution of possible binding outcomes extends *tens of orders of magnitude below what standard simulation provides*. For monitoring the transient time evolution of a system, WE is almost like a "magic bullet."

Obtaining the MFPT from WE simulation is more challenging than characterizing $p(\vec{x},t)$ in many cases. To use the Hill relation Eq. (19), the system must relax to steady state and this relaxation is *not* accelerated by WE for the very reason it is so successful in characterizing $p(\vec{x},t)$; i.e., because it is unbiased. To see this more concretely, let t_{SS} be the average time required for a given system to relax to steady state. Then, because WE runs M copies of the system, the total cost for observing a WE simulation relax to steady state is $\sim M\,t_{SS}$, which will be prohibitive in some though not all systems.[52] Even when $M\,t_{SS}$ is a prohibitive cost, the MFPT can be obtained from transient data ($t < t_{SS}$) available in WE simulation: although the details are beyond the scope of this discussion, the idea is to use much finer-grained and faster-relaxing bins (than were used to run the WE simulation) in a quasi-Markov approximation.[53] Below, we apply WE directly for MFPT calculation in a simple system.

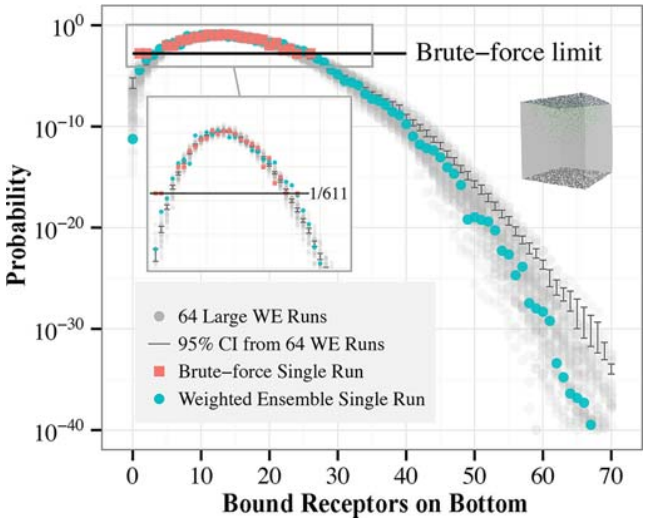


Fig. 5. Weighted ensemble simulation of extremely rare diffusion and binding events (Ref. 51). Particles are initiated at the top of a three-dimensional box (upper right inset) and allowed to diffuse without bias. Any particles that reach the bottom surface of the box can bind to receptors located there. The graph shows the probability distribution of bound receptors after a short time interval—i.e., the likelihood of different outcomes that would result from a single brute-force diffusion simulation. WE enables sampling deep into the tails of the distribution because more trajectories are allotted to rarer outcomes, whereas an equivalent amount of "brute force" sampling cannot detect events rarer than the reciprocal of the number of trajectories, as shown by solid horizontal lines. WE simulations used simulation time equivalent to 611 brute force trajectories, as indicated in the left inset. The grey dots represent independent WE runs (of which green (online) is a representative) and solid vertical bars give the confidence interval based on the grey data—which appears to be skewed upward because of the logarithmic scale. Reproduced with permission from Donovan *et al.*, PLoS Comput. Biol. **12**(2), e1004611 (2016). Copyright 2016 Authors, licensed under a Creative Commons Attribution (CC BY) license.

Like any advanced computational method, WE has its subtleties and limitations. Most important are correlations. Although WE trajectories are independent (non-interacting), exactly as assumed in the trajectory-ensemble definitions, correlations arise in the overall WE protocol due to the splitting and merging steps. After all, when a trajectory is "split," by construction the child trajectories are identical until the split point. Therefore, assessing statistical uncertainty in WE estimates requires great care, even though the method is unbiased.[54]

C. Applying the weighted ensemble to a simple model

To illustrate the power and validity of the weighted ensemble method, we employ it to estimate the transition rate over a high energy barrier in a simple system. We use the WESTPA implementation[55] of WE and apply it to a simple 1D double-well potential under overdamped Langevin dynamics (Eq. (3)) with parameters chosen to approximate the behavior of a small molecule in water. We assume a mass of 100 u, temperature $T = 300$ K, a barrier height of $10k_BT$, and a friction coefficient $\gamma = 24.94\,\text{ps}^{-1}$ which is reasonable for water and corresponds to a diffusion constant of $10^{-6}\,\text{cm}^2/\text{s}$. The simulation is run with a timestep of 3 ps, and all simulation code is available on Github.[56]

The WE simulation is set up with walkers beginning in the rightmost basin, and with the two basin macrostates defined as $x > 20$ nm and $x < -20$ nm, as shown in Fig. 6. Twenty uniform bins of width 2 nm uniformly span from $x = -20.0$ to 20.0 nm, with two additional bins on either end reaching to $\pm\infty$. The WE simulation is run with a resampling time of $\tau = 60$ ps and a target count of ten trajectories per bin, so roughly 200 trajectories will run during each τ. Walkers that reach the left basin are "recycled" and restarted from $x = 20$ nm to generate a non-equilibrium steady state and exploit the Hill relation Eq. (19).

To quantify the effectiveness of WE simulation for this case, we can compare the cost for computing the rate constant (i.e., flux or reciprocal MFPT) from WE simulation to brute-force simulation of overdamped Langevin dynamics. Note from Fig. 6 that WE simulations reach steady values after $\sim$3000 iterations, which corresponds to $\sim 12 \times 10^6$ steps of total simulation (for a single WE run, accounting for all $\sim$200 trajectories) or a total of 36 μs of simulated time. Also from Fig. 6 and Eq. (19), the MFPT is $\sim$1000 μs. Thus, we see that WE simulation has generated the *average* first-passage time using an overall amount of computing that is only a small fraction ($\sim$0.04) of the time needed to yield a *single* transition event via direct simulation, let alone to generate a reliable MFPT estimate from multiple events.

VII. CONCLUDING DISCUSSION

The trajectory arguably is the most fundamental object in classical statistical mechanics, particularly for non-equilibrium phenomena, and this article has attempted to connect trajectory physics with more familiar topics in the traditional physics curriculum. By focusing in depth on the simplest possible example—diffusion—we have been able to formalize and visualize the probabilistic/ensemble picture

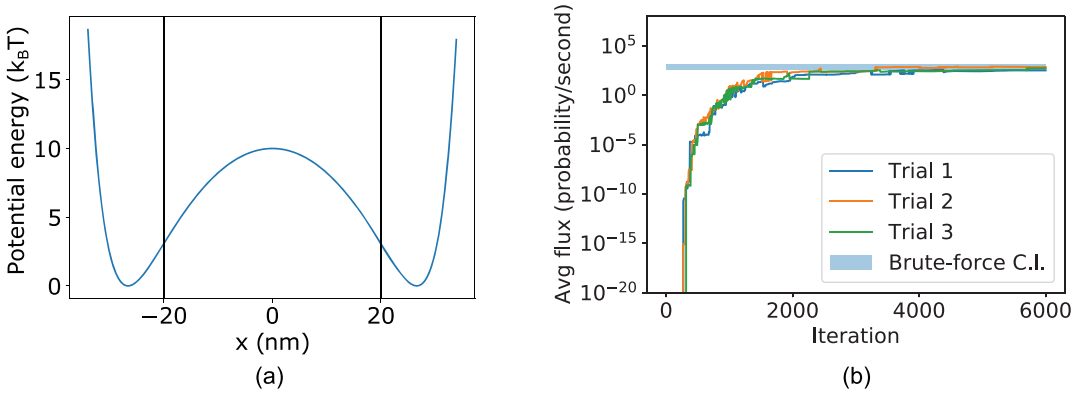


Fig. 6. Weighted ensemble estimation of the rate of a rare event: high-barrier crossing. (a) potential energy function used for double-well simulation with 10 k_BT barrier and state boundaries indicated by the vertical black lines. (b) average flux into the left basin state for simulations started from the right basin, as computed from three independent weighted ensemble simulations (colored lines). The average flux estimates the inverse MFPT by Eq. (19), yielding ∼1 ms. For reference, an independent estimate of the flux is computed using a very long "brute force" simulation (horizontal line). The brute force confidence interval (C.I.) is shown as a blue shaded region, which is ± twice the standard error of the mean based on 11 transitions.

and connect it with simpler spatial distributions. We have further been able to connect these ensembles with observable populations, kinetics, and thermodynamic states, as well as understand a modern, practical path-sampling approach.

A key lesson is that theoretical physics can view a given process at different levels of "magnification," from most microscopic to most averaged (Fig. 7). Trajectories are the most detailed and encompass all system coordinates at all times—which is usually too much to grasp. Trajectories can be averaged spatially at fixed times to yield more familiar probability distributions. Trajectory flows across surfaces of interest can also be averaged to yield probability fluxes: in equilibrium, all such fluxes are zero, whereas in transient regimes or non-equilibrium steady states (NESS's), such flows provide key information. Notably, the Hill relation (Eq. (19)) yields the mean first-passage time (MFPT) from the flux in an appropriate NESS, and furthermore, conditions on reversibility can be derived from flux arguments (Appendix B). Finally, averaging—i.e., integrating—over spatial distributions can yield observable thermodynamic information on state populations;[6,16] see also entropy production and fluctuation relations.[7,42,57,58]

This report has only given a taste of the value of the trajectory picture, which goes much further. Trajectory ideas, for example, are used to develop the Jarzynski relation.[58–60] They provide a direct connection with the path-integral formulation of quantum mechanics.[34] Trajectories offer a unique window into the often misunderstood issue of "reversibility."[61] (See Appendix B.) Not surprisingly, trajectories and their applications are still an area of active research.[7,52,62–65]

VIII. EXERCISES

(1) Confirm by differentiation that Eq. (15) is the exact solution to the diffusion equation Eq. (17), after setting $x_n = x$ and $n = t/\Delta t$. Note that t occurs both in the prefactor *and* the exponent, so differentiation requires the product rule.
(2) Time-discretization generally introduces an error into dynamics computed via Eqs. (4) and (6). Explain why there is an error and how it might be mitigated in computer simulation. For what special case is there no error even if $f \neq 0$?
(3) Implement overdamped dynamics simulation Eq. (4) of the double-well system specified in Sec. VI C. Calculate the MFPT of the system for a range of barrier

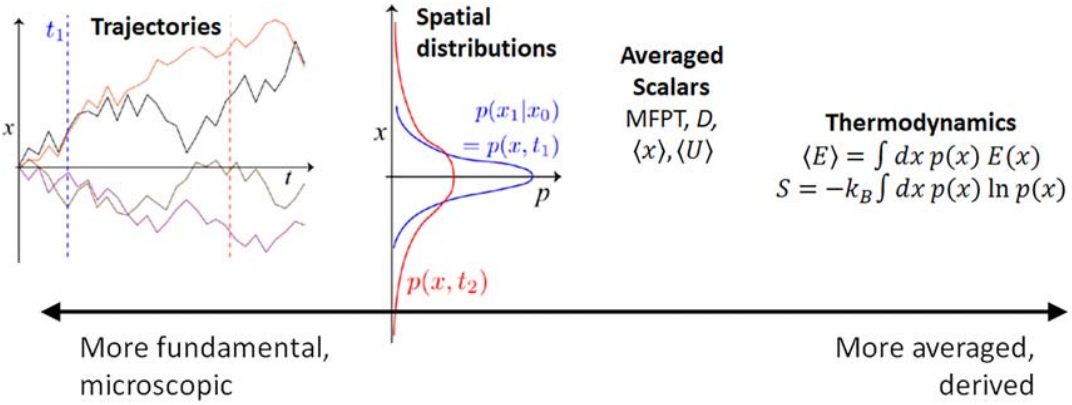


Fig. 7. From the fundamental ensemble of trajectories to more averaged observables. Because trajectories embody the dynamics that fully specifies a system, they are the most fundamental. Averaging or analysis can be performed at fixed time points, including the $t \to \infty$ stationary point. Quantities that can be calculated include the phase-space distribution $p(x, t)$, the mean first-passage time (MFPT), diffusion constant (D), average coordinates or properties (e.g., $\langle x\rangle, \langle U\rangle$), or system-wide thermodynamic properties, in or out of equilibrium. Although simple diffusive trajectories are pictured, the same principles apply in the case of non-zero forces.

heights, starting with a low barrier, by simple averaging of ∼10 observed first-passage times. Compare these values to the expected Arrhenius behavior.[4]

(4) Using the ODLD module of the WESTPA implementation of weighted ensemble, implement a *triple* well system. Consider the left-most basin to be the initial state (A) and the right-most basin the target (B). Examine the relaxation of the probability into the target state as a function of time. For cleanest data, average over multiple WE runs. Vary the depth of the middle well and explain the observed behavior.
(5) Write down the trajectory probability, the analog of Eq. (12), for a system with constant force, sometimes called simple drift. Explain in words the meaning of the distribution. If you can, integrate out intermediate time points to show that the behavior remains Gaussian with constant drift.
(6) For a simple diffusive system described by Eq. (12), obtain the distribution for x_3 by a suitable integration of Eq. (14).
(7) Write down the equations that define (i) a steady state and (ii) equilibrium for a discrete-state system in terms of steady probabilities p_i and state-to-state transition probabilities $T_{i\to j}$ for some fixed time interval. Note that equilibrium is defined by detailed balance. Show that detailed balance implies steady state but not the reverse. A counter-example suffices to disprove a hypothesis.
(8) By studying the theory underlying weighted ensemble,[49] explain in statistical terms why the "resampling" procedure for "merging" trajectories does not bias the time-evolving probability distribution $p(\vec{x}, t)$.
(9) Write pseudocode for a weighted ensemble simulation of an arbitrary system with pre-defined bins. If you are ambitious, implement your pseudocode for 1D overdamped dynamics in the double-well system in Exercise 3.
(10) Understand the continuity equation Eq. (A1) by integrating it over an interval in x from a to b. Integrating the probability density over this region gives the total probability in it. How does this probability change in time, based on the current, and why does the result make sense? Remember the one-dimensional current is defined to be positive in the right-ward direction.
(11) Show that stationary distribution of the Smoluchowski equation Eq. (A3), i.e., when $\partial p/\partial t = 0$, is the expected equilibrium distribution based on the Boltzmann factor.

ACKNOWLEDGMENTS

The authors are grateful for support from the National Science Foundation under Grant No. MCB 1715823 and from the National Institutes of Health under Grant No. GM115805. The authors very much appreciate helpful discussions with Jeremy Copperman and Ernesto Suarez.

APPENDIX A: THE FOKKER–PLANCK PICTURE AND SMOLUCHOWSKI EQUATION IN ONE DIMENSION

The Fokker–Planck and related equations[35,36] are essential for understanding non-equilibrium statistical mechanics. These equations generalize the diffusion equation (17), but they perform essentially the same role: they quantify the way a spatial and/or configurational distribution changes over time based on a given energy landscape. The key point is that this is a very general concept that applies not only to center-of-mass diffusive motion but also to configurational motions internal to a molecule or system. For example, if a protein is started in a certain configuration, where is it likely to be later? The distribution $p(\vec{x}, t)$ quantifies the distribution of configurations $\vec{x}$ at any time t.

Here, we focus on the Smoluchowski equation, which is the Fokker–Planck equation specific for the overdamped, non-inertial dynamics Eq. (3) studied above. The Smoluchowski equation is easiest to grasp starting from the continuity equation, given by

$$\frac{\partial p}{\partial t} = -\frac{\partial J}{\partial x}, \tag{A1}$$

in one dimension, where $p = p(x, t)$ is the probability density at time t and $J = J(x, t)$ is the probability current, i.e., the (average) probability per unit time moving in the $+x$ direction. Note that this is the average over trajectories moving in both directions, so it is the *net* current. The continuity equation simply ensures that the change of probability in any region is the difference between incoming and outgoing probability. For students who are new to the continuity equation, Exercise 10 will clarify its meaning.

To complete the Smoluchowski equation, we need the current corresponding to overdamped dynamics (Eq. (3)). As noted above, overdamped dynamics includes both (simple) diffusion and "drift" (motion due to force). From Eq. (17), we can already infer that the diffusive current is $-D\,\partial p/\partial x$, which is Fick's law indicating that particles/probability will diffuse down their gradients in a linear fashion on average. When a force is present, Eq. (3) indicates that there is also motion linearly proportional to the force, leading to a total current

$$J(x, t) = -D\frac{\partial p}{\partial x} + \frac{D}{k_B T} f(x)\ p(x, t), \tag{A2}$$

where we have assumed $D = k_B T/m\gamma$ is constant in space.

We obtain the full Smoluchowski equation in one-dimension for fixed D by substituting the current Eq. (A2) into the continuity equation Eq. (A1), yielding

$$\frac{\partial p}{\partial t} = D\frac{\partial^2 p}{\partial x^2} - \frac{D}{k_B T}\frac{\partial}{\partial x} f\ p. \tag{A3}$$

The diffusion equation has been augmented by a term dependent on the force. Equation (A3) can be solved to find the steady-state behavior of p both out of or in equilibrium (see Sec. VIII) or to follow the time-dependent behavior as the distribution p relaxes toward its limiting steady profile.

APPENDIX B: ADVANCED TOPIC: MACROSCOPIC REVERSIBILITY BY DECOMPOSING THE EQUILIBRIUM TRAJECTORY ENSEMBLE

Many of us are aware of the intrinsic time reversibility of Newtonian mechanics, whereby any constant-energy trajectory $\vec{x}(t)$ can be "played backwards" to yield another

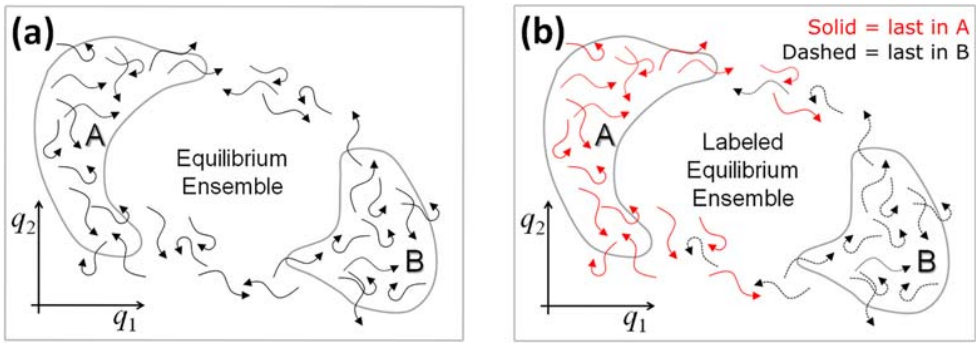


Fig. 8. An exact decomposition of the equilibrium trajectory ensemble. (a) The equilibrium ensemble, consisting of a large number of *independent* trajectories projected onto the schematic coordinates q_1 and q_2. Transitions between macrostates A and B (gray outlines) occur via two pathways, upper and lower. (b) Decomposition of the equilibrium ensemble based on which of two macrostates, A and B, has been visited most recently. These two directional ensembles, "last in A" or "A-to-B" (red, solid lines) and "last in B" or "B-to-A" (black, dashed lines), are non-equilibrium steady states. Arrow tips represent the same time point for all trajectories and arrow tails represent the most recent history, but all history is assumed to be known. Reprinted with permission from Bhatt and Zuckerman, J. Chem. Theory Comput. **7**(8), 2520–2527 (2011). Copyright 2011 American Chemical Society (Ref. 61).

physically valid trajectory. There is an analogous condition on a stochastic trajectory, which can be derived from detailed balance.[60] However, the conditions for reversibility under more realistic circumstances involving a *distribution* of initial and final configurations require the trajectory ensemble picture.[61]

We start by considering an equilibrium ensemble of trajectories: see Fig. 8(a). The equilibrium trajectory ensemble is defined by a set of completely independent systems/trajectories for times $t > t_0$, given that at t_0, the set of phase-space points $\vec{x}(t_0)$ is equilibrium-distributed, i.e., according to the Boltzmann factor. (We don't need to worry about how equilibrium was produced.) If the phase points are equilibrium-distributed at time t_0, they will remain equilibrium-distributed thereafter. This is because the Markovian stochastic dynamics that generates equilibrium also maintains it, which is why we call it equilibrium in the first place.[4]

As sketched in Fig. 8, the equilibrium ensemble at any time t can be *exactly* decomposed into two parts based on a history-labeling process.[46,61] Specifically, based on two arbitrary non-overlapping macrostates A and B, each trajectory can be assigned to the A-to-B set—a.k.a "last-in-A" set—if it currently occupies state A or was more recently in A than B, with the remaining trajectories in the B-to-A direction. This construction requires "omniscience," in the sense of knowing the full history of each trajectory, so it is something of a thought experiment. Note that each of these directional trajectory subsets is automatically maintained as a *non-equilibrium* steady state: when an A-to-B trajectory enters B, its label switches to B-to-A, but the overall equilibrium condition ensures that equal numbers of trajectories will switch labels per unit time.[61]

We are now in a position to understand reversibility, building on the defining process of equilibrium: detailed balance.[4] As a reminder, detailed balance implies there is zero net flow between *any* pair of "microstates," i.e., small phase-space volumes. In the context of the two uni-directional steady states (A-to-B and B-to-A), detailed balance gives us a tool to consider two non-overlapping mechanistic "pathways"—arbitrary tubes of phase points connecting A and B—e.g., upper and lower pathways in Fig. 8. If we place a (hyper-)surface transecting each tube, then there is a certain probability flowing per second through each surface in, say, the A-to-B steady state; call these σ_1 and σ_2. By detailed balance, there is no net flow through either surface in equilibrium, and so the flows in the B-to-A state must be equal and opposite. Mechanistically, the ratio σ_1/σ_2 is the same in both directions: the fraction of events taking each pathway must be the same in both directions. This is mechanistic reversibility. Fuller details and illustrations can be found in earlier work.[61]

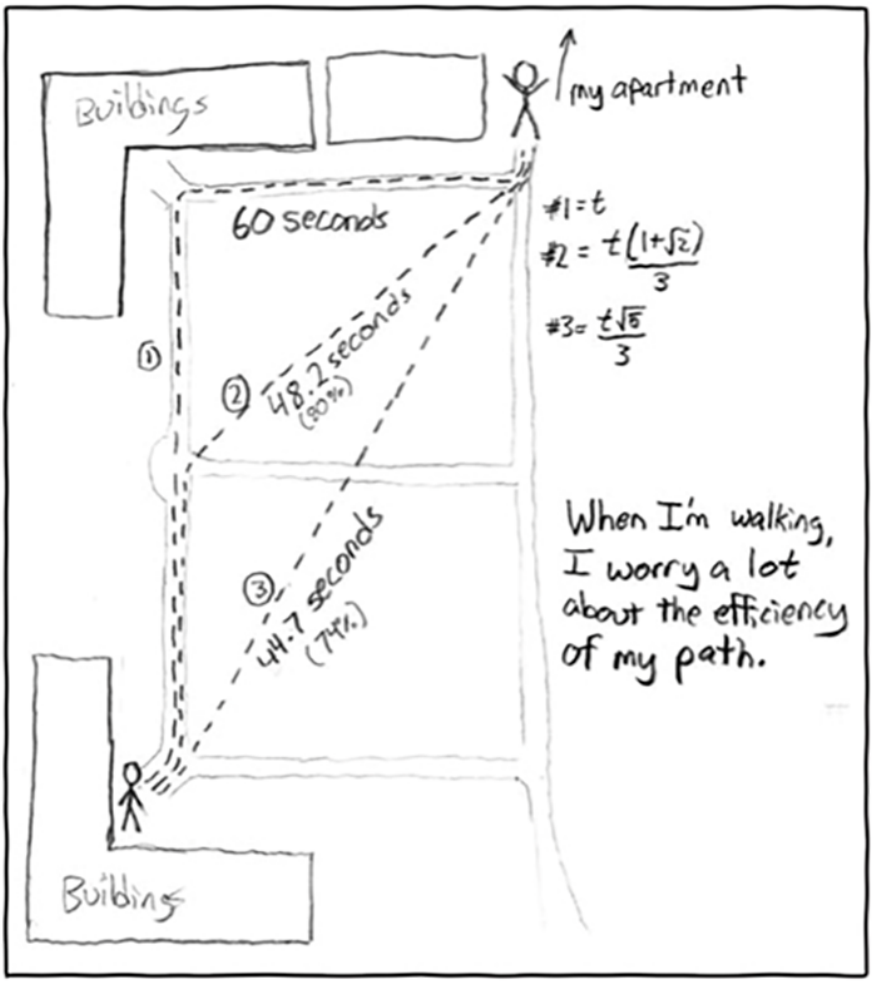
A key point is that the preceding discussion is strictly based on the detailed-balance property of equilibrium. Thus, systems out of equilibrium should *not* be expected to exhibit mechanistic reversibility. This is true experimentally and theoretically. Examples of systems not obeying reversibility would be if A and B states were prepared under different conditions (e.g., temperature, pH,…) or, even under the same conditions, if the initial distribution in A or B did not mimic the process for constructing the directional steady states derived from equilibrium. Specifically, in the A-to-B direction, trajectories should be initiated on the surface of A according to the distribution with which they would arrive from B in equilibrium, which is known as the "EqSurf" construction.[61] To put this informally, state A needs to be "tricked" into behaving as it would in equilibrium, so trajectories are started at the boundary of A as if they had arrived from B (i.e., were last in B) in equilibrium.

[a]Electronic mail: zuckermd@ohsu.edu, ORCID: 0000-0001-7662-2031.

[1]Lars Onsager, "Reciprocal relations in irreversible processes. I," Phys. Rev. **37**(4), 1505–1512 (1931).
[2]Eric Vanden-Eijnden *et al.*, "Transition-path theory and path-finding algorithms for the study of rare events," Annu. Rev. Phys. Chem. **61**, 391–420 (2010).
[3]Ron Elber, Dmitrii E. Makarov, and Henri Orland, *Molecular Kinetics in Condensed Phases: Theory, Simulation, and Analysis* (John Wiley & Sons, New York, 2020).
[4]Daniel M. Zuckerman, *Statistical Physics of Biomolecules: An Introduction* (CRC, Boca Raton, FL, 2010).
[5]Douglas C. Giancoli, *Physics for Scientists and Engineers with Modern Physics* (Pearson Education, London, 2008).
[6]Frederick Reif, *Fundamentals of Statistical and Thermal Physics* (McGraw-Hill, New York, 1965).
[7]Udo Seifert, "Stochastic thermodynamics, fluctuation theorems and molecular machines," Rep. Prog. Phys. **75**(12), 126001 (2012).
[8]Daniel M. Zuckerman, "Key biology you should have learned in physics class: Using ideal-gas mixtures to understand biomolecular machines," Am. J. Phys. **88**(3), 182–193 (2020).
[9]Kresten Lindorff-Larsen, Stefano Piana, Ron O. Dror, and David E. Shaw, "How fast-folding proteins fold," Science **334**(6055), 517–520 (2011).

[10]Alan Fersht *et al.*, *Structure and Mechanism in Protein Science: A Guide to Enzyme Catalysis and Protein Folding* (Macmillan, New York, 1999).
[11]J. Juraszek and P. G. Bolhuis, "Sampling the multiple folding mechanisms of Trp-cage in explicit solvent," Proc. Natl. Acad. Sci. **103**(43), 15859–15864 (2006).
[12]David L. Mobley and Michael K. Gilson, "Predicting binding free energies: Frontiers and benchmarks," Annu. Rev. Biophys. **46**, 531–558 (2017).
[13]K. Gunasekaran, Buyong Ma, and Ruth Nussinov, "Is allostery an intrinsic property of all dynamic proteins?," Proteins **57**(3), 433–443 (2004).
[14]Matthew C. Zwier, Adam J. Pratt, Joshua L. Adelman, Joseph W. Kaus, Daniel M. Zuckerman, and Lillian T. Chong, "Efficient atomistic simulation of pathways and calculation of rate constants for a protein–peptide binding process: Application to the MDM2 protein and an intrinsically disordered p53 peptide," J. Phys. Chem. Lett. **7**(17), 3440–3445 (2016).
[15]Robert A. Copeland, David L. Pompliano, and Thomas D. Meek, "Drug–target residence time and its implications for lead optimization," Nat. Rev. Drug Discovery **5**(9), 730–739 (2006).
[16]Daniel M. Zuckerman and Lillian T. Chong, "Weighted ensemble simulation: Review of methodology, applications, and software," Annu. Rev. Biophys. **46**, 43–57 (2017).
[17]Lawrence R. Pratt, "A statistical method for identifying transition states in high dimensional problems," J. Chem. Phys. **85**(9), 5045–5048 (1986).
[18]Christoph Dellago, Peter G. Bolhuis, Félix S. Csajka, and David Chandler, "Transition path sampling and the calculation of rate constants," J. Chem. Phys. **108**(5), 1964–1977 (1998).
[19]Daniel M. Zuckerman and Thomas B. Woolf, "Dynamic reaction paths and rates through importance-sampled stochastic dynamics," J. Chem. Phys. **111**(21), 9475–9484 (1999).
[20]Martin Bier, Imre Derényi, Marcin Kostur, and R. Dean Astumian, "Intrawell relaxation of overdamped Brownian particles," Phys. Rev. E **59**(6), 6422–6432 (1999).
[21]R. Dean Astumian, "The unreasonable effectiveness of equilibrium theory for interpreting nonequilibrium experiments," Am. J. Phys. **74**(8), 683–688 (2006).
[22]R. Dean Astumian, "Thermodynamics and kinetics of molecular motors," Biophys. J. **98**(11), 2401–2409 (2010).
[23]R. Dean Astumian, Cristian Pezzato, Yuanning Feng, Yunyan Qiu, Paul R. McGonigal, Chuyang Cheng, and J. Fraser Stoddart, "Non-equilibrium kinetics and trajectory thermodynamics of synthetic molecular pumps," Mater. Chem. Front. **4**(5), 1304–1314 (2020).
[24]Terrell L. Hill, "The linear Onsager coefficients for biochemical kinetic diagrams as equilibrium one-way cycle fluxes," Nature **299**(5878), 84–86 (1982).
[25]Terrell L. Hill, *Free Energy Transduction and Biochemical Cycle Kinetics* (Dover, Mineola, NY, 2004).
[26]Kingshuk Ghosh, Ken A. Dill, Mandar M. Inamdar, Effrosyni Seitaridou, and Rob Phillips, "Teaching the principles of statistical dynamics," Am. J. Phys. **74**(2), 123–133 (2006).
[27]Steve Pressé, Kingshuk Ghosh, Julian Lee, and Ken A. Dill, "Principles of maximum entropy and maximum caliber in statistical physics," Rev. Mod. Phys. **85**(3), 1115–1141 (2013).
[28]Kingshuk Ghosh, Purushottam D. Dixit, Luca Agozzino, and Ken A. Dill, "The maximum caliber variational principle for nonequilibria," Annu. Rev. Phys. Chem. **71**, 213–238 (2020).
[29]Robert H. Swendsen, "Explaining irreversibility," Am. J. Phys. **76**(7), 643–648 (2008).
[30]Subrahmanyan Chandrasekhar, "Stochastic problems in physics and astronomy," Rev. Mod. Phys. **15**(1), 1–89 (1943).
[31]Lillian T. Chong, Ali S. Saglam, and Daniel M. Zuckerman, "Path-sampling strategies for simulating rare events in biomolecular systems," Curr. Opin. Struct. Biol. **43**, 88–94 (2017).
[32]Nicolaas Godfried Van Kampen, *Stochastic Processes in Physics and Chemistry* (Elsevier, New York, 1992), Vol. 1.
[33]Lars Onsager and Stefan Machlup, "Fluctuations and irreversible processes," Phys. Rev. **91**(6), 1505–1512 (1953).
[34]J. J. Sakurai, *Modern Quantum Mechanics* (Addison Wesley, Reading, MA, 1985).
[35]Crispin Gardiner, *Stochastic Methods* (Springer-Verlag, Berlin, 2009), Vol. 4.
[36]Hannes Risken and Till Frank, *The Fokker-Planck Equation: Methods of Solution and Applications* (Springer Science & Business Media, New York, 1996), Vol. 18.
[37]Sidney Redner, *A Guide to First-Passage Processes* (Cambridge U. P., Cambridge, 2001).
[38]David Chandler, *Introduction to Modern Statistical Mechanics* (Oxford U. P., Oxford, UK, 1987).
[39]John D. Chodera, Phillip J. Elms, William C. Swope, Jan-Hendrik Prinz, Susan Marqusee, Carlos Bustamante, Frank Noé, and Vijay S. Pande, "A robust approach to estimating rates from time-correlation functions," arXiv:1108.2304 (2011).
[40]Ashok A. Deniz, Samrat Mukhopadhyay, and Edward A. Lemke, "Single-molecule biophysics: At the interface of biology, physics and chemistry," J. R. Soc. Interface **5**(18), 15–45 (2008).
[41]Helen Miller, Zhaokun Zhou, Jack Shepherd, Adam J. M. Wollman, and Mark C. Leake, "Single-molecule techniques in biophysics: A review of the progress in methods and applications," Rep. Prog. Phys. **81**(2), 024601 (2017).
[42]Gavin E. Crooks, "Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences," Phys. Rev. E **60**(3), 2721–2726 (1999).
[43]Ernesto Suárez and Daniel M. Zuckerman, "Pathway histogram analysis of trajectories: A general strategy for quantification of molecular mechanisms," arXiv:1810.10514 (2018).
[44]Rosalind J. Allen, Patrick B. Warren, and Pieter Rein Ten Wolde, "Sampling rare switching events in biochemical networks," Phys. Rev. Lett. **94**(1), 018104 (2005).
[45]Anton K. Faradjian and Ron Elber, "Computing time scales from reaction coordinates by milestoning," J. Chem. Phys. **120**(23), 10880–10889 (2004).
[46]Titus S. van Erp, Daniele Moroni, and Peter G. Bolhuis, "A novel path sampling method for the calculation of rate constants," J. Chem. Phys. **118**(17), 7762–7774 (2003).
[47]Aryeh Warmflash, Prabhakar Bhimalapuram, and Aaron R. Dinner, "Umbrella sampling for nonequilibrium processes," J. Chem. Phys. **127**(15), 114109 (2007).
[48]Gary A. Huber and Sangtae Kim, "Weighted-ensemble Brownian dynamics simulations for protein association reactions," Biophys. J. **70**(1), 97–110 (1996).
[49]Bin W. Zhang, David Jasnow, and Daniel M. Zuckerman, "The "weighted ensemble" path sampling method is statistically exact for a broad class of stochastic processes and binning procedures," J. Chem. Phys. **132**(5), 054107 (2010).
[50]Herman Kahn and Theodore E. Harris, "Estimation of particle transmission by random sampling," Natl. Bur. Stand. Appl. Math. Ser. **12**, 27–30 (1951).
[51]Rory M. Donovan, Jose-Juan Tapia, Devin P. Sullivan, James R. Faeder, Robert F. Murphy, Markus Dittrich, and Daniel M. Zuckerman, "Unbiased rare event sampling in spatial stochastic systems biology models using a weighted ensemble of trajectories," PLoS Comput. Biol. **12**(2), e1004611 (2016).
[52]Upendra Adhikari, Barmak Mostofian, Jeremy Copperman, Sundar Raman Subramanian, Andrew A. Petersen, and Daniel M. Zuckerman, "Computational estimation of microsecond to second atomistic folding times," J. Am. Chem. Soc. **141**(16), 6519–6526 (2019).
[53]Jeremy Copperman and Daniel M. Zuckerman, "Accelerated estimation of long-timescale kinetics from weighted ensemble simulation via non-Markovian 'microbin" analysis,' J. Chem. Theory Comput. **16**(11), 6763–6775 (2020).
[54]Barmak Mostofian and Daniel M. Zuckerman, "Statistical uncertainty analysis for small-sample, high log-variance data: Cautions for bootstrapping and Bayesian bootstrapping," J. Chem. Theory Comput. **15**(6), 3499–3509 (2019).
[55]Matthew C. Zwier, Joshua L. Adelman, Joseph W. Kaus, Adam J. Pratt, Kim F. Wong, Nicholas B. Rego, Ernesto Suárez, Steven Lettieri, David W. Wang, Michael Grabe, Daniel M. Zuckerman, and Lillian T. Chong, "WESTPA: An interoperable, highly scalable software package for weighted ensemble simulation and analysis," J. Chem. Theory Comput. **11**(2), 800–809 (2015).
[56]John Russo (2021). "jdrusso/doublewell: More physical parameters (version 1.1)," Zenodo. https://doi.org/10.5281/zenodo.4706088
[57]G. N. Bochkov and Yu. E. Kuzovlev, "Nonlinear fluctuation-dissipation relations and stochastic models in nonequilibrium thermodynamics: I. Generalized fluctuation-dissipation theorem," Physica A **106**(3), 443–479 (1981).
[58]Christopher Jarzynski, "Nonequilibrium equality for free energy differences," Phys. Rev. Lett. **78**(14), 2690–2693 (1997).
[59]C. Jarzynski, "Equilibrium free energies from nonequilibrium processes," Acta Phys. Pol., Ser. B **29**(6), 1609–1622 (1998).

[60]Gavin E. Crooks, "Nonequilibrium measurements of free energy differences for microscopically reversible Markovian systems," J. Stat. Phys. **90**(5–6), 1481–1487 (1998).
[61]Divesh Bhatt and Daniel M. Zuckerman, "Beyond microscopic reversibility: Are observable nonequilibrium processes precisely reversible?," J. Chem. Theory Comput. **7**(8), 2520–2527 (2011).
[62]Erik H. Thiede, Dimitrios Giannakis, Aaron R. Dinner, and Jonathan Weare, "Galerkin approximation of dynamical quantities using trajectory data," J. Chem. Phys. **150**(24), 244111 (2019).
[63]David Aristoff and Daniel M. Zuckerman, "Optimizing weighted ensemble sampling of steady states," Multiscale Model. Simul. **18**(2), 646–673 (2020).
[64]Grant M. Rotskoff and Eric Vanden-Eijnden, "Dynamical computation of the density of states and Bayes factors using nonequilibrium importance sampling," Phys. Rev. Lett. **122**(15), 150602 (2019).
[65]John D. Russo, Jeremy Copperman, and Daniel M. Zuckerman, "Iterative trajectory reweighting for estimation of equilibrium and non-equilibrium observables," arXiv:2006.09451 (2020).



Paths

It's true, I think about this all the time. (Source: https://xkcd.com/85/)