SHOOTING FOR THE MOON WITH WEIGHTED ENSEMBLE APPLICATIONS

by

Anthony T. Bogetti

B.S., Messiah University, 2017

Submitted to the Graduate Faculty of

the Deitrich School of Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2023

UNIVERSITY OF PITTSBURGH

DEITRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Anthony T. Bogetti

It was defended on

June 29th 2023

and approved by

Lillian T. Chong, Ph. D., Professor of Chemistry

Rob A. Coalson, Ph. D., Professor of Chemistry and Professor of Physics

Geoffrey Hutchison, Ph. D., Associate Professor of Chemistry

Daniel M. Zuckerman, Ph. D., Professor of Biomedical Engineering

Dissertation Director: Lillian T. Chong, Ph. D., Professor of Chemistry

SHOOTING FOR THE MOON WITH WEIGHTED ENSEMBLE APPLICATIONS

Anthony T. Bogetti, PhD

University of Pittsburgh, 2023

Rare events, which are infrequent, but relatively fast once they occur, are ubiquitous in biology. Conventional molecular dynamics (cMD) simulations can only sample rare events up to the microseconds timescale on typical resources because they spend most of their computing time sampling stable states. Path sampling strategies, which exploit the separation of timescales inherent in rare events, focus computing power on the transitions between stable states and are orders of magnitude more efficient than cMD. The weighted ensemble (WE) strategy is a particularly promising path sampling strategy that has not yet reached its full potential. In this dissertation, I describe various advances to the WE strategy that I have developed and demonstrate how those advances have allowed us to "shoot for the moon" by simulating larger systems on longer timescales. In Chapter 1 of this dissertation, I motivate the need for path sampling and discuss the features of WE that set it apart from other path sampling strategies. In Chapter 2, I describe various protein conformational switches and how path sampling strategies can rationally enhance switching kinetics by focusing sampling on the transient states of switches. This chapter highlights WE simulations of the SARS-CoV-2 protein, showcasing the ability of WE to generate complete pathways for systems up to a million atoms and processes as slow as the seconds timescale. In Chapter 3, I present advances to the open-source WESTPA software package (version 2.0) that were motivated by a recent SARS-CoV-2 "stress test." In Chapter 4, I introduce a minimal, adaptive binning (MAB) scheme for WE simulations and showcase the MAB scheme using three systems of varying model resolution. In Chapter 5, I describe LPATH, a general, semi-automated tool for performing bottom-up clustering of simulated pathways into distinct classes using a text-string pattern matching algorithm commonly used in plagiarism detection. Together, the above chapters demonstrate the potential of WE path sampling to "shoot for the moon", tackling larger systems and/or longer timescales.

Table of Contents

List of Tables

List of Figures

Preface

I would like to thank Dr. Chong for her guidance and encouragement and for giving me the opportunity to learn in her lab for the past few years. I would also like to thank my committee for their time and feedback on ideas related to my thesis project, and the Chong lab for helpful discussions. I would especially like to thank Jeremy Leung and Shiv Upadhyay for help getting my citations in the right format. Last but certainly not least, thank you Xiaowei, mom, dad, Eli, Ethan and Mommom and Dandy.

1.0 THE NEED FOR PATH SAMPLING

1.1 INTRODUCTION

Rare events are happening all around us; hurricanes, stock market crashes and even pandemics are some examples of rare events. A rare event is *rare*, in the sense that it happens infrequently, but another important characteristic of a rare event is that once it does occur, it happens quickly. Many important biological processes, such as protein-ligand unbinding and large-scale protein conformational transitions, are rare events and are known to occur on slow timescales of up to seconds and beyond.[1] Because these biological events are rare events, their apparent slow timescales are not due to the event itself, but the time spent waiting for the event to occur.

Molecular dynamics simulations, coupled with all-atom models, are capable of generating molecular movies of biological processes. Viewing a molecular movie is the most direct way to access the mechanism of biological processes such as conformational changes. Typically, molecular dynamics simulations involve the integration of Newton's law of motion over the course of a very short timestep (2 femtoseconds). In order to simulate a biological event on the timescale of microseconds, a total of 500 *million* integrations would be required. While this large number of integrations may seem daunting, advances in computing hardware have enabled all-atom, conventional molecular dynamics (cMD) simulations to generate pathways for biological events of medium-sized proteins ($\sim$100,000 atoms, when including explicit solvent molecules) up to the multi-microseconds timescale on typical computing resources.[1] Specialized hardware, i.e. Anton, can enable the simulation of processes as slow as the milliseconds timescale.[2,3] However, many interesting protein motions–such as large-scale conformational changes–occur on timescales beyond milliseconds.[1] Additional challenges to all-atom cMD, which are summarized in Figure 1, are that massive systems (e.g., larger than one million atoms)[4–10] or the use of either polarizable models (e.g., AMOEBA or DRUDE)[11,12] or sub-atomic models (e.g., hybrid QM/MM potentials)[13–15] will restrict to sampling to much shorter timescales (e.g., hundreds of nanoseconds to a few microseconds).

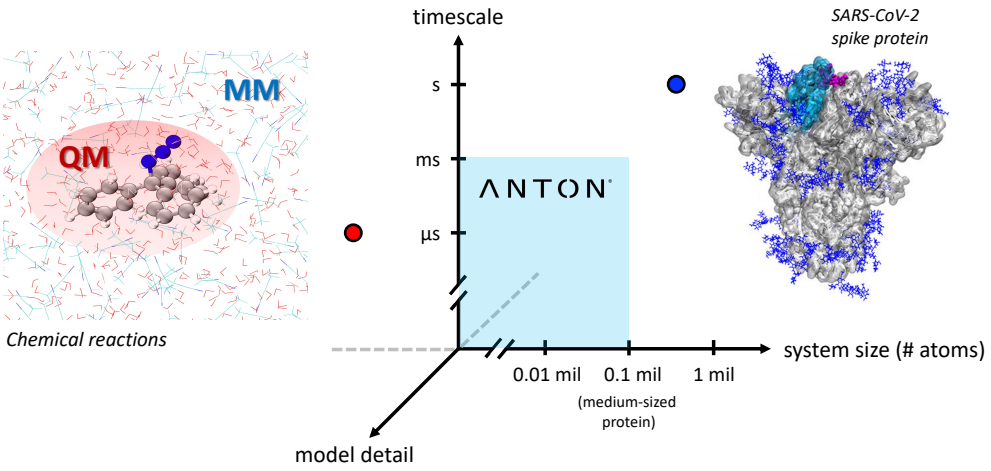

Figure 1: Challenges to all-atom conventional molecular dynamics (cMD) simulations in terms of system size, timescale and model detail. Special-purpose computing resources such as Anton can access up to milliseconds time scale processes for a medium-sized protein and classically-treated models (light blue box). Increasing model detail, such as the use of polarizable force fields (AMOEBA or DRUDE) or addition of a quantum mechanical potential to model reacting atoms, can further increase the computational cost of cMD.

One way to simulate longer-timescale processes relies on the fact that most challenging biological events are rare events. Because rare events are dominated by waiting times, cMD simulations are limited in the timescales they can access due to their inefficiency in simulating rare events: wasting computational power simulating the waiting time. Path sampling simulation strategies recognize this separation of timescales and seek to re-allocate the computational power of MD simulations from the waiting time to the actual event itself. In the past few decades, path sampling strategies have become popular methods for simulating rare events with rigorous kinetics.[16–19] Because the transition time of a rare event is orders of magnitude faster than the time spent waiting in the initial stable state, path sampling strategies are orders of magnitude more efficient than cMD in generating transition pathways. Prominent examples of path sampling strategies that use trajectory segments to generate interface-to-interface transitions include transition interface sampling,[20] forward flux sampling,[21,22] and milestoning.[23–25] In contrast, the weighted ensemble strategy uses trajectory segments to generate region-to-region transitions.[26,27]

In my thesis, I focus on the weighted ensemble (WE) path sampling strategy.[18,26] The goal of the WE strategy, which is illustrated in Figure 2, is to populate empty regions in configurational space, such as bins, through splitting (creating replicas) of promising trajectories, providing an even coverage of trajectories across a progress coordinate. To accomplish this goal, a user defines a progress coordinate, which is a general measure of how far a system has traveled towards a defined goal, and divides that coordinate into bins. The WE strategy assigns statistical weights to an initial set of trajectories and completes two operations in an iterative process: (1) propagate stochastic dynamics (e.g., using a weak Langevin thermostat) for the trajectories in parallel for a short time interval tau (e.g., 100 ps) and (2) apply a resampling procedure, which involves both splitting and merging trajectories. The splitting step, which enriches sampling for success, creates a replica (the coordinates and velocities are copied, but with a different random seed for the thermostat to ensure stochasticity) of a trajectory that transitions to an unoccupied or under-populated bin and is continued until a target number of trajectories per bin is reached. The merging step, which saves computing time, terminates unproductive trajectories in a bin and is applied when the number of trajectories in a given bin is greater than the specified target number of

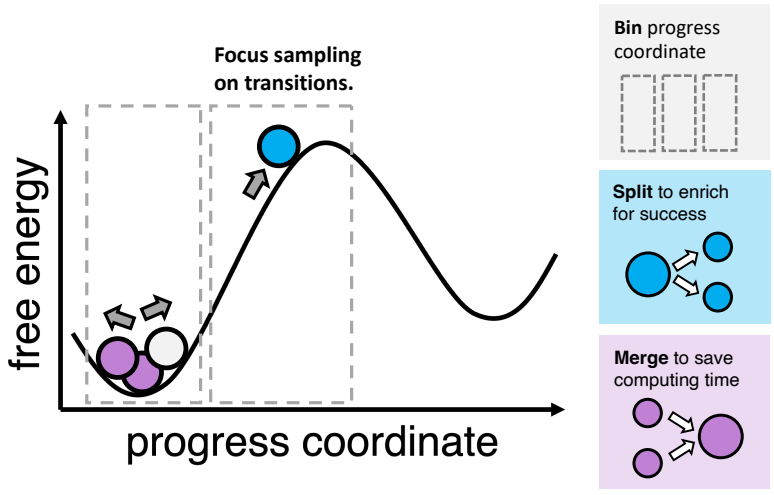

Figure 2: The weighted ensemble path sampling strategy focuses sampling on transitions. First, a progress coordinate is divided into bins. Within each bin, splitting (shown in the blue box) and merging (shown in the purple box) of trajectories occurs to maintain a target number of trajectories per bin. Each trajectory segment is assigned a statistical weight at the start of the WE simulation and the total weight of all trajectories must always sum to one. The trajectory weights are divided among the children trajectories in the case of splitting and transferred from a terminated trajectory to a surviving trajectory in the case of merging.

trajectories per bin. The WE resampler divides a trajectory's weight among child trajectories in the case of splitting, or combines a terminated trajectory's weight with another trajectory in the case of merging. The sum of all trajectory weights, at all times during the simulation, must sum to one. In addition, when merging, the choice of which trajectory to transfer the weight of a terminated trajectory must be random. Thus, throughout the resampling process, the WE strategy remains statistically unbiased, providing continuous pathways and direct calculations of rate constants.[28]

The WE strategy shares many features with transition interface sampling, forward flux sampling and milestoning, such as scalability and the use of trajectory segments to generate transitions between partitions of phase space.[27] However, WE differentiates itself from other path sampling methods by intervening at more frequent intervals in the sampling process. Instead of catching trajectories "in the act" of crossing a new interface, the weighted ensemble strategy splits trajectories that have progressed into newly visited bins after fixed, short time intervals. The more frequent splitting of trajectories in the WE strategy makes monitoring simulation progress easier; further, more frequent opportunities for merging help minimize information loss.[27] Finally, because trajectory weights are independent of the progress coordinate, it is possible to change the progress coordinate and reposition bins at any time during a WE simulation.[18] My thesis highlights the enhanced adaptability of the weighted ensemble strategy, and demonstrates how this adaptability facilitates the simulation of ambitious systems.

1.2 KEY PARAMETERS AND RULES OF THE WEIGHTED ENSEMBLE STRATEGY

The success of a WE simulation can depend heavily on the careful selection of a few key parameters summarized in Figure 3 and discussed in detail below. The progress coordinate is the most impactful parameter in a WE simulation. When a WE simulation begins, an MD engine propagates many trajectory "walkers"–replicates of the system–forward in time. After dynamics propagation, the WE algorithm makes an informed decision about which walkers

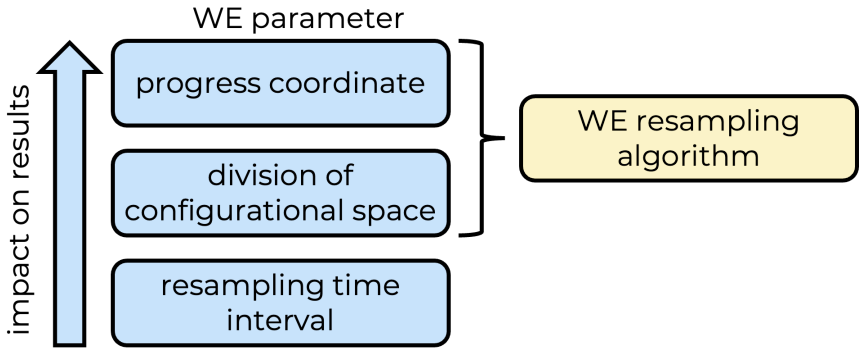

Figure 3: The key parameters of a WE simulation and their varying impact on simulation results. The progress coordinate is the most impactful on a simulation's success followed by the division of configurational space and the resampling time interval. The WE resampling algorithm integrates the progress coordinate and division of configurational space into a decision-making function that is also user-adjustable.

to split and which to merge before the next round of dynamics. The progress coordinate measures how close each walker is to reaching the simulation's desired goal. Progress coordinates can either focus sampling towards a target (such as increasing the RMSD of a certain region of a protein from the starting structure to simulate a conformational change) or can explore phase space with no target (such as increasing the RMSD of the entire protein to simulate a protein's conformational ensemble).[29] A progress coordinate can be multi-dimensional–with secondary and tertiary dimensions providing orthogonal system degrees of freedom that could be helpful in enhancing the simulation of a process of interest. The progress coordinate is intended to describe the slowest relevant motion of the process one wishes to sample by fully capturing the transition of interest from start to finish. It is helpful for the progress coordinate to be continuous (non-discrete) to encourage sampling at intermediate values.

A progress coordinate can be chosen based on physical intuition and need not be perfect, as WE is able to enhance sampling even with an inefficient coordinate. Determining an effective progress coordinate for a system is a common challenge for not only weighted ensemble, but a wide variety of path sampling and other enhanced sampling approaches including umbrella sampling and metadynamics.[30–33] The main caveat of the WE strategy is that, due to the choice of progress coordinate, orthogonal motions in a system may be under-sampled. However, in the worst case, those motions will be sampled in a "brute-force" manner with the same efficiency as cMD.

The second most impactful parameter of a WE simulation is the division of configurational space, which typically involves bins. Binning, or any scheme that groups walkers in configurational space, guides the WE algorithm to split walkers with different progress coordinate values and merge walkers with similar progress coordinate values. Users can define a rectilinear binning scheme in one, two, of three-dimensional space. It is also possible to devise nested binning schemes in which one binning scheme is placed inside of another binning scheme. Both nested binning and adaptive Voronoi binning[28] can focus sampling at specific intersections of high-dimensional phase space. A good binning scheme promotes an even distribution of walkers per bin along the progress coordinate.[34,35] Bin boundaries should be close enough that walkers can transition to empty bins, but far enough apart that walkers cannot vacate the bin too quickly. In addition, a good binning scheme should be

effective in sampling a transition of interest, which usually involves placing bins more finely along the barriers in progress coordinate space.

The WE resampler, a more integrated component, selects walkers for splitting or merging using both the progress coordinate and binning of walkers. While challenging to modify and construct, different resampling algorithms open a wealth of new possibilities for the WE method. For instance, custom resamplers can project multi-dimensional progress coordinates down to one or two dimensions before making split/merge decisions. In addition, modified resampling algorithms allow for the implementation of binless resampling strategies, such as REVO.[36,37] When modifying the WE resampler, one must keep in mind two "unbreakable rules" of the WE strategy. The first rule is that at any given iteration, the weights of all walkers must sum to one. The second rule is that when redistributing walker weights, whether through merging or from a post-merging operation, the redistribution of weight must be random.[26] To be clear for the case of merging, the decision of which walkers to *consider* for merging can be based on any metric, but once walkers are selected, the choice of which one survives must be made randomly based on each walker's weight. Breaking either of these rules would result in statistical bias.

1.3 SHOOTING FOR THE MOON WITH WEIGHTED ENSEMBLE APPLICATIONS

The WE strategy can provide continuous transition pathways and direct calculations of rate constants.[18] However, the effectiveness of WE is limited by large systems ($>$ 500,000 atoms) and/or long timescales ($>$ the ms timescale). Converging simulations of large and/or long timescale systems to a non-equilibrium steady state for rate constant estimates can be extremely challenging.[39,40] However, it is possible to generate continuous pathways for these ambitious systems, which provide powerful insights into the molecular mechanism. Chapter 2 of my thesis showcases mechanistic insight from continuous pathways generated by WE simulations of SARS-CoV-2 spike opening.[41] The SARS-CoV-2 spike protein, detailed in Figure 4, has been a valuable stress test for developing new methods for the WE strategy,

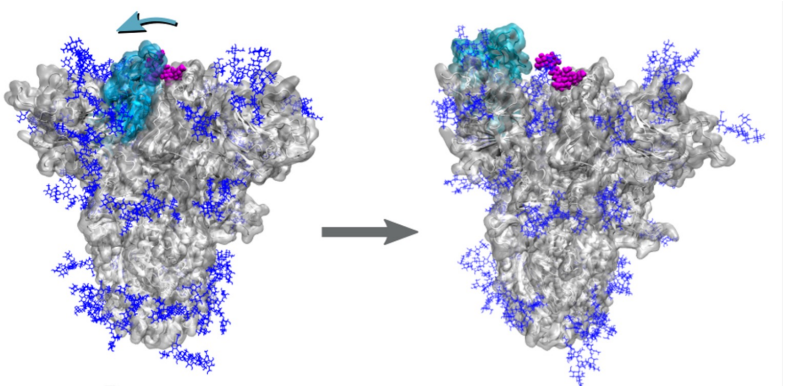

Figure 4: The SARS-CoV-2 spike protein undergoes a large-scale conformational change from RBD-down to a single RBD-up. This conformational change, which occurs on the seconds timescale, has provided an invaluable stress test for the WE strategy. Adapted with permission.[38] Copyright 2021 American Chemical Society.

such as adaptive binning.

1.4 RECENT ADVANCES IN WEIGHTED ENSEMBLE SOFTWARE AND METHODOLOGY

The open-source WESTPA software package[42] has enabled the simulation of numerous, ambitious rare-event processes with the WE strategy.[41,43–47] While we were simulating the SARS-CoV-2 spike protein opening, a stress test for both the WE method and the WESTPA software package, I realized that tackling such ambitious systems over simpler toy models was extremely valuable. We needed to innovate. I began my work by contributing improvements to the WESTPA code, helping to upgrade the code to Python 3, reorganize the core components of the code and add tools for new methods I developed. I discuss my contributions, and the contributions of many others that led to the major release of WESTPA 2.0,[48] in Chapter 3 of my thesis.

One innovation to the WE methodology that I developed in response to stress tests like the simulation of the SARS-CoV2 spike protein was the Minimal Adaptive Binning (MAB) scheme for WE simulations.[49] Choosing an effective–and resource efficient–binning scheme is a challenge in running WE simulations, especially for more complex systems. Binning schemes can be especially challenging to define when progress coordinates are more than two dimensions or when different dimensions of the progress coordinate are important at different times during the simulation. The MAB scheme I devised and implemented in WESTPA 2.0 updates bin positions on-the-fly based on the positions of 1) the walkers that have traveled the furthest along the progress coordinate (in one or both directions) and 2) a bottleneck walker. The MAB scheme, which I discuss in Chapter 4 of this thesis, is more efficient in generating pathways for transitions over high energy barriers when compared with fixed binning schemes and has generated pathways for ambitious systems such as the SARS-CoV-2 spike opening system described in Chapter 2.[41] In addition, the MAB scheme has removed much of the trial and error in placing bins for a given resampling time interval and target number of walkers per bin.

Inspired by the SARS-CoV-2 spike opening simulations, I also developed the LPATH tool which provides a general, semi-automated workflow for analysis of pathway ensembles from both path sampling and cMD simulations. I implemented LPATH and a wealth of plotting tools in WESTPA 2.0. In Chapter 5 of this thesis, I discuss the LPATH tool in more detail, including its relation to the pathway similarity analysis (PSA) and pathway histogram analysis of trajectories (PHAT) approaches.[50,51] The LPATH tool has proven useful in preliminary attempts to analyze cMD and WE pathway ensembles of an alanine dipeptide model system and spike opening of the original and Delta SARS-CoV-2 strains.

For the sake of brevity, I have included only a selection of projects during my PhD in this thesis (among eight primary-author publications). Additional projects that have facilitated my WE studies of ambitious biological processes[41,52] include developing an implicitly polarized force field for proteins and protein mimetics (AMBER ff15ipq-m) that yields reasonable dynamical parameters[53] and developing and implementing a method for more efficient rate constant estimates from WE simulation.[54] I was also a primary contributor to two sets of tutorials for the WESTPA software package[55,56] and an international team effort involving COVID-19 research that resulted in two publications,[9,57] one of which was awarded the 2020 Gordon Bell Special Prize in HPC-based COVID-19 Research.

2.0 THE NEXT FRONTIER FOR DESIGNING SWITCHABLE PROTEINS: RATIONAL ENHANCEMENT OF KINETICS

Reprinted with permission from Bogetti, A. T.,† Presti, M. F.,† Loh, S. N. and Chong, L. T. *J. Phys. Chem. B* 2021, 125 (32), 9069-9077. † denotes co-first authorship.

2.1 INTRODUCTION

Protein conformational switches—proteins that adopt either active (ON) or inactive (OFF) conformations in response to signals such as ligand binding and incident light—have been exploited as the core machinery behind novel biosensors, therapeutic agents, and "smart" biomaterials.[58,59] The fundamental characteristics of a switch include its signal-to-noise ratio (the extent to which the switch converts between ON and OFF states), sensitivity (what levels of effector are required for activation), and response time (the time required for the switch to turn on and off). Signal-to-noise in biological switches can be a complex phenomenon that is sometimes modulated by agonists/antagonists that induce partial or alternate ON/OFF states, and improving signal-to-noise is an active subfield of its own in switch design. The response time has proven to be even more challenging to optimize. Most design strategies focus on stable states, specifically, the ON and OFF conformations. Switching mechanisms can consist of introducing a second stable state in a monomeric protein, creating new protein–protein or protein–ligand binding interfaces, and fusing protein domains such that they achieve input–output communication. These efforts are typically guided by structures of existing proteins or, more recently, by principles of de novo design.[60,61] Either way, they seek to define the structures and optimize the activities of the stable ON/OFF states of the protein. In general, it is left to chance that the stable states interconvert with reasonable rates.

The above scenario is illustrated by the free energy diagrams of Figure 5 by using the

example of a protein biosensor into which two stable conformations have been engineered (represented by OFF and ON states with the latter binding the target ligand; Figure 5A). Some of the basic properties of the switch, for example, signal-to-noise (turn-on/turn-off ratio) and limit of detection, can be optimized by adjusting the relative thermodynamic stabilities of OFF and ON conformations and the ligand binding affinity (K_d) of the ON state, respectively (Figure 5A). Because the structures of OFF and ON states are typically known, these goals can be achieved by using well-established experimental and theoretical approaches. The response time (given by $[k_{ON} + k_{OFF}]^{-1}$) is proportional to the height of the transition state ensemble (TSE) between OFF and ON.Accelerating the response time can be accomplished by introducing interactions that stabilize the TSE but not the ground states (Figure 5B) or, more commonly, by deleting native interactions that are present in the ground states but absent in the TSE (Figure 5C). In either case the TSE must be characterized by experimental and/or computational means.

Here, we examine recent advances in tackling what we regard as the next frontier in the design of switchable proteins: the rational tuning of kinetics (i.e., turn-on and turn-off rates). Oftentimes this means making a switch cycle between ON and OFF sites more rapidly, so that it can react to conditions that change over a wide range of time scales. In other cases, the goal is to make the switch respond more slowly. For instance, decreasing the turn-off rate is useful for enhancing the sensitivity of biosensors because it enables the ON signal to accumulate and for activating optogenetic tools because it allows for a durable biological response that persists well after light is removed, with reduction of photodamage and photobleaching. For the purpose of this review we assume that faster kinetics/lower barrier heights are intended, although the same principles apply if one desires the opposite effect. The main goal is to be able to optimize response times to match that of a given biological process or practical application.

This frontier is a particularly challenging one, requiring the analysis of transient states that experiments typically cannot capture. While these transient states are ideally generated as part of complete, atomically detailed pathways of the switching process from molecular dynamics simulations, such simulations have not been feasible due to the long time scales of switching processes (>milliseconds). Of particular interest is therefore the synergistic use

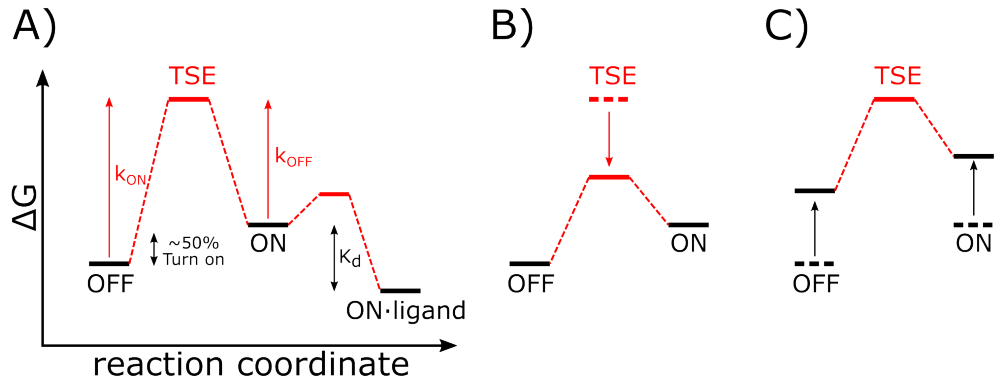

Figure 5: Manipulating barrier heights to accelerate switching rates. (A) Free energy diagram of a protein biosensor with stable OFF and ON states, showing the transition path between them (red). Optimizing the equilibrium properties of the switch (turn-on/turn-off ratio, limit of detection) can be achieved by introducing mutations that shift the relative thermodynamic stabilities of OFF and ON conformations and alter the affinity of the ON state for the target ligand. (B) As in a classic enzyme mechanism, conformational switching can be accelerated by stabilizing the TSE. (C) In practice, it is often more tractable to lower the TSE barrier by destabilizing the ON and OFF folds by introducing mutations that delete interactions that are present in the stable states and absent in the TSE.

of experimental techniques and computational strategies that can enable the generation of detailed structures of transient states for the design of more responsive switchable proteins. We also comment on promising future directions.

We define a protein conformational switch as one in which input and output functionalities are integrated into a single molecule, often by means of fusing receptor and reporter domains in such a way as to facilitate allosteric, interdomain conformational changes. Switch designs can be classified into three broad categories. The first uses an input domain that has naturally evolved the ability to switch between two stable conformations in response to a stimulus. In the second design, the input domain is an existing protein that has but a single fold, with conformational change being achieved via a folding/unfolding reaction that is linked to ligand binding. The third category involves similar mechanisms but employs de novo design principles to generate sequences and structures that may not have existed previously. In each case, the designer is faced with the challenge of converting the binding interaction to an observable signal by means of coupling the input response to an output response. Below we discuss the kinetic barriers that are present in some examples of each category and the experimental approaches that have been used to identify and modulate these barriers.

2.2 EXPERIMENTAL APPROACHES FOR CHARACTERIZING TRANSITION STATES

Central to both experimental and computational approaches for tuning rates is to first identify and characterize TSEs between switch conformations. Depending on the type of switch, as discussed below, these transitions can vary in extent from whole-molecule folding/unfolding to rigid-body domain movement to localized structural rearrangements. The challenge facing experimental methods is to observe sparsely populated states that approximate the relevant TSEs. A protein engineering method known as ϕ-value analysis[62] has been used to map the TSEs of global folding/unfolding. This technique entails introducing a point mutation into a protein and measuring the extent to which it changes the equilibrium

constant between native and unfolded states versus the rates of folding and/or unfolding. ϕ-analysis has been applied to many proteins, and some guiding principles have emerged.[63] NMR- and MS-based methods have been employed to probe more subtle conformational changes, often providing per-residue resolution and rates of interconversion. Recent examples include using ZZ-exchange NMR spectroscopy to measure site-specific folding rates of protein L9,[64] amide hydrogen/deuterium exchange to characterize conformational dynamics of dopamine[65] and XylE membrane-bound transporters,[66] and NMR relaxation dispersion to uncover hidden states and their rates of interconversion in glycotransferase fold switching[67] and dihydrofolate reductase enzymatic function.[68]

2.3 BINDING-INDUCED FOLDING SWITCHES

All native proteins can be made to unfold, and many proteins (including some that are disordered) recognize ligands with high affinity and specificity when they are folded. These features make binding-induced folding a generalizable platform for biosensor engineering. One such example is the alternate frame folding (AFF) design, which was used to convert the small calcium binding protein calbindin D_{9k} into the fluorescent calcium sensor, calbindin-AFF.[69] The AFF modification entailed duplicating the N-terminal EF-hand of calbindin (that contained a calcium-binding residue; cyan in Figure 6A) and fusing it to the protein's C-terminus (magenta). Joining the two polypeptides with a linker long enough to span the N-to-C distance of calbindin allowed calbindin-AFF to switch between two folding "frames", one of which corresponded to the original amino acid sequence (WT frame) and the other to that of a circular permutant (CP frame). The duplicate segments extend from the C-terminus and N-terminus of the WT and CP frames, respectively, as disordered peptides. Binding of calcium to one of the duplicate EF-hands induces it to fold and dock against the shared region of calbindin-AFF (gray), displacing and unfolding its counterpart. The switch was driven in either direction by mutating a calcium-binding residue in one or the other duplicate EF-hand, and the conformational change was reported by strategic placement of donor and quencher fluorophores. The turn-on and turn-off half times were in the 1–10 s

range.

For AFF and other binding-induced folding switch designs, it is reasonable to anticipate that ON/OFF switching times may be accelerated by lowering the barriers to folding and unfolding. Typically, this is done by introducing mutations that raise the free energy of native or denatured states relative to that of the TSE by using the ϕ-value analysis approach. Nevertheless, identifying rate-enhancing mutations by experimental means remains largely a hit-or-miss prospect. Moreover, in the case of calbindin-AFF, the rate-limiting step appears to involve partial unfolding rather than global unfolding,[70] the former of which being more challenging to characterize by traditional ϕ-value analysis. Computational methods were invaluable to improving the response rate of calbindin-AFF (*vide infra*).

2.4 BIOSENSORS WITH PREEXISTING SWITCHABLE INPUT DOMAINS

In contrast to calbindin-AFF, the GCaMP family of genetically encoded calcium indicators (GECIs) employ an input domain (calmodulin, or CaM) that naturally evolved to undergo a dramatic conformational change upon calcium binding. In its calcium-free state, CaM's N-terminal EF-hand, C-terminal EF-hand, and connecting polypeptide adopt a compact, closed conformation. Binding of Ca^{2+} to the EF-hands induces a shift to an extended state that exposes the connecting helix for binding to many protein domains such as the RS20 peptide from myosin light chain kinase. To transduce this change to a fluorescent output, GCaMPs fuse RS20 (magenta in Figure 6B) and CaM (cyan) to the N- and C-termini of GFP (gray) that has been circularly permuted near its chromophore.[71] The interaction between CaM and RS20 protects the chromophore from solvent access, resulting in fluorescent turn-on.

GCaMPs and other GECIs have revolutionized studies of calcium signaling in vivo. To do so, it was necessary to shorten their response times to match those of rapid fluctuations in cellular calcium concentration (1–100 ms). The GCaMP response time is limited by its turn-off rate, which is determined not only by calcium release but also by the extended-to-

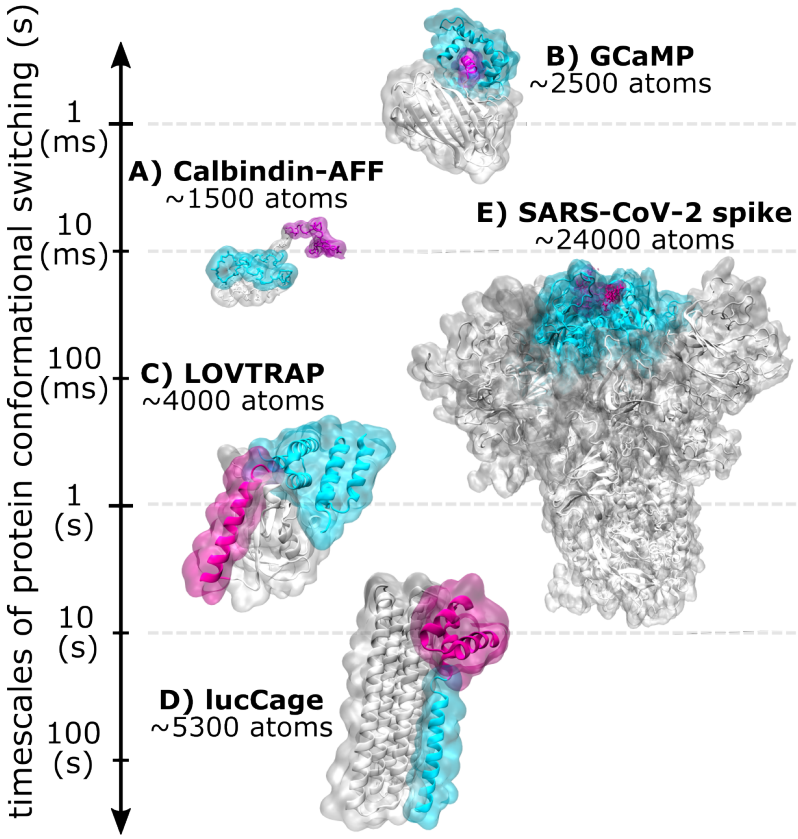

Figure 6: Protein conformational switches and their response times. (A) The calbindin-AFF construct (PDB ID for calbindin D_{9k}: 3ICB) switches via Ca^{2+}-driven unfolding/folding of two duplicate EF-hands (cyan and magenta) and their dissociation/docking with a shared region (gray). (B) The GCaMP calcium sensor (PDB ID: 3EK4) entails Ca^{2+}-induced binding of CaM (cyan) to the RS20 peptide (magenta), which protects the GFP (gray) chromophore from solvent and turns on fluorescence. (C) The LOVTRAP optogenetic construct (PDB ID: 5EFW) involves light-triggered dissociation of the Jα helix (magenta) from LOV2 (gray), resulting in dissociation of Zdark (cyan). (D) The lucCage biosensor (PDB ID: 7CBC) is composed of a cage (gray) and a latch (cyan), to which an analyte recognition domain (magenta) has been fused. Binding of the analyte together with a key (which resembles the latch; not shown) causes the latch to dissociate and expose a sequence in the latch that complements and activates a reporter enzyme. (E) The SARS-CoV-2 spike protein (PDB ID: 6VXX) involves opening of the receptor binding domain (cyan) from the core domain (gray), as gated by a glycan (magenta) attached to the N343 residue.

closed conformational change of CaM that follows. Chemical intuition predicts that mutating residues in the CaM EF-hands (that weaken calcium binding) as well in RS20 (that weaken peptide binding) will accelerate the turn-off rate. Both predictions proved correct; turn-off rates were increased from [2.48 to 4.68 s^{-1}],[72] [5.8 to 21 s^{-1}],[73] and [4.62 to 99 s^{-1}],[74] in various GCaMP GECIs. These results illustrate a central point of this review. Accelerated turn-off rates tended to correlate with higher K_d of the sensors, especially for the EF-hand mutants. This relationship arises because any mutation that weakens ligand binding affinity (RS20 can be considered a second ligand in the GCaMP switch) will likely raise the free energy of the ON state relative to that of the OFF state, thus changing the sensor's equilibrium properties (e.g., sensitivity). If one wishes to optimize response time without perturbing affinity, the mutation(s) should alter the free energies of the ground states relative to the TSE and not with respect to each other (Figure 5B). Rational selection of these mutation sites requires knowledge of the allosteric mechanism gained through experimental or computational means.

The class of photoactivatable proteins exemplified by the second light-oxygen-voltage-sensing domain 2 (LOV2) from *Avena sativa* phototropin 1 is another example of a bioswitch built from pre-existing allosteric domains. Blue light absorption triggers the ON state, in which a covalent bond forms between the flavin mononucleotide (FMN) chromophore and a conserved cysteine.[75] Cys adduct formation is coupled with the rotation of a conserved Gln with concomitant unfolding and dissociation of the N-terminal helix (A'α) and the C-terminal helix (Jα; magenta in Figure 6C) from the LOV2 core domain (gray).[76,77] When the blue light is removed, the photoadduct spontaneously breaks and LOV2 returns to its OFF state, with A'α and Jα folding and rebinding to the core domain. Covalent changes to the FMN chromophore occur on the microsecond time scale, and Jα unfolding proceeds on the millisecond time scale. The ON to OFF reversion, however, requires minutes to hours, making it the rate-limiting step in the photocycle. Random mutagenesis of 7 of the $\sim$20 amino acid sites that comprise the FMN binding pocket identified mutants that exhibit reversion rates from 21-fold faster to 78-fold slower than those of WT LOV2.[78]] The mechanism(s) of rate enhancement remain unclear but may involve destabilization of the Cys-FMN adduct.

An example of LOV2 used as in input domain for a functional switch is the LOV2

Trap and Release of Protein (LOVTRAP) system.[79] LOVTRAP is a two-component switch consisting of LOV2 and Zdark, a 38-residue peptide that was evolved by mRNA display to bind the dark conformation of LOV2. The crystal structure of the dark-state complex revealed that Zdark (cyan in Figure 6C) binds to the LOV2 core domain as well as the tip of Jα. Light-induced unfolding of Jα causes Zdark to dissociate in under a second. As anticipated from earlier LOV2 studies, the dark-to-light conformational change limits the overall response time of LOVTRAP. Reassociation half-times were tuned from 10-fold faster to 26-fold slower (covering a range of 2 s to 9 min) relative to the WT LOV2 construct[79] by mutating two of the FMN-contacting residues previously described.[78,80] LOVTRAP has been used to introduce photocontrol to protein subcellular localization and protein–protein interactions. A protein of interest (POI) is fused to Zdark (or LOV2), and LOV2 (or Zdark) is sequestered to an organelle or anchored to a membrane. Light irradiation causes the POI to dissociate and diffuse to its preferred cellular location and interact with its natural binding partner.[81] In addition to its use in the modular, two-component Zdark system, reversible Jα unfolding has been employed to regulate functions of specific proteins by directly fusing LOV2 to nanobodies,[76] Src kinase,[82] Rac,[83] CamKII,[84] and others. A guide for how to engineer LOV2-based photoswitches, along with tables of the characterized kinetic mutants and potential applications, has been published recently.[85]

2.5 DE NOVO DESIGNED SWITCHES

While building complex allosteric pathways such as those encoded in the CaM and LOV2 sequences is presently out of reach, de novo design of novel protein scaffolds and binding interfaces has reached adolescence, if not maturity. De novo methods thus establish a route for creating binding domains that can be customized to recognize ligands of choice as well as new mechanisms for transducing input to output signals via coupled binding events. The latching orthogonal cage–key (LOCKR) family of protein switches, developed by Baker and colleagues, is composed of a six-helix bundle with the first five helices designated as the "cage" (gray in Figure 6D) and the sixth as the "latch" (cyan).[86] The "key" is an exogenously

added helix that resembles the latch and competes with the latch for docking to the cage. The latch embeds a peptide that can bind a protein partner but is made cryptic by burial in the latch–cage interface. The switch is turned on by addition of the key, which displaces the latch and exposes the peptide for binding its partner. The identity of the peptide establishes the output signal; existing examples are a Bim sequence (programmed cell death) and a degron peptide (protein degradation).

LOCKR switches are activated by a single binding event but are capable of multiple output functionalities. The related lucCage design reverses this relationship to enable biosensors that bind different targets and produce a dedicated output signal (e.g., luminescence).[87] To do so, the latch was modified to contain a domain at its C-terminus (magenta in Figure 6D) that was de novo engineered to possess shared affinity for the cage as well as to any one of a number of analytes to be detected. The activating peptide, also in the latch, was changed to a split luciferase fragment. The complementary luciferase fragment was fused to the key. The combination of analyte binding to the latch and key binding to the cage displaces the latch and allows the luciferase fragments to complement, turning on bioluminescence.

LOCKR and lucCage represent combinations of folds and stabilizing interactions that do not exist in nature. Moreover, the Rosetta-based computational methods used to design them only target the final, lowest-energy structure. They do not consider partially folded structures, pathways, or barriers. A fundamental question thus arises: how do switching rates of de novo designed proteins compare with rates of folding/unfolding and conformational changes of natural proteins? LOCKR and lucCage exhibit turn-on and turn-off times in the minutes-to-hours scale.[87] These times are similar to those of the protein fragment exchange (FREX)-based biosensors.[88] Introduced in 2014, FREX sensors established the analogous unlocking/exchange mechanism to generate output (FRET) but were made from the human fibronectin 3 binding scaffold. This limited comparison suggests that de novo switches already operate with rates in the biological regime even though their designs are based solely on thermodynamic and not kinetic principles. Baker speculates that the folding landscapes of de novo designed proteins tend to be smooth funnels, devoid of large energy barriers, because the design process successfully eliminates competing low-energy states with very different structures.[61] Nevertheless, there is always room for improvement, and response

times of switches based on de novo designs and natural proteins alike can be optimized by using the computational approaches described below.

2.6 COMPUTATIONAL APPROACHES TO TUNING RATES

To our knowledge, only one computational study has reported the rational enhancement of kinetics for a protein conformational switch.[89] The goal of this study was to speed up the slow response time (hundreds of milliseconds) of the engineered protein-based calcium sensor, calbindin-AFF (Figure 6A), by at least an order of magnitude to detect fast physiological Ca^{2+} fluctuations. The computational strategy involved (i) a minimal, residue-level protein model (one bead per residue), (ii) a G$\bar{o}$-type potential[90] that was parametrized to reproduce the thermodynamic stability of each switch component, and (iii) the weighted ensemble (WE) path sampling strategy,[26] which can be orders of magnitude more efficient than standard simulations in generating pathways and rate constants for rare events (e.g., protein folding and protein binding) without introducing any external bias in the dynamics or altering the free energy landscape (Figure 7A).[18] The only prerequisites for this strategy are the structure and experimental folding free energy of the nonpermutant switch component. Despite the simplicity of the simulation model, this strategy identified previously untested mutations that decreased response time by as much as 32-fold (590 to 19 ms) via preferential destabilization of the ground states relative to the transition path ensemble (TPE), which is defined as all transient states in productive pathways, beginning where the trajectory last exited the initial state and ending where the trajectory first entered the target state. In particular, we focused on large, hydrophobic residues that form the most pairwise residue contacts in the initial ground state relative to the TPE-prime candidates for "underpacking" mutations that destabilize the ground state by removing hydrophobic interactions. Importantly, a negative control mutation was correctly predicted to have little effect on the kinetics despite being located near the other mutations. Furthermore, this study demonstrated that the efficiency of the WE strategy relative to standard simulations in estimating rate constants increases *exponentially* with the effective free energy barrier and can therefore be applied to switches

of a similar size (less than a few hundred amino acids) with even slower response times (<100 s).[89]

Although the WE strategy has not been used with atomistic models to rationally manipulate the kinetics for an engineered protein-based conformational switch, the strategy has enabled the generation of rate constants and atomistic pathways for complex biological processes such as protein folding,[43] protein–protein binding,[45] and protein–ligand unbinding.[91] Furthermore, encouraging WE results have been obtained for the activation process of a particularly large natural switchable protein: the glycosylated SARS-CoV-2 spike protein,[4] which must open before binding the human ACE2 receptor to fuse and infect the human host cell. The system for this WE simulation consisted of the head region (residues 16–1140), explicit water molecules, and a physiological ionic strength (150 mM NaCl), totaling almost half a million atoms.

As part of an international team effort that was awarded the 2020 Gordon Bell Special Prize for HPC-Based COVID-19 Research, WE simulations yielded atomically detailed pathways for the opening of the spike receptor binding domain (cyan in Figure 6E) from glycan-shielded state (down) to exposed (up) and open states (Figure 7B).[9] The conformations of the open state align closely with the cryo-EM structure of the ACE2-bound spike protein.[92] While standard MD simulations would require hundreds of years to capture a single, atomically detailed pathway for the opening of the spike—a seconds time scale process[93]—the WE simulations generated hundreds of pathways for spike opening in 45 days by using 100 NVIDIA V100 GPUs in parallel on the TACC Longhorn supercomputer. These pathways reveal that a glycan attached to the N343 residue (magenta in Figure 6E) functions as a gate that controls the opening (switching) process of the spike protein. The functional importance of this glycan has been validated by biolayer interferometry experiments, which revealed a 56% reduction in binding to the ACE2 receptor when N343 is mutated to an alanine. Furthermore, the large-scale collective motions of the spike-opening process are consistent with those observed in two-dimensional cryogenic electron microscopy images of the spike protein.[41] The WE simulations set a new high-water mark for ensemble simulations of atomistic pathways, capturing seconds time scale motions for a massive protein system.

In another simulation study, which was completed on the Folding@home distributing

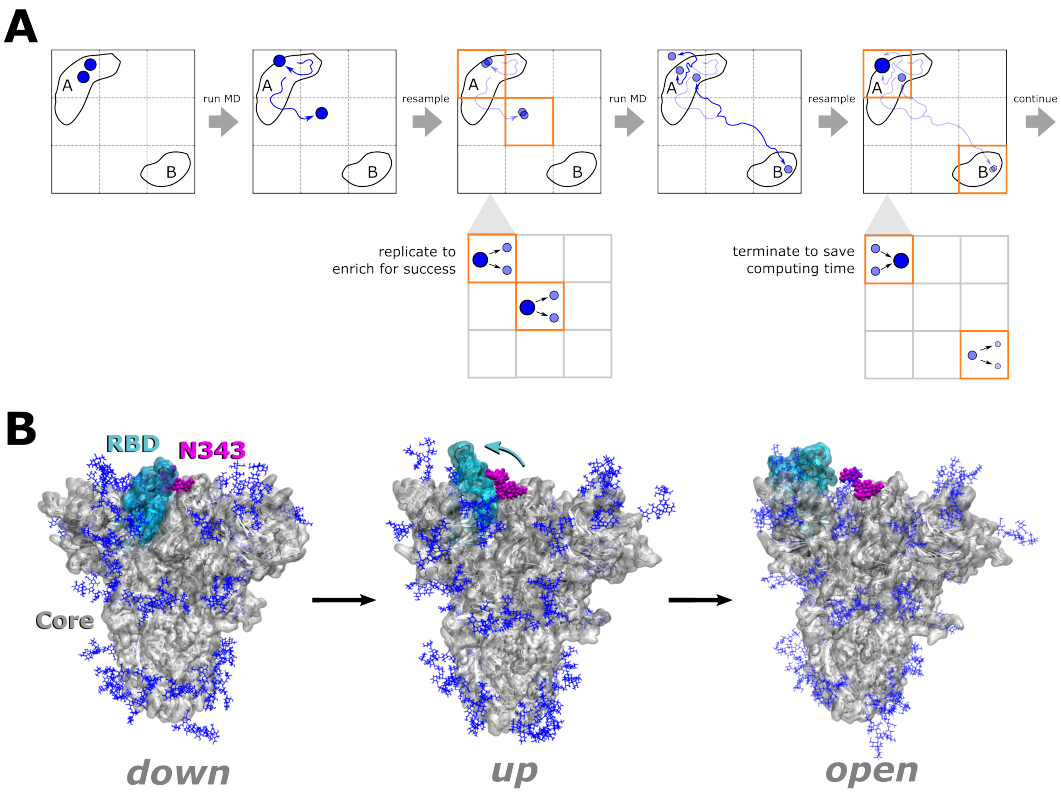

Figure 7: Weighted ensemble simulations of the opening of the SARS-CoV-2 spike protein. (A) Schematic of the weighted ensemble strategy. Trajectories (blue circles) are initiated from state A with equal statistical weights, propagating the dynamics in parallel (blue arrows) for fixed time intervals and applying a resampling procedure after each time interval to ensure equal coverage of configurational space (in this illustration, two trajectories per bin along a two-dimensional progress). The resampling procedure involves replicating trajectories that make transitions to less visited bins and occasionally terminating trajectories that have not made such transitions while rigorously tracking the trajectory weights (indicated by the sizes of the circles). The process of running dynamics and resampling is repeated until a desired number of trajectories have arrived in the target state B. (B) The SARS-CoV-2 spike activation process simulated using the WE strategy. The simulations involved the head region of the spike protein (gray) with full glycosylation (blue) and captured hundreds of switching pathways from the "down" state of the receptor binding domain (cyan) to the "up" and "open" states. Based on these pathways, the glycan attached to the N343 residue (magenta) functions as a gate that controls the switching process.

computing resource, adaptive sampling also captured the open conformations of the spike protein,[94] including the ACE2-bound spike conformation that was sampled by the WE simulations.[41] Like the WE strategy, adaptive sampling is an enhanced sampling strategy that involves iteratively splitting (or replicating) trajectories that have progressed closer to the target state. Together, these results demonstrate the power of "splitting" strategies in sampling switching processes that are beyond the milliseconds time scale and the value of applying such strategies with atomistic models—even when the estimation of rate constants remains a challenge. In contrast to many engineered protein conformational switches, it is not of interest to enhance the switch response time of the spike protein. Rather, the ultimate goal of these studies is to inform strategies for locking the protein in the OFF state, for example, by targeting structures of stable or transient states with small molecules as potential drug inhibitors of COVID-19.

2.7 FUTURE AREAS OF IMPROVEMENT FOR COMPUTATIONAL STRATEGIES

Promising avenues for improving the effectiveness of computational strategies in tuning the kinetics of protein conformational switches include (i) more accurate residue-level, coarse-grained simulation models (force fields) that can offer orders of magnitude speedup over all-atom force fields and (ii) more efficient enhanced sampling strategies to enable faster predictions of mutations that can enhance switching kinetics. Given the slow response times of many engineered protein-based switches, enhanced sampling strategies are essential for capturing switching pathways, even with the use of coarse-grained force fields.

An ongoing challenge of coarse-grained force fields is the ability to simulate protein folding transitions with realistic kinetics. G$\bar{o}$-type potentials[90] on their own have been useful from the perspective of protein engineering in terms of (i) their abilities to reproduce experimental stabilities of individual switch components by optimizing the primary adjustable parameter (the well-depth ϵ) and (ii) their abilities to capture the cooperativity of protein folding, yielding fragment stabilities that are consistent with experimental data.[89] However, such

models yield artificially accelerated dynamics due to the neglect of stabilizing non-native interactions[95] and may not capture non-native, metastable intermediates. On the other hand, the latest generation of coarse-grained force fields that include non-native interactions such as the MARTINI 3[96] and SIRAH 2.0 force fields[97] have not yet matured to the point of being adequate on their own for simulating the folding transitions that can occur for certain protein conformational switches, requiring restraints to maintain secondary structures. To combine the best of both worlds, one might use a hybrid of a G$\bar{o}$-type potential and coarse-grained force field such as MARTINI 3 or SIRAH 2.0.[98] In the very least, electrostatic interactions—both native and non-native—could be used with G$\bar{o}$-type potentials to provide a more realistic, rugged free energy landscape for more quantitative modeling of the protein switching process.

A major challenge of the WE strategy and many other enhanced sampling strategies is the identification of a progress coordinate for the process of interest (e.g., conformational switching). Recent deep learning approaches identify potential progress coordinates by encoding a high-dimensional set of conformational and dynamical features from a training set of trajectory data onto a low-dimensional representation of the features; the progress coordinate can then be decoded to obtain physically relevant details. Two such approaches are the Convolutional Variational Autoencoder (CVAE) method[99] and the Reweighted Autoencoded Variational Bayes for Enhanced Sampling (RAVE) method.[100] The CVAE method has been applied to protein folding, differentiating between various intermediates in the folding process of Fs-peptide,[99] and the RAVE method has been able to detect subtle loop fluctuations in the T4 lysozyme enzyme.[101] Both of these studies highlight the promise of such strategies for aiding the enhanced sampling of large conformational transitions of protein switches. In addition, such strategies could learn effective progress coordinates more efficiently by using complete pathways of the switching processes from WE simulations as training data.[9] These simulations could involve coarse-grained models even if the end goal is to simulate with all-atom models—as long as the coarse-grained simulations have captured the relevant slow motions of the process. Once an effective progress coordinate has been identified, an adaptive binning strategy such as the Minimal Adaptive Binning (MAB) strategy may be applied to automate the placement of bins along the progress coordinate during a simulation

to more efficiently surmount "bottleneck" regions.[49]

Finally, several strategies have been developed for more efficient estimation of rate constants from simulations that have not yet reached a steady state. These strategies include the Rates from Event Duration (RED) scheme, which can estimate rate constants with up to 50% greater efficiency than the original scheme for WE simulations[26] by incorporating the probability distribution of sampled event durations (barrier crossing times).[54] Rate constants can also be estimated more efficiently by constructing a history-augmented Markov state model (haMSM) from completed simulations (e.g., weighted ensemble, adaptive sampling, and standard simulations).[40] In contrast to a standard MSM, an haMSM does not require the use of a long lag time (e.g., $\sim$100 ns) and can therefore provide pathway and kinetics observables for time scales that are both shorter and longer than the lag time.[102] To further accelerate convergence to a steady state, an haMSM could be constructed periodically during a WE simulation to iteratively reweight trajectories.[102] The combination of this on-the-fly reweighting with WE simulation could enable the estimation of rate constants for processes as slow as the seconds time scale, including the switching process of the SARS-CoV-2 spike.[41]

Computational strategies for tuning rates might be applied in two stages. In the first stage, a large set of switch constructs could be virtually screened by using coarse-grained simulations, qualitatively ranking the constructs based on the extent of switching (signal-to-noise) and kinetics (response time). In the second stage, the top one to three switch constructs from the first stage could be characterized by using all-atom simulations to quantitatively identify candidate residues for mutation to improve the response time of the switch. As mentioned above, both stages benefit greatly from the application of enhanced sampling strategies that provide rigorous kinetics (e.g., the WE strategy). To further improve on the efficiency of the computational strategy, deep learning/artificial intelligence strategies could be used to identify more effective progress coordinates for the enhanced sampling and to aid in the detailed analysis of how the protein conformational transitions can occur.

2.8 INTEGRATING EXPERIMENTAL AND COMPUTATIONAL APPROACHES

The power of synergistically combining experimental and computational strategies has been demonstrated for the engineered calbindin-AFF calcium sensor[89]—the only study (to our knowledge) to date that has been successful in rationally enhancing the response time of a protein switch. While time-resolved experiments can measure rate constants for the overall switching process and the thermodynamic stabilities of each switch component (i.e., folding free energies), molecular simulations can provide complete pathways for the switching process, including structures of transient states, which are essential for predicting mutations that could enhance the kinetics. The only prerequisites for these simulations are the structures of the individual switch components and the experimental folding free energy of each component. The latter is used to parametrize the simulation model to yield the expected relative stabilities of the stable states. In the case of a AFF switch construct, only the structure and folding free energy of the parent protein were required for parametrization of the model. The other stable state is a circular permutant of the same protein which can be modeled based on the structure of the parent protein. To further reduce the amount of guesswork and effort required of experiments, these simulations could be used to virtually screen candidate mutations for enhanced response times. Importantly, both thermodynamic and kinetics experiments provide validation of the simulations and help inform the level of detail that is required of the simulation models.

2.9 CONCLUDING THOUGHTS

In closing, experimental and computational strategies have matured to the point where they can be synergistically combined to reduce the amount of guesswork required to engineer protein conformational switches with desired response times. In our own studies, experimentally determined protein structures and thermodynamic stabilities played critical roles in establishing computational simulations and calibrating them. Conversely, theoreti-

cal results inform experiments. For example, de novo approaches alone are seldom sufficient to generate functioning switches. Instead, they typically define structures and amino acid sequences that serve as the starting points for directed or random mutagenesis and library screening experiments. While the prediction of switch response times on time scales beyond milliseconds remain a challenge for atomistic simulations, such simulations of the seconds time scale switching process of the massive SARS-CoV-2 spike protein have demonstrated that the generation of complete pathways for the switching process is in its own right highly valuable, providing direct views of *how* the protein switches from the OFF to the ON state, including the structures of transient states for manipulating the switching kinetics. Given the ever-ongoing advances in computer software and hardware, the future is bright for quantitative predictions of switching kinetics.

2.10 ACKNOWLEDGEMENTS

This work was supported by NIH 1R01GM115805-01 to L.T.C., NIH Grant GM115762 to S.N.L., and the University of Pittsburgh to A.T.B. (Dietrich School of Arts and Sciences Graduate Fellowship). Computational resources were provided by the University of Pittsburgh's Center for Research Computing. We thank Harsimranjit Sekhon, Rommie Amaro, Terra Sztain, and Lorenzo Casalino for helpful discussions.

3.0 WESTPA 2.0: HIGH-PERFORMANCE UPGRADES FOR WEIGHTED ENSEMBLE SIMULATIONS AND ANALYSIS OF LONGER-TIMESCALE APPLICATIONS

Reprinted with permission from Russo, J. D.,† Zhang, S.,† Leung, J. M. G.,† Bogetti, A. T.,† Thompson, J. P., DeGrave, A. J., Torrillo, P. A., Pratt, A. J., Wong, K. F., Xia, J., Copperman, J., Adelman, J. L., Zwier, M. C., LeBard, D. N., Zuckerman, D. M. and Chong, L. T. *J. Chem. Theory Comput.* 2022, 18 (2), 638-649. † denotes co-first authorship.

3.1 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,[103] modern simulations have attacked ever larger complexes[4,5] and even entire virus particles.[7–10] This trend belies the fact that record-setting small-protein simulations in terms of total simulation time remain limited to the millisecond scale on special-purpose resources[2] 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.[17,18,20,23,26,104–109]

Our focus is the weighted ensemble (WE) path sampling approach,[18,26] which has helped 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,[43] coupled protein folding and binding,[44] protein-protein binding,[45] protein-ligand unbinding,[46] and the large-scale opening of the SARS-CoV-2 spike protein.[41] The latter is a significant milestone-both in

the system size (half a million atoms) and timescale (seconds).[41] Instrumental to the success of the above applications have been advances in not only WE methods but also software.[41]

Here, we present the next generation (version 2.0) of the most cited, open-source WE software called WESTPA (WE Simulation Toolkit with Parallelization and Analysis).[42] WESTPA 2.0 is designed to further enhance the efficiency of WE simulations with high-performance algorithms for the following: (i) further enhanced sampling via restarting from reweighted trajectories, adaptive binning, and/or binless strategies, (ii) more efficient handling of large simulation data sets, and (iii) analysis tools for the estimation of first passage time (FPT) distributions and for more efficient estimation of rate constants. Similar to its predecessor, WESTPA 2.0 is a highly scalable, portable, and interoperable Python package that embodies the full range of the WE's capabilities, including a rigorous theory for any type of stochastic dynamics (e.g., MD and Monte Carlo simulations) that is agnostic to the model resolution.[28] In comparison to other open-source WE packages such as accelerated weighted ensemble with a "Work Queue" distributed-computing framework (AWE-WQ)[110] and a weighted ensemble python (wepy) tool,[111] WESTPA is unique in its (i) high scalability with nearly perfect scaling out to thousands of CPU cores[41] and GPUs and (ii) demonstrated ability to interface with a variety of dynamics engines and model resolutions, including atomistic,[45] coarse-grained,[112] whole-cell,[113] and nonspatial system models.[114,115]

After a brief overview of the WE strategy (Section 3.2), we describe the organization of WESTPA 2.0 (Section 3.3) and new analysis tools that further expand the capabilities of the software package (Section 3.4). Together, these features greatly facilitate the execution and analysis of WE simulations of even larger systems and/or slower timescales.

3.2 OVERVIEW OF THE WE PATH SAMPLING STRATEGY

The WE strategy enhances the sampling of rare events (e.g., protein folding, protein binding, and chemical reactions) by orchestrating the periodic resampling of multiple, parallel trajectories at fixed time intervals τ (Figure 8).[26] The statistically rigorous resampling scheme maintains an even coverage of the configurational space by replicating ("splitting")

trajectories that have made transitions to newly visited regions and potentially terminating ("merging") trajectories that have overpopulated previously visited regions. The configurational space is typically defined by a progress coordinate that is divided into bins where an 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.[36] Importantly, trajectories are assigned statistical weights that are rigorously tracked during 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.[28]

WE simulations can be run under equilibrium or nonequilibrium steady-state conditions. To maintain nonequilibrium steady-state conditions, trajectories that reach the target state are "recycled" back to the initial state, retaining the same statistical weight.[39] 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.[116] 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 nonequilibrium steady-state WE simulations in the opposite directions when the historical information is taken into account.[117]

3.3 ORGANIZATION OF WESTPA 2.0

Below, we present the organization of WESTPA 2.0, beginning with code reorganization to facilitate software development (Section 3.3.1) and then proceeding to a description of a Python application programming interface (API) for setting up, running, and analyzing WE simulations (Section 3.3.2); a minimal adaptive binning (MAB) mapper (Section 3.3.3); a generalized resampler module that enables the implementation of both binned and bin-

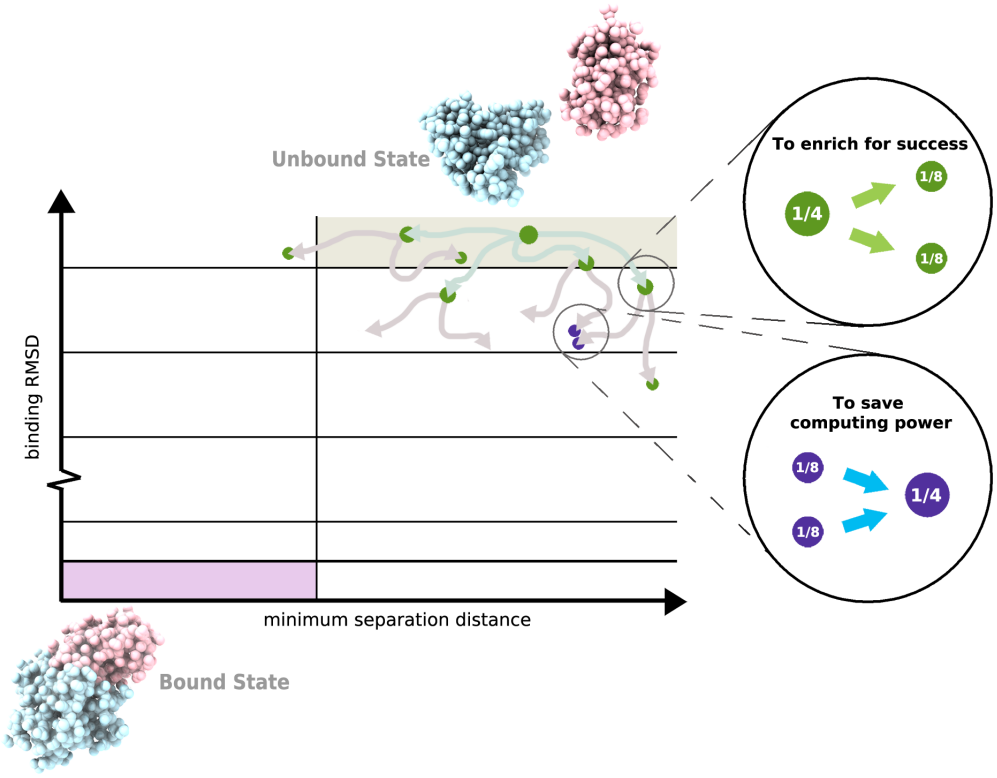

Figure 8: Basic WE protocol. As illustrated for the simulation of a protein–protein binding process, a two-dimensional progress coordinate is divided into bins with the goal of occupying each bin with a target number of four trajectories. Four equally weighted trajectories are initiated from the unbound state and subjected to a resampling procedure at periodic time intervals τ for the following: (i) to enrich for success, trajectories that make transitions to less-visited bins are replicated to generate a target of four trajectories in these 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 these bins (purple spheres). Spheres are sized according to their statistical weights.

less schemes (Section 3.3.4); and an HDF5 framework for more efficient handling of large simulation data sets (Section 3.3.5).

3.3.1 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 the 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 9). 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 precommit hook, alongside flake8 for nonstyle linting. Templates are provided for GitHub issues and pull requests. Both the user's and developer's guides are available on the GitHub wiki along with the Sphinx documentation of key functions with autogenerated docstrings. Further support will continue to be provided through WESTPA users' and developers' email lists hosted on Google Groups (linked on https://westpa.github.io).

3.3.2 Python API for setting up, running, and analyzing 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 10A). 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,[118] or the history-augmented Markov state model (haMSM) restarting plugin (Section 3.4.2), which uses the results of a WESTPA simulation to perform a steady-state analysis then restart the simulation based on the results of that analysis.

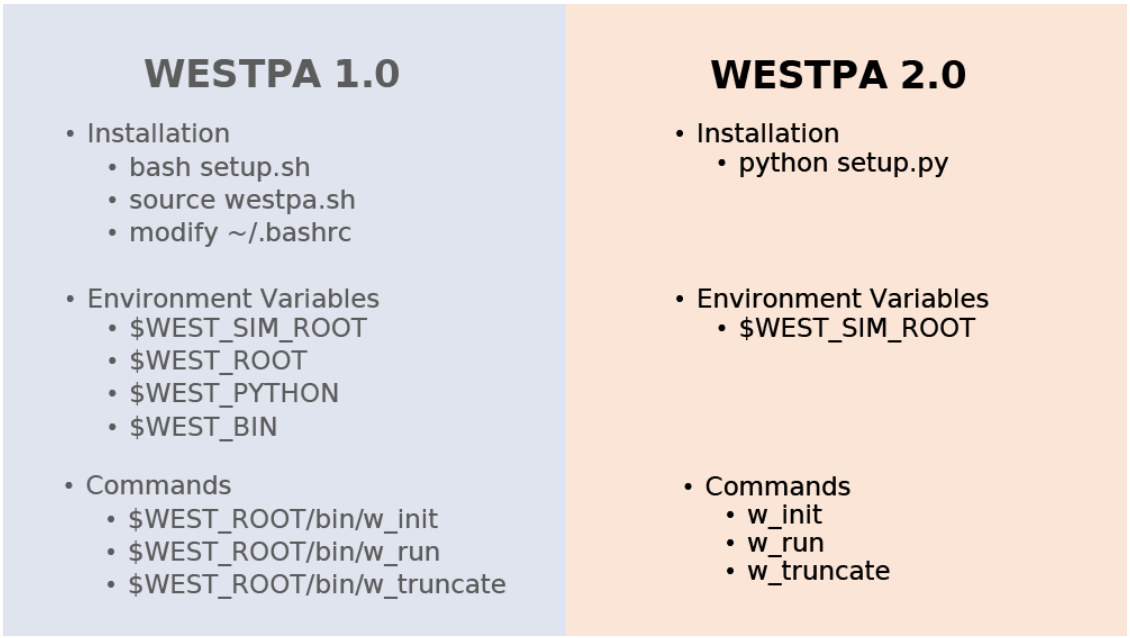

Figure 9: Reorganization of WESTPA 1.0 to WESTPA 2.0. In version 2.0, WESTPA is installed using Python and relies on only a single environment variable such that commands can be called directly through Python. To reflect these changes, we have updated our original suite of WESTPA tutorials for version 2.0 (https://github.com/westpa/westpa_tutorials/tree/westpa-2.0-restruct).[55]

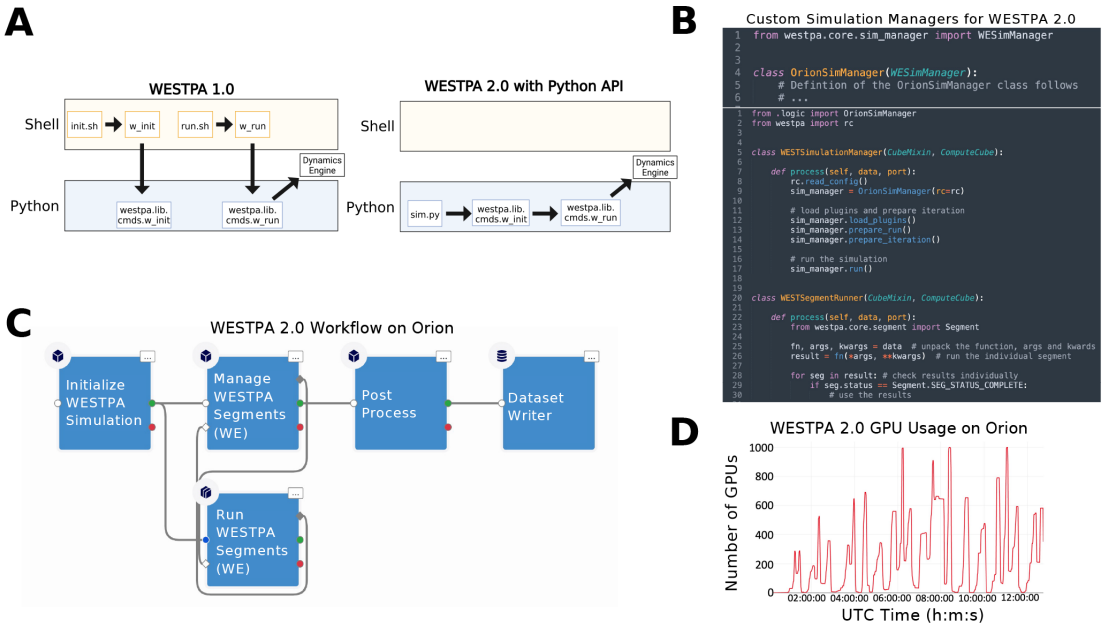

Figure 10: Comparison of workflows for setting up and running WE simulations using WESTPA 1.0 and 2.0, a demonstration of using the Python API for WESTPA 2.0, and GPU performance of the updated API within a cloud computing environment. (A) The Python API in WESTPA 2.0 enables a user to fully define, 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). (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). (C) Example workflow diagram from the Orion user interface using the Python classes constructed from the internal WESTPA APIs. (D) Performance of the WESTPA 2.0 API using the WESTSimulationRunner class within an Amazon Web Services environment using a combination of numerous g4dn instances as a function of the wall clock time in Universal Coordinated Time (UTC) units.

Figure 10B 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 the defined workflow kernels (Figure 10C) and show the GPU performance over wall-clock time (in Coordinated Universal Time; UTC) from a drug-like molecule in a membrane permeability simulation (Figure 10D). Using the internal API, a user's simulation can request large amounts of computational 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).

To facilitate the development of custom analysis workflows in cases where more flexibility is required than that of the existing w_ipa analysis tool,[55] 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) and facilitates retrieval of trajectory data, reducing the overhead of writing custom analysis code in Python and performing quick, interactive analysis of individual 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 values (pcoords), starting conformation (parent), and child trajectories after replication (children) as well as 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).

3.3.3 MAB mapper

To automate the placement of bins along a chosen progress coordinate during WE simulation, we have implemented the MAB scheme[49] 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 and assigning these trajectories to their own bins (Figure 11A,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.[49] 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 19C).

Figure 11D 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 an earlier flux of tacrine through a model cellular membrane bilayer ($\sim$5 vs $\sim$7 ns), and this flux increases more quickly, achieving values that are 2 orders of magnitude higher for the duration of the test.

The MAB scheme provides a general framework for the user creation of more complex adaptive binning schemes.[49] Users can now specify nested binning schemes in the west.cfg file (Figure 11E). To run WESTPA simulations under nonequilibrium steady-state conditions (i.e., with the "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

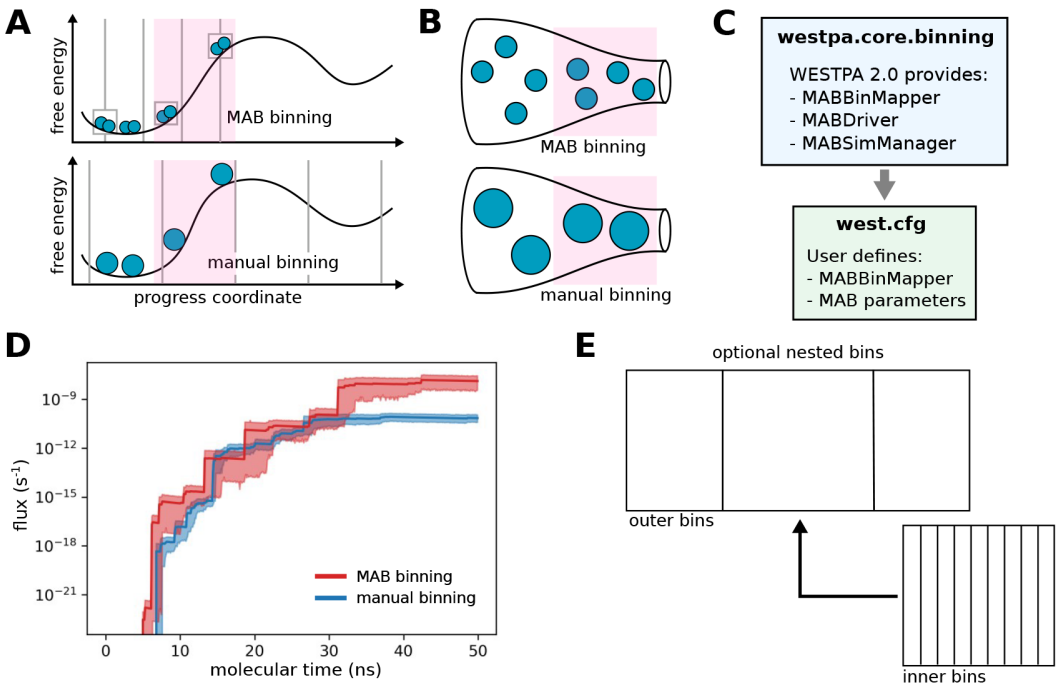

Figure 11: The MAB scheme is more efficient in surmounting free energy barriers than manual fixed binning schemes. (A) Bin positions and trajectories after replication using the MAB scheme vs a manual binning scheme with the same positions of trajectories (blue circles, sized according to statistical weights) along a chosen progress coordinate and a target of two trajectories per bin. (B) Enlarged "bottle" diagrams highlighting the bottleneck region (pink) along with the relative positions and weights of trajectories for the MAB and manual binning schemes in panel (A). (C) MAB scheme options in the westpa.core.binning module and the 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 the 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 Nτ, where N is the number of WE iterations and τ is the fixed time interval (100 ps) of each WE iteration. (E) Schematic of a simple recursive binning case in which closely spaced inner bins are "nested" within a wider outer bin.

the creation of advanced binning schemes.

3.3.4 Generalized resampler module that enables binless schemes

In the original (default) WE resampling scheme, trajectories are split and merged based on a predefined set of bins.[26] In WESTPA 2.0, we introduce a generalized resampler module that enables the 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,[119] our new resampler module provides a more general framework for creating binless schemes by defining a group/reward function of interest; such schemes enable the use of nonlinear progress coordinates that may be identified by machine learning techniques. Following others,[120] 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 12, 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. The basic_nacl_group_by_history example illustrates the grouping of the trajectory based on its "history", that is, 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

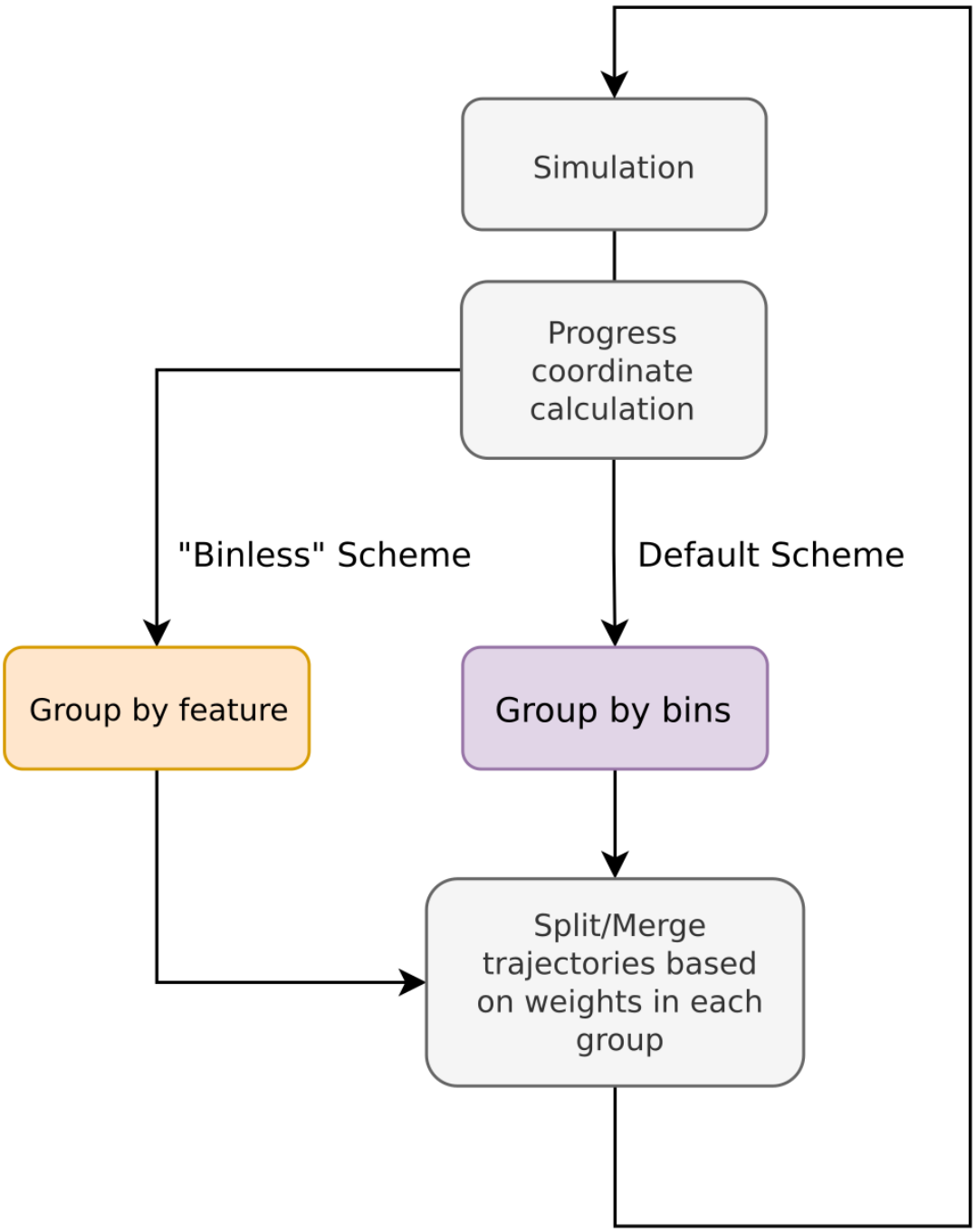

Figure 12: Flowchart for implementing binless resampling schemes in WESTPA 2.0. The implementation involves grouping trajectories by feature (using the group_function keyword defined in the group module) before splitting and merging. The functionality for positioning bins along a chosen progress coordinate remains available in WESTPA 2.0.

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.

3.3.5 HDF5 framework for more efficient handling of large simulation data sets

One major challenge of running WE simulations has been the management of the resulting large data sets, which can amount to tens of terabytes over millions of trajectory files. To address this challenge, we have developed a framework for storing the 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[121] and maintains compatibility with NGLView,[122] an iPython/Jupyter widget for the 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, 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 a potentially large overhead from the file system that results from the storage of numerous small files. For example, a 53% overhead has been observed for a 7.5-GB data set of 103,260 trajectory files generated from NTL9 protein folding simulations (Figure 13), occupying 11.5 GB of actual disk storage on a Lustre file system.

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 (Figure 13A,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 that obtained using 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 (Figure 13C,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 the trajectory coordinate and restart files did not have any significant impact on the WESTPA runtime (Figure 13E), 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 the per-iteration file sizes for the HDF5 run observed in Figure 13D for the p53 peptide. However, the majority of this file size reduction results from efficient HDF5 data compression of trajectory coordinate, restart, and log files. Finally, the 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 3.4 (Figure 13F).

3.4 ANALYSIS TOOLS

WESTPA 2.0 features new analysis tools for estimating rate constants more efficiently using the distribution of "barrier crossing" times (Section 3.4.1), accelerating the convergence using a haMSM to reweight trajectories (Section 3.4.2) and estimating the distribution of FPTs (Section 3.4.3).

3.4.1 RED scheme for rate constant estimation

To more efficiently estimate the 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 ramp-up portion of a

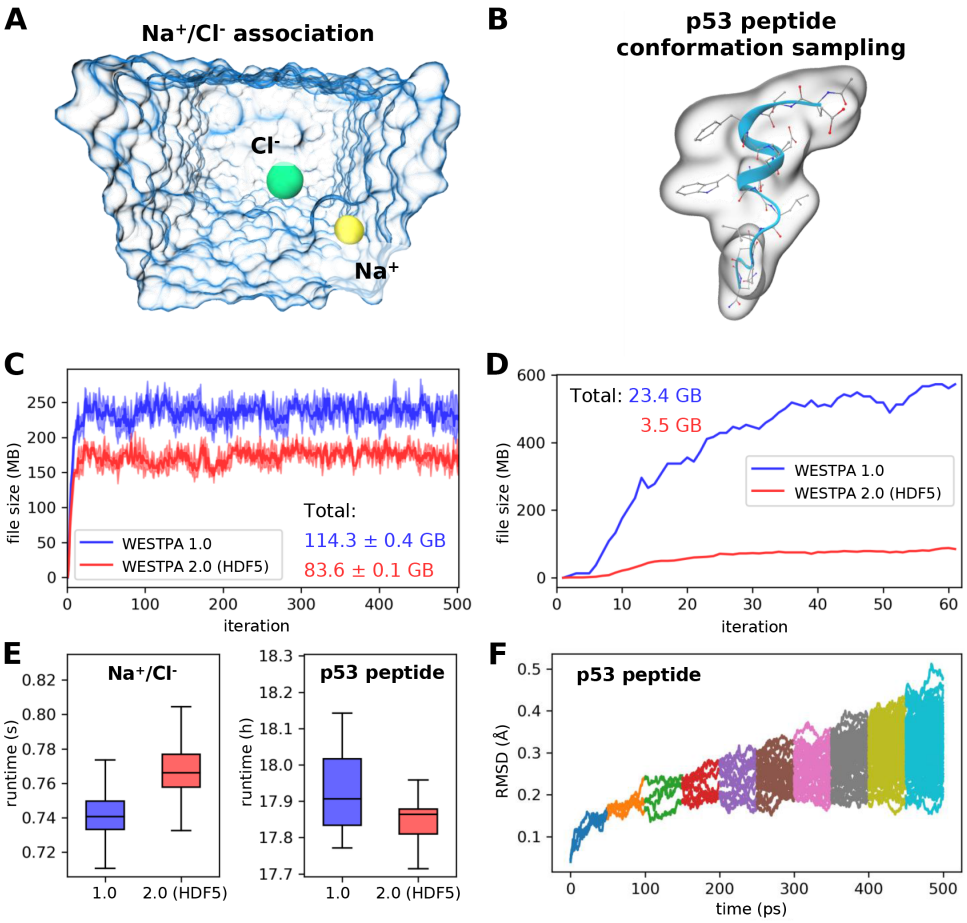

Figure 13: Demonstration of the usage of the HDF5 framework for two example systems. (A) Na^+/Cl^- association simulation where Na^+ (yellow sphere) and Cl^- (green sphere) ions were solvated in explicit water (blue transparent surface). (B) Conformational sampling of a p53 peptide (residues 17-29) in a generalized Born implicit solvent.[44] (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 using trajectories obtained using WESTPA's analysis tools.

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 14A).[54] The correction factor accounts for the systematic error that results from the 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[26] in estimating the association rate constant (Figure 14B).[54]

The code for estimating the rate constants using the RED scheme takes as an 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 that 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 data set called red_flux_evolution to the users' existing direct.h5 file (Figure 14C). The RED rate constant estimates can then be accessed through the Python h5py module and plotted versus time to assess the convergence of the estimates. To estimate the uncertainties in observables calculated from a small number of trials (i.e., the number of independent WE simulations), we recommend using the Bayesian bootstrap approach.[26,124] If it is not feasible to run multiple independent simulations with a certain system due to either the 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.

3.4.2 haMSM restarting plugin

haMSMs provide a powerful tool for the estimation of stationary distributions and rate constants from transient, unconverged WE data.[40] Thus, the approach has a similar motivation to the RED scheme. (48) In haMSM analysis, instead of discretizing trajectories via the WE bins used by WESTPA, as in the WESS and WEED reweighting plugins for a non-equilibrium steady state and equilibrium state, respectively,[39,116] a much finer and more numerous set of "microbins" is employed to calculate the steady-state properties with a higher accuracy. These estimates, in turn, can be used to start new WE simulations from

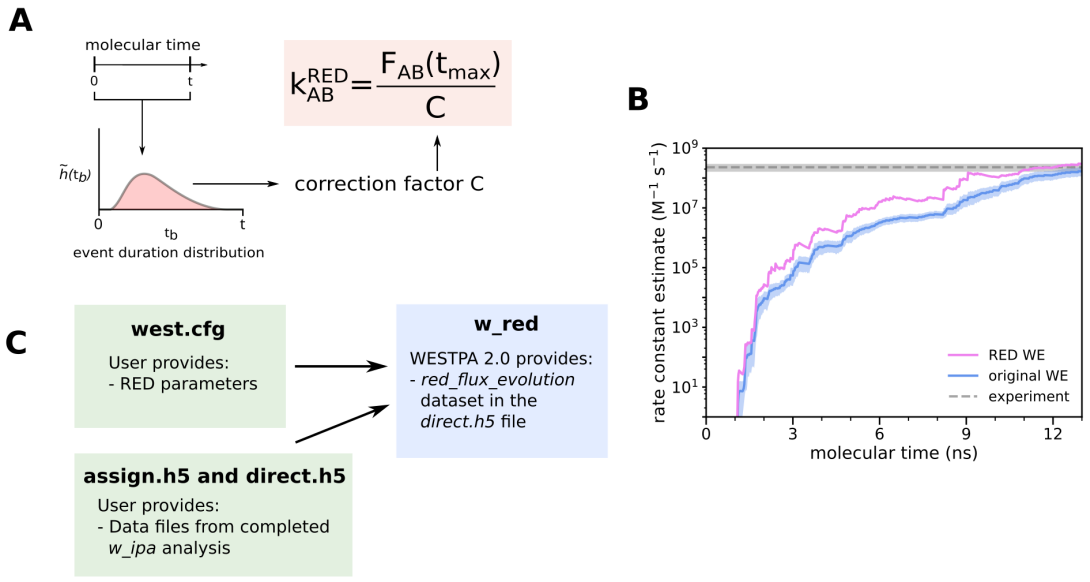

Figure 14: The RED scheme for more efficient rate constant estimation. (A) Schematic illustrating the RED scheme, which incorporates the distribution of event durations as part of a correction factor for rate constant estimates that account for the statistical bias toward the observation of events with 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 the 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.[123] (C) Schematic illustrating how users can generate a data set for calculating the RED scheme correction factor from the simulation data stored in the analysis HDF5 files and apply the correction factor to the rate constant estimate using the new w_red tool.

a steady-state estimate, accelerating the convergence of the simulation.[124] 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 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[125] is used (Figure 15): 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 the flux between them–that is, the weight is 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 the steady state (Figure 16). 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 the steady state and thus converge more quickly. The process can be repeated iteratively, as shown in Figure 16A. The result of this restarting procedure is shown in Figure 16B. 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 α-carbons or even a strided selection of the α-carbons, may be sufficient to capture the important struc-

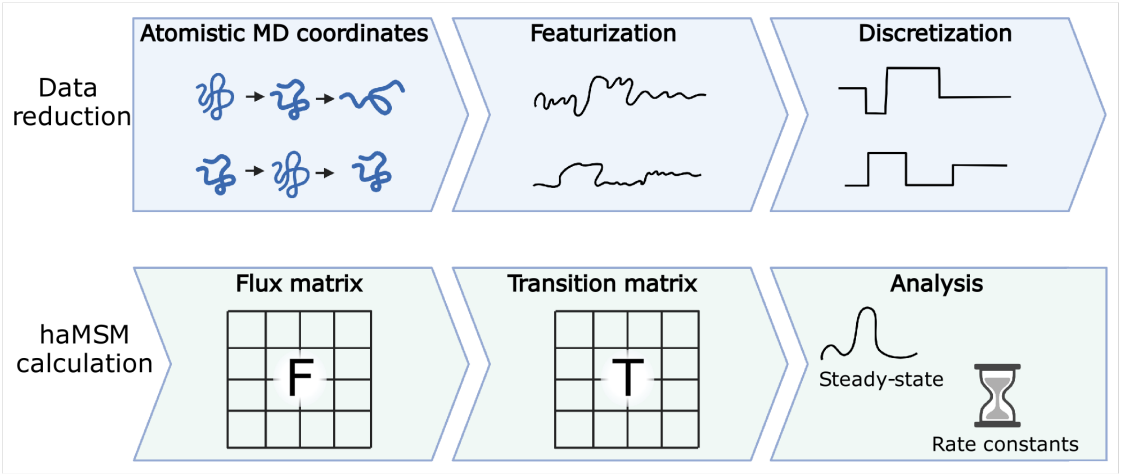

Figure 15: Workflow for constructing an haMSM from trajectories. First, the atomistic trajectories are featurized and discretized. The flux matrix is then computed by computing fluxes between discrete states. The flux matrix is row-normalized into a transition matrix. Estimates of steady-state populations and rate constants are obtained from the analysis of the transition matrix. Figure created with biorender.com.

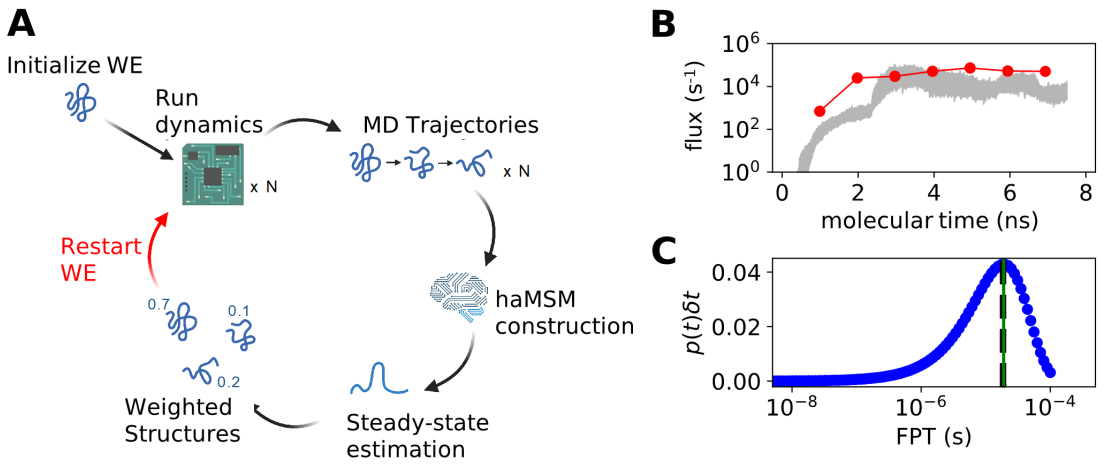

Figure 16: Application of the haMSM restarting plugin to the ms folding process of the NTL9 protein. (A) Diagram of the haMSM restarting plugin's functionality. (B) Example of the restarting plugin functionality in the accelerated convergence of NTL9 folding rate constants from a WESTPA 2.0 simulation using the AMBER 16 MD engine.[126] haMSM estimates at restarting points are shown as dots, WE direct fluxes are shown as red lines, and the 95% credibility region from the direct WE is shown in gray. (C) Distribution of the FPTs for NTL9 folding from the haMSM built at the final restart of the simulation in Figure 16B. The weighted average of the blue FPT distribution is shown in black dashed lines, and the MFPT estimate from the haMSM's steady-state estimate is shown in green.Figure created with biorender.com.

tural information. Choosing the featurization based on rotation-invariant distances, such as pairwise atomic distances instead of atomic positions, can also help capture the structural fluctuations without sensitivity to large-scale motion of the entire system.

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

3.4.3 Estimating FPT distributions

FPTs and their mean values (MFPTs) 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, and diffusion processes of small molecules within crowded environments. WE simulations, via the Hill relation, provide unbiased estimates of the MFPT directly once the steady state is reached[39] or indirectly via non-Markovian haMSM analysis,[116] but the 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.[117] Details on building haMSMs are described above in Section 3.4.2, and more information can be found in the refs[116] and.[117]

Here, we extend and strengthen the earlier FPT distribution analysis from WE data. The original code for calculating the FPT distribution was published on a separate GitHub repository (https://github.com/ZuckermanLab/NMpathAnalysis). Recently, we reorganized and refactored the code in class hierarchical structures: a base class (MatrixFPT) for calculating MFPT and FPT distributions 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, respectively, as mentioned in the haMSM in Section 3.4.2. The updated code has been merged into the msm_we package described

in Section 3.4.2 along with some updates on building a transition matrix from classic MD simulation trajectories.

The new code enables the robust estimation of the FPT distribution. Figure 16C shows the non-Markovian estimation of the FPT distribution of transitions between macrostates A and B from the WE simulation of NTL9 protein folding.

3.5 SUMMARY

WESTPA is an open-source, highly scalable, interoperable software package for applying the WE strategy, which greatly increases the efficiency of simulating rare events (e.g., protein folding and 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 the 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-event sampling, including the identification of slow coordinates using machine learning techniques,[99,100] and the interfacing of the WE strategy with other software involving complementary rare-event sampling strategies (e.g., OpenPathSampling,[127,128] SAFFIRE,[109] and ScMile[129] and analysis tools (e.g., LOOS,[130] MDAnalysis,[131] and PyEmma.[132] WESTPA has also been interfaced with OpenEye Scientific's Orion platform[118] on the Amazon Web Services cloud computing facility. We hope that the above new features of WESTPA will greatly facilitate the 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.

3.6 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, Surl-Hee Ahn and Alan Grossfield for helpful discussions.

4.0 A MINIMAL, ADAPTIVE BINNING SCHEME FOR WEIGHTED ENSEMBLE SIMULATIONS

Adapted with permission from Torillo, P. A.†, Bogetti, A. T.† and Chong, L. T. *J. Phys. Chem. A* 2021, 125, 1642-1649. † denotes co-first authorship.

4.1 INTRODUCTION

Path sampling strategies have been pivotal in enabling the simulation of pathways and kinetics for rare events such as protein un(binding),[44–46,133–135] protein (un)folding,[29,43,136,137] and membrane permeation.[138,139] These strategies exploit the fact that the time required to cross a free energy barrier (t_b) is much shorter than the dwell time in the preceding stable (or metastable) state ($t_b \ll t_{dwell}$) during which the system is "waiting" for a lucky transition over the barrier.[107,140] By focusing the computational power on the actual transitions between stable states rather than on the stable states themselves, path sampling strategies can be orders of magnitude more efficient than standard simulations in sampling the functional transitions of rare events without introducing any bias into the dynamics.[27]

A major challenge for path sampling strategies has been the division of configurational space for a rare-event process. The application of these strategies can therefore be greatly streamlined by schemes that automate the adaptive placement of bins along a chosen progress coordinate. Such adaptive binning schemes have included the use of Voronoi bins[28,119,120] and a variance-reduction approach[34] for the weighted ensemble strategy;[18,26] interfaces have also been used as "bins" to improve flux through bottlenecks between (meta)stable states[109,141,142] for nonequilibrium umbrella sampling[142] and forward flux sampling.[143]

Here, we present a minimal adaptive binning (MAB) scheme within the framework of the weighted ensemble strategy. The scheme can be used with high-dimensional progress coordinates and exhibits the following features: (i) no prior test simulations or training sets

are required as the scheme relies only on the positions of the trailing and leading trajectories along the progress coordinate at chosen fixed time intervals; (ii) fewer bins are required compared with a manual binning scheme due to earlier identification of bottlenecks along the progress coordinate; (iii) the maximum number of CPUs (or GPUs) required is easily estimated prior to running the simulation since a similar number of bins are occupied throughout the simulation; and (iv) the scheme is easily extensible to more sophisticated schemes for adaptive binning. To demonstrate the power of the adaptive binning scheme, we applied the algorithm to simulations of the following processes, in order of increasing complexity: (i) transitions between states in a double-well toy potential, (ii) molecular association of the Na^+ and Cl^- ions, and (iii) conformational transitions of an N-terminal peptide fragment of the p53 tumor suppressor.

4.2 THEORY

4.2.1 The weighted ensemble strategy

The weighted ensemble (WE) strategy involves running many trajectories in parallel and applying a resampling procedure at fixed time intervals τ to populate empty bins in configurational space, typically along a progress coordinate.[18,26] The resampling procedure involves replicating trajectories that advance toward a target state, enriching for success in reaching the target state via a "statistical ratcheting" effect; to save computing time, trajectories that have not made any progress may be terminated, depending on which bin they occupy. Importantly, the rigorous tracking of trajectory weights ensures that no bias is introduced into the dynamics, thereby enabling the calculation of nonequilibrium observables such as rate constants. Furthermore, since the trajectory weights are independent of the progress coordinate, the progress coordinate as well as bin positions can be adjusted "on-the-fly" during a WE simulation.[28]

WE simulations can be carried out under nonequilibrium steady-state or equilibrium conditions.[116] Nonequilibrium steady-state trajectories that reach the target state are "recycled"

by terminating the trajectories and starting a new trajectory from the initial state with the same statistical weight. Equilibrium trajectories are not recycled, which means that target states need not be strictly defined in advance of the simulation.

4.2.2 The MAB scheme

The minimal adaptive binning (MAB) scheme works by first placing a fixed number of evenly spaced bins between "boundary" trajectories: the trailing and leading trajectories along the progress coordinate at a given time. Then, the trailing, leading and "bottleneck" trajectories (described below) are assigned to separate bins. The WE strategy then replicates and prunes trajectories within each bin at fixed time intervals (WE iterations) to maintain a target number of trajectories per bin; trajectory weights are split or merged, respectively, according to rigorous statistical rules.[28]

When running a WE simulation, a steep energy barrier will often slow progress in a particular direction of the progress coordinate. In order to enhance the chance of surmounting that steep barrier, more splitting will need to occur along the barrier. In practical terms, this means that a user will need to space bins more finely along the steep barrier so that there is a higher chance of transitioning into a new bin and therefore splitting. The MAB scheme, a heuristic approach, does not directly detect the location of a steep energy barrier, but does so indirectly. If a leading trajectory is not progressing very far per iteration, the bins that MAB uses to split that leading trajectory will have all been very close to each other, mimicking the fine spacing of bins that would be optimal for surmounting a barrier based on a known potential energy curve or surface.

For the bottleneck trajectory detection, MAB will still have the goal of surmounting the steep energy barrier again (to generate multiple, independent crossing events) by more frequent splitting but cannot rely on the movement of the leading trajectory since that trajectory will already have surmounted the barrier. The bottleneck detection feature will then rely on another indirect indicator of where a steep barrier may be: one based on probability differentials. If a steep barrier is present, it is likely that splitting is the only way trajectories have overcome that barrier and after many rounds of splitting the trajectories

at the top of the barrier will be lower in weight than those at the base of the barrier. The bottleneck trajectory in each uphill direction of interest is identified by calculating the following value of a probability differential Z for each bottleneck candidate and choosing the trajectory that maximizes the following objective function at a given time:

$$Z = log(p_i) - log\left(\sum_{j=1}^{n} p_j\right) \tag{1}$$

where p_i is the log of the weight of trajectory i under consideration and $\sum_{j=1}^{n} p_j$ is the log of the cumulative weight of all n trajectories that have surpassed trajectory i along the progress coordinate in the direction of interest. The log of the weights are used in bottleneck detection due to the fact that WE trajectories, especially near the bottleneck, can have weight differentials in excess of an order of magnitude. Z therefore favors the selection of relatively heavy-weight trajectories with the smallest cumulative weight 'ahead' of them along the progress coordinate. This relative weight differential effectively acts as a proxy for a bottleneck.

Figure 17 illustrates the steps of the MAB scheme for a one-dimensional progress coordinate:

1. Run dynamics for one WE iteration with a fixed interval τ.
2. Tag boundary and bottleneck trajectories with regard to the current WE iteration.
3. Adapt bin positions by dividing the progress coordinate evenly into a specified, fixed number of bins between the positions of the tagged trailing and leading trajectories; assign trailing, leading, and bottleneck trajectories to separate bins.
4. Replicate and prune trajectories to maintain a target number of trajectories in each bin.
5. Run dynamics with updated bins and repeat steps 1–4.

For a multidimensional progress coordinate, steps 2 and 3 are carried out for each dimension of the progress coordinate. When multiple bottlenecks exist, replication of the most major bottleneck trajectory at the current WE iteration enriches for successful transitions over the corresponding bottleneck, enabling later bottlenecks along the landscape to be tackled. To avoid the replication of trajectories outside of the desired configurational space (e.g., regions of unintentional protein unfolding), the MAB scheme includes the option to specify

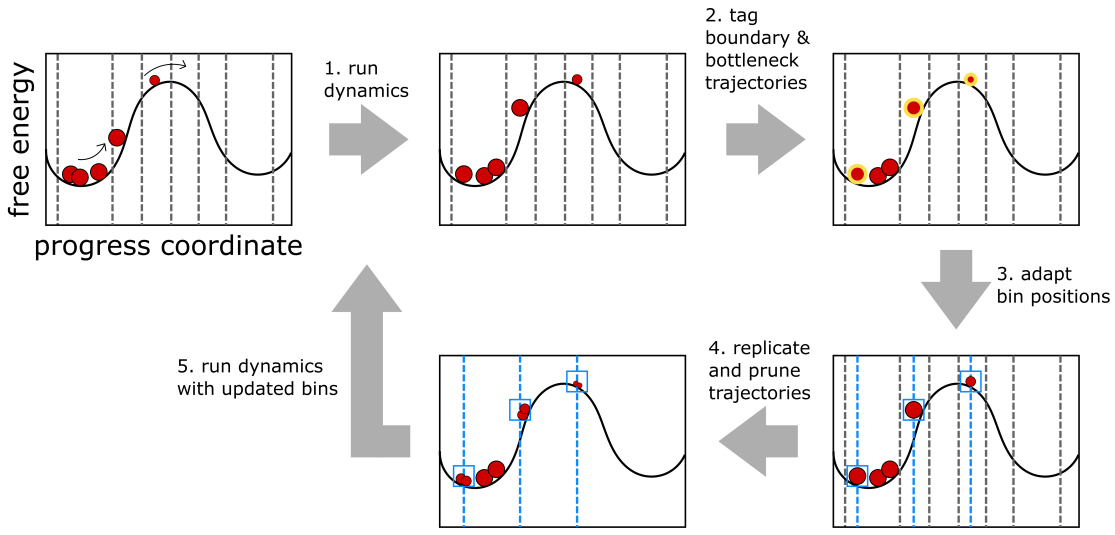

Figure 17: Illustration of the MAB scheme for adaptive placement of bins along a one-dimensional progress coordinate. The scheme involves five steps. (1) Run dynamics for a short, fixed time interval τ using initial bins indicated by gray vertical lines. Trajectories are represented by red circles with sizes that are proportional to their statistical weights. (2) Tag boundary and bottleneck trajectories (highlighted in gold). (3) Adapt bin boundaries (blue vertical lines) by placing a fixed number of bins evenly between the positions of the trailing and leading trajectories along the progress coordinate and assigning each boundary and bottleneck trajectory to a separate bin (blue boxes). (4) Replicate and prune trajectories to maintain a target number of trajectories per bin. (5) Repeat steps 1–4 with updated bin positions until a desired amount of sampling is achieved.

minimum and/or maximum limits of another observable as an additional dimension to the progress coordinate for the replication of trajectories. Since the number of trajectories per bin is fixed and a similar number of bins are occupied throughout the WE simulation, including separate bins for boundary and bottleneck trajectories, we can easily estimate the maximum number of CPUs (or GPUs) required for the simulation. A Python implementation of the MAB scheme is available for use with the WESTPA software package.[42]

4.3 METHODS

4.3.1 WE simulations

All WE simulations were carried out using the open-source WESTPA software package.[42] For each benchmark system, we compared the efficiency of the MAB scheme for adaptive binning to a manual, fixed binning scheme. We present progress coordinates and binning schemes for each benchmark system below.

4.3.2 The double-well toy potential

The double-well toy potential consists of two equally stable states separated by a 34 kT free energy barrier. The potential was defined as

$$V/kt = -60 \times cos^2 x + \frac{3.75}{sin^2 x} \tag{2}$$

For the manual binning scheme, a one-dimensional progress coordinate was divided into 20 bins along a theoretical X position metric ranging from an initial state A at $X = 0.5$ to a target state B at $X = 2.5$; for the MAB scheme, a fixed number of bins ranging from 5 to 20 was used throughout the WE simulation for the same progress coordinate at any given time to determine the impact of the number of bins on the efficiency of generating successful transitions. For each binning scheme, a single WE simulation was run with a fixed time interval τ of 5×10^{-5} for each iteration and a target number of 5 trajectories/bin, yielding a total simulation time of 200 000 δt.

Dynamics were propagated according to the overdamped Langevin equation:

$$X(t+\delta t) = X(t) - \frac{\delta t}{\gamma}\nabla_X V + \delta X^G \tag{3}$$

where γ is the friction coefficient, δt is the time step, and δX^G is a Gaussian random displacement with zero mean and variance $2\frac{kT}{\gamma}\delta t$ with $\delta t = 5 \times 10^{-5}$ and reduced units of $\gamma = 1$ and $kT = 1$.

4.3.3 The Na^+/Cl^- system

To sample Na^+/Cl^- associations in explicit solvent, 5 independent, nonequilibrium steady-state WE simulations were carried out for each of the two binning schemes. A one-dimensional progress coordinate was used which consisted of the Na^+/Cl^- separation distance. A total of 28 bins were equally spaced from a maximum value of 20 Å down to a target state at 2.6 Å. For both binning schemes, 1000 WE iterations were run with a fixed time interval τ of 2 ps for each iteration and a target number of 4 trajectories/bin, yielding an aggregate simulation time of 0.2 μs.

Dynamics were propagated using the AMBER18 software package[144] with the TIP3P water model[145] and corresponding Joung and Cheatham ion parameters.[146] Simulations were started from an unassociated state with a 12 Å Na^+/Cl^- separation and a truncated octahedral box of explicit water molecules that was sufficiently large to provide a minimum 12 Å clearance between the ions and box walls. The temperature and pressure were maintained at 298 K and 1 atm using the Langevin thermostat (collision frequency of 1 ps^{-1}) and Monte Carlo barostat (with 100 fs between attempts to adjust the system volume), respectively. Nonbonded interactions were truncated at 10 Å, and long-range electrostatics were treated using the particle mesh Ewald method.[147]

4.3.4 P53 peptide

To sample alternate conformations of the p53 peptide (residues 17–29), a single equilibrium WE simulation was run using each of the two binning schemes and a two-dimensional progress coordinate that consisted of (i) the heavy-atom RMSD of the peptide from its

MDM2-bound, α-helical conformation, and (ii) the end-to-end distance of the peptide. For both binning schemes, the WE simulations were run using a fixed time interval τ of 50 ps for each iteration and a target number of 4 trajectories/bin. A total simulation time of 2.0 μs was generated for each binning scheme (338 and 200 WE iterations for the MAB and manual binning schemes, respectively). The MAB scheme used a maximum of 44 bins while the manual binning scheme used a maximum of 294 bins that were evenly spaced between an RMSD of 0 and 20 Å and end-to-end distance of 0–26 Å. For the MAB scheme, no other limits were specified for the replication of trajectories.

Dynamics were propagated using the AMBER18 software package[144] with the Amber ff14SBonlysc force field[148] and a generalized Born implicit solvent model (GBneck2 and mbondi3 intrinsic radii).[149] Simulations were started from an energy-minimized conformation of the peptide that was based on the crystal structure of the MDM2-p53 peptide complex (PDB code: 1YCR).[150] The temperature was maintained at 298 K using the Langevin thermostat and a collision frequency of 80 ps^{-1} for water-like viscosity.

4.3.5 Standard simulations

A total of 5 independent 1 μs standard MD simulations were run for the Na^+/Cl^-, and a single 2 μs simulation was carried out for the p53 peptide. Details of dynamics propagation and starting structures for these simulations are the same as those described above for the WE simulations.

4.3.6 Calculation of rate constants

The association rate constant k^{RED} for the Na^+/Cl^- system was directly calculated from the WE simulation using the rate event duration (RED) scheme:[54]

$$k^{RED} = \frac{\hat{F}(t_{max})}{C} \tag{4}$$

where $\hat{F}(t_{max})$ is the cumulative probability of transitions from the unassociated state to the associated state up to the maximum (longest) trajectory length t_{max} of the steady-state WE simulation; C is a correction factor equal to $\int_0^{t_{max}} \int_0^t \tilde{h}(t_b) dt_b dt$, which incorporates the

transient phase of the time evolution of the rate-constant estimate using the distribution $\tilde{h}(t_b)$ of event durations (barrier crossing times) that are less than or equal to tmax.

Uncertainties in the rate constants represent 95% confidence intervals, which is the standard error of the mean for each system multiplied by a critical value. For a large sample size (> 30), this critical value would be 1.96, as obtained from a z-test. However, for the calculations in this study, which involve a smaller sample size (< 30), critical values for determining the confidence interval at 95% were obtained from a t test using the appropriate number of degrees of freedom (number of independent simulations minus 1) for each system.

4.3.7 Estimation of WE efficiency in computing rate constants

The efficiency S_k of WE simulations in computing the association rate constant for the Na^+/Cl^- system was estimated using the following:[26]

$$S_k = \frac{t_{BF}}{t_{WE}} \left(\frac{\Delta k_{BF}^2}{\Delta k_{WE}^2} \right) \tag{5}$$

where $t_{BF/WE}$ is the aggregate simulation time for standard "brute force" (BF) simulation or WE simulation, respectively, and $\Delta k_{BF/WE}$ is the relative error in the rate constants for the corresponding simulations where the absolute error is represented by the 95% confidence interval. Thus, the efficiency of the WE simulation in calculating the rate constant is determined by taking the ratio of aggregate times for the WE and brute force simulations that would be required to estimate the rate constant with the same relative error, with larger values of S_k corresponding to a more efficient simulation. The relative error in the rate constant is assumed to be inversely proportional to the simulation time.

4.4 RESULTS

We demonstrate the power of our minimal adaptive binning (MAB) scheme compared to fixed, manual binning schemes in the weighted ensemble (WE) sampling of rare events. We applied the MAB scheme to the following processes, listed in order of increasing complexity:

(i) transitions between stable states in a double-well toy potential, (ii) molecular associations of the Na^+/Cl^- ions, and (iii) conformational sampling of a peptide fragment of tumor suppressor p53.

4.4.1 Simulations with a double-well toy potential

To test how effectively the MAB scheme performs for a process with a large free energy barrier, we focused on a double-well toy potential in which two equally stable states are separated by a 34 kT (20 kcal/mol at room temperature) barrier. WE simulations with a manual binning scheme (see Figure 18A for bin positions) yielded no pathways from the initial state at $X = 0.5$ to the target state at $X = 2.5$ after 12 000 WE iterations, occupying only 14% of the fixed bins (Figure 18B). In contrast, the MAB scheme generated pathways to the target state in 60 WE iterations, occupying 99% of the bins (Figure 18C). This greater efficiency is due to the identification of bottleneck regions right before the trajectory weights have sharply fallen. These regions correspond to the upward slope of the free energy barrier (Figure 18D) where trajectories that are about to re-cross the high barrier get "stuck" due to the boundary trajectories maintaining their positions far apart from each other.

4.4.2 Simulations of the Na^+/Cl^- association process

To determine the effectiveness of the MAB scheme for a relatively fast process (ns time scale), we simulated the Na^+/Cl^- association process in explicit solvent (Figure 19A). Given the modest free energy barrier for this process,[151] it was feasible to compute association rate constants using standard simulations, providing validation of the rate constants computed using WE simulations and manual/MAB binning schemes.

Table 1 shows the computed rate constants, the efficiencies relative to standard simulations, and the number of successful pathways for WE simulations with the MAB and manual binning schemes. Regardless of the binning scheme, the WE simulations yield rate constants that are within the error of the value from standard simulations (see also Figure 21 in the SI). Given the modest free energy barrier for this process, there is only a 1.6-fold gain in efficiency for the MAB scheme relative to the manual binning scheme (see Figure 19B for

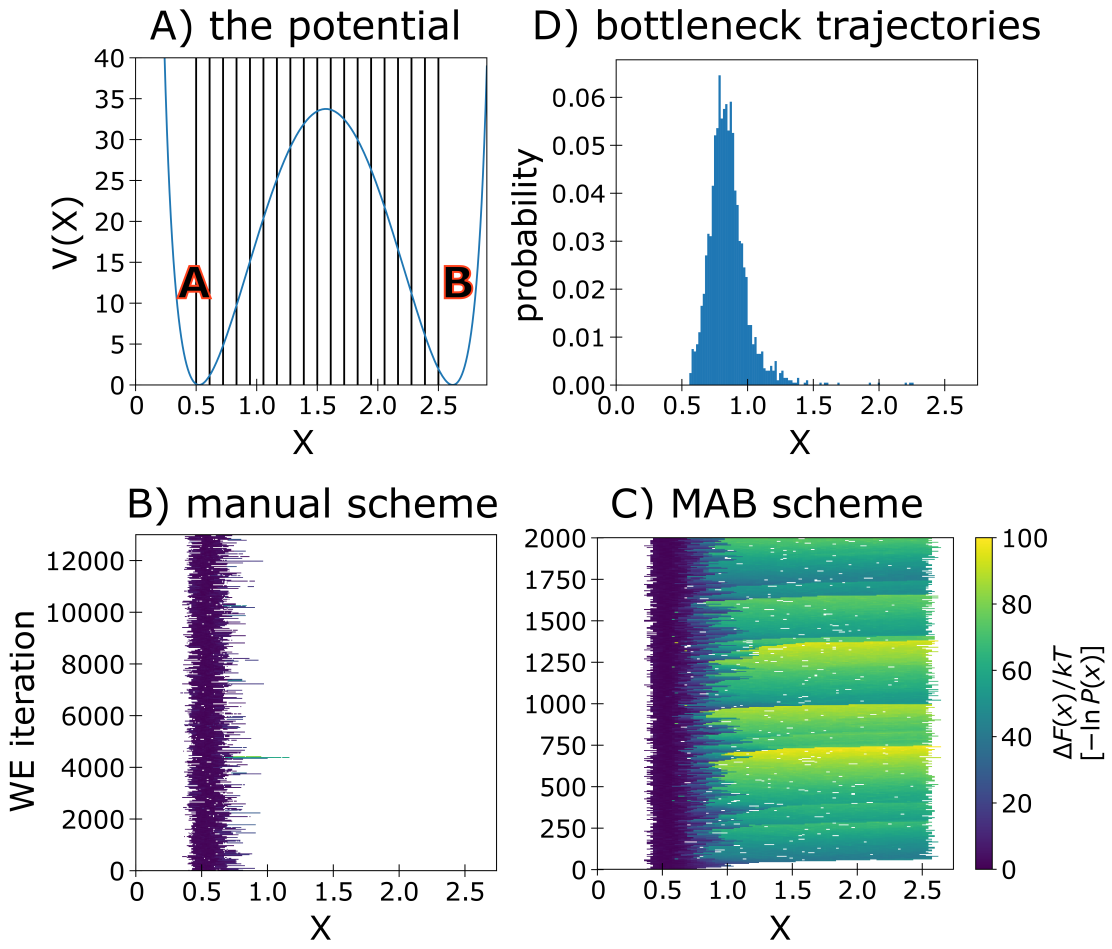

Figure 18: Transitions between stable states of a double-well toy potential. (A) The double-well potential in units of K_BT and manual binning scheme with 20 bins indicated by vertical lines. (B) Probability distribution as a function of the WE iteration for a manual binning scheme. (C) Probability distribution as a function of the WE iteration for the MAB scheme. (D) Probability distribution of bottleneck walkers identified by the MAB scheme using 20 bins at any given time.

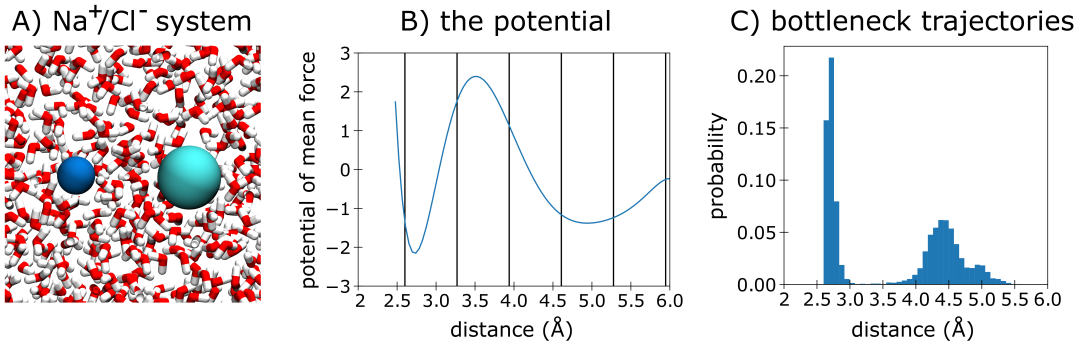

Figure 19: Molecular association of the Na^+ and Cl^- ions. (A) The Na^+/Cl^- system in explicit solvent. (B) Potential of mean force for the Na^+/Cl^- association process in units of K_BT with bin positions for the manual scheme indicated by vertical lines. (C) Probability distribution of the positions of bottleneck trajectories tagged by the MAB scheme along the progress coordinate.

bin positions). The MAB scheme also resulted in a 2-fold gain in the number of successful pathways related to the manual binning scheme. Consistent with our results for the double-well toy potential, the majority (60%) of the bottleneck trajectories occupied bins along the upward slope of the free energy barrier; the remaining bottleneck trajectories (40%) occupied bins located beyond the target state and are only present due to running these simulations without a recycling condition (Figure 19C).

4.4.3 Conformational sampling of the p53 peptide

Given that the WE strategy has previously enhanced the conformational sampling of various biomolecules,[44,152] we applied the MAB scheme to the conformational sampling of a p53 peptide (Figure 20A). As expected, WE simulations using either the MAB scheme or the previously reported manual binning scheme[55] yielded greater coverage of configurational space than standard simulations with the same total computing time (Figure 20B,C). The MAB scheme placed bins more efficiently than the manual binning scheme, resulting in the occupation of 66% of the specified bins (29 out of 44 bins) compared to only 17% (50 out

simulation type	k ($M^{-1}s^{-1}$)	simulation time (μs)	S_k	# successful pathways
WE w/ MAB	$(3.9 \pm 0.3) \times 10^{10}$	1.0	5.1	2498
WE w/ manual bins	$(4.1 \pm 0.4) \times 10^{10}$	1.0	3.1	1226

Table 1: Computed Rate Constants for the Na^+/Cl^- Association Process Using WE Simulations with the MAB Scheme and Manual Binning Scheme. Uncertainties represent 95% confidence intervals determined by a t-test. For each binning scheme, five WE simulations were run with each yielding 0.2 μs of total simulation time. The efficiency S_k of WE relative to standard simulations was calculated as described in Methods. For reference, the computed rate constant based on five 1 μs standard simulations was $(3.9 \pm 0.3) \times 10^{10} M^{-1} s^{-1}$.

of 294 bins) for the manual binning scheme. Notably, the MAB scheme sampled a "horn shaped" region of the probability distribution which consists of primarily low-probability trajectories. This region was not sampled when using the manual binning scheme (or standard simulations) and includes a more extensive set of left-handed helices as well as PPII conformations, which have previously been identified as the dominant state by UV resonance Raman spectroscopy.[153]

4.5 DISCUSSION

On average, the minimal adaptive binning (MAB) scheme replicates more trajectories in steeper regions of the free energy landscape. As mentioned above for the double-well toy potential and Na^+/Cl^- system (Figure 18D and Figure 19C, respectively), our MAB scheme identified bottleneck regions as the upward slopes of the free energy barriers, immediately before the barrier peaks (transition states). In contrast, a recently published variance-reduction strategy, which also seeks the most optimal placement of bins in weighted ensemble simulations, has identified such regions as the vicinity of the transition states; i.e., finer binning in transition-state regions yields the lowest variance in an observable of interest.[34] This slight

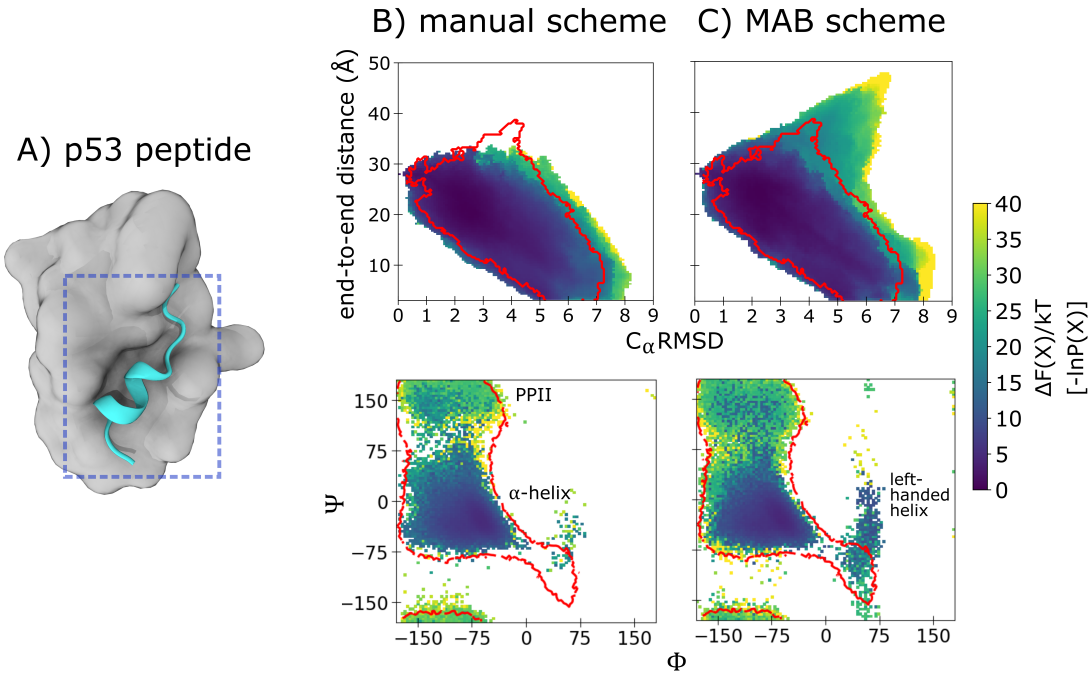

Figure 20: Conformational sampling of a p53 peptide. Probability distributions as a function of the two-dimensional WE progress coordinate from simulations of the p53 peptide (residues 19–23): (A) starting conformation of the p53 peptide (cyan) for the WE simulation, extracted from the crystal structure[150] of its complex with the MDM2 protein (gray); (B) WE simulations using the manual binning scheme; and (C) WE simulations using the MAB scheme. Also shown are Ramachandran plots of the p53 peptide for the manual binning scheme and MAB scheme. Regions sampled by standard simulations are delineated in red.

difference in the locations of the bottleneck regions is likely due to the fact that the goals of the MAB scheme and variance-reduction strategy are different. The MAB scheme aims to surmount free energy barriers whereas the variance-reduction strategy aims to minimize the variance of an observable of interest.[34] Our results suggest that the MAB scheme would be particularly effective in surmounting large barriers when used with a "committor" coordinate,[107,154–157] which tracks the probability that a given system configuration will commit to the target state before returning to the initial state: a nearly optimal, one-dimensional progress coordinate for the rare-event process of interest.[158,159]

The MAB scheme identifies bottleneck trajectories using an objective function that is easily extensible to track any arbitrary value. In its current form, the objective function tracks the probability of the trajectory in question along with the cumulative probability of all trajectories that are further along the progress coordinate of interest, all on a logarithmic scale. This requirement of having some trajectories that have surpassed the trajectory of interest makes it unlikely for identified bottleneck trajectories to be ones that have departed along orthogonal degrees of freedom (i.e., differentiating between a leading trajectory and a bottleneck trajectory). Alternatively, users may modify the objective function to track the average or maximum probability among trajectories that have surpassed the trajectory in question.

By identifying appropriate bin positions for use with other key WE parameters (i.e., resampling interval τ and target number of trajectories per bin), the MAB scheme greatly reduces the need for trial-and-error selection of these parameters, which are highly coupled to one another. To maximize the "statistical ratcheting" effect of the WE strategy, we recommend using the shortest possible τ-value that maintains high scaling of the WESTPA software with the number of GPUs (or CPU cores) on a given computing resource.[55] Furthermore, our results indicate that a target number of either 4 or 5 trajectories per bin is sufficient to surmount large barriers or greatly enhance conformational sampling. If the goal is to simply generate pathways to a target state of interest, we recommend applying the MAB scheme with the minimal number of bins (e.g., 5, with separate bins for the two boundary trajectories and one bottleneck trajectory in the direction of interest, and two bins between the boundary trajectories) to reduce the total computational time required for the simula-

tion. On the basis of our tests with the double-well toy potential, the number of bins does not affect the ability of trajectories to reach the target state using the same total computing time (Figure 22 in the SI). However, if a greater diversity of trajectories or a rate-constant estimate is desired, we recommend applying the MAB scheme with a larger number of bins (15–20) to replicate more trajectories and yield more even coverage of configurational space along the progress coordinate.

4.6 CONCLUSIONS

To streamline the execution of weighted ensemble (WE) simulations, we developed a minimal adaptive binning (MAB) scheme for automatically adjusting the positions of bins along a progress coordinate. Our scheme adjusts bin positions according to the positions of trailing, leading, and "bottleneck" trajectories at the current WE iteration. Despite its simplicity, the MAB scheme results in greater sampling of configurational space relative to manual binning schemes for all three benchmark processes of this study: (i) transitions between states of a double-well toy potential; (ii) Na^+/Cl^- association; and (iii) conformational sampling of a peptide fragment of the tumor suppressor p53. Due to the earlier identification of bottlenecks along the progress coordinate, the MAB scheme enables the simulation of pathways for otherwise prohibitive large-barrier processes and a greater diversity of pathways when desired, all with dramatically fewer bins than manual binning schemes. As demonstrated previously, the efficiency of WE simulations relative to standard simulations is even greater for slower processes, increasing exponentially with the effective free energy barrier when the progress coordinate is appropriately binned.[89]

We recommend the MAB scheme as a general, minimal scheme for automating the placement of bins in combination with any rare-event sampling strategy that requires a progress coordinate. A particularly effective application of the scheme could be its use with a committor coordinate, which is a nearly optimal, one-dimensional progress coordinate for ordering states along simulated pathways for a process of interest according to a "kinetic ruler." Regardless, the MAB scheme provides an ideal launching point for future developments of more

sophisticated binning strategies by yielding initial, promising bins for further optimization.

4.7 ACKNOWLEDGEMENTS

This work was supported by the NIH (1R01GM115805-01) and NSF (CHE-1807301) to L.T.C., and the University of Pittsburgh to P.A.T. (Honors College Brackenridge Undergraduate Research Fellowship and Dietrich School of Arts and Sciences Summer Undergraduate Research Award) and A.T.B (Arts and Sciences Graduate Fellowship). Computational resources were provided by the University of Pittsburgh's Center for Research Computing. We thank Daniel Zuckerman (OHSU) for insightful discussions.

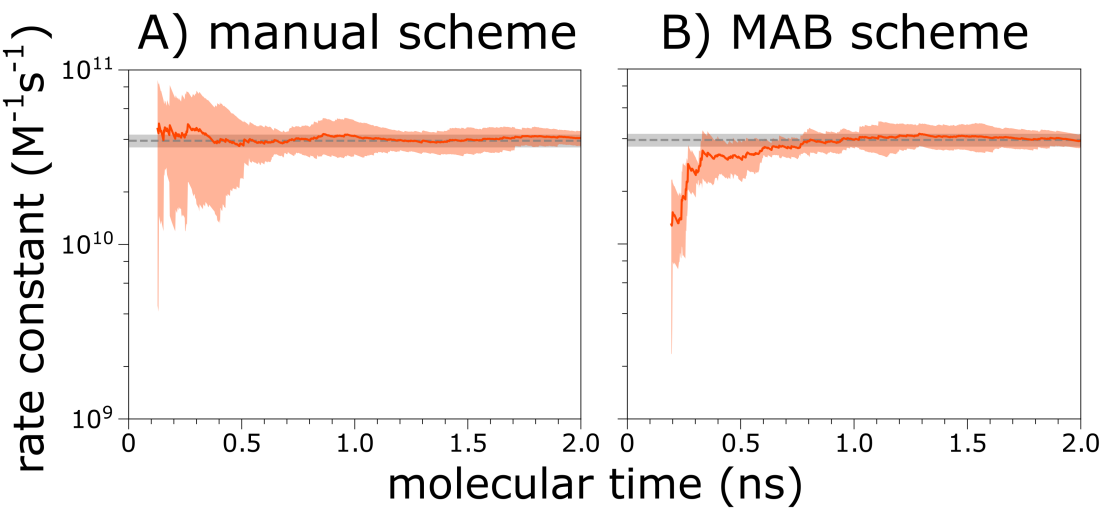

Figure 21: Computed rate constants for the molecular association process involving Na^+ and Cl^- ions in explicit solvent as a function of molecular time $N\tau$ where N is the number of WE iterations and τ is the fixed time interval for WE resampling. The rate constant from standard simulations is shown with the uncertainty as a grey shaded line. Results are shown for A) the manual binning scheme and B) the MAB scheme.

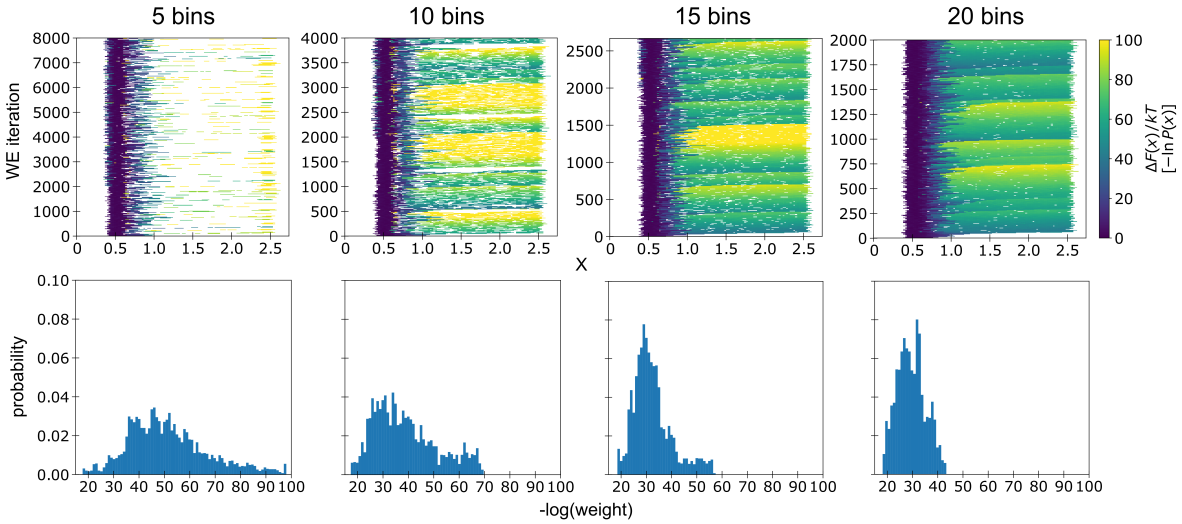

Figure 22: Probability distributions as a function of the progress coordinate X from WE simulations with the double-well potential and MAB scheme using different numbers of bins and the same total computing time (200,000 δt). Distributions of successful trajectories by their corresponding weights (trajectory weights shown on the logscale) are shown in the second row.

5.0 LPATH: A SEMI-AUTOMATED PYTHON TOOL FOR CLUSTERING MOLECULAR PATHWAYS

5.1 INTRODUCTION

Pathways generated by physics-based simulations are the most direct observations of a molecular mechanism. Furthermore, the ensemble of simulated pathways often involves multiple routes through phase space. The identification of these routes is challenging given the diversity and variable lengths of pathways. In addition, the massive amount of trajectory data generated by path sampling strategies (e.g., tens of terabytes) can be unwieldy to analyze.

Current methods for pathway analysis involve two main steps: (1) projecting pathways onto a low-dimensional phase space and (2) clustering pathways based on a similarity metric. The pathway similarity analysis (PSA) method[50] is a "bottom-up" approach that projects pathways onto a low-dimensional phase space consisting of the pairwise root-mean-squared deviation of sampled conformations and then clusters the pathways based on pairwise Hausdorff[160] or Fréchet[161] geometric pathway distances. The pathway histogram analysis of trajectories (PHAT) method[51] presents an approach to quantify pathway diversity via populations of discrete classes. One way that PHAT presents for generating pathway classes is a "bottom-up" approach similar to PSA in which "set similarities" are used to generate similarity scores between pathways (though any similarity metric, such as the geometric distances used in PSA, can be used), which are clustered into distinct classes using Voronoi clustering. Another approach is "top-down" where fundamental sequences, or discrete-state trajectories with loops removed, are calculated from a discrete model (such as a Markov state model) and used to sort pathways.

In this application note, we present the Linguistics Pathway Analysis of Trajectories with Hierarchical clustering (LPATH) tool, which uses a bottom-up approach to clustering pathways based on a similarity metric that is commonly used in computational linguistics for plagiarism detection software.[162] Similar to both the PSA and PHAT methods, LPATH

generates a histogram of pathway classes. Our projection of pathways onto one-dimensional text strings greatly accelerates the hierarchical clustering of pathways and subsequent analysis of path ensembles. While the LPATH tool is designed for simulations run using the weighted ensemble (WE) path sampling method,[18,26] as implemented in the WESTPA 2.0 software package,[48] this tool can also be applied to conventional MD (cMD) simulations. We demonstrate the effectiveness of our LPATH tool in pathway analysis by focusing on a benchmark application involving simulated pathways for a conformational transition of an alanine dipeptide.

5.2 WORKFLOW APPLICATION TO ALANINE DIPEPTIDE

The workflow for the LPATH tool is presented in Figure 23 and involves four steps: 1) discretization, 2) extraction, 3) matching and 4) clustering. An additional step (plotting) is available as part of the tool but not discussed here. Further details of each step are provided below in the context of our benchmark application involving the conformational sampling of alanine dipeptide in generalized Born implicit solvent,[163] i.e., the transition from the $C7_{eq}$ to $C7_{ax}$ conformational states (Figure 24).

5.2.1 System details

In this benchmark application, we demonstrate the use of the LPATH tool for analyzing 20 independent cMD simulations and 5 independent WE simulations–all of which were discretized based on ϕ/ψ angles. The total simulation time of the WE simulations is 14.6 μs and 60 μs for the cMD simulations. All simulations were run with the Amber 22 package.[164] A timestep of 4 fs was enabled by using a hydrogen mass repartitioning (HMR) scheme and coordinates were saved every 1 ps for analysis. The cMD simulations generated 122 successful pathways and the WE simulations generated 83 successful pathways. For optimal pathway clustering analysis results, we recommend the generation of at least 50 pathways.

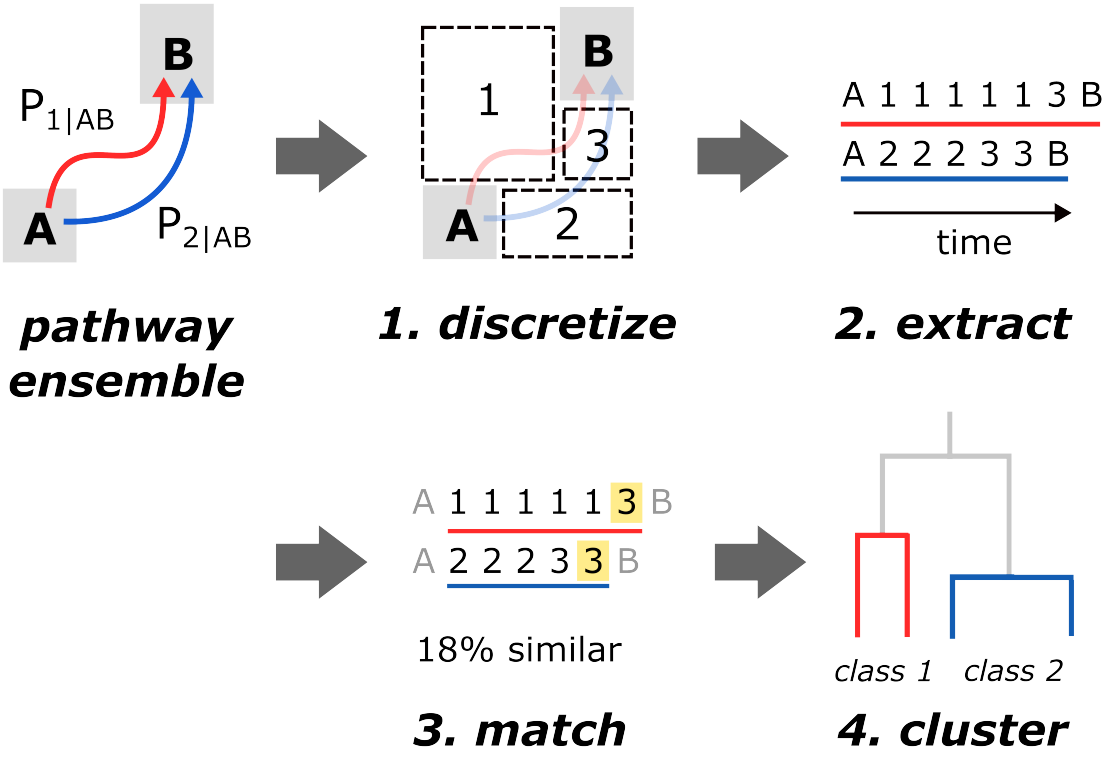

Figure 23: Workflow of the LPATH tool. The workflow consists of five steps that are executed using four command-line options, with matching and clustering both being a part of the "match" option ('lpath match').

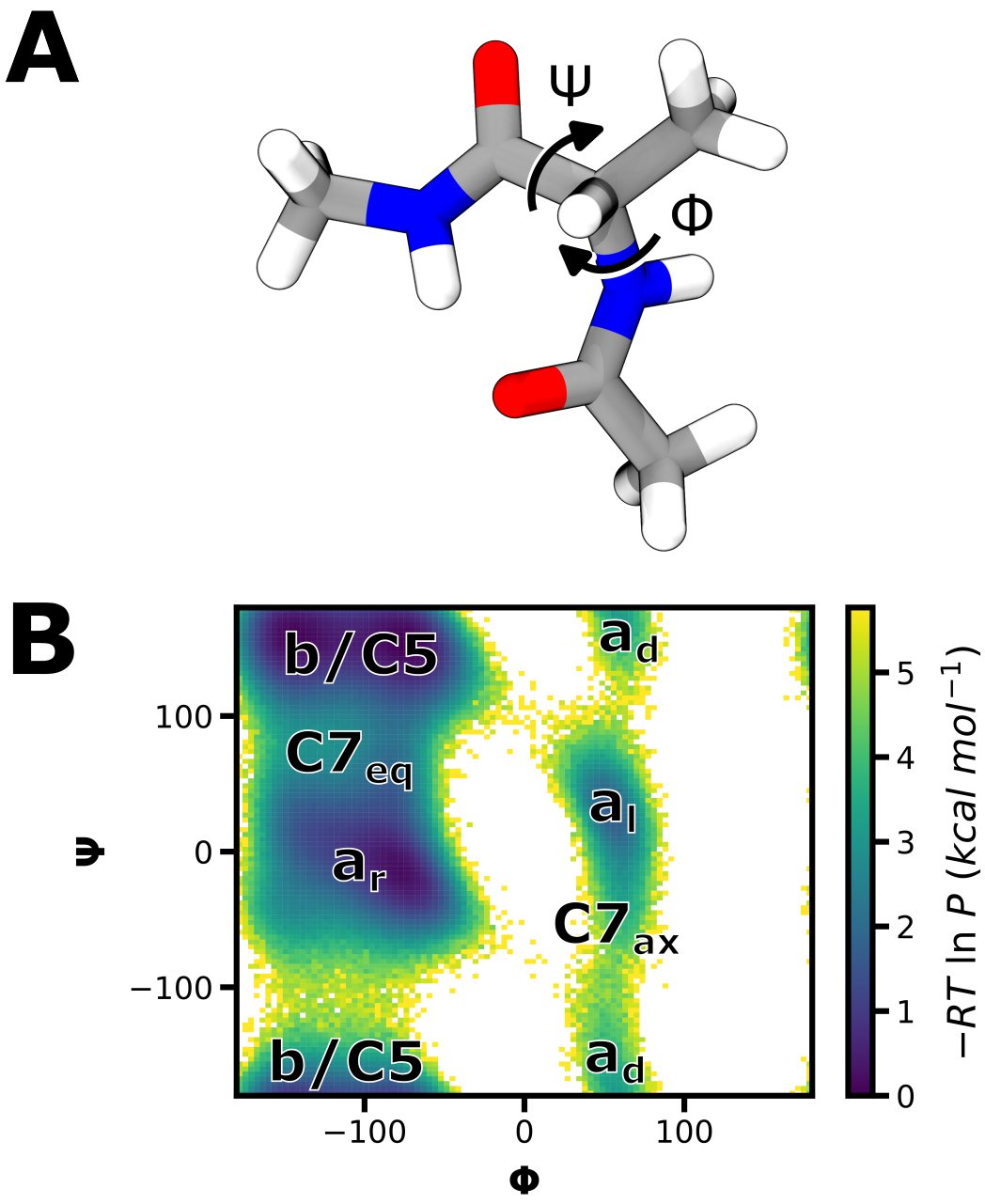

Figure 24: The alanine dipeptide benchmark system. Alanine dipeptide, capped with acetyl and N-methyl groups is shown in A. Rotation around ϕ and ψ angles results in the probability distribution of conformations shown in B. In this workflow application, the transition between $C7_{eq}$ and $C7_{ax}$ is explored, which involves crossing a high energy barrier in the ϕ dimension ($\sim$5 kcal/mol).

5.2.2 Analyzing multiple independent simulations

Whenever possible, we strongly recommend generating multiple independent simulations to assess the variation between runs. Prior to applying the five steps of the LPATH workflow, multiple, independent WE simulations should be first combined with WESTPA's 'w_multi_west' tool with the '–ibstates' flag and multiple independent cMD simulations should be concatenated.

5.2.3 Step 1: discretize phase space

The goal of this step is to separate regions of phase-space into discrete states. If the pathway ensemble was generated by WE simulation, one has the option to discretize based on the WE trajectory segment IDs, unique number identifiers assigned to WE segments at each iteration which serve as general "states." When defining states based on phase-space, we recommend using at least three states in addition to the start and end states. The defined discrete states can consist of metastable intermediates or transient states.

The main choice in this first step is whether to use trajectory segment IDs for state definitions or to use phase-space discretization. As an example of using phase-space discretization, we focus on alanine dipeptide simulations and define 4 intermediate states in terms of ϕ/ψ conformational angles that are labeled in Figure 24. Users should avoid using trajectory segment IDs as a metric if the simulation data is generated from multiple, independent WE simulations, since such IDs are not directly comparable between independent simulations. The use of segment IDs for discretization is beneficial, however, in cases where it is difficult to discretize the phase space based on one or two features. One could also cluster the WE data in phase space using an algorithm such as k-means to automatically identify general states in interest. For a single WE simulation of the alanine dipeptide system, the choice of whether to use discretized states or segment IDs did not significantly influence the conclusions of the pathway analysis (Figure 28 in the SI).

The 'lpath.discretize' function uses WESTPA's 'w_assign' tool to assign segments to states according to a scheme defined in the user's west.cfg file. The resulting assign.h5 file is then used in the subsequent step of successful pathway extraction. To discretize a cMD

trajectory, LPATH requires a text file of molecular features for each trajectory frame and a custom function for assigning states based on these features. The output for this step will be a states.npy file for the next step of successful pathway extraction.

5.2.4 Step 2: extract successful pathways

The goal of this step is to identify all successful pathways (pathways that have reached a state B after being initialized in some state A) and save them in a convenient format for further analysis. Pathway extraction is slightly different between WE and cMD simulations. For WE simulations, a start and end state are provided based on the discretization performed in step 1. Then, all pathways that travel from the start to end states are traced back and added to a list. Additional information about these pathways is also saved such as the weights of trajectory segments and, optionally, progress coordinate or auxiliary (additional) simulation data. The weights can be used later to generate a histogram of pathway class probability and the progress coordinate data for advanced plotting. For cMD simulations, a start and end state are provided based on the discretization done in step 1 and pathways connecting the two states are then extracted and saved to a list. A few key choices must be made in the extraction step, as presented below.

There are two main choices in the pathway extraction step. The first, and perhaps the most important choice is whether to extract–for each pathway–the entire successful pathway (including time spent in the start state) or just the transition portion of the pathway that does not include any "dwell time" in the initial stable state. The first option, to keep the entire pathway ('–trace-basis'), results in longer pathways overall with many shared trajectory segments, as the point of divergence is likely to happen at the barrier between start and end states. Extracting the entire pathway can provide users with a more complete picture of their trajectory ensemble, including more general information about how pathway classes diverge. Note that the use of WE trajectory segment IDs requires the '–trace-basis' flag. On the other hand, extracting solely the transition portion of the pathways leads to overall shorter pathways but reveals more subtle points of differences between pathway classes. For a single WE simulation involving alanine dipeptide, we tested the extraction of the entire

pathway vs only the transition portion and found that both methods identified the same two overall pathway classes. However, the exclusion of pathway points in the start state revealed more minute differences in the pathways taking the bottom route into the $C7_{ax}$ state (Figure 29 in the SI). For most systems, we recommend analyzing only the transition portion of pathways.

The second choice relates to the resolution of points used when the pathways are extracted. The choice of resolution can be controlled through the "stride" option which operates slightly differently for WE vs cMD simulations. By default, analysis on WE simulations are done on conformations extracted at the WE resampling frequency τ. However, users may provide a value with the '–stride' option to dictate how many data points to to access at a sub-τ time resolution. For instance, a '–stride=10' on a WE simulation with τ=100 ps and output of 1 ps would yield conformations at every 10 ps. The resulting pathways are 10x longer than if one had used a τ-resolution for extraction. If a pathway happens to exit the start state or enter the target state mid-τ, only the sub-τ points after the exit or before the entry are kept. For cMD simulations, on the other hand, a stride of 10 would result in a coarser resolution for extracted pathways with 10x fewer points than the default resolution. Note that extracted pathways must consist of a minimum of 10 points for meaningful clustering of the pathways, and the number of points can be adjusted using the '–stride' parameter.

Some additional choices that could influence the clustering of pathways is the decision to impose a threshold for deleting pathways based on pathway length and the decision to combine states that experience many "back-and-forth" transitions. The LPATH tool will automatically alert users if pathways with fewer than 10 total frames exist in the pathway ensemble, due to the fact that shorter pathways could potentially be very fast pathways that skip over intermediate states, obscuring the final pathway class results. It is recommended to remove these shorter pathways by specifying a pathway length threshold with the '–exclude-min-length' parameter. Users may also consider higher thresholds in the case where the pathway ensemble consists of more than 25% pathways below the default 10 frame threshold, but should ensure that after pathway removal, at least 50 pathways remain for matching. The LPATH tool will also alert users to pathways that contain many repeating back-and-

forth transitions between two states. These back-and-forth movements can contribute a lot of noise to matching and result in pathways appearing more dissimilar to each other than they actually are. By re-running the discretization step and combining the states experiencing many back-and-forth transitions, this source of noise can be eliminated.

5.2.5 Step 3: match and cluster pathways into classes

The goal of this step is to compute pairwise pathway similarity and identify distinct pathway classes using hierarchical agglomerative (bottom-up) clustering. First, for each extracted pathway, we construct a text string sequence based on either the discretized states defined in step 1 or WE segment IDs. Next, we perform a pairwise matching of pathway string sequences using the Gestalt pattern matching algorithm[162] to generate a similarity score:

$$similarity_{AB} = \left(\frac{2 * length(longest\_common\_subsequence_{AB})}{length_A + length_B} \right) \tag{6}$$

The Gestalt pattern matching algorithm,[162] borrowed from the field of computational linguistics and commonly used in plagiarism detection software, is a key component of the LPATH tool's matching functionality. In Gestalt pattern matching, shared, non-consecutive substrings in each pair of pathways are normalized by the combined length of both pathways (Equation 6). The normalization step enables the comparison of text strings with different lengths, greatly facilitating analysis of the heterogeneous pathway ensemble. The similarity score returned from the Gestalt pattern matching equation ranges from 0 to 1 and represents the fraction of shared characters present between the two pathways. We convert this similarity score to a distance (necessary for the use of these scores in hierarchical clustering) by subtracting it from 1, such that a similarity of 0.9 will result in a similarity score between the pathways of 0.1. We then construct a pairwise distance matrix from the pathway similarities and perform hierarchical agglomerative ("bottom-up") clustering on the matrix using the Ward linkage.[165] An example of Gestalt pattern matching applied to two example strings is shown in Figure 25.

The main choice for this step is whether to match with the longest common subsequence

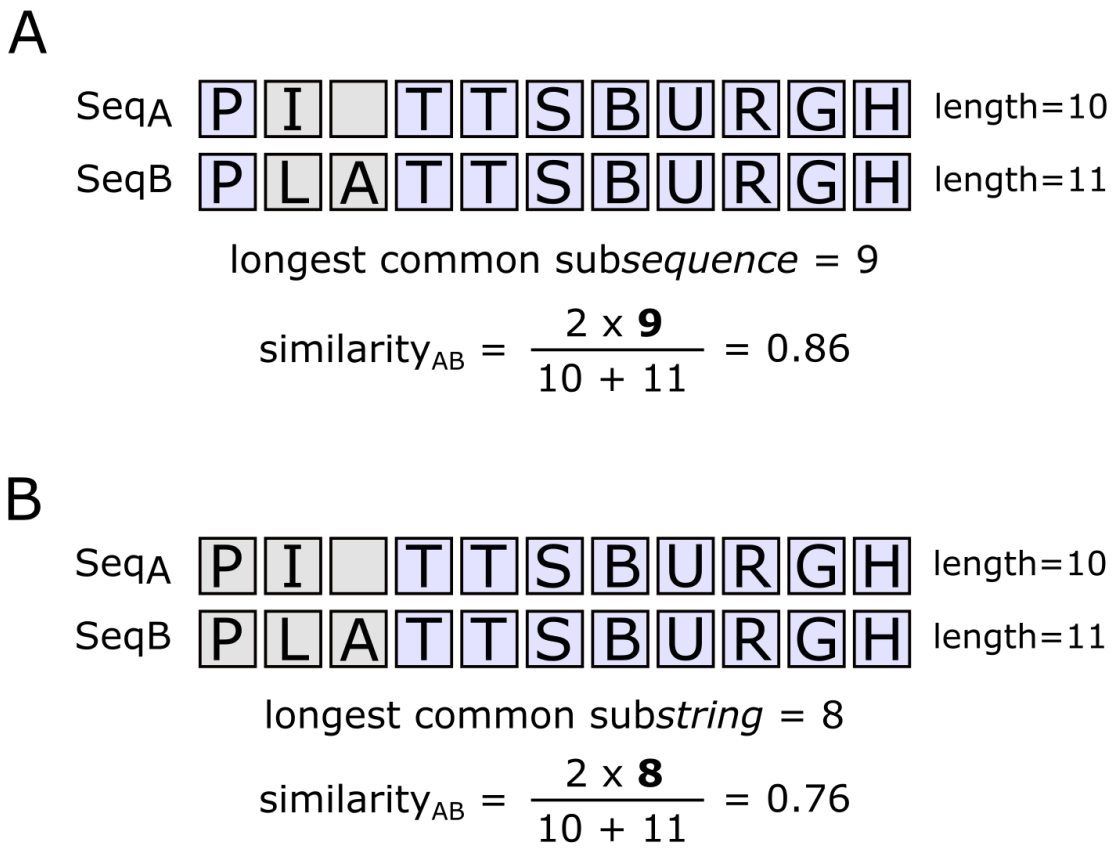

Figure 25: Two example strings being compared with the Gestalt pattern matching algorithm. The algorithm is applied using A) the longest common subsequence, and B) the longest common substring, which is only composed of continuous lengths of states.

or the longest common substring. The subsequence allows for non-continuous points of similarity and should result in higher matching scores overall only if the pathway re-converge after diverging, which is not expected to happen too much during a WE simulation. Note that for the use of trajectory segment IDs for assignment using WE simulation data, the matching can only be performed with the substring option. Varying the use of subsequence vs substring in the case of matching pathways for a single WE simulation of alanine dipeptide did not change the conclusions of the pathway analysis (Figure 30 in the SI).

An additional choice for matching is the ability to "condense" pathway strings before matching. A condensed pathway string eliminates consecutive repeats to provide a fundamental sequence of states visited by each pathway, regardless of how much time is spent in each state. Pathways that visit the same sequence of states, but spend different amounts of time in each state, can appear more dissimilar during matching than they actually are. The ability to condense pathway strings reduces the effect of pathway length on matching and focuses the pattern matching on the fundamental sequence of states visited, further eliminating noise from the pathway ensemble.

5.2.6 Step 4: plotting and interpreting the results

The goal of this step is to identify distinct pathway classes based on a dendrogram (tree diagram) of the clustering results. We first analyzed the set of 5 independent WE simulations in which only the transition portions of successful pathways were discretized in ϕ/ψ space. Figure 26A displays the dendrograms constructed using 32 successful pathways from a WE simulation of alanine dipeptide after removing pathways with fewer than 50 frames. Each vertical "leaf" in the dendrogram represents a pathway, which is connected to other leaves through horizontal "nodes". Dendrogram branches with nodes closer together in the vertical direction are more similar to each other. A horizontal line should be drawn to divide the dendrogram vertically between nodes with a maximum distance separation, representing the most distinct grouping of pathways into distinct classes. In the case of this WE simulation, we draw the horizontal line at y=1.5 to divide the dendrogram into 2 pathway classes. If it is unclear from the dendrogram how many pathway classes are present, it is recommended

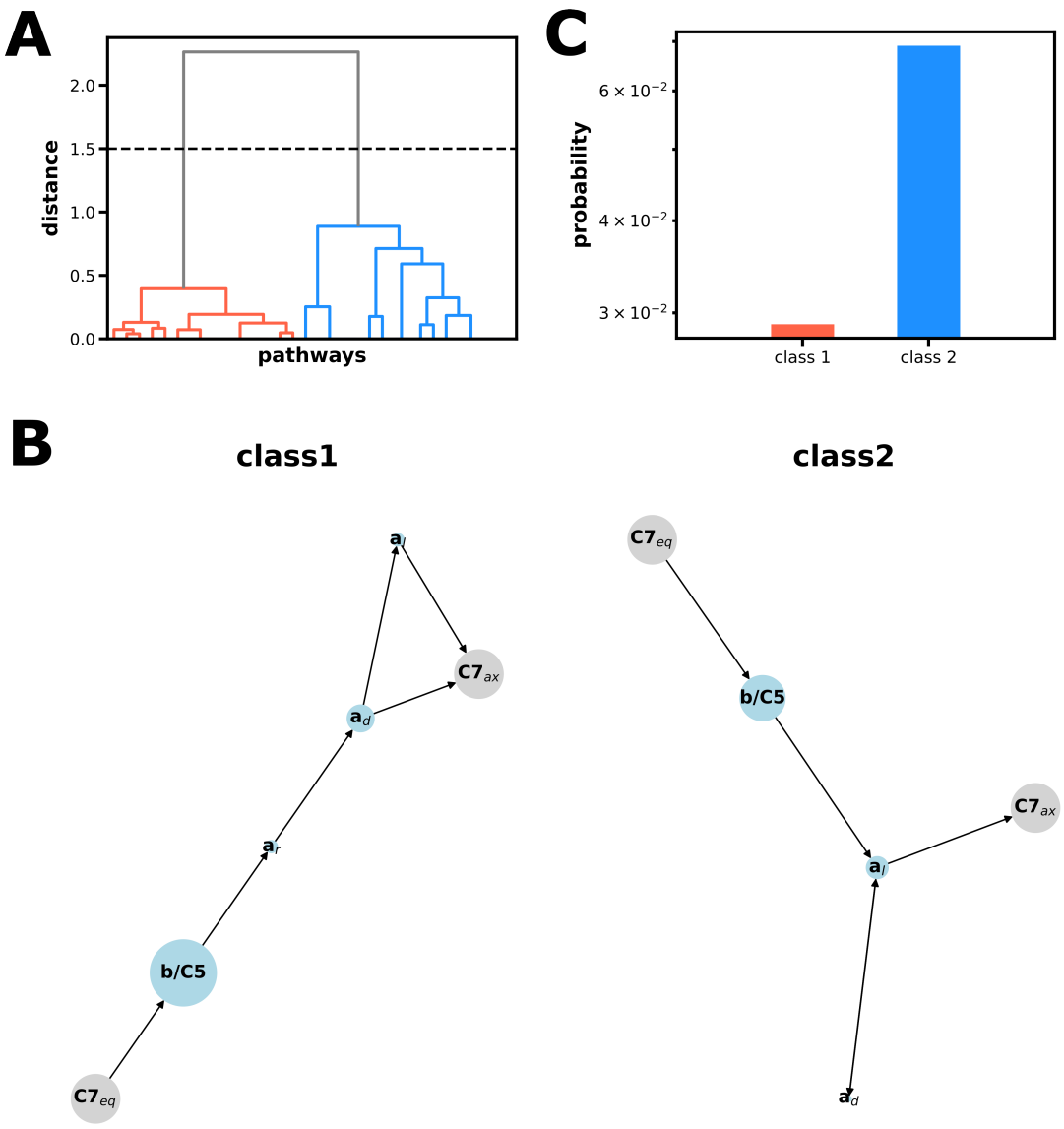

Figure 26: Pathway analysis of 5 independent WE simulations for the $C7_{eq}$ to $C7_{ax}$ transition for alanine dipeptide. A) Dendrograms of successful WE trajectories after removing pathways less than 50 frames (N=32) and matching condensed pathway strings reveal two distinct pathway classes. B) Directed network plots of all the condensed pathways in each pathway class reveal two main routes through phase space. The size of each node is scaled to match the relative time the pathways spend in each state. The upper route crosses when ψ=50 and the lower route when ψ=-100. C) Histograms of distinct pathway classes reveal that the "upper" route is more probable compared to the "lower" route.

to review each previous step in the LPATH workflow to make sure that 1) at least three states were used for discretization of phase space, 2) at least 50 pathways were extracted and 3) each pathway contains at least 10 points for matching. In addition, the decision of how many pathway classes are present should take the total number of pathways into account. To ensure good statistics, we recommend that each pathway class identified from the dendrogram contain at least 10 pathways, so a total of 30 pathways should not be divided into more than 3 pathway classes.

Based on the dendrograms alone, it is not clear how the pathway classes relate to the mechanism of transition from start to end states in the WE simulation. Figures such as those shown in Figure 26B, which can be generated with LPATH's plotting functionality, trace the entire set of pathways in each pathway cluster and plot a directed network of the condensed pathways, revealing the "fundamental" routes pathways take through phase space. Though not necessary, we the network plots we have shown here scale each node by the relative time the pathways spend in each state. Alanine dipeptide's transition from $C7_{eq}$ to $C7_{ax}$ appears to cross the main energy barrier in ϕ along two main "routes," one at ψ=50 (the upper route) and one at ψ=-100 (the lower route). The relative probability of alanine dipeptide pathways being in each pathway class is plotted in Figure 26C. Based on the pathway histograms the lower route is more probable compared to the upper route. The LPATH tool can also determine when during a WE simulation the pathway ensemble diverges into the chosen classes and incorporate this divergence point into the directed network plots.

We next analyzed 20 independent, 3 μs cMD simulations in which successful pathways excluding the time spent in the initial state were discretized in ϕ/ψ space. The cMD pathway dendrogram (Figure 27A) reveals two pathway classes from a total of 122 successful pathways after removing pathways with fewer than 50 frames. The two pathways classes traverse the same two routes through ϕ/ψ space (Figure 27B) discovered in the WE simulation, with the "upper" path still being the most probable compared to the "lower" path (Figure 27C).

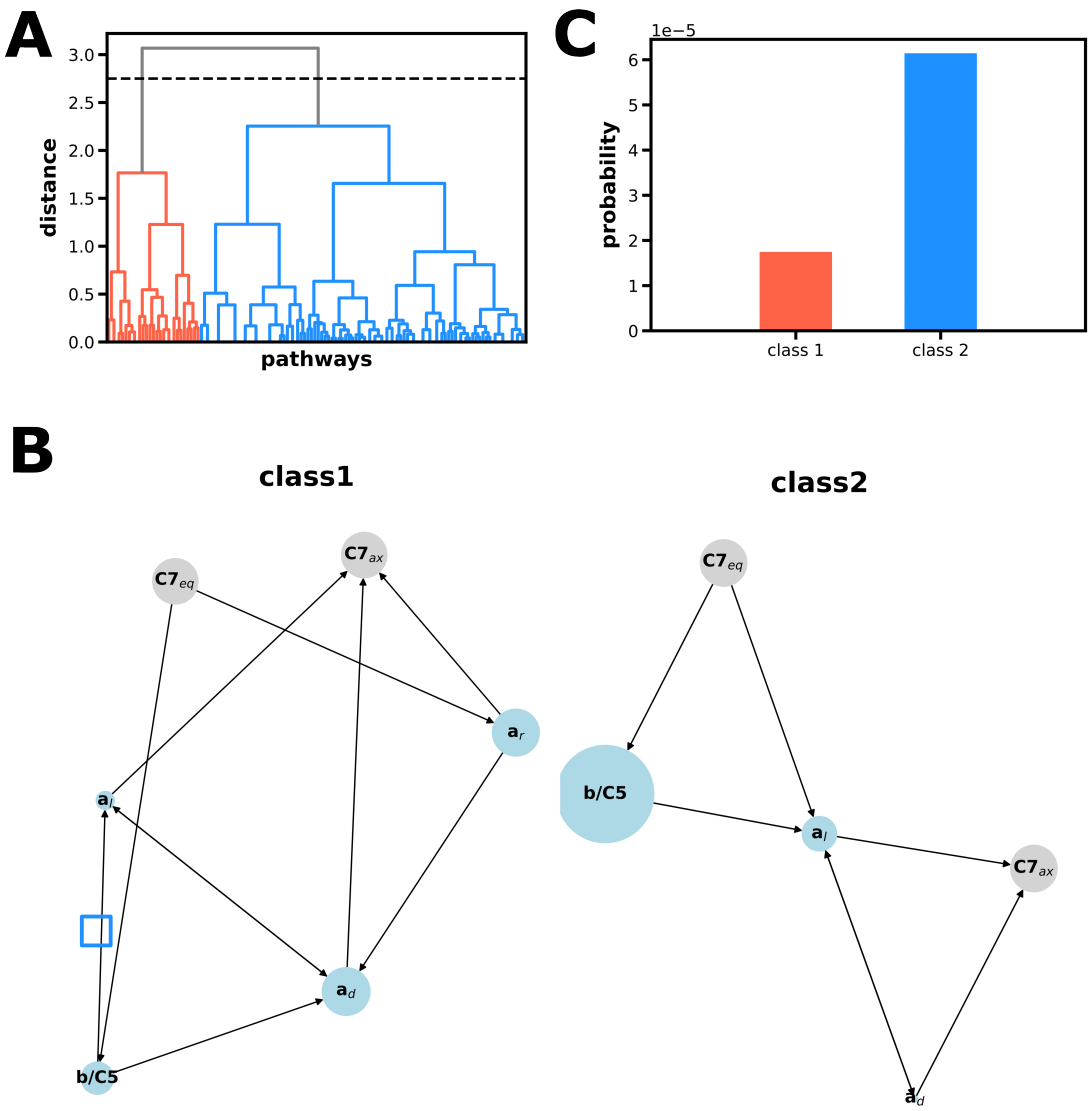

Figure 27: Pathway analysis of 20, 3 μs cMD simulations for the $C7_{eq}$ to $C7_{ax}$ transition for alanine dipeptide. A) Dendrograms of successful WE trajectories (N=122, using only the transition portion of the pathways after exiting the initial state) reveal three distinct pathway classes. B) Traces of all pathways in each pathway class reveal the same two main routes through phase space as were discovered from the WE simulations. The upper route crosses when ψ=50 and the lower route when ψ=-100. However, class 1 appears to contain patwhays that take both routes, suggesting that the dendrogram should be further subdivided into more than 2 classes. C) Histograms of distinct pathway classes reveal that the upper route is more probable compared to the lower routes.

5.3 CONCLUSIONS

The LPATH tool reveals distinct classes in the pathway ensemble by discretizing phase space, extracting successful pathways and clustering those pathways. The heart of the LPATH tool is the use of the Gestalt pattern matching algorithm from computational linguistics which clusters solely based on the matching of text strings representing the pathways. The generality of the pattern matching algorithm, which can match pathways of variable lengths, allows for the semi-automated workflow presented above. We demonstrate the effectiveness of the LPATH tool in analyzing two different pathway ensembles of alanine dipeptide, one from cMD simulations and one from a WE simulation. The distinct pathway classes identified by our tool revealed two distinct pathway routes from the $C7_{eq}$ to $C7_{ax}$ states. The interoperability of the LPATH tool allows for the implementation of alternate methods such as the geometric matching used in the PSA and the Voronoi clustering used in PHAT.

5.4 ACKNOWLEDGEMENTS

We thank Daniel Zuckerman (OHSU) for insightful discussions. We also thank Marion Silvestrini for providing the pattern matching example presented in Figure 25.

5.5 SUPPORTING INFORMATION

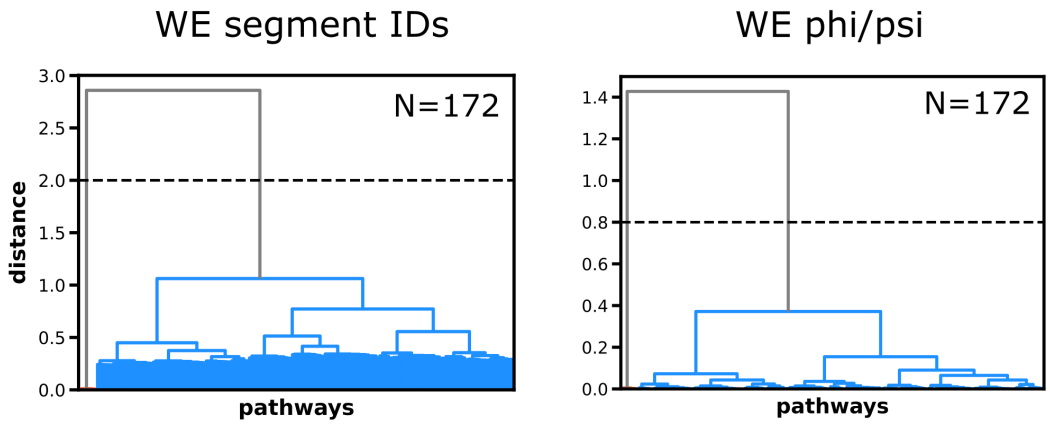

Figure 28: WE segment ID discretization vs ϕ/ψ discretization for a WE simulation of alanine dipeptide. For the alanine dipeptide WE simulation, there is no significant difference between the use of WE segment IDs vs ϕ/ψ angles for discretization. The pathways in each case were assigned to the exact same classes at all levels of the heirarchy. The dendrograms, however, are slightly different in terms of the inter-relatedness of the 172 total pathways.

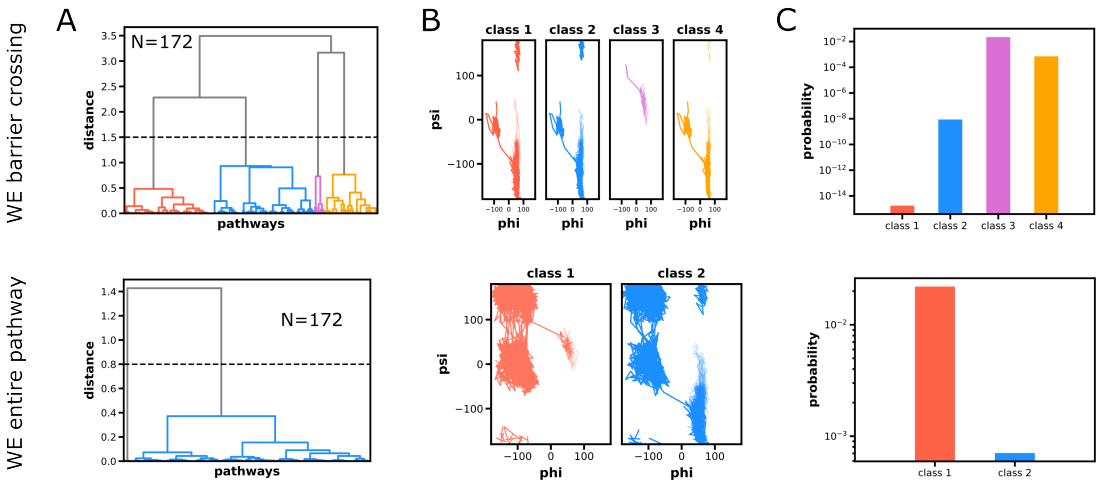

Figure 29: Use of the entire successful pathway vs only the transition portion for a WE simulation of alanine dipeptide. For the alanine dipeptide WE simulation, there is a significant difference between the use of just the transition portion of the pathways (left column) vs the entire pathways (right column) for matching. In A), dendrograms reveal four pathway classes in the case of using just the barrier crossing portions of the successful pathways but only two when using the entire pathways. In B), traces of all pathways in each successful pathway class reveal two main routes through phase space. The upper route crosses when ψ=50 and the lower route when ψ=-100. When just the barrier crossing portions are used for matching, classes 1, 2 and 4 describe the lower route but are clustered into separate classes, likely from subtle differences in time spent sampling the alpha R region. In C) histograms of pathway probabilities reveal that the upper route is more probable compared to the lower route.

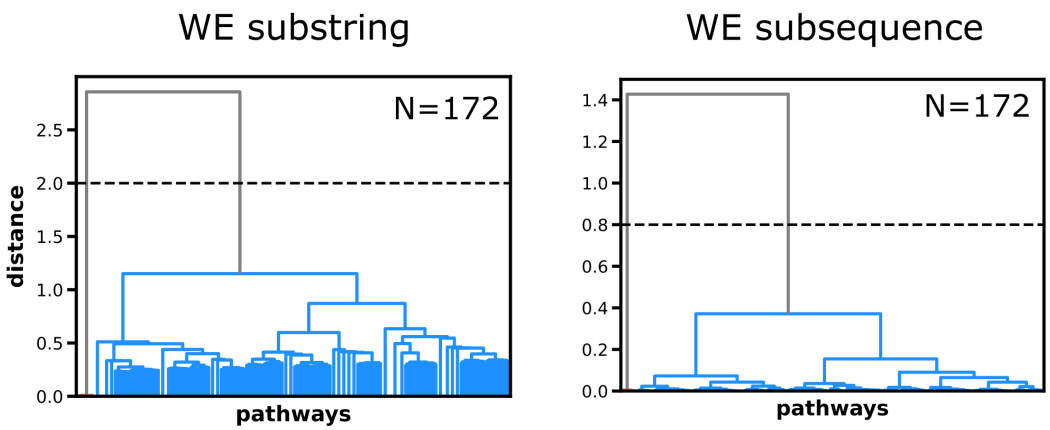

Figure 30: Substring vs subsequence matching for a WE simulation of alanine dipeptide. For the alanine dipeptide WE simulation, there is no significant difference between the use of substring vs subsequence for matching. The pathways in each case were assigned to the exact same classes. The dendrograms, however, are slightly different in terms of the inter-relatedness of the 172 total pathways.

6.0 CONCLUSIONS AND FUTURE DIRECTIONS

In the field of path sampling, large system sizes (greater than one million atoms), long timescales (beyond ms) and/or more detailed models (e.g., polarizable[11,12] or hybrid QM/MM[13–15] models) remain challenges to sampling many rare events. In this thesis, I began in Chapter 1 by motivating the need for path sampling strategies to simulate rare events and introducing the weighted ensemble (WE) strategy, which focuses computing power on functional transitions rather than solely sampling stable states. In Chapter 2, I then presented a perspective on large, biological switches, such as the SARS-CoV-2 spike protein, that undergo their respective transitions on timescales up to and beyond seconds, highlighting the need for innovation to the WE strategy in order to generate transition pathways and calculate rate constants for these ambitious processes. Next, in Chapters 3, I described how I contributed to the development of the WESTPA 2.0 software package for WE simulations, which provides high-performance implementations of new methodologies I present in the subsequent chapters. In Chapter 4, I introduced a minimal adaptive binning scheme for WE simulations, followed in Chapter 5 by introducing LPATH, a general, semi-automated pathway analysis tool for identifying distinct pathway classes from any MD pathway ensemble. The innovations described in this thesis have made WE simulations more efficient, effective and user-friendly.

However, despite the promising advances discussed above, I believe the WE strategy has not yet reached its full potential. For some biological processes, such as protein-ligand unbinding, especially in the case where the ligand is highly charged, it is challenging for WE simulations to generate pathways and out of reach to estimate rate constants. One particular ligand-unbinding process, which involves a charged ADP molecule unbinding from a motor protein receptor called Eg5 is <200,000 atoms and occurs on the seconds timescale.[166,167] Being highly charged, ADP must navigate a rugged energy landscape to unbind, making the selection of an effective progress coordinate especially difficult. Progress coordinates typical for protein-ligand unbinding processes, such as a minimum distance between ligand and receptor, are rendered less effective when high electrostatic attraction pulls ADP to various

sites in the spacious binding pocket. So far, only a three-dimensional combination of an unbinding RMSD, a minimum distance between the ligand and receptor, and the interaction energy between the ligand and receptor has been effective at generating unbinding events in WE simulations. Arriving at this combination of coordinates involved a lot of trial and error. More automated methods for progress coordinate detection are needed not just for WE simulations, but for path sampling simulations as a whole.

Machine learning (ML) strategies have recently shown promise in detecting effective progress coordinates that can be used to enhance a variety of sampling strategies, particularly the RAVE and SGOOP methods.[100,168] These ML techniques exploit some underlying order in an initial trajectory ensemble to decide which motion of the system would be most promising for observing a transition of interest. An example of one such ML technique is the variational autoencoder employed in DeepDriveMD,[99,169] which finds a latent-space representation of a set of trajectories and identifies outliers in that latent space. In DeepDriveMD, outliers in this latent space are most likely system configurations that are undergoing fluctuations in a promising direction related to a transition and are therefore split to enhance for success along that fluctuation. Recent applications of the DeepDrive workflow to WE simulations of Eg5-ADP unbinding have shown promise in automatically identifying a progress coordinate to describe the unbinding process.

Another challenge to WE simulations of Eg5-ADP unbinding involves binning along a high-dimensional progress coordinate. Even with adaptive binning, which removes much of guess-work from the setup of WE simulations, binning in high-dimensional space can lead to inefficient use of computing resources. The use of "checkpointing", or subdividing the entire progress coordinate space into multiple MAB schemes, which is similar to the ideas behind the weighted ensemble-milestoning (WEM) method,[170] can help to more efficiently focus computing power on specific regions of high-dimensional space. This "multi-MAB" approach has already proven useful in simulations of ligand permeation through a membrane,[52] and has successfully generated unbinding pathways for the Eg5-ADP system.

A promising way of performing WE resampling in higher dimensions is through binless schemes. Binless schemes such as REVO[36,37] have performed well for protein-ligand unbinding of small, uncharged ligands. I have re-cast the MAB scheme as a binless resampler to

create a balanced progress resampler (BPR). This resampler collapses a multi-dimensional coordinate into a one-dimensional "coordinate" for resampling purposes, taking the product of a progress score in each progress coordinate dimension (a metric of how close to a dimension's "target" value the trajectory is) and scaling that product by the trajectory weights. The BPR scheme and other binless resampers enable the incorporation of a wide range of information–such as WE trajectory weights and progress coordinate data from previous WE iterations–into the decision on which trajectories to split and merge, making these schemes much more flexible and efficient than current binned schemes.

By combining more automated progress coordinate detection with more efficient resampling in high-dimensional phase space, the WE strategy should soon be able to generate an ensemble of pathways for challenging rare events such as Eg5-ADP unbinding. Recent attempts to identify a progress coordinate for Eg5-ADP unbinding using a variational autoencoder are promising in their ability to separate out conformations from a representative unbinding trajectory in latent space. In addition, recent unbinding pathways generated with the BPR indicate that binless resampling is not only more efficient, but also able to generate unbinding events with less than 50% of the computational time required for a corresponding simulation using a multi-MAB scheme. In conclusion, I anticipate that ligand unbinding processes involving charged ligands, such as the Eg5-ADP ligand unbinding process, will prove invaluable for future methods development. The lack of an obvious, effective, low-dimensional progress coordinate in Eg5-ADP unbinding, and in other processes such as water dissociation from a magnesium ion, provides a challenge to the generation of pathways that can serve as a valuable "stress test" for path sampling methods.

Bibliography

[1] Zwier, M. C.; Chong, L. T. Reaching Biological Timescales with All-Atom Molecular Dynamics Simulations. *Current Opinion in Pharmacology* **2010**, *10*, 745–752, DOI: 10.1016/j.coph.2010.09.008 (page 1).

[2] Shaw, D. E.; Maragakis, P.; Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; Eastwood, M. P.; Bank, J. A.; Jumper, J. M.; Salmon, J. K.; Shan, Y.; Wriggers, W. Atomic-Level Characterization of the Structural Dynamics of Proteins. *Science (New York, N.Y.)* **2010**, *330*, 341–346 (pages 1, 30).

[3] Shaw, D. E.; Adams, P. J.; Azaria, A.; Bank, J. A.; Batson, B.; Bell, A.; Bergdorf, M.; Bhatt, J.; Butts, J. A.; Correia, T.; Dirks, R. M.; Dror, R. O.; Eastwood, M. P.; Edwards, B.; Even, A.; Feldmann, P.; Fenn, M.; Fenton, C. H.; Forte, A.; Gagliardo, J.; Gill, G.; Gorlatova, M.; Greskamp, B.; Grossman, J.; Gullingsrud, J.; Harper, A.; Hasenplaugh, W.; Heily, M.; Heshmat, B. C.; Hunt, J.; Ierardi, D. J.; Iserovich, L.; Jackson, B. L.; Johnson, N. P.; Kirk, M. M.; Klepeis, J. L.; Kuskin, J. S.; Mackenzie, K. M.; Mader, R. J.; McGowen, R.; McLaughlin, A.; Moraes, M. A.; Nasr, M. H.; Nociolo, L. J.; O'Donnell, L.; Parker, A.; Peticolas, J. L.; Pocina, G.; Predescu, C.; Quan, T.; Salmon, J. K.; Schwink, C.; Shim, K. S.; Siddique, N.; Spengler, J.; Szalay, T.; Tabladillo, R.; Tartler, R.; Taube, A. G.; Theobald, M.; Towles, B.; Vick, W.; Wang, S. C.; Wazlowski, M.; Weingarten, M. J.; Williams, J. M.; Yuh, K. A. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ACM: St. Louis Missouri, 2021, pp 1–11, DOI: 10.1145/3458817.3487397 (page 1).

[4] Casalino, L.; Gaieb, Z.; Goldsmith, J. A.; Hjorth, C. K.; Dommer, A. C.; Harbison, A. M.; Fogarty, C. A.; Barros, E. P.; Taylor, B. C.; McLellan, J. S. Beyond Shielding: The Roles of Glycans in the SARS-CoV-2 Spike Protein. *ACS Central Science* **2020**, *6*, 1722–1734 (pages 1, 23, 30).

[5] Anandakrishnan, R.; Zhang, Z.; Donovan-Maiye, R.; Zuckerman, D. M. Biophysical Comparison of ATP Synthesis Mechanisms Shows a Kinetic Advantage for the Ro-

tary Process. *Proceedings of the National Academy of Sciences of the United States of America* **2016**, *113*, 11220 (pages 1, 30).

[6] Durrant, J. D.; Kochanek, S. E.; Casalino, L.; Ieong, P. U.; Dommer, A. C.; Amaro, R. E. Mesoscale All-Atom Influenza Virus Simulations Suggest New Substrate Binding Mechanism. *ACS Central Science* **2020**, *6*, 189–196, DOI: 10.1021/acscentsci.9b01071 (page 1).

[7] Zhao, G.; Perilla, J. R.; Yufenyuy, E. L.; Meng, X.; Chen, B.; Ning, J.; Ahn, J.; Gronenborn, A. M.; Schulten, K.; Aiken, C.; Zhang, P. Mature HIV-1 Capsid Structure by Cryo-Electron Microscopy and All-Atom Molecular Dynamics. *Nature* **2013**, *497*, 643–646 (pages 1, 30).

[8] Perilla, J. R.; Schulten, K. Physical Properties of the HIV-1 Capsid from All-Atom Molecular Dynamics Simulations. *Nature Communications* **2017**, *8*, 15959 (pages 1, 30).

[9] Casalino, L.; Dommer, A. C.; Gaieb, Z.; Barros, E. P.; Sztain, T.; Ahn, S.-H.; Trifan, A.; Brace, A.; Bogetti, A. T.; Clyde, A. AI-Driven Multiscale Simulations Illuminate Mechanisms of SARS-CoV-2 Spike Dynamics. *International Journal of High Performance Computing Applications* **2021**, 1–20 (pages 1, 11, 23, 26, 30).

[10] Jung, J.; Nishima, W.; Daniels, M.; Bascom, G.; Kobayashi, C.; Adedoyin, A.; Wall, M.; Lappala, A.; Phillips, D.; Fischer, W.; Tung, C. S.; Schlick, T.; Sugita, Y.; Sanbonmatsu, K. Y. Scaling Molecular Dynamics beyond 100,000 Processor Cores for Large-Scale Biophysical Simulations. *Journal of Computational Chemistry* **2019**, *40*, 1919–1930 (pages 1, 30).

[11] Ponder, J. W.; Wu, C.; Ren, P.; Pande, V. S.; Chodera, J. D.; Schnieders, M. J.; Haque, I.; Mobley, D. L.; Lambrecht, D. S.; DiStasio, R. A.; Head-Gordon, M.; Clark, G. N. I.; Johnson, M. E.; Head-Gordon, T. Current Status of the AMOEBA Polarizable Force Field. *The Journal of Physical Chemistry B* **2010**, *114*, 2549–2564, DOI: 10.1021/jp910674d (pages 1, 89).

[12] Zhu, X.; Lopes, P. E. M.; MacKerell, A. D. Recent Developments and Applications of the CHARMM Force Fields. *WIREs Computational Molecular Science* **2012**, *2*, 167–185, DOI: 10.1002/wcms.74 (pages 1, 89).

[13] Warshel, A.; Levitt, M. Theoretical Studies of Enzymic Reactions: Dielectric, Electrostatic and Steric Stabilization of the Carbonium Ion in the Reaction of Lysozyme. *Journal of Molecular Biology* **1976**, *103*, 227–249, DOI: 10.1016/0022-2836(76)90311-9 (pages 1, 89).

[14] Senn, H. M.; Thiel, W. QM/MM Methods for Biomolecular Systems. *Angewandte Chemie International Edition* **2009**, *48*, 1198–1229, DOI: 10.1002/anie.200802019 (pages 1, 89).

[15] Bonk, B. M.; Weis, J. W.; Tidor, B. Machine Learning Identifies Chemical Characteristics That Promote Enzyme Catalysis. *Journal of the American Chemical Society* **2019**, *141*, 4108–4118, DOI: 10.1021/jacs.8b13879 (pages 1, 89).

[16] Van Erp, T. S.; Bolhuis, P. G. Elaborating Transition Interface Sampling Methods. *Journal of Computational Physics* **2005**, *205*, 157–181, DOI: 10.1016/j.jcp.2004.11.003 (page 3).

[17] Allen, R. J.; Valeriani, C.; Rein ten Wolde, P. Ten. Forward Flux Sampling for Rare Event Simulations. *Journal of Physics: Condensed Matter* **2009**, *21*, 463102 (pages 3, 30).

[18] Zuckerman, D. M.; Chong, L. T. Weighted Ensemble Simulation: Review of Methodology, Applications, and Software. *Annual Review of Biophysics, Vol 40* **2017**, *46*, 43–57 (pages 3, 5, 8, 22, 30, 53, 54, 73).

[19] Hall, S. W.; Díaz Leines, G.; Sarupria, S.; Rogal, J. Practical Guide to Replica Exchange Transition Interface Sampling and Forward Flux Sampling. *The Journal of Chemical Physics* **2022**, *156*, 200901, DOI: 10.1063/5.0080053 (page 3).

[20] van Erp, T. S.; Moroni, D.; Bolhuis, P. G. A Novel Path Sampling Method for the Calculation of Rate Constants. *Journal of Chemical Physics* **2003**, *118*, 7762–7774 (pages 3, 30).

[21] Allen, R. J.; Warren, P. B.; Ten Wolde, P. R. Sampling Rare Switching Events in Biochemical Networks. *Physical Review Letters* **2005**, *94*, 018104, DOI: 10.1103/PhysRevLett.94.018104 (page 3).

[22] Becker, N. B.; Allen, R. J.; Ten Wolde, P. R. Non-Stationary Forward Flux Sampling. *The Journal of Chemical Physics* **2012**, *136*, 174118, DOI: 10.1063/1.4704810 (page 3).

[23] Faradjian, A. K.; Elber, R. Computing Time Scales from Reaction Coordinates by Milestoning. *Journal of Chemical Physics* **2004**, *120*, 10880–10889 (pages 3, 30).

[24] Majek, P.; Elber, R. Milestoning without a Reaction Coordinate. *Journal of Chemical Theory and Computation* **2010**, *6*, 1805–1817, DOI: 10.1021/ct100114j (page 3).

[25] Bello-Rivas, J. M.; Elber, R. Exact Milestoning. *The Journal of Chemical Physics* **2015**, *142*, 094102, DOI: 10.1063/1.4913399 (page 3).

[26] Huber, G. A.; Kim, S. Weighted-Ensemble Brownian Dynamics Simulations for Protein Association Reactions. *Biophysical Journal* **1996**, *70*, 97–110 (pages 3, 8, 22, 27, 30, 31, 40, 45, 53, 54, 61, 73).

[27] Chong, L. T.; Saglam, A. S.; Zuckerman, D. M. Path-Sampling Strategies for Simulating Rare Events in Biomolecular Systems. *Current Opinion in Structural Biology* **2017**, *43*, 88–94 (pages 3, 5, 53).

[28] Zhang, B. W.; Jasnow, D.; Zuckerman, D. M. The "Weighted Ensemble" Path Sampling Method Is Statistically Exact for a Broad Class of Stochastic Processes and Binning Procedures. *Journal of Chemical Physics* **2010**, *132*, 054107 (pages 5, 7, 31, 32, 53–55).

[29] Zimmerman, M. I.; Bowman, G. R. FAST Conformational Searches by Balancing Exploration/Exploitation Trade-Offs. *Journal of Chemical Theory and Computation* **2015**, *11*, 5747–5757 (pages 7, 53).

[30] Torrie, G.; Valleau, J. Nonphysical Sampling Distributions in Monte Carlo Free-Energy Estimation: Umbrella Sampling. *Journal of Computational Physics* **1977**, *23*, 187–199, DOI: 10.1016/0021-9991(77)90121-8 (page 7).

[31] Kästner, J. Umbrella Sampling: Umbrella Sampling. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2011**, *1*, 932–942, DOI: 10.1002/wcms.66 (page 7).

[32] Laio, A.; Parrinello, M. Escaping Free-Energy Minima. *Proceedings of the National Academy of Sciences* **2002**, *99*, 12562–12566, DOI: 10.1073/pnas.202427399 (page 7).

[33] Bussi, G.; Laio, A. Using Metadynamics to Explore Complex Free-Energy Landscapes. *Nature Reviews Physics* **2020**, *2*, 200–212, DOI: 10.1038/s42254-020-0153-0 (page 7).

[34] Aristoff, D.; Zuckerman, D. M. Optimizing Weighted Ensemble Sampling of Steady States. *Multiscale Modeling & Simulation. A SIAM Interdisciplinary Journal* **2020**, *18*, 646–673 (pages 7, 53, 65, 67).

[35] Aristoff, D.; Copperman, J.; Simpson, G.; Webber, R. J.; Zuckerman, D. M. Weighted Ensemble: Recent Mathematical Developments. *The Journal of Chemical Physics* **2023**, *158*, 014108, DOI: 10.1063/5.0110873 (page 7).

[36] Donyapour, N.; Roussey, N. M.; Dickson, A. REVO: Resampling of Ensembles by Variation Optimization. *Journal of Chemical Physics* **2019**, *150*, 244112 (pages 8, 32, 90).

[37] Roussey, N. M.; Dickson, A. Quality over Quantity: Sampling High Probability Rare Events with the Weighted Ensemble Algorithm. *Journal of Computational Chemistry* **2023**, *44*, 935–947, DOI: 10.1002/jcc.27054 (pages 8, 90).

[38] Bogetti, A. T.; Presti, M. F.; Loh, S. N.; Chong, L. T. The Next Frontier for Designing Switchable Proteins: Rational Enhancement of Kinetics. *The Journal of Physical Chemistry B* **2021**, *125*, 9069–9077, DOI: 10.1021/acs.jpcb.1c04082 (page 9).

[39] Bhatt, D.; Zhang, B. W.; Zuckerman, D. M. Steady-State Simulations Using Weighted Ensemble Path Sampling. *Journal of Chemical Physics* **2010**, *133*, 014110 (pages 8, 32, 45, 50).

[40] Copperman, J.; Zuckerman, D. M. Accelerated Estimation of Long-Timescale Kinetics from Weighted Ensemble Simulation via Non-Markovian "Microbin" Analysis. *Journal of Chemical Theory and Computation* **2020**, *16*, 6763–6775 (pages 8, 27, 45).

[41] Sztain, T.; Ahn, S.-H.; Bogetti, A. T.; Casalino, L.; Goldsmith, J. A.; McCool, R. S.; Kearns, F. L.; McCammon, J. A.; McLellan, J. S.; Chong, L. T.; Amaro, R. E. A Glycan Gate Controls Opening of the SARS-CoV-2 Spike Protein. *Nature Chemistry* **2021**, *13*, 963–968 (pages 8, 10, 11, 23, 25, 27, 30, 31).

[42] Zwier, M. C.; Adelman, J. L.; Kaus, J. W.; Pratt, A. J.; Wong, K. F.; Rego, N. B.; Suárez, E.; Lettieri, S.; Wang, D. W.; Grabe, M. WESTPA: An Interoperable, Highly Scalable Software Package for Weighted Ensemble Simulation and Analysis. *Journal of Chemical Theory and Computation* **2015**, *11*, 800–809 (pages 10, 31, 58).

[43] Adhikari, U.; Mostofian, B.; Copperman, J.; Subramanian, S. R.; Petersen, A. A.; Zuckerman, D. M. Computational Estimation of Microsecond to Second Atomistic Folding Times. *Journal of The American Chemical Society* **2019**, *141*, 6519–6526 (pages 10, 23, 30, 53).

[44] Zwier, M. C.; Pratt, A. J.; Adelman, J. L.; Kaus, J. W.; Zuckerman, D. M.; Chong, L. T. 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. *Journal of Physical Chemistry Letters* **2016**, *7*, 3440–3445 (pages 10, 30, 44, 53, 64).

[45] Saglam, A. S.; Chong, L. T. Protein-Protein Binding Pathways and Calculations of Rate Constants Using Fully-Continuous, Explicit-Solvent Simulations. *Chemical Science* **2019**, *10*, 2360–2372 (pages 10, 23, 30, 31, 53).

[46] Lotz, S. D.; Dickson, A. Unbiased Molecular Dynamics of 11 Min Timescale Drug Unbinding Reveals Transition State Stabilizing Interactions. *Journal of The American Chemical Society* **2018**, *140*, 618–628 (pages 10, 30, 53).

[47] Santhouse, J. R.; Leung, J. M. G.; Chong, L. T.; Horne, W. S. Implications of the Unfolded State in the Folding Energetics of Heterogeneous-Backbone Protein Mimetics. *Chemical Science* **2022**, *13*, 11798–11806, DOI: 10.1039/D2SC04427G (page 10).

[48] Russo, J. D.; Zhang, S.; Leung, J. M. G.; Bogetti, A. T.; Thompson, J. P.; DeGrave, A. J.; Torrillo, P. A.; Pratt, A. J.; Wong, K. F.; Xia, J.; Copperman, J.; Adelman, J. L.; Zwier, M. C.; LeBard, D. N.; Zuckerman, D. M.; Chong, L. T. WESTPA

2.0: High-Performance Upgrades for Weighted Ensemble Simulations and Analysis of Longer-Timescale Applications. *Journal of Chemical Theory and Computation* **2022**, *18*, 638–649, DOI: 10.1021/acs.jctc.1c01154 (pages 10, 73).

[49] Torrillo, P. A.; Bogetti, A. T.; Chong, L. T. A Minimal, Adaptive Binning Scheme for Weighted Ensemble Simulations. *Journal of Physical Chemistry A* **2021**, *125*, 1642–1649 (pages 10, 27, 38).

[50] Seyler, S. L.; Kumar, A.; Thorpe, M. F.; Beckstein, O. Path Similarity Analysis: A Method for Quantifying Macromolecular Pathways. *PLOS Computational Biology* **2015**, *11*, ed. by Tajkhorshid, E., e1004568, DOI: 10.1371/journal.pcbi.1004568 (pages 11, 72).

[51] Suárez, E.; Zuckerman, D. M. Pathway Histogram Analysis of Trajectories: A General Strategy for Quantification of Molecular Mechanisms. **2018** (pages 11, 72).

[52] Zhang, S.; Thompson, J. P.; Xia, J.; Bogetti, A. T.; York, F.; Skillman, A. G.; Chong, L. T.; LeBard, D. N. Mechanistic Insights into Passive Membrane Permeability of Drug-like Molecules from a Weighted Ensemble of Trajectories. *Journal of Chemical Information and Modeling* **2022**, *62*, 1891–1904, DOI: 10.1021/acs.jcim.1c01540 (pages 11, 90).

[53] Bogetti, A. T.; Piston, H. E.; Leung, J. M. G.; Cabalteja, C. C.; Yang, D. T.; DeGrave, A. J.; Debiec, K. T.; Cerutti, D. S.; Case, D. A.; Horne, W. S.; Chong, L. T. A Twist in the Road Less Traveled: The AMBER Ff15ipq-m Force Field for Protein Mimetics. *The Journal of Chemical Physics* **2020**, *153*, 064101, DOI: 10.1063/5.0019054 (page 11).

[54] DeGrave, A. J.; Chong, L. T. The RED Scheme: Rate-constant Estimation from Pre-Steady State Weighted Ensemble Simulations. *Journal of Chemical Physics* **2021**, *154*, 114111 (pages 11, 27, 45, 60).

[55] Bogetti, A. T.; Mostofian, B.; Dickson, A.; Pratt, A.; Saglam, A. S.; Harrison, P. O.; Adelman, J. L.; Dudek, M.; Torrillo, P. A.; DeGrave, A. J.; Adhikari, U.; Zwier, M. C.; Zuckerman, D. M.; Chong, L. T. A Suite of Tutorials for the WESTPA Rare-Events Sampling Software [Article v1.0]. *Living Journal of Computational Molecular Science.* **2019**, *1*, 10607 (pages 11, 35, 37, 64, 67).

[56] Bogetti, A. T.; Leung, J. M. G.; Russo, J. D.; Zhang, S.; Thompson, J. P.; Saglam, A. S.; Ray, D.; Abraham, R. C.; Faeder, J. R.; Andricioaei, I.; Adelman, J. L.; Zwier, M. C.; LeBard, D. N.; Zuckerman, D. M.; Chong, L. T. A Suite of Advanced Tutorials for the WESTPA 2.0 Rare-Events Sampling Software [Article v2.0]. *Living Journal of Computational Molecular Science* **2022**, *5*, DOI: 10.33011/livecoms.5.1.1655 (page 11).

[57] Dommer, A.; Casalino, L.; Kearns, F.; Rosenfeld, M.; Wauer, N.; Ahn, S.-H.; Russo, J.; Oliveira, S.; Morris, C.; Bogetti, A.; Trifan, A.; Brace, A.; Sztain, T.; Clyde, A.; Ma, H.; Chennubhotla, C.; Lee, H.; Turilli, M.; Khalid, S.; Tamayo-Mendoza, T.; Welborn, M.; Christensen, A.; Smith, D. G.; Qiao, Z.; Sirumalla, S. K.; O'Connor, M.; Manby, F.; Anandkumar, A.; Hardy, D.; Phillips, J.; Stern, A.; Romero, J.; Clark, D.; Dorrell, M.; Maiden, T.; Huang, L.; McCalpin, J.; Woods, C.; Gray, A.; Williams, M.; Barker, B.; Rajapaksha, H.; Pitts, R.; Gibbs, T.; Stone, J.; Zuckerman, D. M.; Mulholland, A. J.; Miller, T.; Jha, S.; Ramanathan, A.; Chong, L.; Amaro, R. E. #COVIDisAirborne: AI-enabled Multiscale Computational Microscopy of Delta SARS-CoV-2 in a Respiratory Aerosol. *The International Journal of High Performance Computing Applications* **2023**, *37*, 28–44, DOI: 10.1177/10943420221128233 (page 11).

[58] Adamson, H.; Jeuken, L. J. C. Engineering Protein Switches for Rapid Diagnostic Tests. *ACS Sensors* **2020**, *5*, 3001–3012 (page 12).

[59] Farahani, P. E.; Reed, E. H.; Underhill, E. J.; Aoki, K.; Toettcher, J. E. Signaling, Deconstructed: Using Optogenetics to Dissect and Direct Information Flow in Biological Systems. *Annual Review of Biomedical Engineering, Vol 12* **2021**, *23*, 61–87 (page 12).

[60] Pan, X.; Kortemme, T. Recent Advances in de Novo Protein Design: Principles, Methods, and Applications. *Journal of Biological Chemistry* **2021**, *296*, 100558 (page 12).

[61] Baker, D. What Has de Novo Protein Design Taught Us about Protein Folding and Biophysics? *Protein Science* **2019**, *28*, 678–683 (pages 12, 21).

[62] Fersht, A. R.; Matouschek, A.; Serrano, L. The Folding of an Enzyme. I. Theory of Protein Engineering Analysis of Stability and Pathway of Protein Folding. *Journal of Molecular Biology* **1992**, *224*, 771–782 (page 15).

[63] Naganathan, A. N.; Muñoz, V. Insights into Protein Folding Mechanisms from Large Scale Analysis of Mutational Effects. *Proceedings of the National Academy of Sciences of the United States of America* **2010**, *107*, 8611 (page 16).

[64] Zhang, Y.; Kitazawa, S.; Peran, I.; Stenzoski, N.; McCallum, S. A.; Raleigh, D. P.; Royer, C. A. High Pressure ZZ-Exchange NMR Reveals Key Features of Protein Folding Transition States. *Journal of The American Chemical Society* **2016**, *138*, 15260–15266 (page 16).

[65] Nielsen, A. K.; Möller, I. R.; Wang, Y.; Rasmussen, S. G. F.; Lindorff-Larsen, K.; Rand, K. D.; Loland, C. J. Substrate-Induced Conformational Dynamics of the Dopamine Transporter. *Nature Communications* **2019**, *10*, 2714 (page 16).

[66] Jia, R.; Martens, C.; Shekhar, M.; Pant, S.; Pellowe, G. A.; Lau, A. M.; Findlay, H. E.; Harris, N. J.; Tajkhorshid, E.; Booth, P. J.; Politis, A. Hydrogen-Deuterium Exchange Mass Spectrometry Captures Distinct Dynamics upon Substrate and Inhibitor Binding to a Transporter. *Nature Communications* **2020**, *11*, 6162 (page 16).

[67] Liebau, J.; Tersa, M.; Trastoy, B.; Patrick, J.; Rodrigo-Unzueta, A.; Corzana, F.; Sparrman, T.; Guerin, M. E.; Mäler, L. Unveiling the Activation Dynamics of a Fold-Switch Bacterial Glycosyltransferase by 19F NMR. *Journal of Biological Chemistry* **2020**, *295*, 9868–9878 (page 16).

[68] Fenwick, R. B.; Oyen, D.; van den Bedem, H.; Dyson, H. J.; Wright, P. E. Modeling of Hidden Structures Using Sparse Chemical Shift Data from NMR Relaxation Dispersion. *Biophysical Journal* **2021**, *120*, 296–305 (page 16).

[69] Stratton, M. M.; Mitrea, D. M.; Loh, S. N. A Ca2+-Sensing Molecular Switch Based on Alternate Frame Protein Folding. *Acs Chemical Biology* **2008**, *3*, 723–732 (page 16).

[70] Stratton, M. M.; Loh, S. N. On the Mechanism of Protein Fold-Switching by a Molecular Sensor. *Proteins: Structure, Function, and Genetics* **2010**, *78*, 3260 (page 17).

[71] Nakai, J.; Ohkura, M.; Imoto, K. A High Signal-to-Noise Ca2+ Probe Composed of a Single Green Fluorescent Protein. *Nature Biotechnology* **2001**, *19*, 137–141 (page 17).

[72] Inoue, M.; Takeuchi, A.; Manita, S.; Horigane, S.-I.; Sakamoto, M.; Kawakami, R.; Yamaguchi, K.; Otomo, K.; Yokoyama, H.; Kim, R.; Yokoyama, T. Rational Engineering of XCaMPs, a Multicolor GECI Suite for in Vivo Imaging of Complex Brain Circuit Dynamics. *Cell* **2019**, *177*, 1346–1360 (page 19).

[73] Kerruth, S.; Coates, C.; Dürst, C. D.; Oertner, T. G.; Török, K. The Kinetic Mechanisms of Fast-Decay Red-Fluorescent Genetically Encoded Calcium Indicators. *Journal of Biological Chemistry* **2019**, *294*, 3934–3946 (page 19).

[74] Sun, X. R.; Badura, A.; Pacheco, D. A.; Lynch, L. A.; Schneider, E. R.; Taylor, M. P.; Hogue, I. B.; Enquist, L. W.; Murthy, M.; Wang, S. S.-H. Fast GCaMPs for Improved Tracking of Neuronal Activity. *Nature Communications* **2013**, *4*, 2170 (page 19).

[75] Halavaty, A. S.; Moffat, K. N- and C-terminal Flanking Regions Modulate Light-Induced Signal Transduction in the LOV2 Domain of the Blue Light Sensor Phototropin 1 from Avena Sativa. *Biochemistry* **2007**, *46*, 14001–14009 (page 19).

[76] Iuliano, J. N.; Collado, J. T.; Gil, A. A.; Ravindran, P. T.; Lukacs, A.; Shin, S.; Woroniecka, H. A.; Adamczyk, K.; Aramini, J. M.; Edupuganti, U. R. Unraveling the Mechanism of a LOV Domain Optogenetic Sensor: A Glutamine Lever Induces Unfolding of the Jα Helix. *Acs Chemical Biology* **2020**, *15*, 2752–2765 (pages 19, 20).

[77] Pudasaini, A.; El-Arab, K. K.; Zoltowski, B. D. LOV-Based Optogenetic Devices: Light-driven Modules to Impart Photoregulated Control of Cellular Signaling. **2015**, *2*, 18 (page 19).

[78] Kawano, F.; Aono, Y.; Suzuki, H.; Sato, M. Fluorescence Imaging-Based High-Throughput Screening of Fast- and Slow-Cycling LOV Proteins. *PLoS One* **2013**, *8*, e82693 (pages 19, 20).

[79] Wang, H.; Vilela, M.; Winkler, A.; Tarnawski, M.; Schlichting, I.; Yumerefendi, H.; Kuhlman, B.; Liu, R.; Danuser, G.; Hahn, K. M. LOVTRAP, an Optogenetic System for Photo-Induced Protein Dissociation. *Nature Methods* **2016**, *13*, 755–758 (page 20).

[80] Zoltowski, B. D.; Vaccaro, B.; Crane, B. R. Mechanism-Based Tuning of a LOV Domain Photoreceptor. *Nature Chemical Biology* **2009**, *5*, 827–834 (page 20).

[81] Tischer, D. K.; Weiner, O. D. Light-Based Tuning of Ligand Half-Life Supports Kinetic Proofreading Model of T Cell Signaling. *eLife* **2019**, *8*, e42498 (page 20).

[82] Dagliyan, O.; Tarnawski, M.; Chu, P.-H.; Shirvanyants, D.; Schlichting, I.; Dokholyan, N. V.; Hahn, K. M. Engineering Extrinsic Disorder to Control Protein Activity in Living Cells. *Science (New York, N.Y.)* **2016**, *354*, 1441–1444 (page 20).

[83] Wang, X.; He, L.; Wu, Y. I.; Hahn, K. M.; Montell, D. J. Light-Mediated Activation Reveals a Key Role for Rac in Collective Guidance of Cell Movement in Vivo. *Nature Cell Biology* **2010**, *12*, 591–597 (page 20).

[84] Shibata, A. C. E.; Ueda, H. H.; Eto, K.; Onda, M.; Sato, A.; Ohba, T.; Nabekura, J.; Murakoshi, H. Photoactivatable CaMKII Induces Synaptic Plasticity in Single Synapses. *Nature Communications* **2021**, *12*, 751 (page 20).

[85] Zimmerman, S. P.; Kuhlman, B.; Yumerefendi, H. Engineering and Application of LOV2-Based Photoswitches. *Methods in Enzymology* **2016**, *580*, 169–190 (page 20).

[86] Langan, R. A.; Boyken, S. E.; Ng, A. H.; Samson, J. A.; Dods, G.; Westbrook, A. M.; Nguyen, T. H.; Lajoie, M. J.; Chen, Z.; Berger, S. De Novo Design of Bioactive Protein Switches. *Nature* **2019**, *572*, 205–210 (page 20).

[87] Quijano-Rubio, A.; Yeh, H.-W.; Park, J.; Lee, H.; Langan, R. A.; Boyken, S. E.; Lajoie, M. J.; Cao, L.; Chow, C. M.; Miranda, M. C. De Novo Design of Modular and Tunable Protein Biosensors. *Nature* **2021**, *591*, 482–487 (page 21).

[88] Zheng, H.; Bi, J.; Krendel, M.; Loh, S. N. Converting a Binding Protein into a Biosensing Conformational Switch Using Protein Fragment Exchange. *Biochemistry* **2014**, *53*, 5505–5514 (page 21).

[89] DeGrave, A. J.; Ha, J.-H.; Loh, S. N.; Chong, L. T. Large Enhancement of Response Times of a Protein Conformational Switch by Computational Design. *Nature Communications* **2018**, *9*, 1013 (pages 22, 23, 25, 28, 68).

[90] Go, N. Theoretical Studies of Protein Folding. *Annual Review of Biophysics and Bioengineering* **1983**, *12*, 183–210 (pages 22, 25).

[91] Dixon, T.; Uyar, A.; Ferguson-Miller, S.; Dickson, A. Membrane-Mediated Ligand Unbinding of the PK-11195 Ligand from TSPO. *Biophysical Journal* **2021**, *120*, 158–167 (page 23).

[92] Benton, D. J.; Wrobel, A. G.; Xu, P.; Roustan, C.; Martin, S. R.; Rosenthal, P. B.; Skehel, J. J.; Gamblin, S. J. Receptor Binding and Priming of the Spike Protein of SARS-CoV-2 for Membrane Fusion. *Nature* **2020**, *588*, 327–330 (page 23).

[93] Lu, M.; Uchil, P. D.; Li, W.; Zheng, D.; Terry, D. S.; Gorman, J.; Shi, W.; Zhang, B.; Zhou, T.; Ding, S. Real-Time Conformational Dynamics of SARS-CoV-2 Spikes on Virus Particles. *Cell host & microbe* **2020**, *28*, 880–891 (page 23).

[94] Zimmerman, M. I.; Porter, J. R.; Ward, M. D.; Singh, S.; Vithani, N.; Meller, A.; Mallimadugula, U. L.; Kuhn, C. E.; Borowsky, J. H.; Wiewiora, R. P. SARS-CoV-2 Simulations Go Exascale to Predict Dramatic Spike Opening and Cryptic Pockets across the Proteome. *Nature Chemistry* **2021**, *13*, 651–659 (page 25).

[95] Clementi, C.; Nymeyer, H.; Onuchic, J. N. Topological and Energetic Factors: What Determinesthe Structural Details of the Transition State Ensembleand "En-Route" Intermediates for Protein Folding? An Investigation for Small Globular Proteins. *Journal of Molecular Biology* **2000**, *298*, 937–953 (page 26).

[96] Souza, P. C. T.; Alessandri, R.; Barnoud, J.; Thallmair, S.; Faustino, I.; Grünewald, F.; Patmanidis, I.; Abdizadeh, H.; Bruininks, B. M. H.; Wassenaar, T. A. Martini 3: A General Purpose Force Field for Coarse-Grained Molecular Dynamics. *Nature Methods* **2021**, *18*, 382–388 (page 26).

[97] Machado, M. R.; Barrera, E. E.; Klein, F.; Sóñora, M.; Silva, S.; Pantano, S. The SIRAH 2.0 Force Field: Altius, Fortius, Citius. *Journal of Chemical Theory and Computation* **2019**, *15*, 2719–2733 (page 26).

[98] Poma, A. B.; Cieplak, M.; Theodorakis, P. E. Combining the MARTINI and Structure-Based Coarse-Grained Approaches for the Molecular Dynamics Studies of Conformational Transitions in Proteins. *Journal of Chemical Theory and Computation* **2017**, *13*, 1366–1374, DOI: 10.1021/acs.jctc.6b00986 (page 26).

[99] Bhowmik, D.; Gao, S.; Young, M. T.; Ramanathan, A. Deep Clustering of Protein Folding Simulations. *BMC Bioinformatics* **2018**, *19*, 484 (pages 26, 51, 90).

[100] Ribeiro, J. M. L.; Bravo, P.; Wang, Y.; Tiwary, P. Reweighted Autoencoded Variational Bayes for Enhanced Sampling (RAVE). *Journal of Chemical Physics* **2018**, *149*, 072301 (pages 26, 51, 90).

[101] Smith, Z.; Ravindra, P.; Wang, Y.; Cooley, R.; Tiwary, P. Discovering Protein Conformational Flexibility through Artificial-Intelligence-Aided Molecular Dynamics. *Journal of Physical Chemistry B* **2020**, *124*, 8221–8229 (page 26).

[102] Suárez, E.; Wiewiora, R. P.; Wehmeyer, C.; Noé, F.; Chodera, J. D.; Zuckerman, D. M. What Markov State Models Can and Cannot Do: Correlation versus Path-Based Observables in Protein Folding Models. *Journal of Chemical Theory and Computation* **2021**, *17*, 3119–3133 (page 27).

[103] McCammon, J. A.; Gelin, B. R.; Karplus, M. Dynamics of Folded Proteins. *Nature* **1977**, *267*, 585–590 (page 30).

[104] Zuckerman, D. M.; Woolf, T. B. Efficient Dynamic Importance Sampling of Rare Events in One Dimension. *Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics* **2000**, *63*, 016702 (page 30).

[105] Ray, D.; Stone, S. E.; Andricioaei, I. Markovian Weighted Ensemble Milestoning (M-WEM): Long-time Kinetics from Short Trajectories. **2022**, *18*, 79–95 (page 30).

[106] West, A. M. A.; Elber, R.; Shalloway, D. Extending Molecular Dynamics Time Scales with Milestoning: Example of Complex Kinetics in a Solvated Peptide. *Journal of Chemical Physics* **2007**, *126*, 145104 (page 30).

[107] Pratt, L. R. A Statistical Method for Identifying Transition States in High Dimensional Problems. *Journal of Chemical Physics* **1986**, *85*, 5045–5048 (pages 30, 53, 67).

[108] Swenson, D. W. H.; Bolhuis, P. G. A Replica Exchange Transition Interface Sampling Method with Multiple Interface Sets for Investigating Networks of Rare Events. *Journal of Chemical Physics* **2014**, *141*, 044101 (page 30).

[109] DeFever, R. S.; Sarupria, S. Contour Forward Flux Sampling: Sampling Rare Events along Multiple Collective Variables. *Journal of Chemical Physics* **2019**, *150*, 024103 (pages 30, 51, 53).

[110] Abdul-Wahid, B.; Feng, H.; Rajan, D.; Costaouec, R.; Darve, E.; Thain, D.; Izaguirre, J. A. AWE-WQ: Fast-forwarding Molecular Dynamics Using the Accelerated Weighted Ensemble. *Journal of Chemical Information and Modeling* **2014**, *54*, 3033–3043 (page 31).

[111] Lotz, S. D.; Dickson, A. Wepy: A Flexible Software Framework for Simulating Rare Events with Weighted Ensemble Resampling. *ACS Omega* **2020**, *5*, 31608–31623 (page 31).

[112] Saglam, A. S.; Chong, L. T. Highly Efficient Computation of the Basal Kon Using Direct Simulation of Protein–Protein Association with Flexible Molecular Models. *Journal of Physical Chemistry B* **2016**, *120*, 117–122 (page 31).

[113] Donovan, R. M.; Tapia, J.-J.; Sullivan, D. P.; Faeder, J. R.; Murphy, R. F.; Dittrich, M.; Zuckerman, D. M. Unbiased Rare Event Sampling in Spatial Stochastic Systems Biology Models Using a Weighted Ensemble of Trajectories. *Plos Computational Biology* **2016**, *12*, e1004611 (page 31).

[114] Tapia, J.-J.; Saglam, A. S.; Czech, J.; Kuczewski, R.; Bartol, T. M.; Sejnowski, T. J.; Faeder, J. R. MCell-R: A Particle-Resolution Network-Free Spatial Modeling Framework. *Yeast Genetic Networks: Methods and Protocols* **2019**, *1945*, 203–229 (page 31).

[115] Johnson, M. E.; Chen, A.; Faeder, J. R.; Henning, P.; Moraru, I. I.; Meier-Schellersheim, M.; Murphy, R. F.; Prüstel, T.; Theriot, J. A.; Uhrmacher, A. M. Quantifying the Roles of Space and Stochasticity in Computer Simulations for Cell Biology and Cellular Biochemistry. *MBoC* **2021**, *32*, 186–210 (page 31).

[116] Suárez, E.; Lettieri, S.; Zwier, M. C.; Stringer, C. A.; Subramanian, S. R.; Chong, L. T.; Zuckerman, D. M. Simultaneous Computation of Dynamical and Equilibrium Information Using a Weighted Ensemble of Trajectories. *Journal of Chemical Theory and Computation* **2014**, *10*, 2658–2667 (pages 32, 45, 50, 54).

[117] Suárez, E.; Pratt, A. J.; Chong, L. T.; Zuckerman, D. M. Estimating First-Passage Time Distributions from Weighted Ensemble Simulations and Non-Markovian Analyses. *Protein Science* **2016**, *25*, 67–78 (pages 32, 50).

[118] Grebner, C.; Malmerberg, E.; Shewmaker, A.; Batista, J.; Nicholls, A.; Sadowski, J. Virtual Screening in the Cloud: How Big Is Big Enough? *Journal of Chemical Information and Modeling* **2020**, *60*, 4274–4282 (pages 34, 51).

[119] Adelman, J. L.; Grabe, M. Simulating Rare Events Using a Weighted Ensemble-Based String Method. *Journal of Chemical Physics* **2013**, *138*, 044105 (pages 40, 53).

[120] Dickson, A.; Brooks, C. L. WExplore: Hierarchical Exploration of High-Dimensional Spaces Using the Weighted Ensemble Algorithm. *Journal of Physical Chemistry B* **2014**, *118*, 3532–3542 (pages 40, 53).

[121] McGibbon, R. T.; Beauchamp, K. A.; Harrigan, M. P.; Klein, C.; Swails, J. M.; Hernández, C. X.; Schwantes, C. R.; Wang, L.-P.; Lane, T. J.; Pande, V. S. MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophysical Journal* **2015**, *109*, 1528–1532 (page 42).

[122] Nguyen, H.; Case, D. A.; Rose, A. S. NGLview–Interactive Molecular Graphics for Jupyter Notebooks. *Bioinformatics (Oxford, England)* **2018**, *34*, 1241–1242 (page 42).

[123] Hess, B.; Kutzner, C.; van der Spoel, D.; Lindahl, E. GROMACS 4:Algorithms for Highly Efficient ,Load -Balanced,and Scalable Molecular Simulation. *Journal of Chemical Theory and Computation* **2008**, *4*, 435–447 (page 46).

[124] Mostofian, B.; Zuckerman, D. M. Statistical Uncertainty Analysis for Small-Sample, High Log-Variance Data: Cautions for Bootstrapping and Bayesian Bootstrapping. *Journal of Chemical Theory and Computation* **2019**, *15*, 3499–3509 (pages 45, 47).

[125] Chodera, J. D.; Noé, F. Markov State Models of Biomolecular Conformational Dynamics. *Current Opinion in Structural Biology* **2014**, *25*, 135–144 (page 47).

[126] Salomon-Ferrer, R.; Case, D. A.; Walker, R. C. An Overview of the Amber Biomolecular Simulation Package. *WIREs Computational Molecular Science* **2013**, *3*, 198–210 (page 49).

[127] Swenson, D. W. H.; Prinz, J.-H.; Noé, F.; Chodera, J. D.; Bolhuis, P. G. OpenPathSampling: A Python Framework for Path Sampling Simulations. 1. Basics. *Journal of Chemical Theory and Computation* **2019**, *15*, 813–836 (page 51).

[128] Swenson, D. W. H.; Prinz, J.-H.; Noé, F.; Chodera, J. D.; Bolhuis, P. G. OpenPathSampling: A Python Framework for Path Sampling Simulations. 2. Building and Customizing Path Ensembles and Sample Schemes. *Journal of Chemical Theory and Computation* **2019**, *15*, 837–856 (page 51).

[129] Wei, W.; Elber, R. ScMile: A Script to Investigate Kinetics with Short Time Molecular Dynamics Trajectories and the Milestoning Theory. *Journal of Chemical Theory and Computation* **2020**, *16*, 860–874 (page 51).

[130] Romo, T. D.; Leioatts, N.; Grossfield, A. Lightweight Object Oriented Structure Analysis: Tools for Building Tools to Analyze Molecular Dynamics Simulations. *Journal of Computational Chemistry* **2014**, *35*, 2305–2318 (page 51).

[131] Michaud-Agrawal, N.; Denning, E. J.; Woolf, T. B.; Beckstein, O. MDAnalysis: A Toolkit for the Analysis of Molecular Dynamics Simulations. *Journal of Computational Chemistry* **2011**, *32*, 2319–2327 (page 51).

[132] Scherer, M. K.; Trendelkamp-Schroer, B.; Paul, F.; Pérez-Hernández, G.; Hoffmann, M.; Plattner, N.; Wehmeyer, C.; Prinz, J.-H.; Noé, F. PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models. *Journal of Chemical Theory and Computation* **2015**, *11*, 5525–5542 (page 51).

[133] Teo, I.; Mayne, C. G.; Schulten, K.; Lelièvre, T. Adaptive Multilevel Splitting Method for Molecular Dynamics Calculation of Benzamidine-Trypsin Dissociation Time. *Journal of Chemical Theory and Computation* **2016**, *12*, 2983–2989 (page 53).

[134] Brotzakis, Z. F.; Bolhuis, P. G. Unbiased Atomistic Insight into the Mechanisms and Solvent Role for Globular Protein Dimer Dissociation. *Journal of Physical Chemistry B* **2019**, *123*, 1883–1895 (page 53).

[135] Jagger, B. R.; Ojha, A. A.; Amaro, R. E. Predicting Ligand Binding Kinetics Using a Markovian Milestoning with Voronoi Tessellations Multiscale Approach. *Journal of Chemical Theory and Computation* **2020**, *16*, 5348–5357 (page 53).

[136] Du, W.; Bolhuis, P. G. Sampling the Equilibrium Kinetic Network of Trp-Cage in Explicit Solvent. *Journal of Chemical Physics* **2014**, *140*, 195102 (page 53).

[137] Hruska, E.; Balasubramanian, V.; Lee, H.; Jha, S.; Clementi, C. Extensible and Scalable Adaptive Sampling on Supercomputers. *Journal of Chemical Theory and Computation* **2020**, *16*, 7915–7925 (page 53).

[138] Adelman, J. L.; Grabe, M. Simulating Current-Voltage Relationships for a Narrow Ion Channel Using the Weighted Ensemble Method. *Journal of Chemical Theory and Computation* **2015**, *11*, 1907–1918 (page 53).

[139] Ma, P.; Cardenas, A. E.; Chaudhari, M. I.; Elber, R.; Rempe, S. B. The Impact of Protonation on Early Translocation of Anthrax Lethal Factor: Kinetics from Molecular Dynamics Simulations and Milestoning Theory. *Journal of The American Chemical Society* **2017**, *139*, 14837–14840 (page 53).

[140] Zuckerman, D. M.; Woolf, T. B. Transition Events in Butane Simulations: Similarities across Models. *Journal of Chemical Physics* **2002**, *116*, 2586–2591 (page 53).

[141] Borrero, E. E.; Escobedo, F. A. Optimizing the Sampling and Staging for Simulations of Rare Events via Forward Flux Sampling Schemes. *Journal of Chemical Physics* **2008**, *129*, 024115 (page 53).

[142] Dickson, A.; Warmflash, A.; Dinner, A. R. Nonequilibrium Umbrella Sampling in Spaces of Many Order Parameters. *Journal of Chemical Physics* **2009**, *130*, 074104 (page 53).

[143] Allen, R. J.; Frenkel, D.; Ten Wolde, P. R. Forward Flux Sampling-Type Schemes for Simulating Rare Events: Efficiency Analysis. *Journal of Chemical Physics* **2006**, *124*, 194111 (page 53).

[144] Case D. A., e. a., *AMBER 2018*, 2018 (pages 59, 60).

[145] Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. Comparison of Simple Potential Functions for Simulating Liquid Water. *Journal of Chemical Physics* **1983**, *79*, 926–935 (page 59).

[146] Joung, I. S.; Cheatham, T. E. Determination of Alkali and Halide Monovalent Ion Parameters for Use in Explicitly Solvated Biomolecular Simulations. *Journal of Physical Chemistry B* **2008**, *112*, 9020–9041 (page 59).

[147] Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. A Smooth Particle Mesh Ewald Method. *Journal of Chemical Physics* **1995**, *103*, 8577–8593 (page 59).

[148] Nguyen, H.; Maier, J.; Huang, H.; Perrone, V.; Simmerling, C. Folding Simulations for Proteins with Diverse Topologies Are Accessible in Days with a Physics-Based Force Field and Implicit Solvent. *Journal of The American Chemical Society* **2014**, *136*, 13959–13962 (page 60).

[149] Nguyen, H.; Roe, D. R.; Simmerling, C. Improved Generalized Born Solvent Model Parameters for Protein Simulations. *Journal of Chemical Theory and Computation* **2013**, *9*, 2020–2034 (page 60).

[150] Kussie, P. H.; Gorina, S.; Marechal, V.; Elenbaas, B.; Moreau, J.; Levine, A. J.; Pavletich, N. P. Structure of the MDM2 Oncoprotein Bound to the P53 Tumor Suppressor Transactivation Domain. *Science (New York, N.Y.)* **1996**, *274*, 948–953 (pages 60, 66).

[151] Zwier, M. C.; Kaus, J. W.; Chong, L. T. Efficient Explicit-Solvent Molecular Dynamics Simulations of Molecular Association Kinetics: Methane/Methane, Na^+/Cl^-, Methane/Benzene, and K^+/18-Crown-6 Ether. *Journal of Chemical Theory and Computation* **2011**, *7*, 1189–1197 (page 62).

[152] Dickson, A.; Mustoe, A. M.; Salmon, L.; Brooks, C. L. Efficient in Silico Exploration of RNA Interhelical Conformations Using Euler Angles and WExplore. *Nucleic Acids Research* **2014**, *42*, 12126–12137 (page 64).

[153] Xiong, K.; Zwier, M. C.; Myshakina, N. S.; Burger, V. M.; Asher, S. A.; Chong, L. T. Direct Observations of Conformational Distributions of Intrinsically Disordered P53

Peptides Using UV Raman and Explicit Solvent Simulations. *Journal of Physical Chemistry A* **2011**, *115*, 9520–9527 (page 65).

[154] Onsager, L. Initial Recombination of Ions. *Physical Review* **1938**, *54*, 554–557 (page 67).

[155] Ryter, D. On the Eigenfunctions of the Fokker-Planck Operator and of Its Adjoint. *Physica A* **1987**, *142*, 103–121 (page 67).

[156] Du, R.; Pande, V. S.; Grosberg, A. Y.; Tanaka, T.; Shakhnovich, E. S. On the Transition Coordinate for Protein Folding. *Journal of Chemical Physics* **1998**, *108*, 334–350 (page 67).

[157] Bolhuis, P. G.; Chandler, D.; Dellago, C.; Geissler, P. L. Transition Path Sampling: Throwing Ropes over Rough Mountain Passes, in the Dark. *Annual Review of Physical Chemistry, Vol 62* **2002**, *53*, 291–318 (page 67).

[158] Rhee, Y. M.; Pande, V. S. One-Dimensional Reaction Coordinate and the Corresponding Potential of Mean Force from Commitment Probability Distribution. *Journal of Physical Chemistry B* **2005**, *109*, 6780–6786 (page 67).

[159] Berezhkovskii, A. M.; Szabo, A. Diffusion along the Splitting/Commitment Probability Reaction Coordinate. *Journal of Physical Chemistry B* **2013**, *117*, 13115–13119 (page 67).

[160] Alt, H.; Scharf, L. Computing the Hausdorff Distance Between Curved Objects. *International Journal of Computational Geometry & Applications* **2008**, *18*, 307–320, DOI: 10.1142/S0218195908002647 (page 72).

[161] Alt, H.; Godau, M. Computing the Fréchet Distance Between Two Polygonal Curves. *International Journal of Computational Geometry & Applications* **1995**, *05*, 75–91, DOI: 10.1142/S0218195995000064 (page 72).

[162] Ratcliff, J. W.; Metzener, D. E. Pattern Matching: The Gestalt Approach. *Dr. Dobb's Journal* **1988** (pages 72, 79).

[163] Onufriev, A. V.; Case, D. A. Generalized Born Implicit Solvent Models for Biomolecules. *Annual Review of Biophysics* **2019**, *48*, 275–296, DOI: 10.1146/annurev-biophys-052118-115325 (page 73).

[164] Case D. A., e. a., *AMBER 2022*, 2022 (page 73).

[165] Ward, J. H. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association* **1963**, *58*, 236–244, DOI: 10.1080/01621459.1963.10500845 (page 79).

[166] Turner, J.; Anderson, R.; Guo, J.; Beraud, C.; Fletterick, R.; Sakowicz, R. Crystal Structure of the Mitotic Spindle Kinesin Eg5 Reveals a Novel Conformation of the Neck-linker. *Journal of Biological Chemistry* **2001**, *276*, 25496–25502, DOI: 10.1074/jbc.M100395200 (page 89).

[167] Cochran, J. C.; Gilbert, S. P. ATPase Mechanism of Eg5 in the Absence of Microtubules: Insight into Microtubule Activation and Allosteric Inhibition by Monastrol. *Biochemistry* **2005**, *44*, 16633–16648, DOI: 10.1021/bi051724w (page 89).

[168] Tiwary, P.; Berne, B. J. Spectral Gap Optimization of Order Parameters for Sampling Complex Molecular Systems. *Proceedings of the National Academy of Sciences* **2016**, *113*, 2839–2844, DOI: 10.1073/pnas.1600917113 (page 90).

[169] Lee, H.; Turilli, M.; Jha, S.; Bhowmik, D.; Ma, H.; Ramanathan, A. In *2019 IEEE/ACM Third Workshop on Deep Learning on Supercomputers (DLS)*, IEEE: Denver, CO, USA, 2019, pp 12–19, DOI: 10.1109/DLS49591.2019.00007 (page 90).

[170] Ray, D.; Andricioaei, I. Weighted Ensemble Milestoning (WEM): A Combined Approach for Rare Event Simulations. *The Journal of Chemical Physics* **2020**, *152*, 234114, DOI: 10.1063/5.0008028 (page 90).