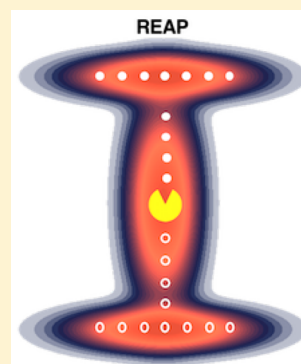


Reinforcement Learning Based Adaptive Sampling: REAPing Rewards by Exploring Protein Conformational Landscapes

Zahra Shamsi,[†] Kevin J. Cheng,[‡] and Diwakar Shukla^{*,†,‡,⊥}[†]Department of Chemical and Biomolecular Engineering, [‡]Center for Biophysics and Quantitative Biology, and [⊥]Department of Plant Biology, University of Illinois at Urbana–Champaign, Urbana, Illinois 61801, United States

Supporting Information

ABSTRACT: One of the key limitations of Molecular Dynamics (MD) simulations is the computational intractability of sampling protein conformational landscapes associated with either large system size or long time scales. To overcome this bottleneck, we present the REinforcement learning based Adaptive samPLing (REAP) algorithm that aims to efficiently sample conformational space by learning the relative importance of each order parameter as it samples the landscape. To achieve this, the algorithm uses concepts from the field of reinforcement learning, a subset of machine learning, which rewards sampling along important degrees of freedom and disregards others that do not facilitate exploration or exploitation. We demonstrate the effectiveness of REAP by comparing the sampling to long continuous MD simulations and least-counts adaptive sampling on two model landscapes (L-shaped and circular) and realistic systems such as alanine dipeptide and Src kinase. In all four systems, the REAP algorithm consistently demonstrates its ability to explore conformational space faster than the other two methods when comparing the expected values of the landscape discovered for a given amount of time. The key advantage of REAP is on-the-fly estimation of the importance of collective variables, which makes it particularly useful for systems with limited structural information.



INTRODUCTION

Molecular dynamics (MD) simulations have rapidly advanced into an invaluable tool for understanding the structure–function relationship in biological molecules.^{1–4} Although they aid our understanding of intricate biomolecular dynamics, the bottleneck lies in the amount of computational resources available to the researcher. In common practice, running MD simulations on nonspecialized computing hardware allows for nanoseconds worth of data per day.¹ The reality is that the salient protein conformational changes can occur at millisecond and even longer timescales; a six or greater order of magnitude difference in terms of nanoseconds,² which can cost up to years worth of simulation time. Examples include the transport cycle for membrane transporter proteins,^{5,6} protein folding,^{4,7–9} and large-scale conformational changes involved in cell signaling.^{10–17} A number of enhanced sampling methods have emerged to address this computational drawback of conventional simulations. Two general classes exist among these methods; one class requires the specification of order parameters, i.e., a function of system degrees of freedom that guides the simulation to reach the desired end state by enhancing sampling along the order parameter. This class can further be broken down into two subclasses, either by biasing the underlying potential along the order parameter (e.g., steered MD,¹⁸ metadynamics,¹⁹ temperature-accelerated MD,²⁰ umbrella sampling²¹) or performing unbiased adaptive sampling using the order parameters as a metric.²² The second class of techniques encourages exploration of the conformational landscape in all directions by modifying the overall

Hamiltonian (e.g., accelerated MD,²³ replica exchange MD,²⁴ or weighted-ensemble simulations²⁵). Depending on the scientific goal, the usefulness of each class of techniques will differ.

Several techniques exist that combine the ideas from these methods to achieve enhanced sampling efficiency. For example, Preto and Clementi introduced a new method called Extended DM-d-MD²⁶ that enhances the sampling of MD trajectories within areas that are typically difficult to sample such as the barriers between metastable regions. It proceeds by iteratively restarting simulations so as to obtain a uniform distribution along the first two diffusion coordinates. Since the diffusion coordinates are obtained from postprocessed simulation data, *a priori* order parameter information is not needed to perform the method. Similarly, the iMapD²⁷ method attempts to efficiently explore the free energy surface of a system using an adaptive exploration strategy; it iteratively starts new simulations at the boundary points of a lower dimensional space (in their case, diffusion coordinates) and outwardly explores the space until new metastable configurations are detected. Another method, named SGOOP,²⁸ attempts to find the best linear combination of a preselected set of order parameters using maximum path entropy estimates. This newly generated coordinate can then be used to sample along using one of the enhanced sampling method mentioned above.

Received: July 8, 2018

Revised: August 17, 2018

Published: August 20, 2018

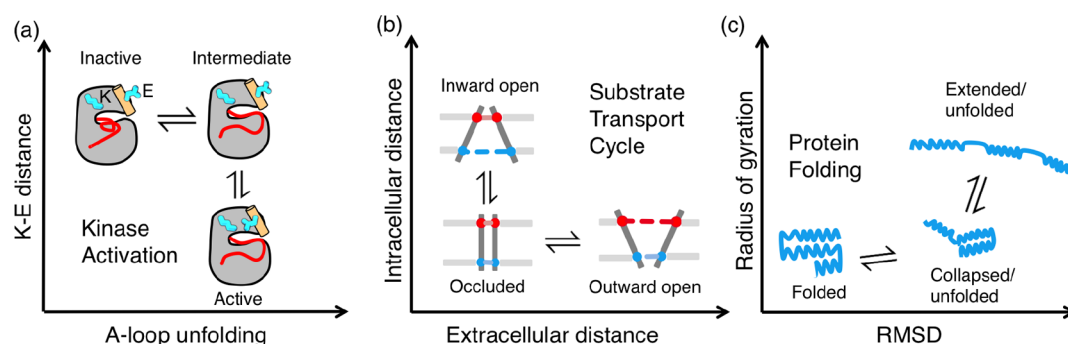


Figure 1. Three significant biological processes can be projected onto an L-shaped landscape, given the selection of appropriate order parameters. These include (a) the activation process in kinases, (b) transport cycle for transporter proteins, and (c) protein folding. In (a), the inactive kinase state is denoted as state 1, the intermediate state as 2, and the activated state as 3.

For this paper, we will focus on the issue with the first class which requires a set order parameters as an input. Essentially, this class of methods can only succeed by knowing which order parameters are relevant for sampling a priori. In the context of biomolecular simulations, order parameters are observables that capture the progress of proteins undergoing the conformational change between different states. For instance, the distance between two residues can serve as the order parameter such that they only approach each other when the protein is in an activated state. These computational techniques work by sampling the conformational space preferentially along the order parameter, “pushing” the protein process toward some final state of interest.

Order parameters have proven useful for researchers as they help reduce the high dimensionality of the system. Since protein dynamics involves highly complex processes, it is desirable to project this high-dimensional space onto order parameters that simplify the simulation data without losing essential information regarding conformational changes. Furthermore, it is not uncommon to characterize the conformational dynamics as a projection onto two order parameters (i.e., on two-dimensional space); nonetheless, one is not limited to a 2-D projection but is obviously preferred as it makes interpretation easier. This is especially useful if the protein under investigation undergoes a series of intermediate steps to achieve some final state. For instance, many protein kinases (proteins involved in signaling via phosphotransfer) only become active after two molecular events occur: the A-loop unfolding and the formation of the K–E salt bridge after the α c-helix rotates to form a K–E salt bridge.^{17,29–31} In other words, while one order parameter changes, the other remains relatively constant. Plotting the progression of these events gives rise to an “L-shaped” landscape. Kinases are not the only biological systems that can be projected onto an L-shaped landscape using two order parameters; these include membrane transporter proteins^{32,33} and protein folding^{1,4,8,9} (see Figure 1 to visualize the landscape).

It is evident from these three landscapes that as one order parameter becomes important for sampling, the other becomes less relevant. To illustrate this point concretely, consider the typical landscape associated with kinase activation.^{11,17,29–31,34} If simulations started with a kinase in the inactive state (Figure 1 a, top left), the optimal sampling strategy would be to first explore in the positive x -direction (rightward) along the A-loop unfolding coordinate. Once state 2 has been reached, the x coordinate no longer becomes relevant for sampling (since the A-loop has completely unfolded) and now the y coordinate

(the K–E bond distance) becomes the optimal direction to sample. By sampling along this L landscape, we can be assured that the most relevant protein conformations have been sampled, i.e., the complete kinase activation cycle. However, the only reason the optimal directions are known beforehand is from closely examining the differences between crystal structures captured in various states. If in this example the protein were sampled without well-established structural information, the only option is to use a “brute-force” approach by which the landscape must be explored in all directions when there is clearly an optimal path in this case.

The concept of optimal sampling along with some order parameter at different points on the landscape has led us to develop the REAP (REinforcement learning based Adaptive samPLing) algorithm that is “smart” enough to determine the relative importance of each order parameter as it explores conformational space. Reinforcement learning (RL) constitutes a significant aspect of the artificial intelligence field with numerous applications ranging from finance to autonomous vehicles.³⁵ It is based on the Pavlovian conditioning and control theory, where the feedback from the environment is learned to maximize the accumulated reward. REAP takes principles from the field of RL^{36,37} by which an agent (or learning system) takes actions in an environment to maximize a reward function. In this study, the action is picking new structures to start a swarm of simulations, while the reward is a mathematical function proportional to how far order parameters sample the landscape (see the REAP Algorithm section for more details). The agent keeps track of which direction is most rewarding, allowing it choose the optimal sampling strategy. In other words, the agent attempts to find the path of least resistance.

The REAP algorithm builds upon a count-based adaptive sampling method. Both methods perform the following: (1) Run a series of short MD simulations from a collection of starting structures. (2) Cluster the proteins based on order parameters of interest. (3) Pick structures from these clusters according to some sampling criterion to start new simulations. The difference between these methods resides in step 3, least count based adaptive sampling chooses new structures based on least populated states, while REAP chooses new structures based on a reward function. This reward function is dependent on weights (a parameter representing how important an order parameter is) and how on the landscape the new simulation data samples compared to the current data. The advantage of REAP over adaptive sampling is that it decreases the chance of choosing structures for the next round of simulations that are

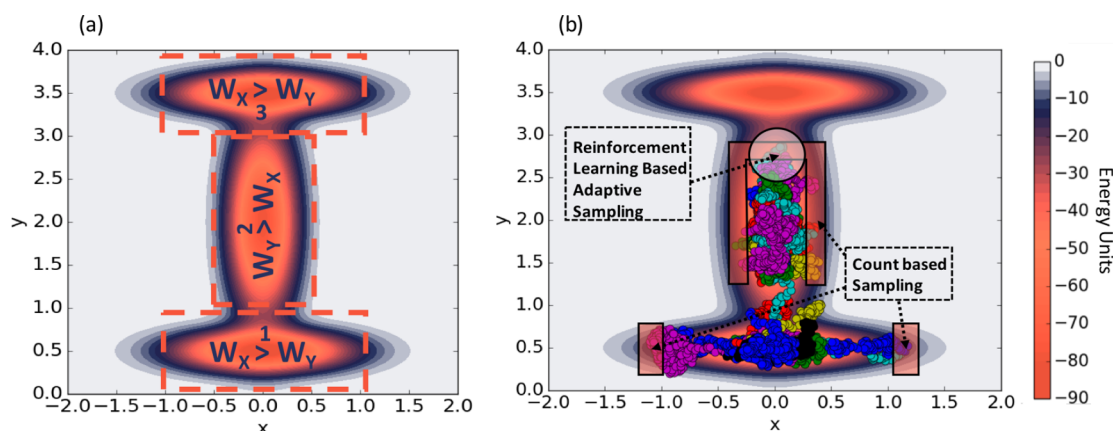


Figure 2. (a) The “I” (Illinois) landscape illustrates that the local optimal sampling strategy changes depending which basin that is currently being sampled (each labeled 1, 2, and 3). The importance of each order parameter is denoted as weights that are updated in each iteration of the REAP algorithm, W_X and W_Y . (b) Given that sampling occurs in basin 2, the orange regions shows structures selected from count based sampling are not optimal for reaching basin 3, more efficiently than the count-based sampling. Instead, REAP is able to identify the appropriate structures (white circular highlight) that facilitate sampling along Y , eventually reaching basin 3.

irrelevant for sampling given the current state. Consider the “I” (Illinois) potential in Figure 2a where the importance of the order parameters X and Y changes in each basin. In the situation, where sampling occurs in basin 2, a count-based^{38,39} adaptive sampling would give equal importance to both X and Y , allowing structures in the orange regions to be chosen for the next round of sampling. The disadvantage is that a lot of these structures are irrelevant toward reaching the final area of sampling, basin 3. The REAP algorithm is able to identify that the most important structures, that is, the white highlighted region (Figure 2b) since they facilitate sampling along the Y direction. As a result, exploring low-energy, biologically relevant regions of the landscape becomes faster, effectively saving precious computational resources for the user.

The use of reward functions to increase the efficiency of sampling has been implemented in other studies as well. For example, Zimmerman and Bowman⁴⁰ have developed one of the first goal-oriented sampling methods named FAST that consider exploration/exploration trade-offs. The reward function, in this case, is a trade-off between maximizing (or minimizing) a single collective variable and favoring poorly sampled states. REAP differs from FAST since it does not require a priori information regarding which single collective variable should be maximized or minimized such as RMSD, residue pair distance, solvent accessible area, etc. The only input needed is a list of possible order parameters. Furthermore, Perez et al.⁴¹ have used the concept relevant to RL such as “explore-and-exploit”⁴² to enhance conformational exploration using data derived from experiments. REAP differs from them as it does not require modification of the original Hamiltonian.

This paper discusses and outlines the basic algorithm of REAP and then evaluates its performance compared to conventional single long trajectories (SL) and least counts sampling (LC) using two idealized potentials; an L-shaped and a circular landscape. The algorithm is then applied to alanine dipeptide MD simulations and Src kinase. The kinase system was sampled using a kinetic Monte Carlo sampling scheme based on Markov state models obtained from a previous study.²⁹ For each case, we plotted the distribution of landscape discovered using repeated simulation trials. The expected values for the REAP distributions were consistently higher than LC and SL, suggesting that REAP is a successful improvement

of LC since it explores new areas of conformational space more efficiently. To avoid terminological confusion, we will interchangeably use “collective variable” (CV) and “order parameter” (OP) for the remainder of this article.

METHODS

REAP Algorithm. Here, we present each step involved in the implementation of the REAP algorithm. We also introduce the RL concept of a *policy* which defines the agent’s way of behaving at a given time. In a mathematical sense, the policy π is the mapping between “states” belonging to the environment and “actions” to achieve the agent’s goal $\pi: S \rightarrow A$. Put differently, the policy tells the agent how to behave at any point in time. The environment is defined as the landscape that is to be explored; with the state S defined as the set of all discovered points on the landscape or simply the current data available. The action A is defined as the agent choosing protein structures to run more simulations on. The user can provide different policies which differ in the OPs provided. By employing the sampling algorithm below, the user can evaluate which of these different policies ensures the most reward while sampling and then evaluate which OPs are relevant for sampling. To avoid any misunderstanding, the definition of “states” (S) here should not be confused with the common usage familiar to biophysicists to represent one particular protein configuration.

(1) Identify some sampling policy π_K and its corresponding set of OPs $K = \{\theta_1, \theta_2, \dots, \theta_k\}$. These OPs could be based on known or likely OPs associated with the conformational transition under investigation. Each policy differs depending on the set of OPs and how new protein structures are chosen for each round of simulation. In our implementation for this work, the sampling policy involves choosing structures based on least populated clusters (performed at step 5, denoted as C_p) and the reward function of each cluster (eq 1).

(2) Set the weight w_i for each $\theta_i \in K$ where $w_i \in [0, 1]$. The initialization of each w_i signifies which OP θ_i is important for the first round of sampling. Of course, if no prior knowledge is available regarding the importance of each weight, each w_i can be fixed to the constant value of $1/k$, where k is the total number of OPs for the given policy π_K . Every iteration of this algorithm produces a new state S (the set of all discovered

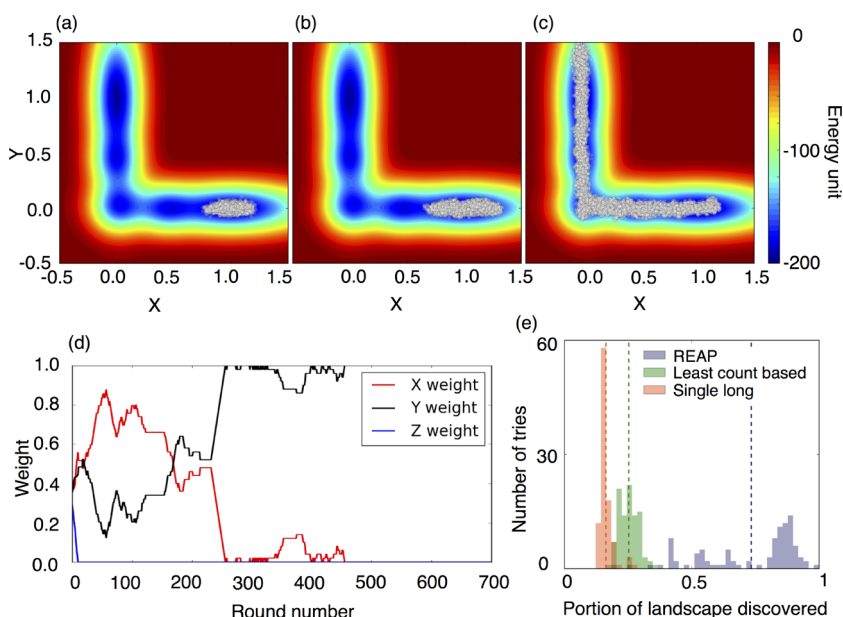


Figure 3. Regions sampled using (a) single long trajectory, (b) least count based adaptive sampling, and (c) REAP algorithm methods performed on L-shaped potentials are shown with white circles on top of the potential. The white circles represent data points generated from eq 3 (d) Weights for each OP signify the importance of each OP depending on the round number (or iteration of the algorithm). The fluctuations of weights show that the algorithm is able to identify the importance of each weight. The weight of an additional OP orthogonal to X and Y, called Z, was expected and shown to go to zero. For more information on what the weights signify, see step 2 in the REAP Algorithm section. (e) Plot showing the distribution for the portion of landscape discovered using REAP, LC, and SL sampling over 100 repeated trials. Dashed lines represent the expected value of each distribution.

points), and since w_i is different for each S , we will introduce a new notation for the weights w_i^S .

(3) Run simulations to generate a series of initial structures. This can be obtained either from a single trajectory or from running short simulations from multiple structures that can be obtained from homology modeling, crystal structures, biased MD methods, etc.

(4) Cluster the data S into a set of L clusters $C = \{c_1, c_2, \dots, c_L\}$. For each cluster $c_j \in C$, identify all the structures that are closest to the cluster c_j . The user could also assign a representative structure to each cluster, e.g., the centroid of each cluster. The goal of this step is to reduce the data size by clumping together structures in OP space. Nonetheless, the clustering method can be arbitrarily chosen during this step.

(5) Identify the set of clusters $C_p \subset C$ which contain the least number of data points. The cardinality (size) of C_p is at the discretion of the user. As mentioned in step 1, the set C_p can be obtained using different criteria.

(6) Given the set of K OPs for policy π_K , calculate the reward for each $c_m \in C_p$.

$$r^K(c_m) = \sum_{i=1}^k w_i^S \frac{|\theta_i(c_m) - \langle \theta_i(C) \rangle|}{\sigma_i(C)} \quad (1)$$

where w_i^S represents the weight or importance of each OP for a given set of discovered points S_{RL} , $\theta_i(c_m)$ is the OP calculated for the cluster c_m , $\langle \theta_i(C) \rangle$ is the arithmetic mean of θ_i for all $c_j \in C$, and $\sigma_i(C)$ represents the standard deviation of θ_i for all $c_j \in C$. Vertical bars indicate the absolute value being taken.

(7) Calculate the cumulative reward.

$$R(C_p) = \sum_{m=1}^{|C_p|} r^K(c_m) \quad (2)$$

where the sum is over each element in the set C_p , and $|C_p|$ is cardinality of C_p .

(8) The next step is to maximize eq 2 by tuning the parameter w_i^S . This can be achieved by choosing from a myriad of optimization algorithms already implemented. In our case, we took advantage of the SciPy python library⁴³ and used the Sequential Least Squares Programming (SLSQP)⁴⁴ to find the optimal weights that maximize the cumulative reward. The following conditions were enforced as a constraint: $\sum_i w_i = 1$ and $|w_i^{t-1} - w_i^t| \leq \delta$, $\forall i$, where $0 < \delta < 1$. t represents the current round of sampling while $t - 1$ represents the previous round. We found these constraints to make the algorithm more robust. The first constraint is to normalize the values of weights, while the second constraint prevents weights from changing too much from round to round. If we do not consider the second constraint, optimization algorithms would always give a weight of one to a single OP and a weight of zero to all other OPs due to the normalization factor (the first constraint). If δ is closer to 1, we let the weights to move freely regardless of the history of the reward function, just to maximize eq 2. Whereas, if δ is closer to 0, we chose the weights mainly based on the history of the reward function. Choice of δ should be a balance between these two factors. Given the updated weights, step 6 is repeated to find the new rewards.

(9) Choose the structures from the clusters that give the highest reward to start new simulations given the updated weights. The two additional parameters, structures and clusters, were chosen with the highest reward, which is up to the discretion of the user.

(10) Repeat steps 3–9 until the user deems the sampling is sufficient enough.

The primary reason for using least count adaptive sampling as specified in step 4 is because it is widely considered as the

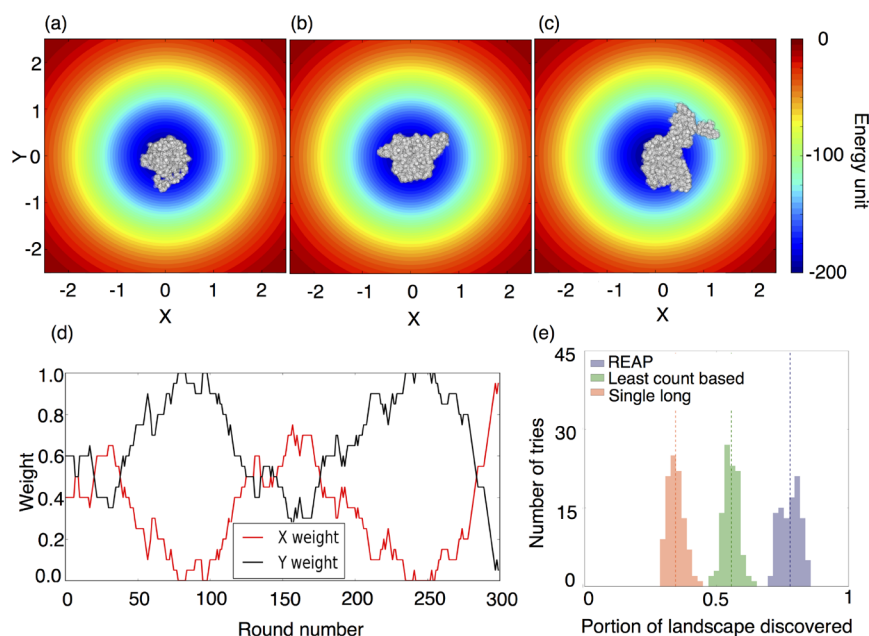


Figure 4. Regions sampled using (a) single long trajectory, (b) least count based adaptive sampling, and (c) REAP algorithm methods performed on circular potentials are shown with white circles on top of the potential. The white circles represent a data point generated from eq 3. (d) Weights for each OP signify the importance of each OP depending on the round number (or iteration of the algorithm). (e) Plot showing the distribution for portion of landscape discovered for REAP, LC, and SL simulations using 100 repeated simulations. The expectation value of each distribution is depicted with dashed lines.

most efficient strategy for exploration of free energy landscapes.^{38,39,45} Even if the reward function cannot properly assign rewards to states to achieve optimal sampling, the least-count adaptive sampling protocol will still be used. The crucial step in this algorithm is step 8, which estimates the relative importance of these OP as the agent explores the landscape. If, for example, some θ_i is provided that gives poor information on conformational changes (i.e., changes little for each round of simulations), the weights will eventually drop to zero from the optimization step. This essentially informs the user that the OP is not important for understanding the conformational changes of the protein. In theory, one can provide different policies π_K with the same reward function to determine which OPs are most relevant for sampling by simply looking how each w_i changes over time.

RESULTS AND DISCUSSION

L-Shaped Potential. To demonstrate that the REAP algorithm outperforms other sampling strategies, we will first consider an idealized system by which the time-evolution of two OPs, X and Y , are governed by the overdamped Langevin equation:

$$\dot{\mathbf{r}}(t) = -1/\gamma \nabla V(\mathbf{r}(t)) + \eta(t) \sqrt{2\beta^{-1}\gamma} \quad (3)$$

Here, $\dot{\mathbf{r}}(t)$ denotes time derivatives of the position vector \mathbf{r} , ∇ is the gradient operator, γ is the friction coefficient, $\beta = 1/k_B T$, where k_B is the Boltzmann constant and T is the temperature in Kelvin, and $\eta(t)$ represents a random force that models the collisions of molecules in a fluid. The random force obeys a Gaussian distribution with a zero mean and satisfies the following autocorrelation condition $\langle \eta_i(t) \eta_j(t') \rangle = \delta_{ij} \delta(t - t')$.⁴⁶

The goal is to study the performance of this sampling method for a model potential representing landscapes common in protein conformational change, such as ion transports,

activation processes, and protein folding (see Figure 1 for examples). We will first assume that the potential is L-shaped with five metastable states, i.e., long-lived intermediate states between the initial and final state. To compare each sampling method, we performed the same amount of simulation time on the L-shaped landscape for each sampling procedure: traditional simulation (single long trajectory, SL), least count based adaptive sampling (LC), and the REAP algorithm (Figure 3a, b, and c, respectively). All three simulations were initiated at the bottom right corner of the landscape (point (1.1, 0) in Figure 3; details of the simulations and REAP parameters are provided in Table S1). These model simulations clearly demonstrate the advantage of using REAP for exploring the landscape. This is due to the algorithm's ability to quickly identify the important directions of sampling. These directions are quantified as weights for each OP (see eq 1), facilitating immediate exploration to the left. This learning of directional importance can be visualized by looking at the change in weight values over time (see Figure 3d). We additionally included an unvarying OP, Z , to demonstrate the algorithm's capability to identify its insignificance. At the very start of the simulations, REAP quickly assigns this additional OP with zero weight, thereby preventing further sampling along this coordinate. REAP's ability to identify unimportant OPs in a more realistic system, such as Src kinase, will be elaborated on in a later section.

In the first rounds of simulations, both X and Y directions are equally important (Figure 3d up to round 10) as sampling in all directions gives equal rewards. It is not until trajectories reach the high-energy states in the Y direction where the X direction becomes more rewarding than Y . As a result, the X weight increases while the Y weight decreases. When the trajectories reach the point where X no longer becomes rewarding near round 300 (point (0, 0) in Figure 3c), then the X weight decreases and Y weight dramatically increases until

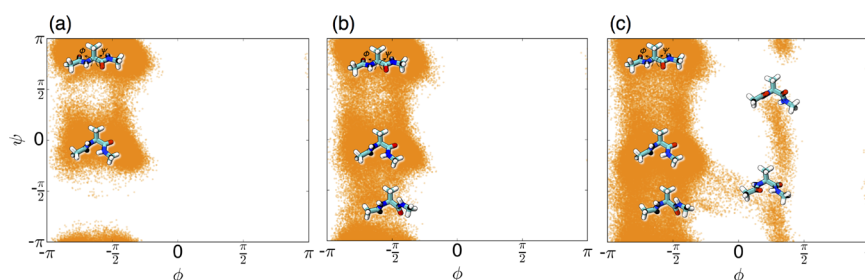


Figure 5. Alanine dipeptide landscapes were generated using three simulation methods: (a) single long trajectory simulation, (b) least counts adaptive, and (c) the REAP algorithm. Representative molecular structures are shown on top of the landscape to illustrate the new regions sampled using REAP within the same simulation time.

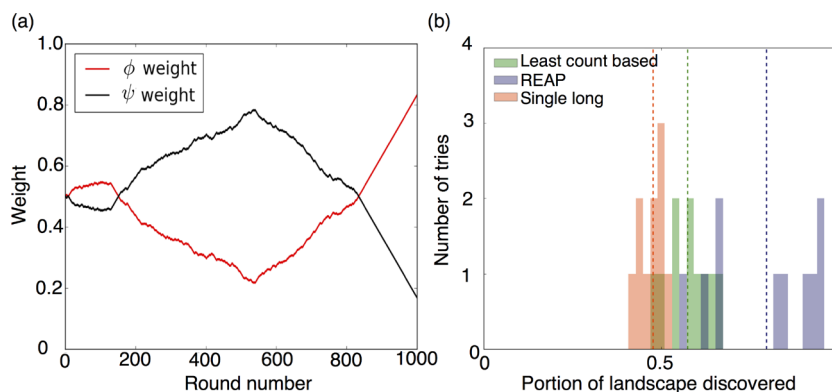


Figure 6. (a) The weights corresponding to alanine dipeptide OPs (ϕ and ψ) change over each simulation round. As a result, these change in weights demonstrate how the algorithm identifies important direction of sampling as it explores the landscape (b) A plot showing the distribution of portion of landscape discovered over 10 repeated simulation trials. Dashed lines indicate the expected value for each distribution.

the trajectories reach the fifth basin (top left corner, point (0, 1.1) in Figure 3c). It is worth mentioning that while running simulation, if a weight goes to zero, it means that the corresponding OP has not become relevant so far, but there is no way to tell if it is because of inadequate sampling or it is really irrelevant. For example, if one stopped REAP at around 50 rounds in Figure 3d, then one might conclude that the Y direction is irrelevant, but in this case, we can see that the weight on the Y direction ends up climbing to almost 1 around round 300.

To compare the performances between REAP, LC, and SL, we plotted the distribution of landscape discovered using 100 repeated simulation trials (Figure 3e). The distribution for SL and LC were concentrated around 0.2 and 0.25, respectively, while REAP is populated mostly at 0.8 with more distribution spread. Furthermore, the expected value for REAP is, at least, twice as much as LC and SL, suggesting that REAP is expected to perform better than the latter two. Additional plots showing the distribution for each individual OPs were calculated (SI Figures 1 and 2) and the time evolution of average smallest value of X_s (SI Figure 3). These different metrics provide an alternative perspective for showing how REAP outperforms the other two methods. We further assessed how REAP performs with an additional, nonfunctional OP (i.e., an OP that does not change in time). Our results suggest that the distributions were very similar compared to using two OPs (SI Figure 4).

Circular Potential. In the second model system, we tested the performance of the algorithm in which there is no important direction of sampling at any given time. Thus, we considered a single circular potential with a single metastable state. The reason for choosing this model potential is to show

that the REAP algorithm performance is the same as the least-count adaptive sampling for a system with no preferred direction of sampling. The dynamics were governed by eq 3 with a single circular metastable state (Figure 4).

We performed three sampling procedures, SL, LC, and REAP, with the same amount of sampling (details of the simulations and REAP parameters is provided in Table S1). The majority of generated points from traditional SL simulation and least count adaptive sampling were mostly confined to the stable basin (Figure 4a,b), while the REAP algorithm explored more regions of the landscape (Figure 4c). The trajectories using REAP demonstrated some directionality, attempting to sample one direction at a time. The importance of each coordinate as a function of round number is illustrated in Figure 4d. We again found that REAP outperforms both LC and SL for this landscape. This is evident from the distribution plots for portion of landscape discovered in Figure 4e. Three distinct distributions arise from this landscape, with REAP's distribution shifted toward higher values than LC and SL. REAP again demonstrates better performance relative to the other sampling methods. Since model potentials are only insofar useful for conceptual demonstrations, we will further consider two molecular systems such as alanine dipeptide, a ubiquitous benchmark in the enhanced sampling literature, and Src kinase.

Alanine Dipeptide. To illustrate that our algorithm remains effective when using MD to sample, we applied the REAP algorithm to alanine dipeptide. We performed a total of 2 ns simulations to sample the dihedral angle landscape using SL trajectories, LC, and REAP (Figure 5, details of the simulations and REAP parameters are provided in Table S1).

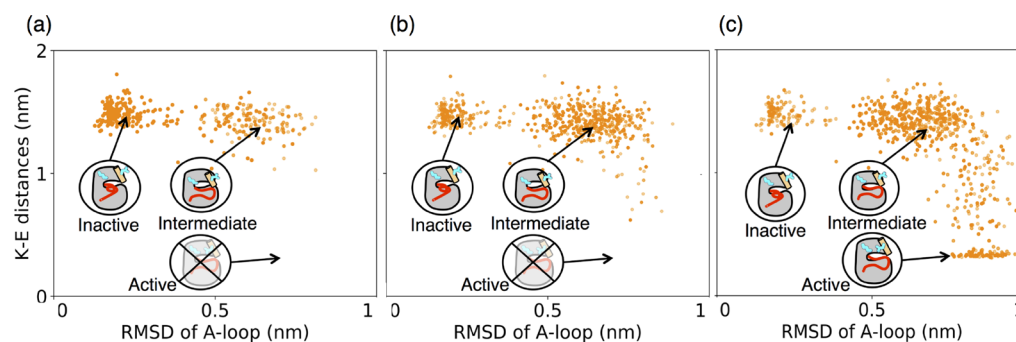


Figure 7. (a) Simulation data of Src kinase using a single trajectory approach is plotted using two OPs: K–E distances and RMSD of the A loop. Simulations started in the inactive conformation and ran for a total time of 15 μ s. (b) Simulations data using least counts adaptive sampling. Sampling improves on (a), but fails to reach the active state (c) This plot shows how the REAP sampling algorithm outperforms that of the single trajectory approach in (a). The algorithm was able to facilitate sampling of the active kinase conformation.

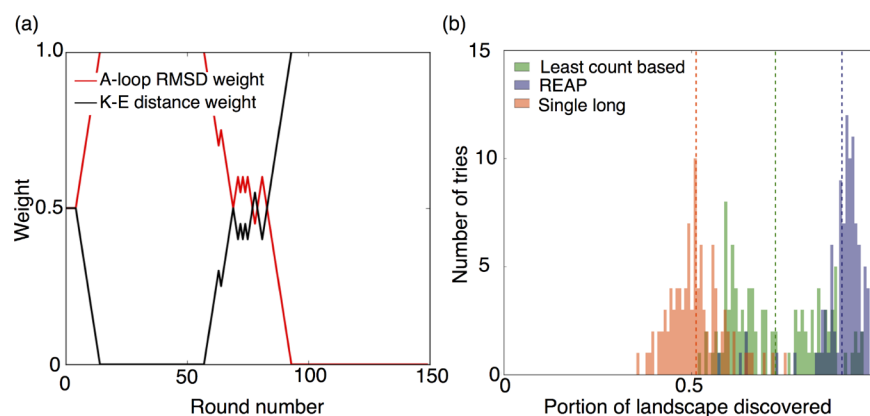


Figure 8. (a) Weights for each OP signify the importance of each OP depending on the round number (or iteration of the algorithm). (b) A plot showing the frequency of portion of landscape discovered for REAP, LC, and SL simulations for 100 trials. Dashed lines represent the expected value of each distribution.

With regard to simulation details, the starting structure of alanine dipeptide was obtained from the Python package MSMBuilder 3.8.⁴⁷ The simulation was carried out using OpenMM.⁴⁸ The AMBER99SB force field⁴⁹ was used along with the TIP3P⁵⁰ water model. A cubic box with periodic boundaries was employed to model bulk solvent. The final system consisted of 1831 atoms and simulated at temperature of 300 K while using a Langevin integrator to propagate particle motion and regulate temperature. A friction coefficient of 1.0/ps. A Monte Carlo barostat⁵¹ was used to maintain a pressure of 1 bar. To deal with long-range interactions, a nonbonded cutoff of 10 Å was used with the Particle Mesh Ewald (PME)⁵² method to calculate long distance interactions. The system was initially minimized for 1 ps (500 steps) then equilibrated for 200 ps. A total of 2 ns was generated from production MD runs. Trajectories were saved every 0.1 ps. A time step of 2 fs and hydrogen bonds were constrained for the entire simulation.

Simulations started from the most stable state of the peptide at $(-\pi/2, \pi)$ (Figure 5). After 2 ns, SL of alanine dipeptide captured two metastable state in the landscape but failed to sample new metastable states at $(\pi/4, -\pi/2)$ revealed using REAP. LC improves on SL, as it is able to better sample the regions for $\psi = \pi/4$ and $\psi = -\pi/2$. Additional landscapes are provided to visualize how the majority of REAP trials explored more regions (SI Figures 5–7). With regard to directionality, the algorithm initially starts to ϕ direction, but quickly learns that the OP that is best to sample along is along the ψ

coordinate (Figure 6a). This change in weights effectively allowed the discovery of new regions along ψ . At the second intersection shown in the plot in Figure 6 near round number 800, the least-counts sampling built into REAP (step 5) allows for sampling of an entirely new region (with ϕ taking a value near $\pi/4$). The distribution in Figure 6b illustrates that the REAP algorithm can be expected to outperform LC and SL upon inspecting the resulting expected values. It appears that two distributions arise from using REAP; one centers around 0.85 and the other around 0.6 near the LC distribution. This can be interpreted to indicate that it is unlikely that REAP will ever perform worse than LC. In addition, distributions of the individual angles, ϕ and ψ , were provided to demonstrate that REAP samples transitions better (SI Figures 8 and 9).

We considered the performance of REAP using two extra OPs, θ , and ζ . Much like the results with the L landscape (SI Figure 4), sampling of four OPs is comparable to that with two OPs (SI Figure 10). Moreover, distributions with respect to ϕ and ψ remained alike (SI Figures 11 and 12). These data indicate that REAP's performance does not diminish as extra OPs are added.

Src Kinase. We furthermore demonstrated that the REAP algorithm's effectiveness in a protein system that has implications in cancer drug discovery. To provide some background, protein kinases are a family of enzymes that catalyze the transfer of phosphate group to serine, threonine, or tyrosine residues.^{29,53,54} When the activity of this kinase becomes deregulated (perhaps due to a genetic mutation), it

can cause uncontrolled cell proliferation leading to tumor development. Extensive analysis has been done previously using markov state models (MSMs)^{3,31,55} to characterize the kinase dynamics.²⁹ Furthermore, it has been shown that the conformational dynamics can be projected onto two OPs, forming an L-shaped landscape.²⁹ This system is therefore appropriate to apply REAP, as it is a natural extension of the aforementioned L model potential. Using the same MSM from this work,²⁹ we generated trajectories using kinetic Monte Carlo (KMC) sampling. This sampling scheme uses the kinetics derived from the MSM to stochastically propagate the dynamics over time as opposed to integrating Newton's laws of motion.

We used the python package MSMBuilder version 3.6⁴⁷ to carry out our stochastic simulations. Using KMC, we generated 15 μ s of long continuous simulations and compared that to using the REAP algorithm. All of the simulations started from the same MSM state (the inactive kinase conformation), and we showed that the long simulation approach could not sample the active state within the given amount of time (see Figure 7a and SI Figure 13 for additional plots; details of the simulations and REAP parameters is provided in Table S1). The LC strategy (Figure 7b and SI Figure 14) demonstrated improved sampling for the intermediate state and the region between the active and intermediate. REAP, on the other hand, was able to discover an entire new area of the landscape corresponding to the active kinase conformation (Figure 7c and SI Figure 15), a region that the other two methods failed to sample.

A plot of the weight fluctuations is shown in Figure 8a. It shows how the algorithm initially finds sampling along the A-loop RMSD more important than the K–E distance (round 20–60). Afterward, the weights fluctuate about 0.5 until the K–E distance then becomes relevant for sampling from 90 and onward to reach the active conformation. The efficacy of the REAP algorithm is demonstrated in Figure 8b, which shows a distribution of portion of landscape discovered for 100 repeated trials. The expected values are nearly equally spaced, with REAP's distribution aggregating around 0.9. The distribution for LC is split, centering at 0.6 and 0.8. On the other hand, SL is shifted to the left, concentrated at 0.5. Given this data, REAP can be expected to explore the landscape faster than SL and LC. Not only is REAP efficient at landscape exploration, but it also reaches the active state of Src kinase in less time (SI Figure 16, 17, and 18). REAP and LC both perform better than SL, with about half the median value for the time to reach active. REAP is further shifted toward shorter times to reach the active state compared to LC, accompanied by a difference of about 5 μ s (SI Figure 19). Furthermore, REAP samples the active state and transition better than LC and SL. This is evident from the distribution of K–E distances shown in SI Figure 20.

We were further interested in determining the efficiency of our algorithm when extra nonfunctional order parameters (i.e., distances that do not change significantly over time) are considered. This was achieved by first providing one insignificant distance on the α E helix (three OPs total, SI Figure 21) then, in a separate case, 10 additional distances situated on the α E and α F helix⁵⁶ (12 OPs total, SI Figure 22). The time to reach active using three and 12 OPs are comparable with using only two OPs by a median difference of 2 and 3 μ s, respectively (SI Figure 23). Despite this difference, three and 12 OPs still maintain a faster median time than LC. We moreover plotted the time series of the average smallest

K–E distance for each instance, illustrating that introducing these nonproductive OPs will still perform better than LC over time (compare SI Figures 17 and 26). Similarly, distributions of the K–E distance for cases with multiple OPs still sample active state better than LC (compare SI Figures 18 and 27).

When contrasting the distribution for the portion of landscape discovered using three and 12 OPs (SI Figure 24), we found that the expectation values are still greater than that of the LC sampling with two OPs. Further, the distributions contain overlapping regions, with an expectation value difference of only ~ 0.1 . These results, as well as the ones of alanine dipeptide and L-shaped landscape (SI Figures 10 and 4, respectively), suggest that the introduction of additional OPs will not dramatically decrease the performance of the algorithm and can still be expected to perform better than LC. With regard to how active state sampling is affected when introducing multiple OPs, our results indicate that these superfluous distances bear no affect (compare SI Figure 20c and SI Figure 25). REAP successfully attributes low weights to these additional/irrelevant distances as their values remain low throughout the simulations (SI Figures 28 and 29).

CONCLUSION

The proposed algorithm, REAP, has been shown to efficiently sample landscapes in the case of model potentials (L-shaped, circular), alanine dipeptide, and Src kinase. It achieves this efficiency by identifying which OPs maximizes a reward function that encourages exploratory behavior. This is mathematically represented as weights, and we have demonstrated that the algorithm is able to determine which OPs are preferable while exploring the conformational landscape. In all systems that were studied, REAP consistently outperformed the traditional simulation approach and LC sampling when examining the distribution landscape discovered for the same simulation time. A rule of thumb that the slowest dynamic process captured by an MSM based adaptive sampling is comparable by the order of magnitude to the aggregate duration of all MD trajectories used to build the MSM is employed to describe ability of adaptive sampling to explore conformational landscapes.⁵⁷ In this study, we show that REAP enhances efficiency of adaptive sampling by an order of magnitude. Therefore, we expect processes that are an order of magnitude slower than the aggregate simulation time could be explored using REAP.

Regarding algorithm improvement, it is possible to introduce multiple structures at different positions along the landscape, essentially allowing the simulations to be explored from different starting points. This idea is motivated from the concept of “multi-agent reinforcement learning”^{58–60} by which agents can either interact in a cooperative or competitive fashion. One drawback of REAP is that the selection of initial OPs will most likely depend on structural or biophysical data. However, the algorithm allows for a use of a large number of OPs and the algorithm reduces the weight associated with the fast directions to zero within a few rounds of sampling. Another possible way of choosing OPs is using the approach outlined by Tajkhorshid and co-workers that involves estimating the work done by performing short pulling simulations along the OP directions associated with the conformational change process.^{6,61} If none are available, then one possibility is to use evolutionary coupling (EC) pairs⁶² as OPs (distance between residues that evolve together over time) for the given sampling policy.

In our recent work,^{22,63} we have shown that using evolutionary coupling distances as a criteria for least-counts adaptive sampling can enhance the exploration of the landscape. Given that the REAP algorithm uses this count based sampling strategy, we expect that using ECs as OPs for REAP will not only sample the landscape faster, but also differentiate between ECs that are actually relevant for conformational dynamics from those that are only important for protein folding. The separation between these two types of EC is still an open scientific question to the community.

We believe this algorithm will be particularly beneficial for those interested in building MSMs. This is because it uses the swarm of simulation approach, which runs many small trajectories in parallel. MSMs are the preferred theoretical framework at the moment to merge these discontinuous simulations and accurately reproduce the same observables as traditional MD. Additionally, REAP has advantages other than building MSMs. The principle challenge with most biased MD methods is that the original Hamiltonian is altered to preferentially sample some subset of the high dimensional space of proteins. The result of this alteration will then modify the probability distributions of protein configurations, with the possibility of favoring states that are less likely in actual biological systems. Therefore, observations on the dynamics of these biased simulations may not be useful for predicting the detailed kinetic or thermodynamic mechanism of conformational change in proteins.

■ ASSOCIATED CONTENT

● Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: [10.1021/acs.jpcb.8b06521](https://doi.org/10.1021/acs.jpcb.8b06521).

Supplementary Figures S1–S29 and supplementary Table S1 (simulations details) (PDF)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: diwakar@illinois.edu.

ORCID

Diwakar Shukla: [0000-0003-4079-5381](https://orcid.org/0000-0003-4079-5381)

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

Authors acknowledge the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Z.S. was supported by Widiger and 3M fellowships from the Department of Chemical & Biomolecular Engineering at University of Illinois, Urbana–Champaign. K.J.C. is supported by the Ford Foundation fellowship from the National Academies of Arts, Sciences and Medicine. We also thank Stanford Digital Repository for providing the Src kinase MD simulation data.

■ REFERENCES

- (1) Dror, R. O.; Dirks, R. M.; Grossman, J.; Xu, H.; Shaw, D. E. Biomolecular simulation: a computational microscope for molecular biology. *Annu. Rev. Biophys.* **2012**, *41*, 429–452.
- (2) Adcock, S. A.; McCammon, J. A. Molecular dynamics: survey of methods for simulating the activity of proteins. *Chem. Rev.* **2006**, *106*, 1589–1615.

- (3) Shukla, D.; Hernández, C. X.; Weber, J. K.; Pande, V. S. Markov State Models Provide Insights into Dynamic Modulation of Protein Function. *Acc. Chem. Res.* **2015**, *48*, 414–422.
- (4) Lane, T.; Shukla, D.; Beauchamp, K.; Pande, V. To milliseconds and beyond: challenges in the simulation of protein folding. *Curr. Opin. Struct. Biol.* **2013**, *23*, 58–65.
- (5) Jiang, J.; Shrivastava, I. H.; Watts, S. D.; Bahar, I.; Amara, S. G. Large collective motions regulate the functional properties of glutamate transporter trimers. *Proc. Natl. Acad. Sci. U. S. A.* **2011**, *108*, 15141–15146.
- (6) Moradi, M.; Tajkhorshid, E. Mechanistic picture for conformational transition of a membrane transporter at atomic resolution. *Proc. Natl. Acad. Sci. U. S. A.* **2013**, *110*, 18916–18921.
- (7) Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; Shaw, D. E. How fast-folding proteins fold. *Science* **2011**, *334*, 517–520.
- (8) Voeltz, V.; Bowman, G.; Beauchamp, K.; Pande, V. Molecular Simulation of ab Initio Protein Folding for a Millisecond Folder NTL9 (1–39). *J. Am. Chem. Soc.* **2010**, *132*, 1526–1528.
- (9) Lapidus, L. J.; Acharya, S.; Schwantes, C. R.; Wu, L.; Shukla, D.; King, M.; DeCamp, S. J.; Pande, V. S. Complex pathways in folding of protein G explored by simulation and experiment. *Biophys. J.* **2014**, *107*, 947–955.
- (10) Selvam, B.; Shamsi, Z.; Shukla, D. Universality of the Sodium Ion Binding Mechanism in Class A G-Protein-Coupled Receptors. *Angew. Chem., Int. Ed.* **2018**, *57*, 3048–3053.
- (11) Moffett, A. S.; Bender, K. W.; Huber, S. C.; Shukla, D. Allosteric Control of a Plant Receptor Kinase through S-Glutathionylation. *Biophys. J.* **2017**, *113*, 2354–2363.
- (12) Kohlhoff, K.; Shukla, D.; Lawrenz, M.; Bowman, G.; Konerding, D. E.; Belov, D.; Altman, R. B.; Pande, V. S. Cloud-based simulations on Google Exacycle reveal ligand modulation of GPCR activation pathways. *Nat. Chem.* **2014**, *6*, 15–21.
- (13) Lawrenz, M.; Shukla, D.; Pande, V. S. Cloud computing approaches for prediction of ligand binding poses and pathways. *Sci. Rep.* **2015**, *5*, 7918.
- (14) Shukla, D.; Lawrenz, M.; Pande, V. S. Elucidating Ligand-Modulated Conformational Landscape of GPCRs Using Cloud-Computing Approaches. *Methods Enzymol.* **2015**, *557*, 551–572.
- (15) Shukla, D.; Peck, A.; Pande, V. S. Conformational heterogeneity of the calmodulin binding interface. *Nat. Commun.* **2016**, *7*, 10910.
- (16) Vanatta, D. K.; Shukla, D.; Lawrenz, M.; Pande, V. S. A network of molecular switches controls the activation of the two-component response regulator NtrC. *Nat. Commun.* **2015**, *6*, 7283.
- (17) Moffett, A. S.; Bender, K. W.; Huber, S. C.; Shukla, D. Molecular dynamics simulations reveal the conformational dynamics of Arabidopsis thaliana BRI1 and BAK1 receptor-like kinases. *J. Biol. Chem.* **2017**, *292*, 12643–12652.
- (18) Grubmüller, H.; Heymann, B.; Tavan, P. Ligand binding: molecular mechanics calculation of the streptavidin-biotin rupture force. *Science* **1996**, *271*, 997–999.
- (19) Huber, T.; Torda, A. E.; van Gunsteren, W. F. Local elevation: A method for improving the searching properties of molecular dynamics simulation. *J. Comput.-Aided Mol. Des.* **1994**, *8*, 695–708.
- (20) Abrams, C. F.; Vanden-Eijnden, E. Large-scale conformational sampling of proteins using temperature-accelerated molecular dynamics. *Proc. Natl. Acad. Sci. U. S. A.* **2010**, *107*, 4961–4966.
- (21) Kästner, J. Umbrella sampling. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2011**, *1*, 932–942.
- (22) Shamsi, Z.; Moffett, A. S.; Shukla, D. Enhanced unbiased sampling of protein dynamics using evolutionary coupling information. *Sci. Rep.* **2017**, *7*, 12700.
- (23) Hamelberg, D.; Mongan, J.; McCammon, J. A. Accelerated molecular dynamics: a promising and efficient simulation method for biomolecules. *J. Chem. Phys.* **2004**, *120*, 11919–11929.
- (24) Sugita, Y.; Okamoto, Y. Replica-exchange molecular dynamics method for protein folding. *Chem. Phys. Lett.* **1999**, *314*, 141–151.

- (25) Huber, G. A.; Kim, S. Weighted-ensemble Brownian dynamics simulations for protein association reactions. *Biophys. J.* **1996**, *70*, 97–110.
- (26) Preto, J.; Clementi, C. Fast recovery of free energy landscapes via diffusion-map-directed molecular dynamics. *Phys. Chem. Chem. Phys.* **2014**, *16*, 19181–19191.
- (27) Chiavazzo, E.; Coifman, R. R.; Covino, R.; Gear, C. W.; Georgiou, A. S.; Hummer, G.; Kevrekidis, I. G. iMapD: intrinsic Map Dynamics exploration for uncharted effective free energy landscapes. *Proc. Natl. Acad. Sci. U. S. A.* **2016**, *114*, E5494–E5503.
- (28) Tiwary, P.; Berne, B. J. Spectral gap optimization of order parameters for sampling complex molecular systems. *Proc. Natl. Acad. Sci. U. S. A.* **2016**, *113*, 2839–2844.
- (29) Shukla, D.; Meng, Y.; Roux, B.; Pande, V. S. Activation pathway of Src kinase reveals intermediate states as targets for drug design. *Nat. Commun.* **2014**, *5*, 3397.
- (30) Meng, Y.; Roux, B. Locking the Active Conformation of c-Src Kinase through the Phosphorylation of the Activation Loop. *J. Mol. Biol.* **2014**, *426*, 423–435.
- (31) Meng, Y.; Shukla, D.; Pande, V. S.; Roux, B. Transition path theory analysis of c-Src kinase activation. *Proc. Natl. Acad. Sci. U. S. A.* **2016**, *113*, 9193–9198.
- (32) Chen, L.-Q.; Cheung, L. S.; Feng, L.; Tanner, W.; Frommer, W. B. Transport of Sugars. *Annu. Rev. Biochem.* **2015**, *84*, 865–894.
- (33) Wang, K. H.; Penmatsa, A.; Gouaux, E. Neurotransmitter and psychostimulant recognition by the dopamine transporter. *Nature* **2015**, *521*, 322.
- (34) Moffett, A. S.; Shukla, D. Using molecular simulation to explore the nanoscale dynamics of the plant kinome. *Biochem. J.* **2018**, *475*, 905–921.
- (35) Wulfmeier, M.; Rao, D.; Wang, D. Z.; Ondruska, P.; Posner, I. Large-scale cost function learning for path planning using deep inverse reinforcement learning. *Int. J. Rob. Res.* **2017**, *36*, 1073–1087.
- (36) Szepesvári, C. *Algorithms for reinforcement learning*; Morgan & Claypool Publishers, 2010.
- (37) Gosavi, A. Reinforcement learning: A tutorial survey and recent advances. *INFORMS J. Comput.* **2009**, *21*, 178–192.
- (38) Weber, J.; Pande, V. Characterization and Rapid Sampling of Protein Folding Markov State Model Topologies. *J. Chem. Theory Comput.* **2011**, *7*, 3405–3411.
- (39) Singhal, N.; Snow, C.; Pande, V. Using path sampling to build better Markovian state models: Predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. *J. Chem. Phys.* **2004**, *121*, 415.
- (40) Zimmerman, M. I.; Bowman, G. R. FAST Conformational Searches by Balancing Exploration/Exploitation Trade-Offs. *J. Chem. Theory Comput.* **2015**, *11*, 5747–5757.
- (41) Perez, A.; Morrone, J. A.; Dill, K. A. Accelerating physical simulations of proteins by leveraging external knowledge. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2017**, *7*, e1309.
- (42) Berry, D. A.; Chen, R. W.; Zame, A.; Heath, D. C.; Shepp, L. A. Bandit problems with infinitely many arms. *Ann. Stat.* **1997**, *25*, 2103–2116.
- (43) Jones, T. P. P.; Oliphant, E. *SciPy: Open source scientific tools for Python*; 2001–present; <http://www.scipy.org/>.
- (44) Kraft, D. DFVR, A software package for sequential quadratic programming; Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt, (DFVLR), 1988, <https://www.tib.eu/en/search/id/TIBKAT%3A016896521/A-software-package-for-sequential-quadratic-programming/>.
- (45) Bowman, G.; Ensign, D.; Pande, V. Enhanced modeling via network theory: Adaptive sampling of markov state models. *J. Chem. Theory Comput.* **2010**, *6*, 787–794.
- (46) Fogedby, H. C. Langevin equations for continuous time Lévy flights. *Phys. Rev. E: Stat. Phys., Plasmas, Fluids, Relat. Interdiscip. Top.* **1994**, *50*, 1657.
- (47) Harrigan, M. P.; Sultan, M. M.; Hernández, C. X.; Husic, B. E.; Eastman, P.; Schwantes, C. R.; Beauchamp, K. A.; McGibbon, R. T.; Pande, V. S. MSMBuilder: Statistical Models for Biomolecular Dynamics. *Biophys. J.* **2017**, *112*, 10–15.
- (48) Eastman, P.; Friedrichs, M. S.; Chodera, J. D.; Radmer, R. J.; Bruns, C. M.; Ku, J. P.; Beauchamp, K. A.; Lane, T. J.; Wang, L.-P.; Shukla, D.; et al. OpenMM 4: A Reusable, Extensible, Hardware Independent Library for High Performance Molecular Simulation. *J. Chem. Theory Comput.* **2013**, *9*, 461–469.
- (49) Hornak, V.; Abel, R.; Okur, A.; Strockbine, B.; Roitberg, A.; Simmerling, C. Comparison of multiple Amber force fields and development of improved protein backbone parameters. *Proteins: Struct., Funct., Genet.* **2006**, *65*, 712–725.
- (50) Mark, P.; Nilsson, L. Structure and dynamics of the TIP3P, SPC, and SPC/E water models at 298 K. *J. Phys. Chem. A* **2001**, *105*, 9954–9960.
- (51) Chow, K. H.; Ferguson, D. M. Isothermal-isobaric molecular dynamics simulations with Monte Carlo volume sampling. *Comput. Phys. Commun.* **1995**, *91*, 283–289.
- (52) Darden, T.; York, D.; Pedersen, L. Particle mesh Ewald: An Nlog(N) method for Ewald sums in large systems. *J. Chem. Phys.* **1993**, *98*, 10089.
- (53) Shan, Y.; Gnanasambandan, K.; Ungureanu, D.; Kim, E. T.; Hammarén, H.; Yamashita, K.; Silvennoinen, O.; Shaw, D. E.; Hubbard, S. R. Molecular basis for pseudokinase-dependent auto-inhibition of JAK2 tyrosine kinase. *Nat. Struct. Mol. Biol.* **2014**, *21*, 579–584.
- (54) Lin, Y.; Meng, Y.; Jiang, W.; Roux, B. Explaining why Gleevec is a specific and potent inhibitor of Abl kinase. *Proc. Natl. Acad. Sci. U. S. A.* **2013**, *110*, 1664–1669.
- (55) Weber, J. K.; Shukla, D.; Pande, V. S. Heat dissipation guides activation in signaling proteins. *Proc. Natl. Acad. Sci. U. S. A.* **2015**, *112*, 10377–10382.
- (56) Roskoski, R. Src protein-tyrosine kinase structure, mechanism, and small molecule inhibitors. *Pharmacol. Res.* **2015**, *94*, 9–25.
- (57) Sinititskiy, A. V.; Pande, V. S. Theoretical restrictions on longest implicit time scales in Markov state models of biomolecular dynamics. *J. Chem. Phys.* **2018**, *148*, 044111.
- (58) Watkins, C. J.; Dayan, P. Q-learning. *Machine Learning* **1992**, *8*, 279–292.
- (59) Littman, M. L. Markov games as a framework for multi-agent reinforcement learning. *Proceedings of the Eleventh International Conference On Machine Learning*. **1994**, 157–163.
- (60) Hu, J.; Wellman, M. P. Multiagent reinforcement learning: theoretical framework and an algorithm. *Int. Conf. Machine Learning*. **1998**, 242–250.
- (61) Li, J.; Wen, P.-C.; Moradi, M.; Tajkhorshid, E. Computational characterization of structural dynamics underlying function in active membrane transporters. *Curr. Opin. Struct. Biol.* **2015**, *31*, 96–105.
- (62) Hopf, T. A.; Schärfe, C. P.; Rodrigues, J. P.; Green, A. G.; Kohlbacher, O.; Sander, C.; Bonvin, A. M.; Marks, D. S. Sequence co-evolution gives 3D contacts and structures of protein complexes. *eLife* **2014**, *3*, e03430.
- (63) Feng, J.; Shukla, D. Characterizing conformational dynamics of proteins using evolutionary couplings. *J. Phys. Chem. B* **2018**, *122*, 1017–1025.