JCTC Journal of Chemical Theory and Computation

Article

Cite This: J. Chem. Theory Comput. 2018, 14, 3365−3380

pubs.acs.org/JCTC

A Probabilistic Framework for Constructing Temporal Relations in Replica Exchange Molecular Trajectories

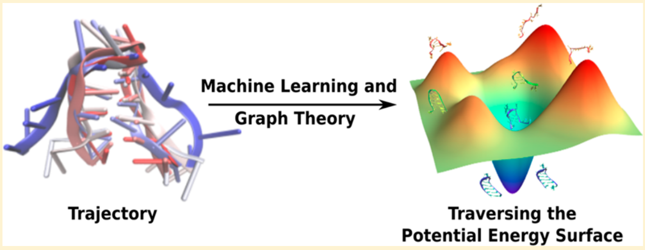
Aditya Chattopadhyay,[†] Min Zheng,[‡] Mark P. Waller,[§] and U. Deva Priyakumar*,[†]

[†]Centre for Computational Natural Sciences and Bioinformatics, International Institute of Information Technology, Hyderabad 500032, India

[‡]Centre for Multiscale Theory and Computation, Westfälische Wilhelms-Universität Münster, Münster, Germany

[§]Department of Physics and International Centre for Quantum and Molecular Structures, Shanghai University, Shanghai, 200444, People's Republic of China

Supporting Information



ABSTRACT: Knowledge of the structure and dynamics of biomolecules is essential for elucidating the underlying mechanisms of biological processes. Given the stochastic nature of many biological processes, like protein unfolding, it is almost impossible that two independent simulations will generate the exact same sequence of events, which makes direct analysis of simulations difficult. Statistical models like Markov chains, transition networks, etc. help in shedding some light on the mechanistic nature of such processes by predicting long-time dynamics of these systems from short simulations. However, such methods fall short in analyzing trajectories with partial or no temporal information, for example, replica exchange molecular dynamics or Monte Carlo simulations. In this work, we propose a probabilistic algorithm, borrowing concepts from graph theory and machine learning, to extract reactive pathways from molecular trajectories in the absence of temporal data. A suitable vector representation was chosen to represent each frame in the macromolecular trajectory (as a series of interaction and conformational energies), and dimensionality reduction was performed using principal component analysis (PCA). The trajectory was then clustered using a density-based clustering algorithm, where each cluster represents a metastable state on the potential energy surface (PES) of the biomolecule under study. A graph was created with these clusters as nodes with the edges learned using an iterative expectation maximization algorithm. The most reactive path is conceived as the widest path along this graph. We have tested our method on RNA hairpin unfolding trajectory in aqueous urea solution. Our method makes the understanding of the mechanism of unfolding in the RNA hairpin molecule more tractable. As this method does not rely on temporal data, it can be used to analyze trajectories from Monte Carlo sampling techniques and replica exchange molecular dynamics (REMD).

1. INTRODUCTION

Molecular dynamics simulations (MD) provide atomistic explanations of different phenomena exhibited by complex systems like protein (un)folding,[1−5] drug-receptor interactions,[6−10] rapid internal fluctuations, or conformational changes within macromolecules.[11−15] Given initial positions and velocities, these simulations closely follow the temporal evolution of a system in its energetically accessible phase space. This time evolution of the system is stored as positional coordinates of its atoms corresponding to each time step. Development of better theoretical algorithms and computer hardware has been conducive in extensive sampling of the phase space of these biological systems.[16−18] However, this also results in a raft of raw simulation data, which in its unprocessed form provides very little insight into the structure and dynamics of the underlying system.

Traditional methods, aimed at making these data more tractable, drastically reduce the complexity of the problem by projecting these high-dimensional positional coordinates onto a low-dimensional manifold. The characterization of this manifold is highly dependent on the chemists' expertise. For instance, protein unfolding trajectories are often analyzed by observing the time evolution of certain order parameters such as root-mean-square deviation (RMSD), radius of gyration (RGYR), fraction of native contacts, etc. The effectiveness of these methods depends heavily on the quality of the order parameters used.[19] In a more system agnostic direction, dimensionality reduction techniques like PCA have been commonly employed to reduce the degrees of freedom from a highly correlated atomic position's configuration space to a more manageable low dimensional space.[20−22] While these techniques are credible in their own right, they show a one-dimensional view of a multidimensional problem. For instance, by observing the RMSD of a protein (with respect to its native state) vs time, one can understand how much the molecule

Received: December 13, 2017
Published: May 23, 2018

DOI: 10.1021/acs.jctc.7b01245

unfolds with respect to time but stay oblivious to finer details like how different molecular configurations interacted with each other in this process. Similarly, the free energy landscape of a system undergoing a structural change like unfolding can be constructed by binning the low-dimensional coordinates along its principal components. However, this PCA projection only faintly captures the essence of the high-dimensional free energy landscape.[23]

These limitations have led to a paradigm shift in studying the energetics of a system in terms of its *metastable* states.[24,25] The dynamical evolution of a system in its configuration space can be thought of as traversing its potential energy surface (PES). Topological features like energy basins and transition states connecting them form a simple representation for analyzing the PES of a biomolecule for pathways, kinetics, etc.[26,27] The first step toward identifying these *metastable* states involves grouping different molecular configurations in the configuration space into clusters depending on a "similarity" metric.[28−31] These clusters help in identifying the different conformational substates visited by the MD simulation.[32] The "similarity" metric depends on the clustering algorithm[33] employed (k-means,[34] self-organizing maps,[35] average-linkage etc.). A good clustering algorithm should partition the molecular configuration space into distinct groups which are in good agreement with the energy basins present in its PES in an unbiased manner. Shao et al., through extensive analysis, showed that there is no such single clustering algorithm.[36] All the algorithms operate under a certain set of assumptions about the underlying data (say, number of clusters in the case of k-means, or the bandwidth of the Gaussian distributions within the data in the case of mean-shift[37]). These assumptions bias the results thus obtained.

A search for more robust techniques has led to the development of a second layer of clustering techniques over the geometric ones. These methods are called dynamic or kinetic clustering methods, as they directly tap into the temporal information available in MD trajectories of molecular systems. This kinetic information helps to differentiate between microstates that are kinetically inaccessible due to an energy barrier between them, thus reproducing the natural energy basins. Most probable path (MPP)[23] and robust Perron cluster analysis (PCCA+)[38] are some of the state-of-the-art dynamic clustering algorithms.

Having identified these *metastable* states, Markov state models (MSMs) provide a very natural way of extracting both thermodynamic and kinetic information from simulations of biological systems.[39−42] One of the successes of MSMs is that they allow the possibility of a high resolution description of the intrinsic dynamics of a system in terms of "microstates," as compared to a handful of important "states" defined by an experimental chemist. However, the efficacy of a MSM is limited to the amount of PES sampled by a MD simulation. Due to the danger of a system being trapped in a single energy basin, it is computationally infeasible to map the entire PES from a single MD simulation. Attempts have been made to construct MSMs from several short MD simulations starting from different parts of the PES.[43] However, the expanse of the PES traversed by these simulations will depend on the conformational variation of the starting structures. This is limited by the computational chemist's expertise and understanding of how the PES behaves with respect to different conformations of the system under study.

Replica exchange molecular dynamics (REMD) is an enhanced sampling method originally introduced by Swendsen and Wang.[44] In this method, several independent MD simulations are run at different temperatures with periodic exchanges between configurations belonging to adjacent temperatures. This method circumvents the problem of a system being trapped in an energy basin by performing periodic exchanges with higher temperature systems according to the Metropolis criteria. As a result, the system is able to cover a much larger portion of the PES without the requirement of several independent simulations starting from different parts of the PES. Unlike other enhanced sampling methods like umbrella sampling[45] or metadynamics,[46] REMD does not require the specification of an appropriate "reaction coordinate" and thus is useful in studying conformational changes in an unbiased manner.

The lack of temporal information between configurations, before and after a replica exchange, impedes the construction of the transition matrix for a MSM. In recent years, there have been some efforts to extract kinetic information from such discontinuous trajectories.[47−49] Buchete and Hummer[48] exploited the property that REMD allows accurate calculation of Hamiltonian dynamics on a short time scale, δ_{REMD}, between replica exchanges to obtain solutions to the master equations, which can subsequently be used to construct MSMs. One drawback of this work is that it only accounts for fast interstate transitions with a time-scale smaller than δ_{REMD}. Conformational dynamics in real systems typically exhibit relatively longer characteristic time scales. Setzl and Hummer[50] proposed a probabilistic framework of estimating interstate transitions in real systems from REMD simulations for constructing the master equations. Their method involves evaluating histograms of the "reaction coordinate" over all REMD transition paths and equilibrium trajectories. Identification of a good "reaction coordinate" for a complex biological process is an active field of research.[51−54] The transition-based reweighting analysis method developed by Wu et al.,[55] which allows for calculation of thermodynamic and kinetic properties of a system from multiensemble trajectories, can be used to successfully extract kinetics from REMD simulations. Nevertheless, it harnesses the partial temporal information available in trajectory data (simulation data between temperature swaps) and hence cannot be used in the scenario where absolutely no temporal information is available, for instance, in the case of Monte Carlo trajectories. Even in the case of REMD simulations, sometimes interstate transitions of macromolecules have a time-scale of about 50 ps.[56] The likelihood of observing this transition in a continuous segment (no temperature swap) of a standard REMD simulation, employing an exchange attempt every 2 ps, is extremely small. Let us assume that the spacing between adjacent temperature replicas was chosen such that there is significant energy overlap between them to ensure an acceptance probability of 0.2. Now, the probability for observing a continuous segment for at least 50 ps would be $0.8^{25} = 0.004$, which is negligible. Thus, this information would not likely be accounted for while estimating an MSM using the TRAM approach for such a system.

In this paper, we try to address a more general problem: "given an equilibrium sampling of the configuration space of any system, is it possible to extract the most reactive path between any two configurations?" Such a method will be beneficial in analyzing any biological process in an unbiased way without the requirement of in-depth knowledge about the

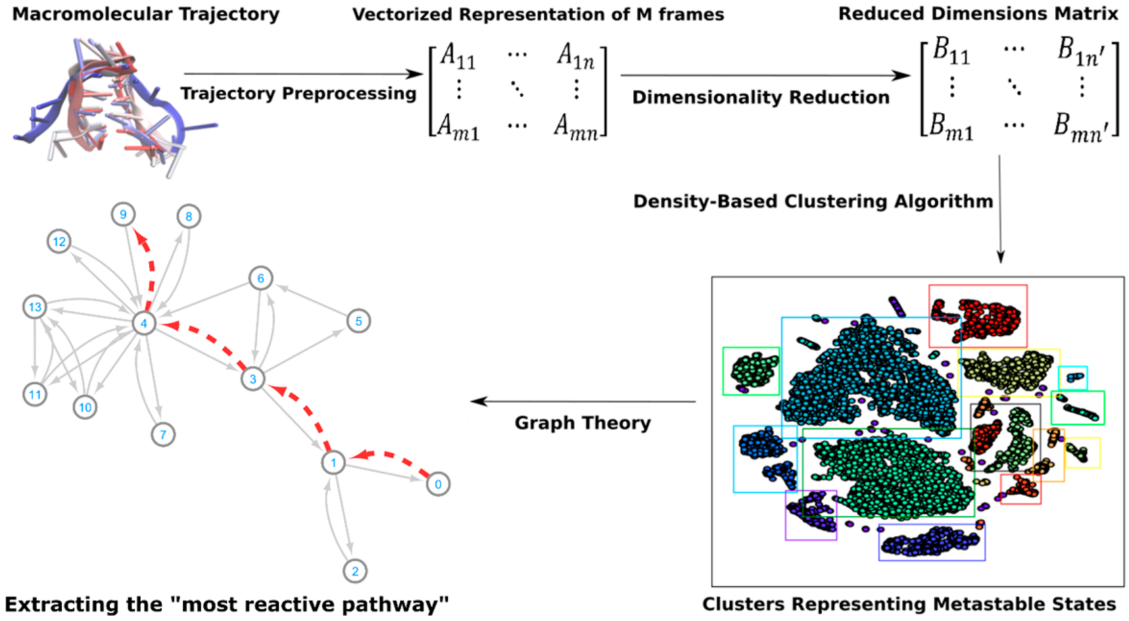

Macromolecular Trajectory
Vectorized Representation of M frames
Reduced Dimensions Matrix
Trajectory Preprocessing
$\begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix}$
Dimensionality Reduction
$\begin{bmatrix} B_{11} & \cdots & B_{1n'} \\ \vdots & \ddots & \vdots \\ B_{m1} & \cdots & B_{mn'} \end{bmatrix}$
Density-Based Clustering Algorithm
9
8
12
6
13
4
5
3
11
10
7
1
0
2
Graph Theory
Extracting the "most reactive pathway"
Clusters Representing Metastable States

Figure 1. A bird's eye-view of the overall pipeline for the proposed method.

system. Our method can also be applied to Monte Carlo simulation techniques,[57−59] where, unlike REMD, even partial temporal knowledge of the system is unavailable. In this work, we demonstrate the application of our method in analyzing RNA unfolding trajectories. The outline of the paper is as follows: First, we build the theory behind the proposed methodology. Second, we demonstrate the utility of our method for studying the dynamics of a system by comparing the results of our algorithm with that obtained from MSM analysis of a standard MD simulation of RNA hairpin in 8 M urea solution at 300 K. Third, as a proof of concept, we use our method to analyze a REMD simulation of the same RNA molecule.

2. THEORY AND METHODOLOGY

Our method for extracting dynamic information from trajectories involves (i) choosing an appropriate vector representation for the trajectory, (ii) identifying metastable states (energy basins) from the MD trajectories by clustering, (iii) creating a network with these metastable states as nodes, and (iv) constructing the most probable conformational path of the system. A bird's eye view of our methodology pipeline is illustrated in Figure 1, with each part explained in greater detail in subsequent subsections.

2.1. Choosing an Appropriate Vector Representation.

Molecular dynamics simulations produce a vast amount of data in the form of trajectories which enumerate the positions of every particle in the system with respect to time. If a trajectory has N snapshots with M atoms, it will have $N \times M \times 3$ values (factor 3 comes from x, y, and z coordinates of each atom). These positional coordinates contain all the information necessary for extracting dynamic and structural features of the system under study, within the limitations of sampling, generality of force field, etc. Using these coordinates directly to identify energy basins would introduce artifacts due to global translations and rotations of the system during the simulation. The energy of the system is only affected by internal motions within the system, like conformational changes.

Depending on the application, a suitable basis vector representation should be chosen for subsequent processing. We call this part "preprocessing of the trajectory data." Any vector basis ideally should satisfy the following properties:

i. There exists a scalar function $f{:}X \rightarrow Y$, where X represents the vector representation $(x_1, x_2, x_3, ..., x_n)$ assuming an n-dimensional representation. The range of this function f must be equal to the entire accessible PES of the molecular system. This ensures comprehensiveness of the basis chosen.

ii. There must exist a mapping function $g{:}X' \rightarrow X$, where X' represents the original Cartesian position coordinates obtained from MD trajectories. This function g must be one-to-one to ensure specificity of the representation, i.e., each unique MD snapshot X', must map to one unique point in the vector basis X chosen.

iii. The representation must be robust to changes in the system such as global translations and rotations of the system, which does not change the energetics of the system.

As the proposed algorithm in this paper involves topological features of the vector space representing the trajectory, small changes along the dimensions of the feature vector must correspond to small changes along the energy surface.

2.2. Identifying Metastable States.

Stillinger and Weber[60] showed that the partition function can be rewritten as a summation of separate contributions from quench regions along a PES. A quench region is defined by a local minimum and its immediate neighborhood in the configuration space (core region). Thus, identifying energy basins is sufficient to describe the thermodynamics of the system. This method has since been successfully applied to many different systems like condensed silicon,[61] water,[62] amorphous metal−metalloid alloys,[63] organic molecules, and proteins.[64]

The first step of our pipeline involves identifying these energy basins or metastable states from a given conformational sampling of a natural system. A system oscillates in an energy basin until it gets enough energy to cross the energy barrier and hop into an adjacent basin. T_{ii} represents transition probability that the system is in state i at time = t and again at state i at time = $t + \tau$, for any lag time τ. Internal motions within a molecular system arise from oscillations between these energy basins.[65] When a system is in any of these local minima, the transition probability can be defined as $T_{ii} \approx 0$. This is the condition for a metastable state; i.e., the system appears to be at a stable minimum if seen for a short period of time but eventually escapes to some other location on the PES.

Directly applying geometric clustering techniques like k-means, Gaussian mixture models, mean-shift, etc. to identify these metastable states can lead to erroneous partitioning of the configuration space. Without a loss of generality, let us consider a toy system described by a 1D position coordinate r. Given enough sampling, the geometric clustering algorithm will group the configurations into two clusters at 4.0 Å, the midway point. However, as shown in Figure S1 in the Supporting Information, the geometric criteria need not coincide with the energy barrier which truly separates the two energy basins, which is at 4.8 Å for the toy system under consideration. It is difficult to construct a transition matrix of different microstates or a Markov state model for dynamic clustering without information on how the system evolved over time. Boltzmann sampling of a molecular system using REMD and hybrid Monte Carlo methods[66,67] lacks this temporal knowledge for dynamic clustering.

To address this problem, we rely on certain assumptions to identify metastable states in such situations. Intuitively, when a system fluctuates in an energy basin, it gives rise to a sampling of conformations which are structurally similar. Here, by structurally we do not necessarily refer to the 3N Cartesian coordinates describing a system with N atoms. This structure can be any descriptor used to denote every configuration the system can be in, as explained in subsection 2.1. We then perform PCA on the trajectory data so as to reduce to dimensions that describe 99% of the data. The idea behind this is that as Euclidean distances between two different snapshot vector configurations are indicative of how much they differ from each other, keeping 99% of the variance would ensure that PCA nearly preserves the Euclidean distances between snapshots as the contribution of the other 1% is negligible. For all subsequent purposes, this reduced dimensional representation (subspace spanned by the principal components) is referred to as the high-dimensional data. For example, 57 principal components contribute to 99% of the variance in trajectory data corresponding to an RNA hairpin molecule at 410 K in an aq. urea solution. The projection from original 93D (this is delineated in subsection 4.1) to reduced 57D (spanned by the principal components) is referred to as the high-dimensional data.

Consider every single snapshot in a trajectory to be a point in some low dimensional manifold. We assume that the local topology of a given snapshot in this manifold has all the information necessary to characterize an energy basin. We project the nD configurations to a 2D plane using a nonlinear dimensionality reduction technique, t-distributed Stochastic Neighbor Embedding (t-SNE).[68] In this method, the dimensionality reduction is formulated as an optimization problem. The MD trajectory is comprised of N snapshots of high-dimensional objects $(x_1, x_2, x_3, ..., x_n)$.

t-SNE defines neighborhood probabilities p_{ij} as

$$p_{(j|i)} = \frac{\exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{k\neq i} \exp\left(\frac{-\|x_i - x_k\|^2}{2\sigma_i^2}\right)} \tag{1}$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \tag{2}$$

The parameter σ_i determines the bandwidth of the Gaussian distribution for the data point x_i. For each x_i, σ_i is determined using a binary search such that the induced probability distribution P_i has a fixed user-defined perplexity, $2^{H(P_i)}$. $H(P_i)$ refers to the Shannon entropy of P_i, defined as $\sum_j p_{(j|i)} \log_2 p_{(j|i)}$. It then creates a one-to-one mapping to a 2D space $(y_1, y_2, y_3, ..., y_n)$ with the local neighborhood probability q_{ij} defined as

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k\neq l} (1 + \|y_k - y_l\|^2)^{-1}} \tag{3}$$

The algorithm then tries to reproduce the joint Gaussian probabilities in high-dimensional space with a heavy tail Student t-distribution in 2D space. This is formulated by the following optimization problem:

$$\min C = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{4}$$

Intuitively, the minimum value of this function C is 0, as probabilities cannot be negative. If $p_{ij} = q_{ij}\ \forall\ i,j$, then $C = 0$. Due to higher volume, it is impossible to encode all the information encapsulated in high dimension data to a lower dimension manifold. This is known as the "crowding problem."[68] As we are only interested in the immediate local neighborhood of a point, t-SNE ensures that this information is retained. The heavy-tailed distribution attracts points in the immediate neighborhood of a reference point closer to it while all other points are repelled further away. This is delineated in Figure S2 in the Supporting Information. The figure illustrates a toy 1D example with a standard Gaussian distribution (μ = 0 and σ = 1) and a Student's t-distribution with 1 degree of freedom (Cauchy distribution). Due to the nature of these distributions used in the t-SNE algorithm, the probability of a point is positively correlated to the similarity (spatial closeness) of these points. The Cauchy distribution tries to reproduce the Gaussian distribution in lower dimension. Without a loss of generality, observing the 1D toy example it is evident that due to the heavy tails of the Cauchy distribution a value of 0.05 will take a larger r value compared to that of the Gaussian distribution (repelling force). However, a high probability value like 0.20 has a smaller r value for the Cauchy distribution compared to the Gaussian distribution (attractive force). This ensures that points close to each other in the 2D manifold are structurally very close. We chose a 2D manifold as with higher dimensions the "crowding problem" is not properly compensated by the heavy-tail distribution. The t-SNE algorithm presents the memory bottleneck for our pipeline. It requires computation of pairwise similarities. Thus, the memory requirement scales as $O(N^2)$ with N being the number of data points. This necessitates a greater time-lag between adjacent frames than that saved by the original trajectory (usually 2 ps), so as to fit the whole trajectory in the main memory. This is further expanded upon in subsection 4.3.

Functional motions in large biomolecules, like proteins, often involve small fluctuations in structures. Inadequate sampling of some states might lead to their misassignment to one of the geometrically nearby energy basins that are well-sampled. The t-SNE preprocessing step ensures that only extremely close points in the high-dimensional configuration space lie in close proximity to each other in the 2D reduced space. We claim that such a high degree of spatial similarity necessitates that these points belong to the same energy basin, thus eliminating the need for a secondary dynamic clustering technique to align the cluster boundaries with those of the energy barriers along the PES.

These metastable energy basins can be of any arbitrary shape or size. Geometric clustering algorithms like k-means or more generalized Gaussian mixture models assumes that these clusters are Gaussian/hyperspheres in shape. Superposition of several Gaussians take any arbitrary shape in trajectory vector hyperspaces.[69] However, in the absence of a second kinetic clustering wrapper around the initial geometric criteria, this is not possible. Unlike other algorithms, density-based clustering algorithms have recently found success in analyzing simulation trajectories.[70,71] An advantage of density-based methods is that it automatically identifies the number of clusters from data and thus is adaptive to the system under study. The DBSCAN algorithm[72] identifies metastable energy basins as regions of high density points in the configuration space sampled by the trajectory. We apply this algorithm on the 2D map of the trajectory data generated by t-SNE. As proximal points in the t-SNE 2D manifold are much better indicators of membership to the same energy basin than proximal points in the original high-dimensional space, applying DBSCAN in the original configuration space would give inferior results.

The DBSCAN algorithm requires two parameters *minPts* and *eps*. The parameter *minPts* is the minimum number of points required to form a cluster and is derived from this minimum lifetime of a metastable state. The parameter *eps* can be thought of as the radius of the hypersphere that defines the neighborhood of each point. A system oscillates in an energy basin, gets enough energy, and suddenly hops to another energy basin. Subsequently, it can also hop back to a previously visited energy basin. However, in order to be classified as a metastable state, the system has to have a minimum lifetime in an energy basin. This lifetime depends on the nature of the system. A point belongs to any cluster only if at least one other point lies inside its *eps* neighborhood, else the point is classified as an outlier/noise. In this work, a maximum of about 1% of the trajectory data was labeled as noise. We estimate *eps* using the method outlined by Sawant.[73] A more detailed discussion on the choice of these two parameters (*minPts* and *eps*) employed in this work is presented in subsection 1 in the Supporting Information. Note that any density-based clustering algorithm such as HDBSCAN and DBSCAN* should achieve similar results; the crucial step here is the t-SNE preprocessing step that enables the geometric cluster boundaries to align closer to the natural energy barriers without the need for a secondary kinetic clustering wrapper. This assumption is verified in subsection 4.2, where we compare the quality of our clusters to those obtained by a kinetic clustering algorithm.

Once each cluster is identified, we perform a kernel density estimation,[74,75] using the Gaussian kernel on the data points in original high-dimension space. From the learned nonparametric probability distribution for each cluster, we perform a Monte Carlo estimation of the average structure as the "representative element" for that cluster. Refer to subsection S.2 in the Supporting Information for short notes on Monte Carlo average and kernel density estimation.

This is carried out in high-dimensional space because t-SNE only preserves local topology of the vector space while totally ignoring the global arrangement of these data points. This was sufficient for identifying clusters, where only the local topology was relevant, but in order to determine how these metastable states interact with each other dynamically, their relative arrangement globally is important.

2.3. Creating a Network.

After having successfully mined the *metastable* states from the trajectory, one needs a method to connect these energy basins together and map the PES for the system. To this end, we turn to statistics and graph theory. Consider the "representative element" from each of these clusters as nodes in a graph. A graph G(V, E) is a mathematical structure consisting of V, vertices/nodes, and E, edges connecting these nodes. One could construct a Markov state model to easily connect these *metastable* states with edges representing the transition probability between states, but in the absence of any temporal data, this is not possible.

Imagine the trajectory to be a set of N data points with M identified metastable states. Each of these N data points represents a state the system was in through the course of its simulation. So, say the system was in state i at time t. It has to be in one of the M metastable states at time $(t - \Delta t)$. It can be the same metastable state that state i belongs to, or some other metastable state with enough energy to jump the energy barrier in Δt time. We follow the Markovian assumption that in order to know the probability that the system is in state i at time t, we need to look no further than $(t - \Delta t)$. We adopt a probabilistic model to deal with the uncertainties regarding which metastable state the current trajectory snapshot came from.

Our method draws inspiration from a coin toss experiment with a biased and an unbiased coin.[76] Given 100 samples of heads and tails with the information of which coin resulted in which coin toss result, we can easily find the probabilities of getting heads from each coin—namely, 0.8 (biased) and 0.5 (unbiased). This is easy to estimate:

$$p(\text{biased coin gives heads}) = \frac{\#\text{ of heads by biased coin}}{\#\text{ of biased coin tosses}} \tag{5}$$

$$p(\text{unbiased coin gives heads}) = \frac{\#\text{ of heads by unbiased coin}}{\#\text{ of unbiased coin tosses}} \tag{6}$$

However, if the same data set is given to us without information regarding which toss came from which coin, these become hidden variables. We assume each toss came with some probability from each of these coins and iteratively learn these parameters using an algorithm called Expectation Maximization (EM).[77] Similarly, assume each MD snapshot can probabilistically come from each of the metastable states, the hidden variables.

Let x_i be the ith frame of the trajectory and z_j represent the jth metastable node. $p(x_i, z_j|\theta)$, represents that the system was in energy basin j at time = t and will attain snapshot i in time = $t + \tau$, τ being the Markovian lag time for the process. θ refers to the underlying parameters of the probabilistic model. These probabilities are estimated using the EM algorithm, which iteratively maximizes the log-likelihood of data. The log-likelihood of the trajectory data is

$$G = \sum_i \log \sum_j p(x_i, z_j|\theta) \tag{7}$$

Maximizing this directly is difficult due to the summation over hidden variables being inside the logarithm function. So, EM maximizes the expectation value of the log-likelihood $F =$

Expectation(G) instead. It has been shown that maximizing this F monotonically maximizes G[78] (defined in eq 7):

$$F = \sum_i \sum_j M_{ij} \log\left(\frac{p(x_i, z_j|\theta)}{M_{ij}}\right) \tag{8}$$

Here, M_{ij} represents the membership probability that x_i came from hidden variable z_j.

The algorithm involves two steps, the M step and the E step, as an alternating maximization procedure. In the E step, the probability parameters θ are kept constant while argmax_M $F(\theta,M)$ is calculated. M refers to M_{ij} $\forall$ i,j. In the M step, the membership probabilities M are kept constant while argmax_θ $F(\theta,M)$ is calculated. This is done iteratively until the expected log-likelihood F converges within a threshold value epsilon. For our method, we chose an epsilon value of 10^{-6}.

Having laid down the basic idea of the EM algorithm, we now define the nature of the probability density function used, $p(x_i, z_j|\theta)$.

$$\begin{aligned} p(x_i, z_j|\theta) &= \varphi_j \frac{\exp\left(\frac{1}{2}(x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j)\right)}{\sqrt{(2\pi)^n|\Sigma_j|}} \quad \text{if } E_i \le \epsilon_j \\ &= \varphi_j \frac{\exp\left(\frac{1}{2}(x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j)\right)}{\sqrt{(2\pi)^n|\Sigma_j|}} \quad \text{if } E_i > \epsilon_j \\ &\quad \times \beta \exp(-\beta(E_i - \varepsilon_j)) \end{aligned} \tag{9}$$

The parameters to learn θ are $\sum_j$, β, φ_j $\forall$ i, j. μ_j, E_i, and ε_j $\forall$ i, j are hyperparameters for the algorithm and are set by the user depending on the nature of the system under study. Hyperparameters refer to parameters that are user-defined and not explicitly learned by the learning algorithm. The parameter E_i depends on the total energy of the ith frame of the trajectory, ε_j depends on the "representative energy" of the jth *metastable* node, and μ_j is taken to be the vector descriptor of the "representative element" for each cluster, which is defined in the previous subsection. The "representative energy" for each cluster (*metastable* node) can be obtained in a similar way. Instead of learning a kernel density over the vector descriptors for all the structures in a cluster, we learn a probability distribution over the energies' E_i's associated with each conformation in a particular cluster. The "representative energy" is then simply the Monte Carlo average estimated from a random sampling of 10 000 points from the learned probability density for each cluster.

This probability density function is a hybrid of a Gaussian distribution and the Boltzmann distribution. Intuitively, a transition probability from a metastable energy basin j to a particular configuration i can be defined as

$$p(x_i, z_j|\theta) = p(z_j)\, g(x_i|z_j)\, a(x_i|z_j) \tag{10}$$

where $p(z_j)$ defines equilibrium probability of the metastable cluster j, used as φ_j. Without any prior knowledge about the nature of these metastable states, this value is initialized to a uniform value of 1/(# of metastable clusters) $\forall$ j.

$g(x_i|z_j)$ defines the probability of generating a trial move for obtaining configuration i from energy basin j. This function probabilistically incorporates the fact that spatially closer configurations of natural systems are more probable within a small lag time τ. Most natural systems change in a well-behaved manner without sudden jumps along the PES. We model this $g(x_i|z_j)$ with a Gaussian distribution with its mean centered at μ_j, with the covariance matrix $\sum_j$ being the only learnable parameter.

$$g(x_i|z_j) = \frac{\exp\left(\frac{1}{2}(x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j)\right)}{\sqrt{(2\pi)^n|\Sigma_j|}} \tag{11}$$

For all j, $\sum_j$ is initialized to be the covariance matrix of the entirety of the trajectory data. To prevent instabilities in the covariance matrix, it is added with a small regularization factor.

The final part $a(x_i|z_j)$ is the acceptance probability of accepting a configuration i to come from energy basin j. The Metropolis criteria for accepting a new configuration are min(1, $\exp(-\beta(E_i - \varepsilon_j))$). Thus, if the energy of a given snapshot, E_i, has lower energy than the metastable cluster "representative energy," ε_j, it is always an accepted move. If the snapshot energy is higher, it is accepted by a probability of $\exp(-\beta(E_i - \varepsilon_j))$, with β being the normalization constant.

$$a(x_i|z_j) = \min(1, \exp(-\beta(E_i - \varepsilon_j))) \tag{12}$$

This in some way ensures that the equilibrium probability density learned by the EM algorithm is consistent with the Boltzmann distribution; i.e., the dynamics of the most probable path are being constructed from an NVT ensemble. The learnable parameter here is β, which is initialized to $1/(k_\mathrm{B}T)$; k_B is the Boltzmann constant and T is the temperature of the system under study. However, as we have no *a priori* information about the system and choose a maximum likelihood estimation of parameters to come up with an optimal distribution for the given sampling of the system, the β value need not be directly related to the temperature of the system.

The log-likelihood of this function, from eq 9:

$$\begin{aligned} \log p(x_i, z_j|\theta) = \log(\varphi_j) - \frac{n}{2}\log(2\pi) - \frac{1}{2}\log(\det(\Sigma_j)) \\ - (x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j) + \\ \max\left(0, \frac{E_i - \varepsilon_j}{|E_i - \varepsilon_j|}\right)(\log \beta - \beta(E_i - \varepsilon_j)) \end{aligned} \tag{13}$$

In each iteration, EM alternates between an E step and an M step until convergence. The overall idea of the algorithm is outlined as a flowchart in Figure 2. The update rules are as follows:

E step

$$M_{ij} = \frac{p(x_i, z_j|\theta)}{\sum_k p(x_i, z_k|\theta)} \tag{14}$$

M step

$$\beta = \frac{\sum_{i,j} M_{ij}}{\sum_{i,j} M_{ij}(E_i - \varepsilon_j)} \quad \forall\ i, j \text{ where } E_i > \varepsilon_j \tag{15}$$

$$\varphi_j = \frac{\sum_i M_{ij}}{\text{\# of trajectory snapshots}} \tag{16}$$

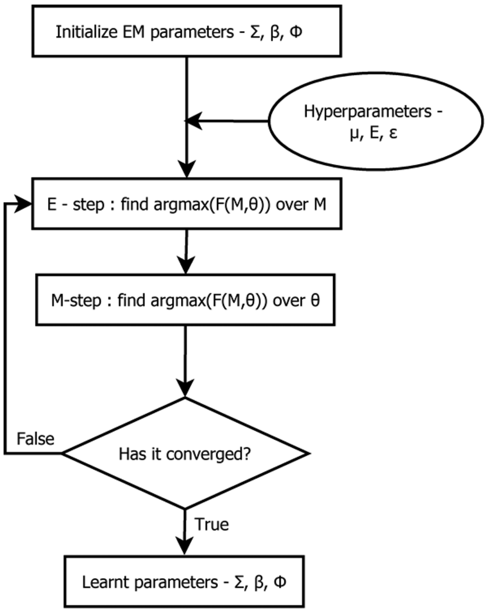

Initialize EM parameters - Σ, β, Φ
Hyperparameters - μ, E, ε
E - step : find argmax(F(M,θ)) over M
M-step : find argmax(F(M,θ)) over θ
False
Has it converged?
True
Learnt parameters - Σ, β, Φ

Figure 2. Flowchart depicting the overall flow of logic of the expectation maximization (EM) algorithm. Here μ, E, and ε are the user supplied parameters to the algorithm. E refers to the total energy of the snapshots, and μ and ε refer to the "representative element" and the "representative energy" of each cluster, respectively (refer to subsections 2.2 and 2.3 for definitions). The parameters $\sum$, β, and φ are learned from data. For initialization of these parameters, refer to subsection 2.3.

$$\Sigma_j = \frac{\sum_i M_{ij}(x_i - \mu_j)(x_i - \mu_j)^T}{\sum_i M_{ij}} \tag{17}$$

Having learned the probabilities $p(x_i,z_j|\theta)$ for all configurations sampled i and clusters identified j, we estimate the transition probabilities $\Pi(a|b)$. $\Pi(a|b)$ is defined as the probability the system will transition to energy basin a from energy basin b, in lag time τ:

$$\Pi(a|b) = \sum_i 1_a(x_i)\, p(x_i, z_b|\theta) \tag{18}$$

The index i runs over all the conformations sampled in the trajectory, i.e., the number of snapshots in the trajectory.

$1_a(x_i)$ is an indicator function with a being a subset of all frames of the trajectory belonging to metastable cluster a.

$$\begin{aligned} 1_a(x_i) &= 1 \quad \text{if } x \in a \\ &= 0 \quad \text{if } x \notin a \end{aligned} \tag{19}$$

These transition probabilities, $\Pi(a|b)$, denote directional edges between metastable node b to metastable node a in the digraph $G = (V, E)$.

2.4. Extracting Most Probable Conformational Paths.

Finally, having obtained a graph with metastable states as nodes and transition probability weighted edges connecting them, one needs a way to extract some dynamical information from this graph. Over the years, numerous efforts have been made to study macromolecular dynamics using graphs.[79−81] Here, we present a simple way to extract some relevant information about the conformational dynamics of the system with ideas inspired from transition path theory.[82]

Any conformational change can be characterized by an initial state, final state, and intermediates connecting them. The final state is not necessarily unique. There is an ensemble of pathways possible from a given initial state to a final state. This method provides a systematic way for extracting the "most reactive" conformational path. Every molecular process can be visualized as a finite sequence of edges from a starting state to a destination state. The rate of such a reaction would depend on the rates at which these edges (transition between intermediate states) are traversed. The edge with the slowest rate is the rate limiting step of this process and thus the "bottleneck" of the said reaction. We define the "most reactive" path as that sequence of edges from the given digraph $G = (V, E)$ with the fastest rate-limiting step. The edges of the digraph represent transition probabilities from one energy basin to another as described in the previous section. Let us assume $\Pi(a|b)$ (directed edge from b to a) is of value 0.4. This means that if 100 molecules are fluctuating at energy basin b at a given time t, about 40 of them would transition to energy basin a in time $t + \tau$. With this knowledge, we define the "bottleneck" of a reactive path as the smallest edge along that path. The "most reactive path" would then be a path p whose smallest path is the largest possible from a set of all candidate paths connecting the initial state to the final state. This is known as the "widest path problem." The widest path need not be unique as multiple paths can have the same smallest edge. To ensure uniqueness of the "most reactive path," we define paths recursively. Let **P** be a set of candidate widest paths from A to B. The set **P** can be partitioned into two paths (P_l, P_r) by removing the smallest edge. For instance, if a candidate path was A, ..., x, y, ..., B, with edge $E(x,y)$ being the smallest edge, P_l will include the paths with A as the initial state and x as the final state and P_r would include the paths with y as the initial state and B as the final state. We recursively compute the "widest path" with these modified start and end states and prune paths from P_l and P_r. We continue this process until only one path remains, the "most reactive path." The rationale behind this pruning of candidate paths is that in a stepwise molecular process, the fastest process will be the one where every step is the fastest possible. This is achieved using a modified version of Dijkstra's algorithm[83] as explained below. We use a max heap priority queue, which is an efficient data structure for querying and popping the element with the highest key value for a given array of elements.

Input: Digraph $G(V, E)$, start vertex $s \in V$, matrix c containing edge weights for all $(u,v) \in E$.

Output: Most reactive path P

```
• Initialize dist[s] = $\infty$, prev[$s$] = $-1$
• For each vertex $v \in V - \{s\}$, initialize dist[$v$] = $-\infty$, prev[$v$] = $v$
• Insert all vertices in a priority queue (max heap) $Q$ with [key,value] = [dist[$v$],($v$, prev)]
• While $Q$ is not empty −
    • Remove topmost element from heap $u$.
    • For all vertices $v$ such that edge $(v,u) \in E$ −
        • new_dist = max(min(dist[$u$], $c(u,v)$), dist[$v$])
        • Update corresponding key values in the priority queue, with corresponding parent node
• Retrace the prev array backward from the end state to the start node $s$ to find the edges involved in the widest path.
```

The greedy aspect of Dijkstra's algorithm ensures that each subpath u_1, u_2, ..., u_k is no better than the path u_1, u_2, ..., u_l ..., u_k

selected by the algorithm $l < k$ in terms of the cost (path, with largest possible value of smallest edge, connecting node 1 to node l). Greedy algorithms is a class of computational algorithms which, at each step, locally makes the choice which provides maximum benefit to the cost it optimizes.[84] The algorithmic complexity of Dijkstra's widest path algorithm is $O(|E| + |V| \log|V|)$, where $|E|$ denotes the number of edges in the graph and $|V|$ represents the number of vertices/nodes.

The implementation of our algorithm is available free of charge at https://bitbucket.org/adityababiblue1994/macromolecule_unfolding. We utilized the scikit-learn[85] and Theano[86] machine-learning frameworks. SciPy,[87] numPy,[88] mdtraj,[89] and matplotlib[90] modules were also used. A standard .dcd file, along with the corresponding .pdb file is needed as input.

3. SIMULATION METHODS

To test the validity of our method, we simulated three unbiased MD trajectories of a RNA hairpin loop (with sequence GGGCGAAAGCCU) in 8 M urea solution for 300, 100, and 100 ns at 300, 360, and 410 K, respectively. The secondary structure of the RNA hairpin is provided in Figure 3. The RNA hairpin unfolds within 20 ns above 360 K but takes about 189 ns to unfold at 300 K; hence, a 300 ns time scale was used at this temperature alone to properly sample the unfolding phenomena. CHARMM all-atom force field for nucleic acids[91] and the CHARMM general force field (CGenFF) for urea[92] were employed, and NAMD 2.12[93] was used to run these simulations. All simulations were carried at a constant pressure of 1 atm using a Nosé−Hoover Langevin piston. The piston period and decay parameters were set to 100 and 50 fs, respectively. Initially, the RNA molecule in the aqueous urea solution was subjected to an initial 5000 step energy minimization, followed by, first, a 200 ps equilibration run at constant temperature, with constraints on the heavy atoms in the RNA moiety, and a subsequent 1 ns NVT run without any constraints.

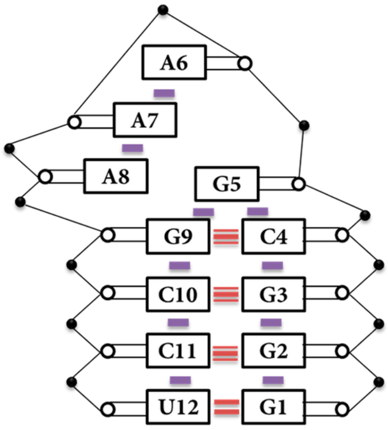


Figure 3. Secondary structure representation of the RNA hairpin moiety that is used as a model system to verify the utility of our algorithm. The naming convention of the nucleobases adopted in this paper are illustrated in this figure. Hydrogen bonding and stacking interactions found in the experimental structure are shown in red and purple colors, respectively.

For proof of concept, we also performed a REMD simulation of the same RNA hairpin molecule in 8 M aq. urea solution using the NAMD engine. We simulated 48 replicas distributed over a temperature range −[300 K, 400 K]. The spacing between the temperatures of each individual replica was chosen such that there is significant energy overlap between them to ensure an acceptance ratio of about 20%. Each replica was simulated at constant volume and temperature (using a Langevin thermostat) for 30 ns with an exchange attempt every 2 ps. The overall simulation time was about 1.4 μs. The final replica trajectories were unshuffled and the trajectory corresponding to 400 K was chosen for our analysis, as complete denaturation of the hairpin loop takes place at this temperature.

4. RESULTS AND DISCUSSION

4.1. Appropriate Descriptors for a RNA Hairpin in 8 M aq. Urea Solution.

The first step of our algorithm involves specifying a suitable vector basis to describe each snapshot in a trajectory, consistent with the properties outlined in subsection 2.1. The model system chosen was an RNA hairpin in 8 M urea solution at 300, 360, and 410 K, respectively. Hydrogen-bonding and base stacking are the driving forces behind stabilization of RNA molecules in their native state.[94−96] We calculate interaction energies between each possible base pair, the self-energy of the backbone using the CHARMM force field. Urea is a well-known chemical denaturant and has been shown to assist in the unfolding of proteins and RNA.[97−101] It stabilizes the unfolded structure compared to the native folded state. In order to incorporate solvent effects into our basis representation, urea−base, water−base, urea−backbone, and water−backbone nonbonded interactions were chosen. We note in passing that we also tried a distance-based vector representation as that too satisfies properties i, ii, and iii (given in subsection 2.1). A 12 × 12 matrix D was constructed for each MD snapshot. Each element D_{ij} represents the distance between center of geometry of base i and base j. These matrix elements were then concatenated to give the desired vector basis for each frame of the trajectory. However, this representation did not give desirable results. This inaccuracy can be attributed to the fact that small changes in any matrix element D_{ij} do not give rise to similar changes along the PES. For example, if a GUA base flips out, such that the hydrogen bonding between a GUA-CYT pair and stacking interactions with adjacent bases get compromised, there will be a considerable change along the PES.

Assume

$$V' - V = \Delta \tag{20}$$

V' represents the new distance vector basis after the GUA base flips outside, and V represents the original distance vector basis before the GUA base flips outside

Now if the RNA molecule is fully unfolded, a same shift in the basis vector by Δ will lead to a negligible change in the energy along the PES. This is because once the molecule is fully unfolded, strong base−base canonical interactions are absent. This problem does not exist if interaction energies are directly taken as dimensions of the descriptive vector for each MD snapshot.

Each MD snapshot is represented by a 93 dimension vector. The hairpin moiety has 12 nucleobases, which gives rise to 66 dimensions for each possible base-pair combination, 12 dimensions involving nonbonded interactions of urea with each of the 12 nucleobase and similarly 12 dimensions for water−base nonbonded interactions. Note that all the solvent interactions are averaged for every nucleobase. The remaining

DOI: 10.1021/acs.jctc.7b01245

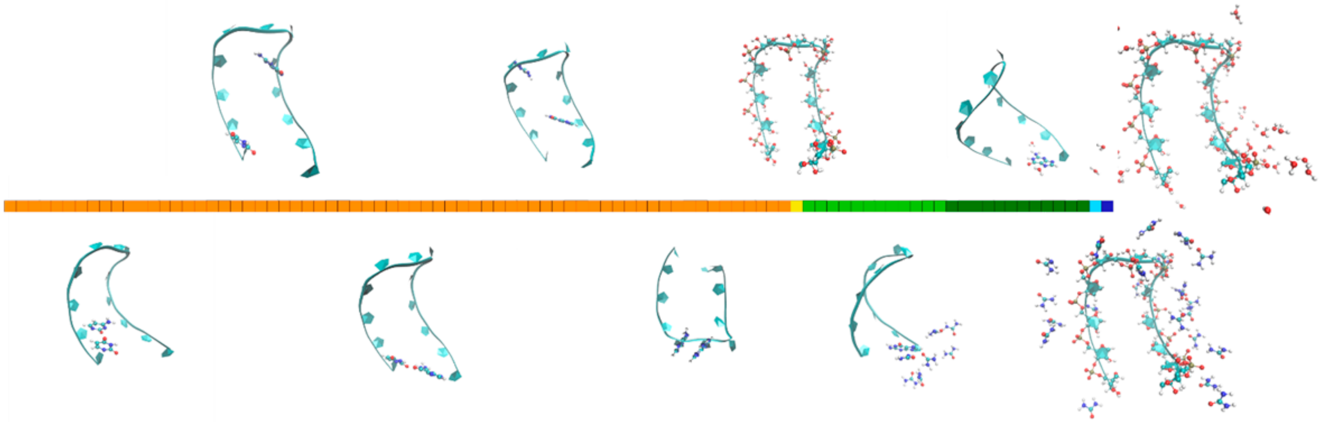


Figure 4. Each frame in the trajectory is represented by the above vector representation. Each square represents a dimension and is color coded according to the normalization groups (see subsection 4.1). The atoms represented by the ball-and-stick model denote the interacting entities. Orange, base−base interaction energy; yellow, backbone conformation energy; light green, base−urea interaction energy; deep green, base−water interaction energy; light blue, backbone−urea interaction energy; deep blue, backbone−water interaction energy.

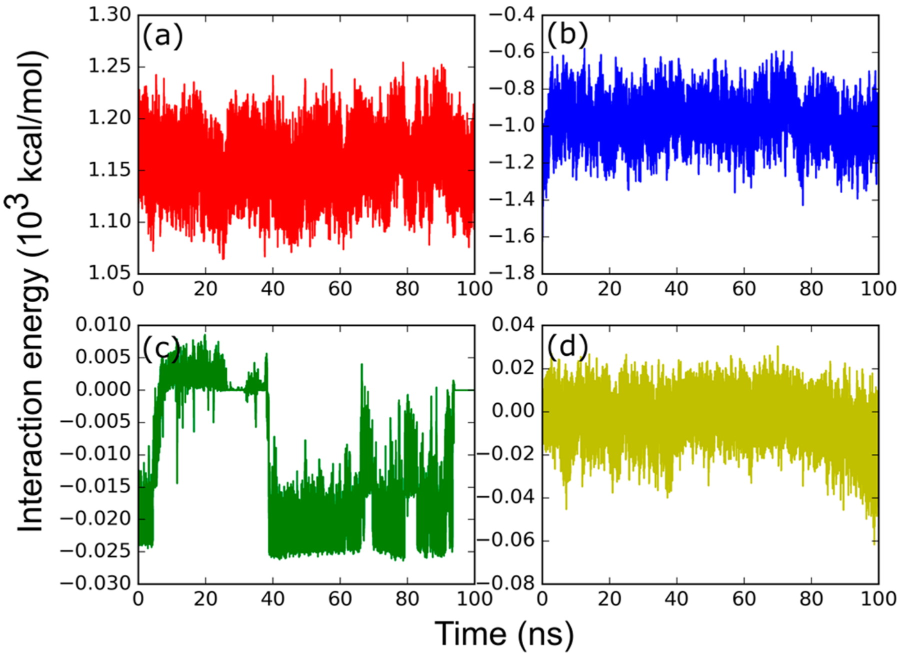


Figure 5. Interaction energy vs. time plot for the RNA hairpin system in 8 M aq. urea solution as the unfolding event progresses. (a) RNA backbone conformational energy, (b) RNA backbone−urea interaction energy, (c) base-pair interaction (CYT-GUA), (d) base−urea interaction (GUA).

three dimensions come from conformational energy of the RNA backbone, urea−backbone interactions, and water−backbone interactions. For the solvent−backbone interactions (namely, urea−backbone interactions and water−backbone interactions), the sum total of the nonbonded interactions (CHARMM all-atom force field) between each solvent atom and each atom that constitutes the backbone of the RNA molecule, i.e., alternating sugar and phosphate groups, was considered. The entire trajectory having N frames can be represented by N vectors of 93 dimensions each. Figure 4 represents the basis vector representation used for describing the molecular system. It is trivial to show that such a representation is consistent with the properties mentioned in subsection 2.1:

i. Usually, the total potential energy of the system can be parametrized over its individual bonded and nonbonded interactions in an additive way. As each dimension of our basis corresponds to an agglomerative version of these energy components, the existence a scalar function f:$X \rightarrow Y$ is guaranteed. Here, X refers to the vector basis and $Y \in$ entire accessible PES of the system.

ii. The MD force fields typically parametrize every interaction over the position coordinates X' of the system, either explicitly in the case of nonbonded interactions or implicitly in the case of bonded interactions. It directly follows from this that as individual energy components are functions of X', there must exist a mapping function g:$X' \rightarrow X$, where individual dimensions of X are coarse-grained versions of these fine-grained components.

iii. The bonded and nonbonded energy terms of a MD force-field depend on the relative positions of the participating atoms, rather than their global coordinates. This ensures that our basis representation involving force field energies is translational and rotational invariant.

This vector basis is not normalized and can lead to a bias toward dimensions with larger values during subsequent clustering. This is because the absolute value of base-pair interaction energies are on the order of 0−30 kcal/mol, while the backbone conformational energy is on the order of (1.1−1.3) × 10^3 kcal/mol. Similar variation is observed in the base−solvent and backbone−solvent interaction energies. Intuitively, the folded state, partially unfolded, and fully unfolded state all belong to different energy basins. The feature vector associated with each of these configurations of the RNA molecule will have maximum variation in the dimensions representing base−base interactions (hydrogen bonding and stacking) as compared to changes in the backbone self-energy. Similarly, base−solvent interactions show most variation compared to backbone−solvent interactions. Figure 5 shows how different interactions involving the RNA molecule evolve with time (as the unfolding event progresses). We perform a min−max normalization such that all dimensions are within [−1, 1] range. This is done by dividing the vector into four normalization groups:

- Dimensions (1−66): base−base interactions
- Dimension (67): self-energy of backbone
- Dimensions (68−91): solvent-base interactions (solvent includes both urea and water)
- Dimensions (92−93): solvent-backbone interactions

Min−max normalization was applied separately to each of these groups using the following formula:

$$y' = \frac{2y - (\max + \min)}{\max - \min} \tag{21}$$

y' represents the normalized value of that dimension of a given vector, and y represents the original value of that dimension of a given vector. The parameters max and min, represent the maximum and minimum value of all dimensions in that group from all the vectors in the trajectory, respectively. This normalization ensures that the clustering algorithm looks at all dimensions of the feature vector equally.

4.2. Partitioning of the Configuration Space.

Having arrived at a suitable vector description of the system, the next step involves grouping the conformations into metastable regions. This is achieved via the clustering algorithm described in subsection 2.2. Python's machine learning module scikit-learn's t-SNE and DBSCAN implementation[85] was used for this paper. All default parameters were used except the ones explained below. For all experiments, the perplexity parameter for t-SNE was set to 40, owing to the highly dense simulation data. To estimate the *eps* parameter of DBSCAN, we followed the method outlined by Savant.[73] The fourth nearest neighbor distance was calculated for all data points in 2D reduced conformational space. The 99th percentile distance value of this data set was set as the *eps* value for that system. For our hairpin loop RNA system, we used a 100 ps minimum lifetime for defining a metastable state. How this translates into a *minPts* (another DBSCAN hyperparameter) value is explained later.

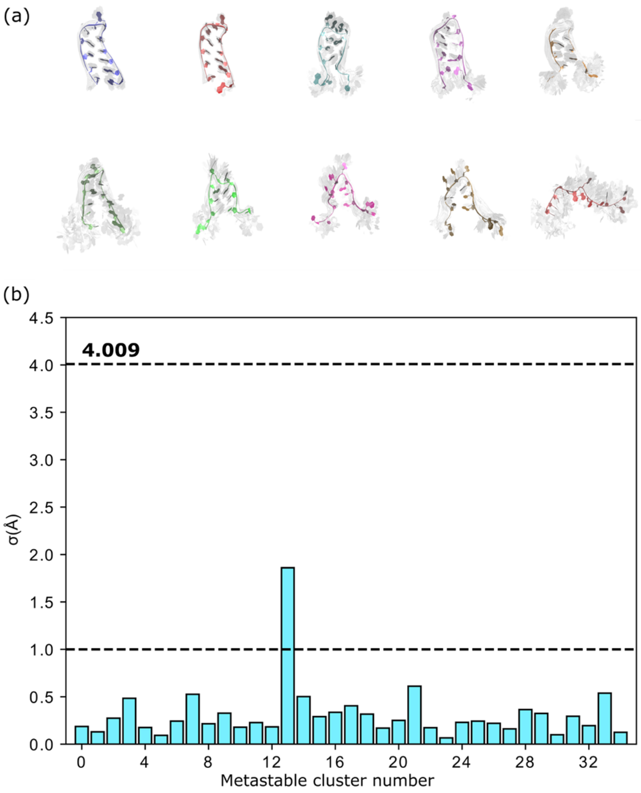


Figure 6. (a) Select metastable states of RNA hairpin molecule in 8 M urea at 300 K. The colored structure represents the local minima of each of these clusters, while the gray structures represent internal fluctuations within each cluster. (b) σ(Å) refers to the intracluster standard deviation in RMSD values of all frames that belong to a certain cluster. For each cluster, the local minimum was chosen as the reference for calculating the RMSD values. Each bar in the plot represents a metastable cluster identified by the "probabilistic" algorithm. The dotted line at 4.01 Å represents the standard deviation in RMSD values for the entire trajectory. Only one of the clusters identified has σ(Å) > 1.0 Å, the metastable cluster number 13 (1.86 Å).

The metastable states visited by a 300 ns MD run of a RNA hairpin molecule in 8 M urea at 300 K is depicted in Figure 6a. The colored structure represents the local minima of each of these clusters, while the gray structures represent internal fluctuations within each cluster. This local minimum for each cluster was determined using the conjugate-gradient descent minimization module of the NAMD program[93] by choosing the trajectory snapshot closest to the cluster's "representative element" as the starting points and running a 20 000 step minimization. As evident from the figure, there are much fewer fluctuations from the local minima for each of these clusters. In a more statistical direction, Figure 6b shows the intracluster standard deviation in RMSD values for all the identified metastable clusters for a 300 ns unbiased MD run for the RNA hairpin loop in 8 M urea and 300 K. For calculating RMSD values, hydrogen atoms were not considered. For each cluster, the local minimum was taken as the reference structure. Except metastable cluster 13, all other states have an intracluster RMSD standard deviation value less than 1.0 Å. Cluster number 13 has the highest standard deviation of 1.86 Å and represents a completely unfolded RNA hairpin molecule. Fluctuations in the backbone in its unfolded state result in a negligible traversal along the PES as all stable base-pair hydrogen bonding and stacking interactions have been effectively broken. Thus, although the spatial structure has a comparatively large deviation, the structures are very close in the PES. The black line at 4.009 Å represents the standard deviation of the RMSD values for the entire trajectory taken as a whole (here, the initial native state was chosen as a reference). As expected, this value is much larger than any intracluster RMSD standard deviation. These results suggest that our hypothesis that "structures in the same energy basin exhibit similar structural characteristics" is credible.

To test the quality of our algorithm, we employ the most probable path (MPP) algorithm introduced by Jain and Stock to partition the given trajectories into dynamic clusters.[23] These dynamic clusters were treated as the gold standard, and adjusted mutual information (AMI)[102] was used to measure the quality of clusters obtained from our method. An AMI value of 1.0 represents perfect matching of the two sets of clusters, and a value of 0.0 denotes that the two sets are dissimilar or similar by random chance. Refer to subsection S.3 in the Supporting Information for a short note on the intuition and formula of the AMI metric. To eliminate ambiguity, we will refer to the clusters identified by our method as "density clusters" as opposed to the "dynamic clusters" used for quality evaluation.

As mentioned in subsection 2.2, the memory bottleneck for our method is the t-SNE algorithm. In the original trajectories, snapshots were saved every 2 ps. However, for an in-memory calculation of the similarity matrix (required for the t-SNE optimization), we skip frames to ensure the number of snapshots analyzed is about 9000. MPP has no such memory limitation and so was applied on the entire trajectory. In other words, the lag time for the MPP Markov chains was taken to be 2 ps for all the experiments. The $Q_{\min}$ parameter of MPP was set by cross-validation, maximizing the AMI score with the "density clusters." $Q_{\min}$ can be thought of as the level of granularity of these "dynamic clusters." As our method lacks any sort of hierarchical clustering, we tune the $Q_{\min}$ parameter such that the level of description of the PES is similar in both the methods. A major drawback of the dynamic clustering method is that it relies on the assumption that intracluster transitions are more probable than intercluster translations. The validity of this assumption is directly correlated to the barrier heights between energy basins along the PES. At higher temperatures, these barrier heights decrease, allowing the system to easily sample a wider conformational space. This is reflected in a lower $Q_{\min}$ value at 360 and 410 K as compared to 300 K. The AMI value, which shows the amount of overlap between the "density clusters" and the "dynamic clusters," also decreases with an increase in temperature: 0.655 at 410 K < 0.774 at 360 K < 0.843 at 300 K. An AMI value of 0.843 at 300 K shows an excellent agreement between the "density clusters" and the gold standard "dynamic clusters."

4.3. Comparative Analysis of the Probabilistic Framework with MSMs.

The final transition matrix obtained from the MPP algorithm was processed using the definition of "most reactive path" in subsection 2.4. In all the systems, the cluster pertaining to the starting snapshot was taken as the initial node, and the cluster whose "representative element" had the greatest SASA (Solvent Accessible Surface Area) was considered to be the destination node. The pathway predicted by the MPP algorithm is denoted as the "Markovian" path, and the pathway predicted by our algorithm (using the method outlined in subsections 2.3 and 2.4) is referred to as the "probabilistic" path. Note that while both the "Markovian" and the "probabilistic" path are evaluated over the same MD trajectory, the computation of the "Markovian" path utilizes the sequential information in the simulation, whereas the "probabilistic" path uses only the spatial information in each frame for its analysis. Refer to Figure 3 for the naming convention of the nucleobases used in this subsection.

To meet the memory requirements of the t-SNE preprocessing step, for the RNA hairpin in 8 M aq. urea solution at 300 K trajectory, we skip every 16 frames, making the lag time $\tau = 32$ ps. This translates into a *minPts* value of 4, as $4\tau = 128$ ps. We project both the pathways on a 2D grid with a number of base pairs and number of bases stacking as the order parameters, as shown in Figure 7. This 2D representation can be used to uniquely represent each metastable state as the RNA hairpin does not contain any internal bulges or loops. Both the "probabilistic" and the "Markovian" pathway demonstrate that, at 300 K, the unfolding mechanism proceeds with unzipping of the stem from the free end. The G1−U12 base pair opens first, followed by the G2−C11 pair. This is accompanied by a simultaneous destabilization of the hairpin loop structure with base A8 flipping out and destacking of bases A6 and A7. From this intermediate state of two open base pairs from the free end and a destabilized loop structure, the "probabilistic" pathway predicts a direct transition to a completely unfolded state. The "Markovian" pathway elucidates a more fine-grained path with an intermediary jump to a state with a further unzipping of the stem from the loop end of the C4−G9 base pair. Figure 7 shows the "representative element" of each cluster visited by the "most reactive path." The "probabilistic" pathway is color coded in blue, while the "Markovian" pathway is color coded in red.

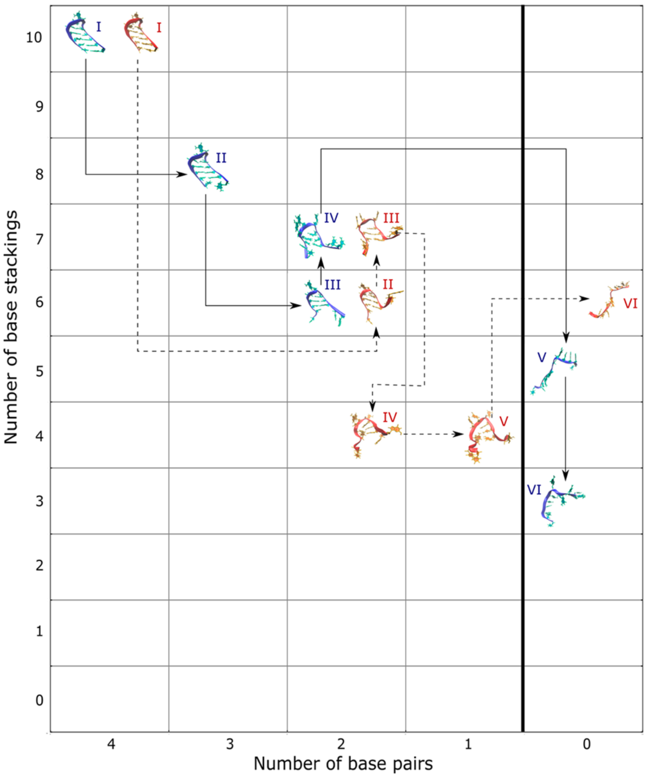


Figure 7. The "most reactive path" identified by the "probabilistic" pathway (blue) and the "Markovian" pathway (red). The structures represented are the "representative elements" of their respective clusters. Every square in the 2D grid is associated with a particular metastable state by the order parameters—number of base stacking and number of base pairs. This figure corresponds to MD simulation at 300 K.

The input parameters for both the 410 and 360 K trajectories are taken to be the same, as both these MD simulations are 100-ns-long. From the trajectory data, every six frames were skipped, making the lag time $\tau = 12$ ps. To ensure a minimum lifetime of at least 100 ps, a metastable cluster should have a minimum of nine points, $9\tau = 108$ ps. The *minPts* parameter of DBSCAN was set to 9. Both pathways highlight a very similar mechanism in both cases as illustrated in Figures S3 and S4 in the Supporting Information, for 360 and 410 K, respectively. At 360 K, the "Markovian" pathway again shows a more nuanced mechanism compared to the "probabilistic" pathway.

Thus, in all three systems, by purely using the spatial information in the MD trajectories, we were able to qualitatively reconstruct the most probable path to an appreciable degree of accuracy. Refer to subsection S.4 in the Supporting Information for a detailed discussion on the "quality" of the predicted "probabilistic" paths. This makes our method suitable for kinetic analysis of trajectories which involve sampling from a distribution and not explicit integration of Newton's equations of motions like hybrid Monte Carlo methods[103] or Replica Exchange Molecular Dynamics (REMD). The finer details in the "Markovian" path (like states IV and V in Figure 7) can be attributed to a smaller lag time (2 ps) used in the construction of the transition matrix in the MPP algorithm, which does not have the memory bottleneck t-SNE step.

We analyzed an REMD simulation of the same RNA hairpin in 8 M aq. urea solution with the framework introduced in this paper. The "probabilistic" pathway corresponding to a temperature of 400 K is shown in Figure 8. We used the same procedure for calculating the input parameters for our algorithm as explained in the previous sections. Taking memory constraints into consideration, every two frames were skipped, making the lag time $\tau = 4$ ps. Consistent with our 100 ps lifetime requirement for a state to be metastable, the *minPts* parameter of DBSCAN for this system was set to 25. For this system, 69 metastable states or clusters were identified, which is much larger than the 14 identified for the MD trajectory at 410 K. This is expected as REMD enables the system to explore a much larger conformational area along the PES as compared to conventional unbiased MD. Akin to the MD pathway at 410 K, at 400 K unfolding proceeds by unzipping of the stem from the free end (opening of the U12−G1 and the C11−G2 base pairs). This is followed by a destabilization of the loop structure at an intermediate metastable state with only the G9−C4 base pair intact. This transitionary state is absent at 410 K, where the system has enough energy to directly proceed to an unfolded state.

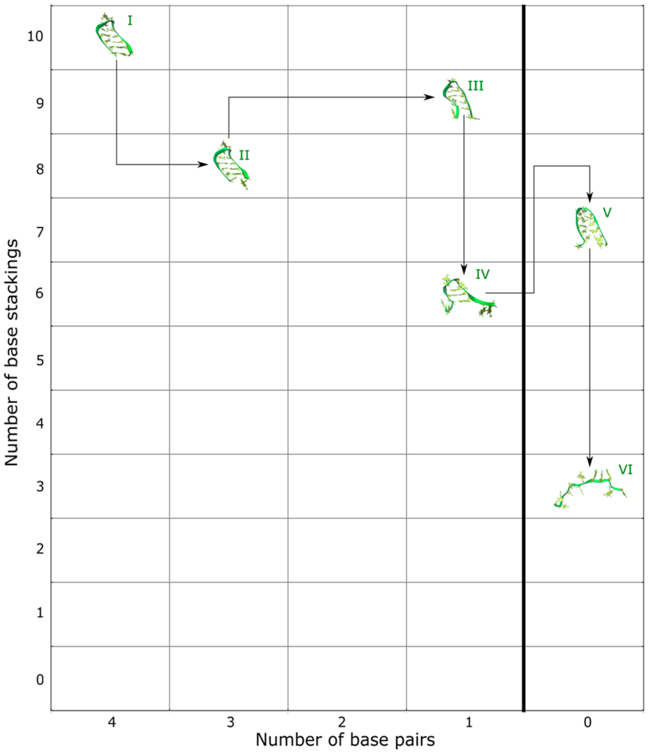


Figure 8. The "most reactive path" identified by the "probabilistic" pathway. The structures represented are the "representative elements" of their respective clusters. Every square in the 2D grid is associated with a particular metastable state by the order parameters—number of base stacking and number of base pairs. This figure corresponds to REMD simulation at 400 K.

A common test for Markovianity of a process is the slowest implied time scale vs lag-time plot. If the system is Markovian, then after a certain lag time *k*, the implied time scale must converge to a constant value independent of the lag time.[104] For calculating the implied time scale in REMD simulations, only the transitions in the unbiased trajectories are counted (Figure S5). It is evident that while the MD simulation is Markovian in nature, the REMD simulation is not, possibly due to insufficient sampling of the interstate transitions, which makes MSM analysis difficult in the latter case.

4.4. Correctness of the "Widest" Path Hypothesis. In this subsection, we provide some empirical evidence that our assumption of the most probable path being the "widest" path has merit. We show that the "most reactive path" identified for all three systems, without using any temporal connections between conformations, also exists in their respective MD trajectories. We coarse-grain the trajectories as a sequence of unique clusters. A directed network is created for each temperature with the metastable states as nodes and directed edges between them denoting a transition. If the system is in cluster 1 at time *t* and in cluster 2 in time $t + \Delta t$ (with Δt being the lag time between adjacent frames of the trajectory, or the integration time step of the MD trajectory, which is 2 ps in this case), there exists a directed edge from node 1 to node 2 in the network. The networks for the MD run at 300 K are shown in Figure 9, the networks corresponding to the other two trajectories, namely 360 and 410 K, are illustrated in Figures S6 and S7 in the Supporting Information. The "probabilistic" path is highlighted by red dotted arrows in all three networks. If the edge predicted by the "probabilistic" path does not exist in the network, it is highlighted in blue. At both 360 and 410 K, the most probable unfolding path predicted exists in the network. At 300 K, the optimum path identified by our algorithm deviates slightly by proposing a direct transition between state 31 and state 13 (blue dotted edge), which is absent in the trajectory.

5. CONCLUSION

In this study, we propose a novel method for extracting kinetic information from molecular trajectories without using the sequential information available in these trajectories. This method involves four different stages, which enables extraction of such data starting from a collection of sampled configuration space points: (a) choosing an appropriate vector representation for the trajectory; (b) identifying metastable states using PCA, t-SNE, and DBSCAN; (c) creating a network connecting the metastable states using EM; and (d) extracting most probable paths using Dijkstra's widest path algorithm. In the case of

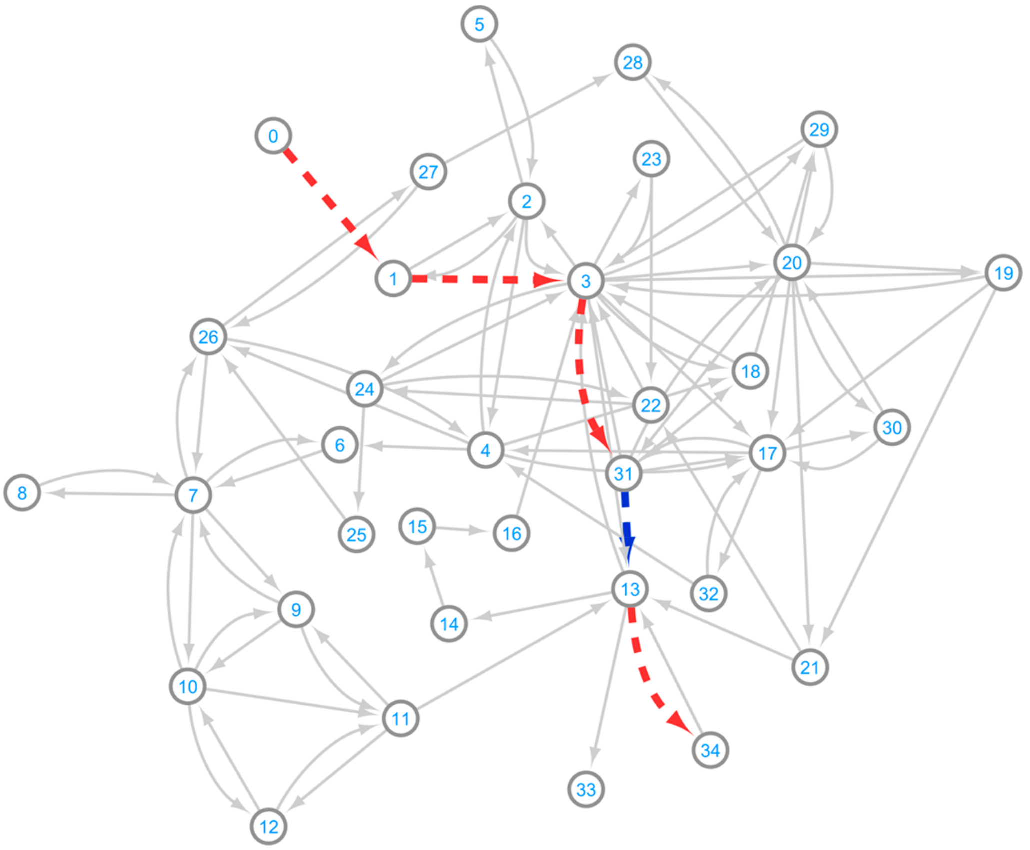


Figure 9. Networks for the MD run at 300 K. Each node u in the graph represents a metastable cluster, and a directed edge (u_1, u_2) denotes a transition from cluster u_1 at time t to cluster u_2 at time $t + \Delta t$, Δt being the integration time step of the MD simulation (2 ps in this case). The above graph is for a MD simulation of the RNA hairpin in 8 M aq. urea solution at 300 K. The red dotted directed edge represents the ***most reactive path*** identified by our method. The blue dotted directed edge represents a transition proposed by our algorithm, which is absent in the trajectory.

trajectories from Monte Carlo and REMD simulations, temporal information is not available, and hence traditional methods such as MSMs cannot be used. We use two key assumptions. First, molecular trajectories follow the Markovian assumption; that is, the system's current state at time t in its configuration space solely depends on its previous state at time $t - \tau$. Second, natural systems evolve gradually over time, i.e., in a short time τ, a system is most likely to visit energy basins that are in close proximity to its current energy basin (this includes staying in its own energy basin). Instead of a hard assignment (selecting the closest energy basin as the most likely destination), we adopt a soft assignment protocol and let the algorithm learn these transitions from the simulation data itself. This is accomplished using an iterative expectation maximization algorithm within the maximum likelihood estimation framework. As proof of concept, we first show the effectiveness of our proposed method in unraveling temporal connections between metastable states that are consistent with those found in an unbiased MD trajectory of the same RNA hairpin in an 8 M aq. urea system. We then go on to show the utility of our method in the extraction of a similar temporal pathway in REMD simulations of the same RNA hairpin molecule. In theory, this method can be used to analyze trajectories obtained from any sampling algorithm, be it unbiased MD, REMD, hybrid Monte Carlo, or Replica exchange umbrella sampling. This work is an effort in the direction toward harnessing the full power of enhanced sampling methods, which until now was mostly limited to evaluating equilibrium properties of the system.

■ ASSOCIATED CONTENT

Ⓢ Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jctc.7b01245.

> Plots showing a toy 1D example illustrating the flaw of geometric clustering techniques, a toy 1D example depicting a Gaussian and a Cauchy distribution to illustrate the core idea behind the t-SNE algorithm, "most reactive paths" identified by the "Markovian" and "probabilistic" pathways at both 360 and 410 K, network showing the presence of the "most reactive path" in MD simulations at 360 and 410 K, and descriptions of some of the technical details (PDF)

■ AUTHOR INFORMATION

Corresponding Author

*FAX: +91 40 6653 1413. E-mail: deva@iiit.ac.in.

ORCID

Mark P. Waller: 0000-0003-1650-5161

U. Deva Priyakumar: 0000-0001-7114-3955

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

U.D.P. thanks DST SERB for financial assistance (Grant No. EMR/2016/007697). We thank Prof. Peter Comba (Universität Heidelberg) and Dr. G. Narahari Sastry (IICT

Hyderabad) for their support. Prof. C.V. Jawahar and Dr. Prabhakar Bhimalapuram are acknowledged for fruitful discussions. M.Z. thanks IIIT Hyderabad for supporting her stay in India.

■ REFERENCES

(1) Simmerling, C.; Strockbine, B.; Roitberg, A. E. All-Atom Structure Prediction and Folding Simulations of a Stable Protein. *J. Am. Chem. Soc.* **2002**, *124* (38), 11258−11259.

(2) Daggett, V. Molecular Dynamics Simulations of the Protein Unfolding/Folding Reaction. *Acc. Chem. Res.* **2002**, *35* (6), 422−429.

(3) Duan, Y.; Kollman, P. A. Pathways to a Protein Folding Intermediate Observed in a 1-Microsecond Simulation in Aqueous Solution. *Science* **1998**, *282* (5389), 740−744.

(4) Brooks, C. L. Protein and Peptide Folding Explored with Molecular Simulations. *Acc. Chem. Res.* **2002**, *35* (6), 447−454.

(5) Scheraga, H. A.; Khalili, M.; Liwo, A. Protein-Folding Dynamics: Overview of Molecular Simulation Techniques. *Annu. Rev. Phys. Chem.* **2007**, *58*, 57−83.

(6) de Jonge, M. R.; Koymans, L. H.; Guillemont, J. E.; Koul, A.; Andries, K. A Computational Model of the Inhibition of Mycobacterium Tuberculosis ATPase by a New Drug Candidate R207910. *Proteins: Struct., Funct., Genet.* **2007**, *67* (4), 971−980.

(7) Kollman, P. A.; Massova, I.; Reyes, C.; Kuhn, B.; Huo, S.; Chong, L.; Lee, M.; Lee, T.; Duan, Y.; Wang, W.; et al. Calculating Structures and Free Energies of Complex Molecules: Combining Molecular Mechanics and Continuum Models. *Acc. Chem. Res.* **2000**, *33* (12), 889−897.

(8) Špačková, N.; Cheatham, T. E.; Ryjáček, F.; Lankaš, F.; Van Meervelt, L.; Hobza, P.; Šponer, J. Molecular Dynamics Simulations and Thermodynamics Analysis of DNA-Drug Complexes. Minor Groove Binding between 4 ‘, 6-Diamidino-2-Phenylindole and DNA Duplexes in Solution. *J. Am. Chem. Soc.* **2003**, *125* (7), 1759−1769.

(9) Bui, J. M.; McCammon, J. A. Protein Complex Formation by Acetylcholinesterase and the Neurotoxin Fasciculin-2 Appears to Involve an Induced-Fit Mechanism. *Proc. Natl. Acad. Sci. U. S. A.* **2006**, *103* (42), 15451−15456.

(10) Lu, Y.; Yang, C.-Y.; Wang, S. Binding Free Energy Contributions of Interfacial Waters in HIV-1 Protease/Inhibitor Complexes. *J. Am. Chem. Soc.* **2006**, *128* (36), 11830−11839.

(11) Hornak, V.; Okur, A.; Rizzo, R. C.; Simmerling, C. HIV-1 Protease Flaps Spontaneously Open and Reclose in Molecular Dynamics Simulations. *Proc. Natl. Acad. Sci. U. S. A.* **2006**, *103* (4), 915−920.

(12) Noy, A.; Pérez, A.; Laughton, C. A.; Orozco, M. Theoretical Study of Large Conformational Transitions in DNA: The B↔ A Conformational Change in Water and Ethanol/Water. *Nucleic Acids Res.* **2007**, *35* (10), 3330−3338.

(13) Sefcikova, J.; Krasovska, M. V.; Šponer, J.; Walter, N. G. The Genomic HDV Ribozyme Utilizes a Previously Unnoticed U-Turn Motif to Accomplish Fast Site-Specific Catalysis. *Nucleic Acids Res.* **2007**, *35* (6), 1933−1946.

(14) Kormos, B. L.; Baranger, A. M.; Beveridge, D. L. A Study of Collective Atomic Fluctuations and Cooperativity in the U1A−RNA Complex Based on Molecular Dynamics Simulations. *J. Struct. Biol.* **2007**, *157* (3), 500−513.

(15) van der Vaart, A.; Karplus, M. Minimum Free Energy Pathways and Free Energy Profiles for Conformational Transitions Based on Atomistic Molecular Dynamics Simulations. *J. Chem. Phys.* **2007**, *126* (16), 164106.

(16) Freddolino, P. L.; Liu, F.; Gruebele, M.; Schulten, K. Ten-Microsecond Molecular Dynamics Simulation of a Fast-Folding WW Domain. *Biophys. J.* **2008**, *94* (10), L75−L77.

(17) Shaw, D. E.; Dror, R. O.; Salmon, J. K.; Grossman, J.; Mackenzie, K. M.; Bank, J. A.; Young, C.; Deneroff, M. M.; Batson, B.; Bowers, K. J. Millisecond-Scale Molecular Dynamics Simulations on Anton. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*; IEEE, 2009; pp 1−11.

(18) Salomon-Ferrer, R.; Götz, A. W.; Poole, D.; Le Grand, S.; Walker, R. C. Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 2. Explicit Solvent Particle Mesh Ewald. *J. Chem. Theory Comput.* **2013**, *9* (9), 3878−3888.

(19) Krivov, S. V.; Karplus, M. Hidden Complexity of Free Energy Surfaces for Peptide (Protein) Folding. *Proc. Natl. Acad. Sci. U. S. A.* **2004**, *101* (41), 14766−14770.

(20) Das, P.; Moll, M.; Stamati, H.; Kavraki, L. E.; Clementi, C. Low-Dimensional, Free-Energy Landscapes of Protein-Folding Reactions by Nonlinear Dimensionality Reduction. *Proc. Natl. Acad. Sci. U. S. A.* **2006**, *103* (26), 9885−9890.

(21) Mu, Y.; Nguyen, P. H.; Stock, G. Energy Landscape of a Small Peptide Revealed by Dihedral Angle Principal Component Analysis. *Proteins: Struct., Funct., Genet.* **2005**, *58* (1), 45−52.

(22) Ernst, M.; Sittel, F.; Stock, G. Contact- and Distance-Based Principal Component Analysis of Protein Dynamics. *J. Chem. Phys.* **2015**, *143* (24), 244114.

(23) Jain, A.; Stock, G. Identifying Metastable States of Folding Proteins. *J. Chem. Theory Comput.* **2012**, *8* (10), 3810−3819.

(24) Rao, F.; Caflisch, A. The Protein Folding Network. *J. Mol. Biol.* **2004**, *342* (1), 299−306.

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

(26) Wales, D. J. Energy Landscapes: Some New Horizons. *Curr. Opin. Struct. Biol.* **2010**, *20* (1), 3−10.

(27) Prentiss, M. C.; Wales, D. J.; Wolynes, P. G. The Energy Landscape, Folding Pathways and the Kinetics of a Knotted Protein. *PLoS Comput. Biol.* **2010**, *6* (7), e1000835.

(28) Duran, B. S.; Odell, P. L. *Cluster Analysis: A Survey*; Springer Science & Business Media, 2013; Vol. *100*.

(29) Kaufman, L.; Rousseeuw, P. J. *Finding Groups in Data: An Introduction to Cluster Analysis*; John Wiley & Sons, 2009; Vol. *344*.

(30) Jain, A.; Dubes, R. *Algorithms for Clustering Data*; Prentice-Hall, Inc: Upper Saddle River, NJ, 1988.

(31) Kogan, J.; Nicholas, C.; Teboulle, M. *Grouping Multidimensional Data*; Springer, 2006.

(32) Poncin, M.; Hartmann, B.; Lavery, R. Conformational Sub-States in B-DNA. *J. Mol. Biol.* **1992**, *226* (3), 775−794.

(33) Jain, A. K.; Murty, M. N.; Flynn, P. J. Data Clustering: A Review. *ACM Comput. Surv. CSUR* **1999**, *31* (3), 264−323.

(34) Hartigan, J. A.; Wong, M. A. Algorithm AS 136: A k-Means Clustering Algorithm. *J. R. Stat. Soc. Ser. C Appl. Stat.* **1979**, *28* (1), 100−108.

(35) Kohonen, T. The Self-Organizing Map. *Neurocomputing* **1998**, *21* (1), 1−6.

(36) Shao, J.; Tanner, S. W.; Thompson, N.; Cheatham, T. E. Clustering Molecular Dynamics Trajectories: 1. Characterizing the Performance of Different Clustering Algorithms. *J. Chem. Theory Comput.* **2007**, *3* (6), 2312−2334.

(37) Comaniciu, D.; Meer, P. Mean Shift: A Robust Approach toward Feature Space Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24* (5), 603−619.

(38) Deuflhard, P.; Weber, M. Robust Perron Cluster Analysis in Conformation Dynamics. *Linear Algebra Its Appl.* **2005**, *398*, 161−184.

(39) Noé, F.; Horenko, I.; Schütte, C.; Smith, J. C. Hierarchical Analysis of Conformational Dynamics in Biomolecules: Transition Networks of Metastable States. *J. Chem. Phys.* **2007**, *126* (15), 155102.

(40) Pan, A. C.; Roux, B. Building Markov State Models along Pathways to Determine Free Energies and Rates of Transitions. *J. Chem. Phys.* **2008**, *129* (6), 064107.

(41) Yang, S.; Roux, B. Src Kinase Conformational Activation: Thermodynamics, Pathways, and Mechanisms. *PLoS Comput. Biol.* **2008**, *4* (3), e1000047.

(42) Zeng, X.; Zhang, L.; Xiao, X.; Jiang, Y.; Guo, Y.; Yu, X.; Pu, X.; Li, M. Unfolding Mechanism of Thrombin-Binding Aptamer Revealed by Molecular Dynamics Simulation and Markov State Model. *Sci. Rep.* **2016**, *6*, 24065.

(43) Noé, F.; Schütte, C.; Vanden-Eijnden, E.; Reich, L.; Weikl, T. R. Constructing the Equilibrium Ensemble of Folding Pathways from

Short Off-Equilibrium Simulations. *Proc. Natl. Acad. Sci. U. S. A.* **2009**, *106* (45), 19011−19016.

(44) Swendsen, R. H.; Wang, J.-S. Replica Monte Carlo Simulation of Spin-Glasses. *Phys. Rev. Lett.* **1986**, *57* (21), 2607.

(45) Torrie, G. M.; Valleau, J. P. Nonphysical Sampling Distributions in Monte Carlo Free-Energy Estimation: Umbrella Sampling. *J. Comput. Phys.* **1977**, *23* (2), 187−199.

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

(47) Yang, S.; Onuchic, J. N.; García, A. E.; Levine, H. Folding Time Predictions from All-Atom Replica Exchange Simulations. *J. Mol. Biol.* **2007**, *372* (3), 756−763.

(48) Buchete, N.-V.; Hummer, G. Peptide Folding Kinetics from Replica Exchange Molecular Dynamics. *Phys. Rev. E* **2008**, *77* (3), 030902.

(49) Leahy, C. T.; Murphy, R. D.; Hummer, G.; Rosta, E.; Buchete, N.-V. Coarse Master Equations for Binding Kinetics of Amyloid Peptide Dimers. *J. Phys. Chem. Lett.* **2016**, *7* (14), 2676−2682.

(50) Stelzl, L. S.; Hummer, G. Kinetics from Replica Exchange Molecular Dynamics Simulations. *J. Chem. Theory Comput.* **2017**, *13* (8), 3927−3935.

(51) Ernst, M.; Wolf, S.; Stock, G. Identification and Validation of Reaction Coordinates Describing Protein Functional Motion: Hierarchical Dynamics of T4 Lysozyme. *J. Chem. Theory Comput.* **2017**, *13* (10), 5076−5088.

(52) Ma, A.; Dinner, A. R. Automatic Method for Identifying Reaction Coordinates in Complex Systems. *J. Phys. Chem. B* **2005**, *109* (14), 6769−6779.

(53) Antoniou, D.; Schwartz, S. D. Toward Identification of the Reaction Coordinate Directly from the Transition State Ensemble Using the Kernel PCA Method. *J. Phys. Chem. B* **2011**, *115* (10), 2465−2469.

(54) McGibbon, R. T.; Husic, B. E.; Pande, V. S. Identification of Simple Reaction Coordinates from Complex Dynamics. *J. Chem. Phys.* **2017**, *146* (4), 044109.

(55) Wu, H.; Paul, F.; Wehmeyer, C.; Noé, F. Multiensemble Markov Models of Molecular Thermodynamics and Kinetics. *Proc. Natl. Acad. Sci. U. S. A.* **2016**, *113* (23), E3221−E3230.

(56) Ishikawa, H.; Kwak, K.; Chung, J. K.; Kim, S.; Fayer, M. D. Direct Observation of Fast Protein Conformational Switching. *Proc. Natl. Acad. Sci. U. S. A.* **2008**, *105* (25), 8619−8624.

(57) Kolinski, A.; Skolnick, J. Monte Carlo Simulations of Protein Folding. I. Lattice Model and Interaction Scheme. *Proteins: Struct., Funct., Genet.* **1994**, *18* (4), 338−352.

(58) Pereira de Araújo, A. F.; Pochapsky, T. C. Monte Carlo Simulations of Protein Folding Using Inexact Potentials: How Accurate Must Parameters Be in Order to Preserve the Essential Features of the Energy Landscape? *Folding Des.* **1996**, *1* (4), 299−314.

(59) Lotan, I.; Schwarzer, F.; Latombe, J.-C. Efficient Energy Computation for Monte Carlo Simulation of Proteins. *Lect. Notes Comput. Sci.* **2003**, *2812*, 354−373.

(60) Stillinger, F. H.; Weber, T. A. Hidden Structure in Liquids. *Phys. Rev. A: At., Mol., Opt. Phys.* **1982**, *25* (2), 978.

(61) Stillinger, F. H.; Weber, T. A. Computer Simulation of Local Order in Condensed Phases of Silicon. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1985**, *31* (8), 5262.

(62) Stillinger, F. H.; Weber, T. A. Inherent Structure in Water. *J. Phys. Chem.* **1983**, *87* (15), 2833−2840.

(63) Weber, T. A.; Stillinger, F. H. Local Order and Structural Transitions in Amorphous Metal-Metalloid Alloys. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1985**, *31* (4), 1954.

(64) Elber, R.; Karplus, M. A Method for Determining Reaction Paths in Large Molecules: Application to Myoglobin. *Chem. Phys. Lett.* **1987**, *139* (5), 375−380.

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

(66) Duane, S.; Kennedy, A. D.; Pendleton, B. J.; Roweth, D. Hybrid Monte Carlo. *Phys. Lett. B* **1987**, *195* (2), 216−222.

(67) Widom, M.; Huhn, W. P.; Maiti, S.; Steurer, W. Hybrid Monte Carlo/Molecular Dynamics Simulation of a Refractory Metal High Entropy Alloy. *Metall. Mater. Trans. A* **2014**, *45* (1), 196−200.

(68) van der Maaten, L.; Hinton, G. Visualizing Data Using T-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579−2605.

(69) Sheather, S. J. Density Estimation. *Stat. Sci.* **2004**, *19* (4), 588−597.

(70) Melvin, R. L.; Godwin, R. C.; Xiao, J.; Thompson, W. G.; Berenhaut, K. S.; Salsbury, F. R., Jr Uncovering Large-Scale Conformational Change in Molecular Dynamics without Prior Knowledge. *J. Chem. Theory Comput.* **2016**, *12* (12), 6130−6146.

(71) Sittel, F.; Stock, G. Robust Density-Based Clustering to Identify Metastable Conformational States of Proteins. *J. Chem. Theory Comput.* **2016**, *12* (5), 2426−2435.

(72) Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *KDD-96 Proceedings*; AAAI, 1996; vol *96*, pp 226−231.

(73) Sawant, K. Adaptive Methods for Determining Dbscan Parameters. *Int. J. Innov. Sci. Eng. Technol.* **2014**, *1* (4).

(74) Rosenblatt, M. others. Remarks on Some Nonparametric Estimates of a Density Function. *Ann. Math. Stat.* **1956**, *27* (3), 832−837.

(75) Parzen, E. On Estimation of a Probability Density Function and Mode. *Ann. Math. Stat.* **1962**, *33* (3), 1065−1076.

(76) Do, C. B.; Batzoglou, S. What Is the Expectation Maximization Algorithm? *Nat. Biotechnol.* **2008**, *26* (8), 897−899.

(77) Dempster, A. P.; Laird, N. M.; Rubin, D. B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. R. Stat. Soc. Ser. B Methodol.* **1977**, 1−38.

(78) Little, R. J.; Rubin, D. B. *Statistical Analysis with Missing Data*; John Wiley & Sons, 2014.

(79) Amato, N. M.; Song, G. Using Motion Planning to Study Protein Folding Pathways. *J. Comput. Biol.* **2002**, *9* (2), 149−168.

(80) Tang, X.; Kirkpatrick, B.; Thomas, S.; Song, G.; Amato, N. M. Using Motion Planning to Study RNA Folding Kinetics. *J. Comput. Biol.* **2005**, *12* (6), 862−881.

(81) Su, J. G.; Li, C. H.; Hao, R.; Chen, W. Z.; Xin Wang, C. Protein Unfolding Behavior Studied by Elastic Network Model. *Biophys. J.* **2008**, *94* (12), 4586−4596.

(82) Metzner, P.; Schütte, C.; Vanden-Eijnden, E. Transition Path Theory for Markov Jump Processes. *Multiscale Model. Simul.* **2009**, *7* (3), 1192−1219.

(83) Pollack, M. Letter to the Editor—The Maximum Capacity Through a Network. *Oper. Res.* **1960**, *8* (5), 733−736.

(84) Black, P. E. *Dictionary of Algorithms and Data Structures*; NIST, 1998.

(85) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12* (Oct), 2825−2830.

(86) Bergstra, J.; Breuleux, O.; Bastien, F.; Lamblin, P.; Pascanu, R.; Desjardins, G.; Turian, J.; Warde-Farley, D.; Bengio, Y. Theano: A CPU and GPU Math Compiler in Python. In *Proceedings of the 9th Python in Science Conference (SciPy 2010)* **2010**; pp 1−7.

(87) Jones, E.; Oliphant, T.; Peterson, P. *SciPy: Open Source Scientific Tools for Python*, 2014.

(88) van der Walt, S.; Colbert, S. C.; Varoquaux, G. The NumPy Array: A Structure for Efficient Numerical Computation. *Comput. Sci. Eng.* **2011**, *13* (2), 22−30.

(89) 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. *Biophys. J.* **2015**, *109* (8), 1528−1532.

(90) Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9* (3), 90−95.

(91) Foloppe, N.; MacKerell, A. D., Jr All-Atom Empirical Force Fiel bd for Nucleic Acids: I. Parameter Optimization Based on Small

Molecule and Condensed Phase Macromolecular Target Data. *J. Comput. Chem.* **2000**, *21* (2), 86−104.

(92) Vanommeslaeghe, K.; Hatcher, E.; Acharya, C.; Kundu, S.; Zhong, S.; Shim, J.; Darian, E.; Guvench, O.; Lopes, P.; Vorobyov, I.; MacKerell, A. D. CHARMM General Force Field: A Force Field for Drug-like Molecules Compatible with the CHARMM All-Atom Additive Biological Force Fields. *J. Comput. Chem.* **2010**, *31* (4), 671−690.

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

(94) Yoon, J.; Thirumalai, D.; Hyeon, C. Urea-Induced Denaturation of PreQ1-Riboswitch. *J. Am. Chem. Soc.* **2013**, *135* (32), 12112−12121.

(95) Doudna, J. A.; Doherty, E. A. Emerging Themes in RNA Folding. *Folding Des.* **1997**, *2* (5), R65−R70.

(96) Sarkar, K.; Nguyen, D. A.; Gruebele, M. Loop and Stem Dynamics during RNA Hairpin Folding and Unfolding. *RNA* **2010**, *16* (12), 2427−2434.

(97) Goyal, S.; Chattopadhyay, A.; Kasavajhala, K.; Priyakumar, U. D. Role of Urea−Aromatic Stacking Interactions in Stabilizing the Aromatic Residues of the Protein in Urea-Induced Denatured State. *J. Am. Chem. Soc.* **2017**, *139* (42), 14931−14946.

(98) Priyakumar, U. D.; Hyeon, C.; Thirumalai, D.; MacKerell, A. D., Jr Urea Destabilizes RNA by Forming Stacking Interactions and Multiple Hydrogen Bonds with Nucleic Acid Bases. *J. Am. Chem. Soc.* **2009**, *131* (49), 17759−17761.

(99) Miner, J. C.; García, A. E. Equilibrium Denaturation and Preferential Interactions of an RNA Tetraloop with Urea. *J. Phys. Chem. B* **2017**, *121* (15), 3734−3746.

(100) Kasavajhala, K.; Bikkina, S.; Patil, I.; MacKerell, A. D., Jr; Priyakumar, U. D. Dispersion Interactions between Urea and Nucleobases Contribute to the Destabilization of RNA by Urea in Aqueous Solution. *J. Phys. Chem. B* **2015**, *119* (9), 3755−3761.

(101) Canchi, D. R.; García, A. E. Cosolvent Effects on Protein Stability. *Annu. Rev. Phys. Chem.* **2013**, *64*, 273−293.

(102) Vinh, N. X.; Epps, J.; Bailey, J. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *J. Mach. Learn. Res.* **2010**, *11* (Oct), 2837−2854.

(103) Hansmann, U. H.; Okamoto, Y.; Eisenmenger, F. Molecular Dynamics, Langevin and Hydrid Monte Carlo Simulations in a Multicanonical Ensemble. *Chem. Phys. Lett.* **1996**, *259* (3−4), 321−330.

(104) Swope, W. C.; Pitera, J. W.; Suits, F. Describing protein folding kinetics by molecular dynamics simulations. 1. *J. Phys. Chem. B* **2004**, *108*, 6571−6581.