From Structural Clustering to Enhanced Sampling: A Data-Driven Approach for Exploring Protein Conformational Ensembles

by

Subarna Sasmal

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Chemistry

New York University

May, 2025

Dr. Glen M. Hocky

Dedication

To My family; Madhu Basanta Sasmal, Rinku Sasmal and Susmita Rout

Acknowledgements

I am deeply grateful to the many individuals who have supported me throughout my doctoral journey at New York University, culminating in this dissertation.

First and foremost, I extend my sincere gratitude to my advisor, Dr. Glen Hocky, whose mentorship has been truly inspiring. His patience, encouragement, and unwavering support have shaped me as a researcher. I feel incredibly fortunate to have had the opportunity to work with him. His expertise and guidance have been instrumental in my research, providing me with the resources and confidence to explore new directions and grow as a scientist. His constant encouragement to pursue my own interests has made this journey both fulfilling and enriching.

I am also thankful to all the members of the Hocky group for their invaluable discussions, collaboration, and huge support, which have greatly contributed to my work. I would also like to extend my thanks to my colleague, Nicodemo Mazzaferro, for the opportunity to collaborate on a project developing methods for accurate kinetic rates measurements, which has been both inspiring and impactful.

My heartfelt appreciation goes to my dissertation committee— Prof. Mark E. Tuckerman, Prof. Nathaniel J. Traaseth, and Prof. Yingkai Zhang for their insightful guidance, constructive feedback, and invaluable suggestions throughout my doctoral studies.

I would also like to express my gratitude to my collaborator, Dr. Martin McCullagh from Oklahoma State University. Working with him has been an immensely rewarding experience, and I have learned a great deal from his expertise. His guidance was crucial to the success of our

projects.

I want to extend my gratitude to Dr. Gareth A Tribello from Queen's University Belfast for his guidance and support during our time working together on the project described in Section 4.6. His vast knowledge, patience, and encouragement were instrumental in helping me make significant progress.

On a personal note, I am deeply grateful to my parents, Mr. Madhu Basanta Sasmal and Mrs. Rinku Sasmal, for their unconditional love, support and guidance in every step of my life. Their belief in me has been my greatest source of strength. I am also immensely thankful to my wife, Mrs. Susmita Rout, for being an incredible partner. Her patience, understanding, and constant encouragement have been my anchor throughout this journey.

I acknowledge with gratitude the financial support that made my research possible, including funding from the National Institutes of Health (NIH), the Simons Center Graduate Fellowship, and the McCracken Fellowship. I also appreciate the resources provided by NYU High-Performance Computing (HPC), which enabled me to conduct extensive simulations and data analysis. A special thanks to Dr. Shenglong Wang for his assistance in resolving technical challenges related to running simulations on the cluster.

To all who have contributed to my academic and personal growth during this journey—thank you. This dissertation is a reflection of your support and encouragement.

Finally, I am grateful to God for everything.

Abstract

Proteins are a class of biomolecules that are one of the most important building blocks of living organisms. While it is common knowledge that the function of a protein is determined by its three dimensional structure, in reality proteins exist in multiple metastable states with different energy and specific functions. Molecular dynamics simulations is an approach by which we can use computational modeling to characterize the proteins conformational ensemble with atomistic detail. In practice, simple simulations do not allow us to access relevant conformations in accessible amounts of conformational time. Enhanced sampling algorithms allow us to more rapidly explore a system's conformational ensemble, but typically requires guessing a set of collective coordinates that, if biased, would allow us to observe all relevant configurations with an inference of their correct likelihoods. In this work, I describe an approach for simultaneously characterizing and exploring conformational ensembles of proteins. Our approach relies on a probabilistic clustering model called ShapeGMM, where configurations are used to learn a Gaussian mixture model in cartesian coordinate space. In my work, we demonstrated that the technique Linear Discriminant Analysis can be used to form a coordinate that separates states of a molecule and allows us to sample between them using enhanced sampling. We then showed that we can train a ShapeGMM model with samples generated by such a bias approach. This gives an approach by which conformational ensembles can be quantitatively characterized. Finally, we show that this allows us to perform iteration, in which case we can develop better coordinates by alternating sampling and fitting.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Molecular Dynamics and Rare Events

Molecular Dynamics (MD) simulations are a powerful tool for uncovering the functions of biomolecules in cellular environments. Biomolecules—including proteins, nucleotides, lipids, and carbohydrates—play essential roles in sustaining all living organisms. These macromolecules exist in multiple metastable states, and their functions are strongly influenced by their spatial configurations. Over the years, MD simulations have been extensively used to investigate a wide range of biological processes, such as protein conformational transitions from inactive to active state [1–4], drug binding to enzyme active sites [5–8], phase transitions in chemical systems at critical temperatures [9–11], allosteric regulation [12–15] and the protein folding problem [16–19].

In conventional MD studies, a system is described by a force field which defines the potential energy of interactions between all the atoms of the system. Each particle experiences forces due to interactions with surrounding particles. By calculating these forces and solving Newton's equation of motion for a given initial condition, MD simulations generate an ensemble of system configurations. This atomistic representation, along with the underlying dynamics, provides critical insights into microscopic mechanisms. For instance, designing an effective drug that binds strongly to a protein's active site requires a thorough understanding of the protein's conformational ensembles. Data obtained from MD simulations help compute thermodynamic properties

such as binding free energy differences, configurational entropies of metastable states, and free energy barriers for protein folding and unfolding. Additionally, MD enables the calculation of kinetic properties, including state transition rates and drug residence times in enzyme active sites. By bridging atomistic dynamics with macroscopic physical properties, MD simulations offer an invaluable tool for understanding biomolecular behavior.

Despite the broad applicability of MD, certain challenges limit its efficiency. The computational expense of MD simulations increases rapidly with system size. While increased computational resources help mitigate this issue to some extent, there is a deeper challenge that remains.

One of the most critical limitations of conventional MD is the need to sample rare events. Many biophysical processes exhibit this problem, where the system becomes trapped in a local metastable basin, unable to reach the other states due to large energy barriers ($>> kT$) separating them in high-dimensional configurational space [20–22]. These transitions often occur on millisecond to second timescales in real-time, which is far beyond the reach of standard simulations, especially for large biomolecular systems. Addressing these challenges requires the development of enhanced sampling techniques and more efficient computational approaches to extend the timescales accessible by MD simulations, thereby improving their predictive power and applicability to complex biological systems.

1.2 Enhanced Sampling

In last few decades, scientists have come up with different enhanced sampling approaches that is designed to deal with rare events and ensures better sampling of Free Energy Surface (FES). Enhanced sampling algorithms can be briefly classified in two categories - (**a**) collective variable (CV) based algorithms which depend on biasing few selected degrees of freedom that is supposed to capture the slow modes of the system. And by enhancing the fluctuations of the CVs, it forces the system out from the initial metastable state to explore the FES. Umbrella Sam-

pling (US) [23], Metadynamics (MetaD) [20], Well Tempered Metadynamics (WT-MetaD) [24], On-the-fly-Probability-Enhanced-Sampling (OPES) [25], driven Adiabatic Free Energy Dynamics (d-AFED)/Temperature Accelerated Molecular Dynamics (TAMD) [26, 27] are some popular CV based methods which are widely used to study biophysical systems. (**b**) Not CV based algorithms which rely on enhancing sampling without adding any external biases such as Replica Exchange Molecular Dynamics (REMD) [28], Replica Exchange with Solute Tempering (REST) [29], or Transition Path Sampling (TPS) [30] are some well known techniques of this category.

In this thesis, I have mainly focused on studying transitions between different conformational ensembles of proteins using only CV based enhanced sampling methods- WT-MetaD and OPES Metadynamics (OPES-MetaD) to be specific. In WT-MetaD simulations, time dependent gaussian hills are deposited along few chosen CVs with time. The amount of bias deposited at time t is given by

$$V(s,t) = \sum_{t'=0,\tau,2\tau,\ldots,t} w(t') \exp\Big(- \sum_{i=1}^{d} \frac{(s_i(\boldsymbol{x},t') - s_i)^2}{2{\sigma_i}^2} \Big) \tag{1.1}$$

Where, $s = \{s_1, s_1, \ldots, s_d\}$ are CVs which are function of atomic coordinates. $\{{\sigma_i}^2\}$ are measure of variance of each CV. $w(t') = w_0 \exp\left(-V(s,t)/k_B\Delta T\right)$ is the time dependent height of gaussian hill with w_0 as the initial height and it decreases exponentially with time. Unlike the original metadynamics, use of this scaling factor $w(t')$ ensures the smooth convergence of free energy surface. ΔT is an input parameter with the dimension of temperature that controls the effective sampling temperature of CVs, T+ΔT. Rather than setting ΔT, one specifies a parameter called biasfactor, $\gamma = T + \Delta T/T$. γ controls the smoothness of sampled distribution. After sufficiently long time, the FES can be computed from the total amount of bias deposited as $F(s) = -\frac{\gamma}{\gamma-1} V(s, t \to \infty)$.

OPES method attempts to sample from a target probability distribution which is different from Boltzmann's distribution. In OPES-MetaD variant, the target distribution is the marginal probability distribution as a function of some CVs obtained from metadynamics, $P_{tg}(s) = [P(s)]^{1/\gamma}$ where $P(s)$ is the ground truth. It builds the bias potential on-the-fly that is implemented by

reweighting the kernel density estimation of $P(s)$. The bias at time t is given by

$$V(s,t) = k_B T\left(\frac{\gamma - 1}{\gamma}\right) \log\left(\frac{P_t(s)}{Z_t} + \epsilon\right) \tag{1.2}$$

where T is the temperature, γ is the biasfactor and $P_t(s)$ is estimate of unbiased probability distribution at time t. Z_t is a normalizing factor which is obtained by integrating over the explored CV space in time t and ϵ is regularization parameter that controls the maximum amount of bias that can be added to the system. In contrast to basin filling approach of metadynamics by dropping gaussian hills with time, OPES quickly approximates the bias required to sample from the target distribution. So, for OPES, there is an initial fast exploration phase of system followed by slower refinement of details in the deposited bias which becomes stable after certain amount of time. OPES-MetaD has proven to be highly efficient in studying large biophysical systems due to its faster convergence and rapid sampling over conformational space. At convergence, FES can be calculated from the deposited bias in same way as for metadynamics.

1.3 Methods for Analyzing Molecular Simulation Data

MD simulations of biophysical systems produce huge amount of high dimensional data, which is essentially cartesian coordinates associated with all atoms in each time frame. All the configurations are sampled from an underlying Boltzmann's distribution. This time continuous trajectory characterizes different metastable states and transitions between those states which could be explored in the given simulation time. If a system has N atoms then the resulting configurational space will be $3N$ dimensional and there are no possible ways to visualize metastable states embedded in such high dimensions, this is also known as curse of dimensionality. To properly analyze MD simulations we need to perform three operations: (1) Featurization, (2) Clustering and (3) Dimensionality Reduction. These are described next.

1.3.1 Featurization

Each configuration $\boldsymbol{X} \in \mathbb{R}^D$ is a point in $D = 3N$ dimensional space. To represent a configuration in a lower dimension without losing too much information, we need to define a transformation as $\boldsymbol{X} : \mathbb{R}^D \rightarrow \mathbb{R}^d$, where $d << D$. There are many ways in which one can define desired features, such as using internal coordinates (pairwise distances, angles and dihedrals) or positions of selected group of particles (e.g. alpha carbons, heavy atoms, backbone atoms) etc. While internal coordinates have been extensively used to describe a configuration due to their translational and rotational invariance property but it can be overwhelming for large systems because they scale as $O(N^2)$. On the other hand, use of atomic positions as features are a direct representation of macromolecule in high dimensional space and it doesn't overestimate system size since it scales as $O(N)$. But atomic positions are dependent on reference frame and not invariant to translation or rotation. We can say that atomic positions might be an better option for describing macromolecular configurations if translational and rotational invariance problem can be dealt with proper weighted alignment techniques. In this work, we have demonstrated the use of atomic positions as input features in both clustering and reducing dimensionality of data. Alternatively, one can also use sophisticated dimensionality reduction approaches to transform the data into lower dimensions and then use lower dimensional feature vector to describe the system configurations.

1.3.2 Clustering algorithms

In MD simulations, configurations are sampled from Boltzmann's distribution where probability peaks corresponding to different metastable states (high probability regions) are separated from each other by large free energy barriers (low probability regions). Therefore naturally in MD simulations, samples come mostly from metastable states and less from transition regions. Clustering algorithms are powerful techniques that can classify these configurations generated

from MD simulation in a small number of groups/clusters based on some similarity measurement. Each cluster is a group of configurations that represent a particular state of the system embedded in high dimensional probability distribution of configurational space. The use of clustering techniques to analyze MD simulation data is very common and it helps to identify key structural differences in metastable states and to explain the dynamics between them. There is a wide range of clustering algorithms which can be classified in three major categories - (**1**) Partitioning Clustering, (**2**) Density Based Clustering and (**3**) Kinetic Clustering.

In Partitioning schemes, clusters are represented as a group of configurations which are structurally similar. Members belonging to different clusters are far away from each other. The partitioning schemes can be further divided in four classes (**a**) Centroid based algorithms (k-means and k-medoids) [31], (**b**) Neighbor based algorithms (nearest neighbor and common nearest neighbor) [31], (**c**) Model based (Gaussian Mixture Models) [31] and (**d**) Hierarchial clustering (agglomerative and divisive algorithms) [32].

k-means and k-medoids Algorithms

In the centroid based algorithms, each cluster is associated with a centroid that represents the content of that cluster. Number of clusters is chosen a priori and a distance metric is defined to measure similarity between data points. The samples are assigned to the clusters based on their proximity to the centroids. The configurations within the same cluster exhibit low pairwise distances, while those in different clusters are separated by large distances. Additionally, a distance cutoff is often applied that restricts the configurations that can be included in a specific cluster. Essentially, these algorithms tend to partition the hyperplane into distinct groups, separated by boundaries, similar to the structure of a Voronoi diagram. k-means and k-medoids are two most popular algorithms which are used in many studies. Centroids for k given clusters, $\boldsymbol{\mu}_c, c = 1, \ldots, k$ are defined as the mean of configurations that belong to same cluster. A Loss function $L(\boldsymbol{\mu}_c)$ is defined as the sum of square distances of all the configurations to the mean of the cluster to

which it is assigned. The objective is to find optimal cluster centers by minimizing $L(\boldsymbol{\mu}_c)$. This optimization problem can be solved in an iterative manner- 1. initialize by randomly selecting any k configurations as cluster means, 2. assign every sample in data set to a cluster to which it is closest, 3. recalculate the new cluster means with new assignments and 4. repeat steps 2 and 3 until the cluster means don't change anymore or converges within a threshold value. Due to random initialization scheme of cluster means, quality of clustering strongly depends on the initial choice of configurations as cluster means and algorithm might need to be run multiple times with different initial choices to achieve reasonable results. To choose the appropriate number of clusters, one has to run the algorithm multiple times with increasing values of k and make a plot of $L(\boldsymbol{\mu}_c)$ with k to look for an elbow/kink which indicates a sharp positive change in the slope. The computational cost associated with k-means algorithm scale as $\mathcal{O}(Mkl)$, where M is number of configurations in the data and l is number of iterations needed. Unlike k-means, in k-medoids algorithm the centroid of a cluster is chosen as one of the cluster members which minimizes the sum of square distances of all the members to the centroid. Hence, in k-medoids, cluster means are true physical conformations of the system.

Neighbor based Algorithms

In neighbor algorithm, clusters are obtained iteratively based on the density of neighbors in the feature space. There is no need to specify the number of clusters at the beginning, only a nearest neighbor distance cutoff (d_{cut}) is required to evaluate the proximity of configurations to each other. For a given data set, the list of nearest neighbors are calculated for all samples based on the d_{cut}, then the sample with maximum number of neighbors represent a cluster center and all its neighbors are members of that cluster. After finding first cluster, all the members of that cluster are taken out from the data set and continue the same process of finding neighbors with remaining data, until all the samples are assigned to a cluster. This is a very simple and fast algorithm because one doesn't need to keep track of the entire distance matrix just keeping the record of

list of neighbors for each sample would do the job. In common nearest neighbor algorithm, there is one additional input parameter, called nearest neighbor number cutoff (n_{cut}) that allows one configuration to be included in an existing cluster if its number of common nearest neighbors compared to any of the samples from the same cluster is greater than or equal to n_{cut}. Due to this inclusion property of common nearest neighbor algorithm, a cluster can keep expanding in size until no more samples can be added to it. Hence, the resulting clusters represent the regions of high data point density in feature space, separated by low probability regions.

Gaussian Mixtue Models

Gaussian Mixture Model (GMM) [33] attempts to approximate the probability distribution of feature space as a sum of multivariate gaussians. The mean and covariance associated with each multivariate gaussian represents the center and shape of a cluster. The configurations of the same cluster are structurally close to each other, while members from different class are too different. GMM has been widely used to understand the high dimensional probability distribution of complex data sets from various applications including MD simulations. For a given data matrix $\boldsymbol{X}_{M\times d}$, where M is number of samples and d is the dimension of feature space, the probability of a configuration can be written as sum of K clusters-

$$P(\boldsymbol{x}_i) = \sum_{j=1}^{K} \phi_j N(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \Sigma_j), \qquad \forall i = 1, \ldots, M \tag{1.3}$$

Where $\boldsymbol{x}_i$ is a vector of dimension d, corresponding to the i^{th} configuration in feature space, $N(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \Sigma_j)$ is the j^{th} normalized, multivariate gaussian with mean $\boldsymbol{\mu}_j$, covariance matrix Σ_j, and weights ϕ_j (weights are normalized such that $\sum_{j=1}^{K} \phi_j = 1$). Considering that all samples in the data set arise from this probability density, we can ascribe the likelihood L to the data as -

$$L = \prod_{i=1}^{M} P(\boldsymbol{x}_i) = \prod_{i=1}^{M} \left(\sum_{j=1}^{K} \phi_j N(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \Sigma_j) \right) \tag{1.4}$$

In maximum likelihood estimation, the log likelihood function for this probability distribution function is given by-

$$\ln(L) = \sum_{i=1}^{M} \ln\left(\sum_{j=1}^{K} \phi_j N(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \Sigma_j)\right) \tag{1.5}$$

The objective is to maximize the log likelihood value by optimizing these parameters $\{\phi_j\}$, $\{\boldsymbol{\mu}_j\}$ and $\{\Sigma_j\}$ for K clusters. The Expectation Maximization (EM) algorithm [34] is widely used to find the solutions for the maximum likelihood problem in an iterative manner. EM algorithm starts with an initial guess $\{\phi_j\}$, $\{\boldsymbol{\mu}_j\}$ and $\{\Sigma_j\}$. This can be achieved in multiple ways, such as breaking the trajectory into K parts or randomly choosing K frames as cluster centers and assign remaining samples to their closest cluster center based on some distance metric (like RMSD). Then in expectation step, the posterior probability for each configuration in the data set is computed. Posterior probability for i^{th} configuration belonging to j^{th} cluster is given by,

$$\gamma_{Z_i}(j) = \frac{\hat{\phi}_j N(\boldsymbol{x_i} \mid \hat{\boldsymbol{\mu}}_j, \hat{\Sigma}_j)}{\sum_{j=1}^{K} \hat{\phi}_j N(\boldsymbol{x_i} \mid \hat{\boldsymbol{\mu}}_j, \hat{\Sigma}_j)} \tag{1.6}$$

where $Z_i \in (1, \ldots, K)$ are the latent variables. Expectation step is followed by Maximization step, where the values of the parameters $\{\phi_j\}$, $\{\boldsymbol{\mu}_j\}$ and $\{\Sigma_j\}$ are updated [31]. To learn the optimized parameters one has to iterate between expectation and maximization steps until the log likelihood converges within some given threshold. In GMM, a data point is assigned to a cluster in which it has largest log likelihood (i.e. largest value of $\gamma_{Z_i}(j)$). After the model has been trained with the input data and parameters are learned, GMM can be used to predict the cluster assignments of new data sets sampled from the same underlying probability distribution.

In density based clustering algorithms, the clusters are considered as existing peaks in a high dimensional probability distribution. Clusters illustrate multiple metastable states in free energy landscape, separated from each other by high energy barriers. Samples that belong to same cluster do not necessarily have similar structure because probability peaks can be asymmetric in shape. Unlike partitioning schemes, here one doesn't need to specify the number of clusters.

It is particularly useful for analyzing MD simulation trajectories where sampled configurations mostly come from free energy minimums and only few belong to the transition regions with low probability density. The density is measured for each configuration in the data set followed by finding out the correct probability peaks to which a particular sample belong. Some popular density based algorithms are - Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Hierarchial Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), Density Peak Clustering etc.

DBSCAN and HDBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [35] identifies clusters by analyzing the density of data points within a specified neighborhood. It uses a distance metric (typically Euclidean) to determine the number of neighbors within a given radius ϵ. If a point has at least a predefined number of neighbors *min_samples*, it is classified as a core point. The choice of ϵ and *min_samples* is crucial, as it directly affects the clustering outcome and must be carefully tuned for each dataset. In DBSCAN, a point m is directly reachable from a core point n if it lies within ϵ of n. A point m is reachable from n if there exists a chain of core points connecting them. Notably, reachability is not symmetric— m being reachable from n does not necessarily mean that n is reachable from m. To address this, DBSCAN introduces the concept of density-connected points, where two points m and n are density connected if there exists a third point s from which both m and n are reachable. The algorithm effectively identifies high-density regions, treating them as clusters while labeling points in low-density areas as noise points. A notable variant, HDBSCAN (Hierarchical DBSCAN) [36], has gained popularity in analyzing molecular dynamics (MD) simulation trajectories. Unlike DBSCAN, HDBSCAN handles clusters with varying densities, a common characteristic in data governed by the Boltzmann distribution. Additionally, HDBSCAN can uncover hierarchical structures within the data, making it particularly useful for complex clustering tasks.

Kinetic Clustering Algorithms

In kinetic clustering algorithms, the clusters are obtained based on kinetic proximity. The members of same clusters are kinetically close, meaning they can easily interconvert to each other in a short time scale, while members from two different clusters have relatively lower probability of transition. This is in accordance with MD simulations where a metastable state is an ensemble of configurations that are geometrically similar and make quick transitions among each other. But transitions between members of two metastable states which are separated by high free energy barrier is very rare, and happens on a characteristic long time scale, hence kinetically far away. Thus, Kinetic clustering techniques is an ideal approach to understand long time scale behavior of a biomolecular system by breaking down its dynamics into a number of clusters and probing transitions between them. MD simulation provides us with a time continuous trajectory of configurations, $\boldsymbol{x}(t) = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$. It is assumed that the entire trajectory of the system can be described by a number of discrete microstates. Each microstate is a group of configurations that are either kinetically or geometrically similar. To find the appropriate number of microstates from the data, one can start with an initial structural clustering like k-means, nearest neighbor etc. Similar to Markov State Modeling (MSM) [37], a Transition Probability Matrix ($\boldsymbol{T}$) is constructed, where $T_{ij}(\tau) = p(\boldsymbol{x}_j(t+\tau)|\boldsymbol{x}_i(t))$ is the probability of transition from state i to j in lag time τ. $\boldsymbol{T}(\tau)$ is a row stochastic matrix, where $\sum_j T_{ij} = 1$ and $T_{ij} \geq 0, \forall i, j$. It is assumed here, that the system is Markovian, which means the probability of current state depends only on the previous state but not on the history of how it arrived at the current state, i.e. $p(\boldsymbol{x}_j, t+\tau) = p(\boldsymbol{x}_j, t+\tau|\boldsymbol{x}_i, t)p(\boldsymbol{x}_i) = T_{ij}(\tau)p(\boldsymbol{x}_j)$, it is known as conditional independence property. The dynamics of the system can be described by using transition probability matrix if and only if the system in markovian. It is thereby also assumed that transition probabilities between the states or the elements of $\boldsymbol{T}(\tau)$) do not change with time, otherwise it won't be valid. If $\boldsymbol{\pi}(t)$ is a row vector representing the probabilities over microstates at time t , then time evolution of the

probability distribution is written as, $\boldsymbol{\pi}(t+\tau) = \boldsymbol{\pi}(t)\boldsymbol{T}(\tau)$. The aim is find a perfect permutation of the transition probability matrix such that it transforms into a almost block diagonal form where elements within a block has much higher value than off-block elements. The eigen value problem $\boldsymbol{TV} = \boldsymbol{V\Lambda}$ cab be solved to obtain $\boldsymbol{V}$ as the eigen vector matrix and $\boldsymbol{\Lambda}$ with eigen values. These eigen values represent specific relaxation timescales for the system, $t_i = -\frac{\tau}{\ln \lambda_i}$. PCCA [38] and PCCA+ [39] are two well known methods for performing the spectral clustering of the transition matrix. Using the eigen vectors corresponding to positive and non zero eigen values, microstates are further projected onto a reduced dimensional space. The final step involves grouping these microstates into a small number of clusters that represent the actual metastable states and it can be done using any geometric clustering. Choice of the lag time τ, is a system specific parameter which controls the quality of kinetic modeling. It should be sufficiently large so that system loses memory of its history but not too large that the system is no more markovian.

Comparing K-means, GMM and HDBSCAN

To illustrate how these different methdos compare, I here test three popular clustering methods- K-means, GMM and HDBSCAN using Wine dataset from `scikit-learn` [40]. The Wine dataset contains 178 samples with 13 features each, representing 3 types of wine [41]. For K-means and GMM, we set the initial number of clusters to 3, while HDBSCAN does not require a predefined cluster count. In case of K-means and GMM, I used the k-means++ initialization scheme, a tolerance value of 0.0001, a maximum of 300 iterations, and ran 20 times to find the best fit. For HDBSCAN, we set the minimum cluster size to 8 and used the Euclidean distance metric. Figure 1.1 illustrates the clustering results, where data points are projected onto two dimensions using the first two LDA components (LD1 and LD2). The samples are color-coded based on their true class labels (ground truth) and the predicted cluster assignments from each method. Among the three methods, K-means performed best, accurately assigning most samples to their correct clusters. GMM successfully identified two clusters but struggled with the third. HDBSCAN exhibited a

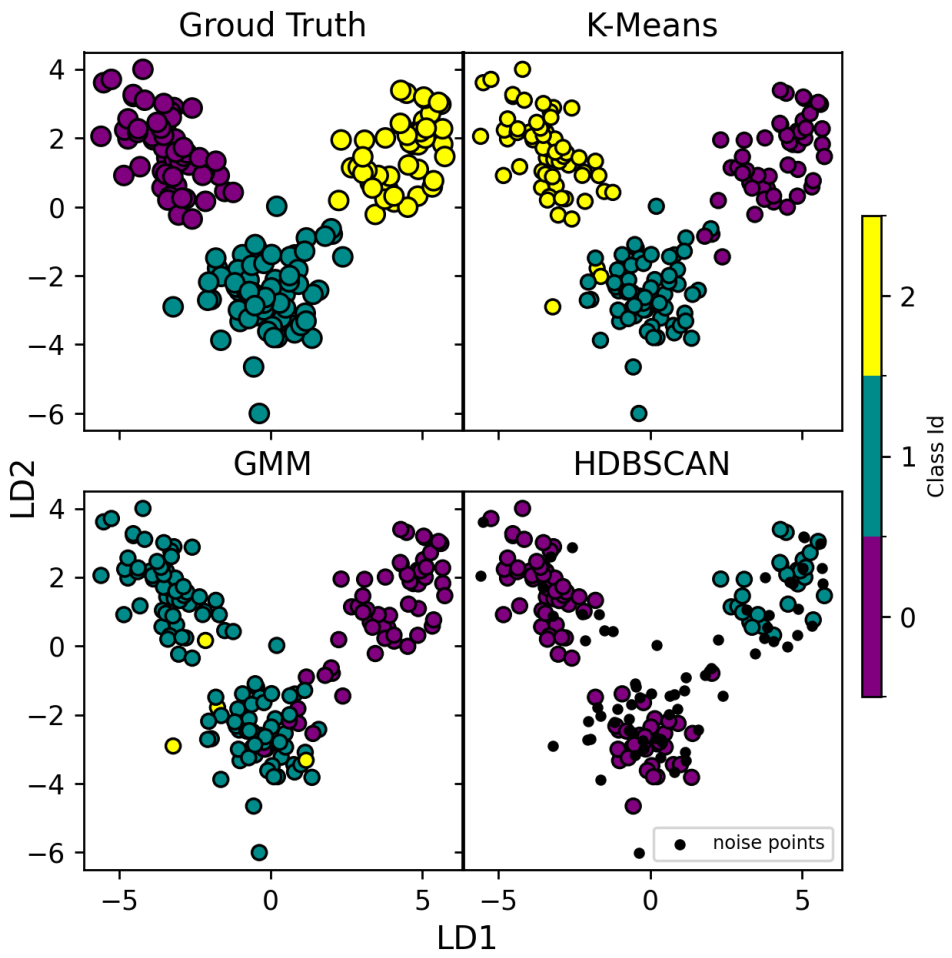

Figure 1.1: Comparison of Clustering Algorithms. Wine data set from UC Irvine Machine Learning Repository benchmark datasets, is used to analyze the performance of K-means, GMM and HDBSCAN. The data points are projected along first two LDA components and colored according to their cluster assignments. Ground truth represents true class labels for data points.

similar trend, grouping the data into two clusters while designating the remaining samples as noise points. The effectiveness of a clustering algorithm is highly dependent on the complexity and characteristics of the dataset. While K-means performed well in this case, there are scenarios where GMM or HDBSCAN might be more suitable. To quantify clustering performance, we computed the Silhouette Score [42], a metric that evaluates similarity of samples is to its own cluster compared to other clusters. The score ranges from -1 to 1, where 1 indicates well separated clusters, values near 0 suggest closely spaced but distinct clusters, and -1 represents poor clustering

results. The scores are 0.285, 0.250 and 0.098 respectively for k-means, GMM and HDBSCAN. This simple comparison on the Wine dataset aims to demonstrate how different clustering algorithms operate rather than to declare a definitive best method.

1.3.3 Dimensionality Reduction Techniques

Dimensionality reduction techniques are widely used in analyzing MD simulations by projecting the data from a high dimensional space to a much lower dimension that can still retain relevant information about the system. The underlying idea is to transform the data matrix $\boldsymbol{X}_{n\times d}$ into a new representation $\boldsymbol{X}^{'}$ of dimension $n \times k$, where n is number of samples and $k << d$. Quality of this transformation depends on how much information from high dimensional space can be preserved in reduced dimension. Dimensionality reduction methods are necessary for visualizing high dimensional data in a lower dimension by means of projection and very often used as compulsory preliminary step in clustering algorithms to obtain distinct states from MD simulation data. In some cases, the transformed coordinates can also be used as collective variables in enhanced sampling simulations to characterize transitions between metastable states and to enhance the fluctuations along slow degrees of freedom thereby increasing chances of rare event sampling. There are many dimensionality reduction algorithms that have been used in MD studies; they can be separated in two major categories - (**1**) Linear Dimensionality reduction techniques such as Linear Discriminant Analysis (LDA) [31], Principal Component Analysis (PCA) [31], Time lagged Independent Component Analysis (t-ICA) [43]. These methods aim to find a optimal projection by using a linear combination of input features, along which the data can be best understood. (**2**) Non-linear dimensionality reduction techniques such as t-distributed Stochastic Neighbor Embedding (t-SNE) [44], Uniform Manifold Approximation and Projection (UMAP) [45], diffusion maps [46], Isomap [47], Kernel PCA [48] etc. They focus on finding the best low dimensional embedding as some non linear complex functions of input features, that can efficiently describe both the local and global structures in the data set.

Principal Component Ananlysis

Principal Component Analysis (PCA) is a very simple and popular dimensionality reduction technique that has been used to analyze different types of data. PCA aims to find a set of orthogonal coordinates (called Principal Components or PCs) in the direction of maximum variance of the data set. PCs are also frequently used as collective variables in enhanced sampling simulations of biomolecules. For a given data set $\boldsymbol{X}_{M\times d}$, where M is number of samples in the data and d is number of features that describe a single configuration in high dimensional space, first the data set is centered by subtracting the average, $\boldsymbol{X}_c = \boldsymbol{X} - \boldsymbol{\mu}$, where $\mu = \sum_{i=1}^{M} \boldsymbol{x}_i$ is a vector representing feature means. Centering the data makes sure that PCs will be translationally invariant. Then the covariance matrix is computed as, $\boldsymbol{C} = \frac{1}{M}\boldsymbol{X}_c{}^T\boldsymbol{X}_c$, which has dimension of $d \times d$. By diagonalizing the covariance matrix as $\boldsymbol{CY} = \boldsymbol{Y\Lambda}$, we can get the eigen vectors $\boldsymbol{y}_i, \forall i = 1, \ldots, d$ that form the columns of matrix $\boldsymbol{Y}$. The eigen vectors are PCs which are then arranged according to decreasing order of eigen values ($\lambda_1 > \lambda_2 > \cdots > \lambda_d$). The component with largest eigen value is called PC1 that always incorporates maximum amount of variance in the data set. If one decides to use first j PCs, then the transformed data is given by, $\boldsymbol{X}^{'} = \boldsymbol{X}_c\boldsymbol{W}$, where $\boldsymbol{W} = \left[\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_j\right]_{d\times j}$. In practice $j << d$, so that dimensionality of the data is reduced significantly. For example, to capture 90% of total variance of the data set in low dimensional space, the number of PCs is chosen such that, $\sum_{a=1}^{j} v_a \geq 0.90$ where $v_a = \frac{\lambda_a}{\sum_{b=1}^{d}\lambda_b}$, is called explained variance ratio that gives the amount of variance captured by each component.

Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a supervised classification technique. It is mainly used for dimensionality reduction and classification problems in Machine Learning (ML). It reduces the dimensionality of the data by projecting them in a direction along which different clusters are best separated from each other. For K clusters, LDA provides $K - 1$ linearly independent projections

of the input data, among which the one with largest eigen value is always the best to capture patterns in data. LDA finds such linear transformation by maximizing between the class scatter matrix ($\tilde{S_b}$) and minimizing within the class scatter matrix ($\tilde{S_w}$) in the projected space. Unlike PCA, here the data set is labeled, that means each configuration in the data set is a member of a particular class. For K classes, between the class scatter matrix of feature space is defined as,

$$S_b = \sum_{i=1}^{K} n_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T \tag{1.7}$$

and within the class scatter matrix as.

$$S_w = \sum_{i=1}^{K} \sum_{j \in n_i} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T \tag{1.8}$$

Here, M is total number of samples in the data set, n_i is number of samples in cluster i, $\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{j \in n_i} \boldsymbol{x}_i$, is the mean of features for i^{th} cluster and $\boldsymbol{\mu} = \frac{1}{M} \sum_{i=1}^{M} \boldsymbol{x}_i$ is the global mean of the entire data set. There is a linear dependence of the cluster means, directly related to the definition of global mean by, $\sum_{i=1}^{K} n_i(\boldsymbol{\mu} - \boldsymbol{\mu_i}) = 0$. S_b describes the spread of the cluster means from global mean and S_w captures the variance of the data within a class. Here the objective is to find a optimal linear transformation G which will transform the data as $\boldsymbol{y}_i = G^T \boldsymbol{x}_i$, such that classes are best separated in projected space. The fisher's objective function is defined as, $J(G) = \frac{\tilde{S_b}}{\tilde{S_w}}$, that has to be maximized for an optimal G^*. It can be shown that, $J(G)$ can be expressed in terms of scatter matrices S_b and S_w in feature space,

$$J(G) = \frac{G^T S_b G}{G^T S_w G} \tag{1.9}$$

this is also known as Rayleigh quotient. Maximizing this ratio with respect to G is equivalent to solving the generalized eigen value problem, $S_b G = \lambda S_w G$. If S_w is non-singular, which means the inverse (${S_w}^{-1}$) exists, it is reduced to a simple eigen value problem. Solving this eigen value

problem, returns $K-1$ linearly independent eigen vectors with non-zero eigen values. For $K=2$, only a single LDA coordinate is obtained and it is given as,

$$\boldsymbol{y}(\boldsymbol{x}) = \frac{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T}{\Sigma_1 + \Sigma_2}\boldsymbol{x} \tag{1.10}$$

LDA coordinates have also been used as potential collective variable in enhanced sampling simulations. It has been shown that biasing LDA coordinate can promote better sampling of FES which results in faster convergence of free energy path. Different versions of LDA method have also been proposed such as deep-LDA [49, 50], HLDA (harmonic LDA) [51, 52].

t-Distributed Stochastic Neighbor embedding

It is a non linear dimensionality reduction algorithm, primarily used for visualizing high dimensional complex manifolds of data in lower dimensions (typically 2D or 3D). t-SNE is very effective in preserving both local and global structures in the data. It operates by translating similarities among data points in high dimensional space into probabilities and then mapping them onto a lower dimensional representation while preserving these probabilities. For a given data set $\boldsymbol{X}_{M\times d}$, pairwise similarities are calculated as,

$$p_{j|i} = \frac{\exp\left(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2 / 2{\sigma_i}^2\right)}{\sum_{k\neq i}\exp\left(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_k\right\|^2 / 2{\sigma_i}^2\right)} \tag{1.11}$$

$\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2$ is the euclidean distance between points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. $p_{j|i}$ is the conditional probability, that $\boldsymbol{x}_i$ would pick $\boldsymbol{x}_j$ as its neighbor. $p_{j|i}$ is described by a gaussian centered at $\boldsymbol{x}_i$, it has higher values for the data points which are nearby and very low values for those which are widely separated. The denominator in the above exxpression is a normalization factor that ensures $\sum_{j\neq i} p_{j|i} = 1$. And σ_i is the variance of the gaussian centered at $\boldsymbol{x}_i$, values of them are chosen such that perplexity matches with the user specified value. Perplexity is a hyperparam-

eter in t-SNE algorithm, it is defined as, $Perp(P_i) = 2^{H(P_i)}$. $H(P_i) = \sum_{j \neq i} p_{j|i} \log_2^{p_{j|i}}$, is called the Shannon entropy and P_i is the set of all conditional probabilities $\{p_{j|i}, \forall j \neq i\}$. $Perp$ controls the balance between preserving local and global structures in the data. While a low value of $Perp$ makes t-SNE focuses on very local structure in the data, increasing its value can change the behavior by considering a large number of neighbors thereby capturing global structure better. It has been shown that, a $Perp$ value of 5 to 50 works substantially good for any kind of data sets [44]. In practice, for each data point $\boldsymbol{x}_i$, t-SNE performs a binary search to find the value of σ_i that satisfies, $Perp(P_i) = Perp_{target}$. The joint probability p_{ij} is then symmetrized as,

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2M} \tag{1.12}$$

In the lower dimensional space, the Student t-distribution with one degree of freedom (also known as Cauchy distribution) is used to model the pairwise similarities between the data points. The joint probability distribution is defined as,

$$q_{ij} = \frac{\left(1 + \left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \left\|\boldsymbol{y}_k - \boldsymbol{y}_l\right\|^2\right)^{-1}} \tag{1.13}$$

where, $\{\boldsymbol{y}_i \, \forall i = 1, \ldots, M\}$ are the configurations in projected space. The use of Cauchy distribution in projected space instead of gaussian (like in case of SNE), alleviates the problem of crowding, where the data points from different clusters get very close to each other when projected in 2D or 3D from very high dimensional space [44]. The aim is to find the optimal mapping such that pairwise similarity in the high dimensional space can be perfectly preserved in lower dimensional representation. The cost function is defined using Kullback-Leibler divergence (KL) as,

$$C = \mathrm{KL}(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{1.14}$$

Cost function gives a measure of overlap between probabilities p_{ij} and q_{ij} for all $i, j = 1, \ldots, M$. It has the property that $p_{ij} = p_{ji}$ and $q_{ij} = q_{ji} \forall i, j$. The cost function is minimized with respect to $\boldsymbol{y}_i$ using gradient descent method. The gradient of the cost function is given as [44],

$$\frac{\delta C}{\delta \boldsymbol{y}_i} = 4 \sum_j (p_{ij} - q_{ij})(\boldsymbol{y}_i - \boldsymbol{y}_j)\left(1 + \left\|\boldsymbol{y}_i - \boldsymbol{y}_j{}^2\right\|\right)^{-1} \tag{1.15}$$

After initializing, the projected variables $\{\boldsymbol{y}_i\}$ form a multivariate gaussian distribution with zero mean and unit covariance, the embeddings are updated iteratively using the equation below,

$$\boldsymbol{y}_i{}^{(t)} = \boldsymbol{y}_i{}^{(t-1)} + \eta \frac{\delta C}{\delta \boldsymbol{y}_i} + \alpha(t)(\boldsymbol{y}_i{}^{(t-1)} - \boldsymbol{y}_i{}^{(t-2)}) \tag{1.16}$$

where η is learning rate and $\alpha(t)$ is momentum term. So, the hyperparameters for t-SNE are $Perp$, η, α and number of iterations. This algorithm is efficient in understanding the patters or clusters of complex manifolds but computationally very expensive (scales as $O(M^2)$), especially for large data sets.

Comparing PCA, LDA and t-SNE

To illustrate how the dimensionality reduction works, here I conducted a comparative analysis of three popular dimensionality reduction techniques— PCA, LDA, and t-SNE using the MNIST dataset with scikit-learn [40]. The MNIST dataset [53] consists of 70,000 handwritten digits (0–9), each represented as a 28×28 pixel image with 784 features. It is widely used in machine learning to benchmark various methods due to its high-dimensional and complex nature. For simplicity, we selected only three classes randomly (2, 5, and 9), resulting in a dataset of approximately 20,000 samples. The input data was first rescaled such that it has zero mean and unit variance for features. Figure 1.2(A) presents the projection of these samples onto two dimensions, with points color-coded by class. Among the three methods, t-SNE performed the best, effectively

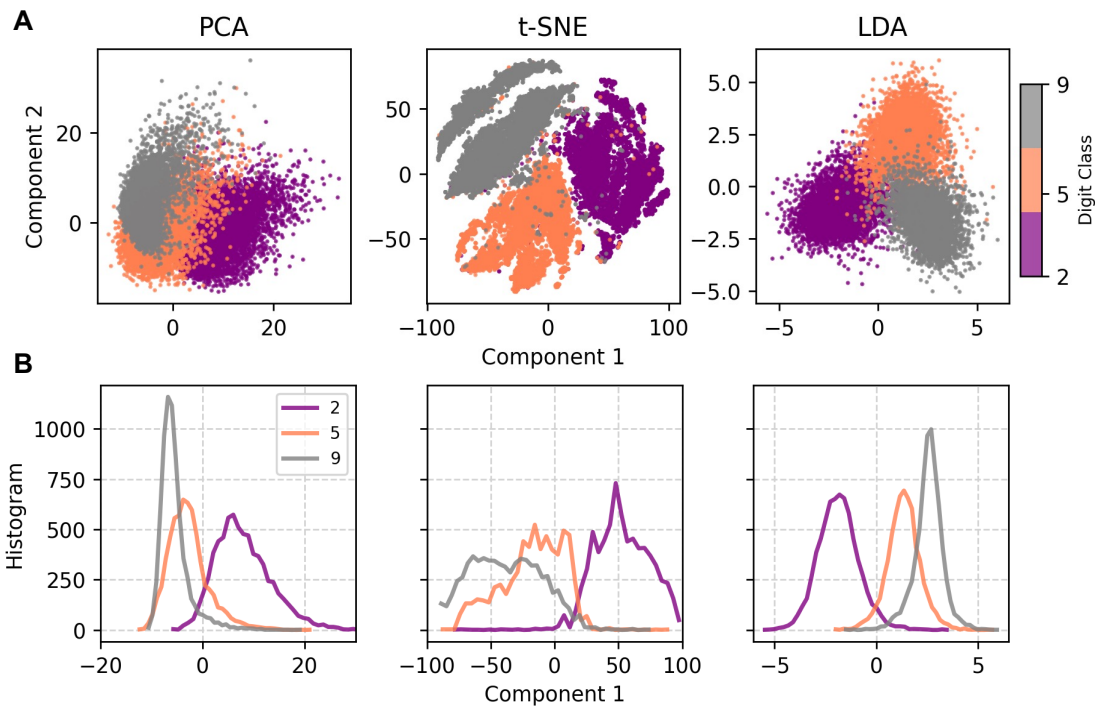

Figure 1.2: Comparison of Dimensionality Reduction techniques. Analyzed performance of PCA, LDA and t-SNE on a subset of the MNIST dataset (digits 2, 5, and 9). (**A**) Scatter plots showing the projections of the data points along first two components from each method. Data points are colored according to their true digit class labels. (**B**) Histogram of three classes along the first component in each case.

identifying and distinguishing clusters in lower dimensions with minimum overlap between the classes. LDA performed quite well, particularly considering that it is a linear method and computationally less expensive than t-SNE. This is expected, as LDA is a supervised technique that maximizes class separation while minimizing variance within a class. PCA, on the other hand, struggled to separate clusters, which aligns with its nature as an unsupervised method that focuses on maximizing variance rather than class separability. Figure 1.2(B) further illustrates how the samples are distributed along the first component in each method. Notably, the first linear discriminant (LD1) in LDA provided a lot better smooth separation between clusters compared to the first principal component in PCA or t-SNE's first axis. This also suggests that LD1 could be particularly useful in identifying state transitions between metastable states in biomolecular simulations.

2 Reaction coordinates for conformational transitions using Linear Discriminant Analysis on Positions

This chapter has been adapted from Ref. [54]

2.1 Introduction

A large class of enhanced sampling techniques work by biasing a system to explore along a low dimensional set of collective variables (CVs) [55]. These methods allow us, in principle, to use the known applied bias to reconstruct the free energy landscape in that low dimensional space. In practice, the choice of the CVs is crucial, with an ideal set of CVs allowing the system to explore all relevant states within available simulation time [55]. Recently, extensive effort has been invested in using a variety of machine learning approaches, from very simple to very sophisticated, to determine optimal coordinates for sampling from molecular dynamics (MD) simulation data (Refs. [49, 51, 52, 56–72] provide a representative but not exhaustive sample).

One commonly encountered challenge is to compute the free energy path of a transition between two states along a linear dimension that chemists term a reaction coordinate (RC). For a macromolecule such as a protein, the two states could be configurations for which we have known structures (e.g. the PDB structure of a protein solved with and without a bound ligand),

or processes for which one state is known and the other can be at least qualitatively defined (e.g. folding/unfolding or binding/unbinding). If a long MD trajectory containing multiple transitions between these states is available, then reaction coordinates could be trained based on the idea that we want to enhance sampling along the slowest modes in the system [50, 58, 62, 65, 66, 73]. However, having this data is rare, in which case one can try iterative enhanced sampling and learning reaction coordinates with the goal of maximizing the number of transitions between the two states in a fixed amount of simulation time [58, 59, 63, 65, 67, 74].

An alternative approach which has shown some success is to train reaction coordinates based on short simulations within the two states, and use a method that produces a coordinate representing the difference between the two sets of data. Linear dimensionality reduction techniques such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are the simplest approaches for combining a large set of variables that describe a system of interest to produce a small set of CVs that characterize the available data. While PCA, which produces coordinates that capture the most variance in the data, has been used to promote exploration in enhanced sampling simulations, LDA seems to hold more promise as an RC since it is a supervised approach designed to maximally separate different labeled classes of data (i.e. reactants and products). We describe LDA in full detail in the next section. In one study, Mendels et al. [52] produced a modified approach to LDA termed harmonic LDA (HLDA, because the covariance matrices in the two different states of interest are combined by a harmonic average rather than a simple sum), and in that work and subsequent ones,[51, 61] combined it with Metadynamics (MetaD) to effectively enhance sampling between two states in several different systems. Later, a neural network was used to combine features before training LDA vectors to produce the reaction coordinate [49].

In the prior examples of reaction coordinate design for free energy sampling of biomolecules that we are aware of, the input features to the method were internal coordinates, or a function of internal coordinates, for the molecule(s) of interest—for example, distances, angles, and dihedrals.

Often, these could be CVs based not on atomic positions directly, but on coarse-grained (CG) representations of the biomolecule, such as the distance between the centers of masses (COMs) of two different domains, or the distance between the COM of a ligand and certain atoms in its binding pocket. This is not surprising, because these often correspond to our physical intuition about the biomolecular reaction coordinate. Moreover, internal coordinates are invariant to translation and rotation of the molecule, and thus bias forces applied to these coordinates do not depend on the position or orientation of the molecule.

Recently, we presented atomic coordinates as an alternative set of features to use in the context of clustering biomolecular data [75]. Atomic coordinates of a subset of atoms, or of beads corresponding to a CG representation of a molecule, offer an alternative to internal coordinates with the advantage that there is little choice in selecting the features to use. Using a protein as an example, we need only make the standard choice between C_α atoms, backbone, all heavy atoms, and so on. Moreover, only $3N-6$ atomic coordinates essentially describe the state of a biomolecular system with N important atoms (but ignoring contributions of solvent, salt, etc.), whereas use of internal coordinates often results in an over-determined set of features, such as all $O(N^2)$ pairs of distances. In Ref. [75] we developed a procedure for clustering molecular configurations into a Gaussian mixture model (GMM) using atomic positions that overcomes challenges of orientational dependence that prevented their use earlier, as described below. Because a Gaussian mixture model in positions is a natural way to coarse-grain a free energy landscape,[75–78] with locally harmonic bins around metastable states, the resulting clustering is a physically appealing definition of the "states" a molecule can adopt.

However, our Gaussian mixture model still relies on a very high $(3N-6)$ dimensional representation of our molecule. Given that the output of our clustering algorithm is a set of states each defined by a multivariate Gaussian distribution, LDA is a natural approach to produce a low dimensional representation of our data with large separation between states. In this work, we first apply LDA to the folded and unfolded states determined from shapeGMM clustering of a long

unbiased MD trajectory of a fast-folding protein, and demonstrate that it produces a physically reasonable ordering of states from folded to unfolded. We then show that this coordinate is a "good" reaction coordinate because the position of the barrier separating folded and unfolded is very close to the location where the system is equally likely to proceed to folded or unfolded (in terms of a committor function to be defined below). We implement this position LDA coordinate in the PLUMED sampling library, and demonstrate that biased sampling along this coordinate can accelerate transitions between the folded and unfolded states, and produce a qualitatively similar free energy surface as compared to the unbiased trajectory in 3% of the simulation time, without any additional tuning of the CV. Finally, we train a position LDA coordinate on an achiral helical system where data is only available in the left and right-handed states, and show that this coordinate also allows us to readily sample between the two states, despite there being no information about the transition provided during training.

2.2 Theory and Methods

2.2.1 Molecules in Size-and-Shape Space

Consistent with our previous work on structural alignment and clustering,[75] we consider structures from a MD simulation to be associated with Gaussian distributions in atomic positions. Structures are represented by N particles (a subset of atoms) using a vector $\boldsymbol{x}$ of dimension $N \times 3$ which is a member of an equivalence class,

$$[\boldsymbol{x}_i] = \{\boldsymbol{x}_i \boldsymbol{R}_i + \mathbf{1}_N \vec{\xi}_i^T : \vec{\xi}_i \in \mathbb{R}^3, \boldsymbol{R}_i \in \mathrm{SO}(3)\}, \tag{2.1}$$

where $\vec{\xi}_i$ is a translation in $\mathbb{R}^3$, $\boldsymbol{R}_i$ is a rotation $\mathbb{R}^3 \to \mathbb{R}^3$, and $\mathbf{1}_N$ is the $N \times 1$ vector of ones. $[\boldsymbol{x}_i]$ is a point in size-and-shape space[79] which has dimension $3N - 6$ and is defined as $S\Sigma_N^3 = \mathbb{R}^{3N}/G$ where $G = \mathbb{R}^3 \times \mathrm{SO}(3)$ is the group of all rigid-body transformations for each frame with elements

$\boldsymbol{g} = (\vec{\xi}, \boldsymbol{R})$.

Within the shapeGMM framework, the probability density of particle positions is assumed to be a Gaussian mixture,

$$P(\boldsymbol{x}_i) = \sum_{j=1}^{K} \phi_j N(\boldsymbol{x}_i \boldsymbol{g}_{i,j} \mid \boldsymbol{\mu}_j, \Sigma_j), \tag{2.2}$$

where $N(\boldsymbol{x}_i \boldsymbol{g}_{i,j} \mid \boldsymbol{\mu}_j, \Sigma_j)$ is the jth normalized, multivariate Gaussian with mean $\boldsymbol{\mu}_j$, covariance matrix Σ_j, and weight ϕ_j (the weights are normalized such that $\sum_{j=1}^{K} \phi_j = 1$). $\boldsymbol{g}_{i,j}$ is the element of G that minimizes the Mahalanobis distance between $\boldsymbol{x}_i$ and $\boldsymbol{\mu}_j$. Iterative determination of $\boldsymbol{g}_{i,j}$ and $\boldsymbol{\mu}_j$ is performed in a Maximum Likelihood procedure [75].

In the current work, we will consider LDA coordinates learned using data from only two states. Additionally, we will only consider "weighted" alignment of particle positions, which equates to using a Kronecker product covariance (where $\Sigma_j = \Sigma_N \otimes I_3$, for Σ_N the $N \times N$ covariance of particle positions) in defining the Mahalanobis distance between frame and average structure as described in detail in Ref. [75].

2.2.2 Dimensionality Reduction using Linear Discriminant Analysis on Particle Positions

We propose to use LDA directly on aligned particle positions as a reaction coordinate. LDA for two states produces the linear model with the maximal inter-average variance while minimizing intra-cluster variance[31]. For K different clusters, this is achieved by first computing the within-cluster scatter matrix,

$$\boldsymbol{S}_w = \sum_{i=1}^{K} \sum_{j \in N_i} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T, \tag{2.3}$$

and the between-cluster scatter matrix,

$$\boldsymbol{S}_b = \sum_{i=1}^{K} (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T, \tag{2.4}$$

where $\boldsymbol{\mu}_i$ is the average structure of cluster i and $\boldsymbol{\mu}$ is the global average. The simultaneous minimization of within-cluster scatter and maximization of between cluster scatter can be achieved by finding the transformation G that maximizes the quantity

$$\mathrm{Tr}\left((G^T \boldsymbol{S}_w G)^{-1} G^T \boldsymbol{S}_b G\right). \tag{2.5}$$

This maximization can be achieved through an eigenvalue/eigenvector decomposition but such a procedure is only applicable when $\boldsymbol{S}_w$ is non-singular. The LDA method was reformulated in terms of the generalized singular value decomposition (SVD) [80] extending the applicability of the method to singular $\boldsymbol{S}_w$ matrices such as those encountered when using particle positions.

In addition to employing the SVD solution to the LDA approach, care must be taken in how particle positions are aligned when performing LDA. This is evident when one considers the scatter matrices in (2.3) and (2.4). The values and null spaces of these scatter matrices will depend on the specific alignment procedure chosen. There are three obvious choices for structural alignment prior to LDA: (1) alignment of each frame to its respective cluster mean/covariance, (2) alignment to one cluster or another, and (3) alignment to a global average. The first choice will lead to scatter matrices with different null spaces for each cluster making their addition in (2.3) unsatisfactory. Alignment to a cluster mean will yield consistent null spaces for each cluster but requires distinct alignment reference and global average structures. Additionally, aligning to a cluster mean yields to an undesirable ambiguity (and asymmetry) in the choice of cluster. Alignment to a single global average overcomes all of these issues and, as we show in the Appendix 2.6, yields a sampling coordinate that is at least as good as alignment to a cluster mean for the systems tested here.

The result of an LDA procedure on two labeled states will be a vector, $\boldsymbol{v}$, of coefficients that best separate the two states. These vectors are similar in nature to the eigenvectors from PCA, a procedure more familiar to the bio-simulation field.

2.2.3 Biasing a linear combination of positions

The value of the LDA coordinate after this procedure is a dot product of the vector $\boldsymbol{v}$ with the atomic coordinates $\boldsymbol{x} - \boldsymbol{\mu}$. When computing this value on the fly within an MD simulation, we need to consider the value of $\boldsymbol{[x(t)]}$, the equivalence class of the position at time t, translated and rotated to a reference $\{\boldsymbol{\mu}, \Sigma\}$.

Therefore, to compute the value of the LDA coordinate l, we first translate $\boldsymbol{x(t)}$ by $\vec{\xi}(t) = \frac{1}{N}\sum_{i=1}^{N}\vec{x}_i(t) - \frac{1}{N}\sum_{i=1}^{N}\vec{\mu}_i(t)$, the difference in the geometric mean of the current frame and that of the reference configuration. Then, we compute $\boldsymbol{R}(t)$, the rotation matrix which minimizes the Mahalanobis difference between $\boldsymbol{x(t)} - \vec{\xi}$ and $\boldsymbol{\mu}$, for a given Σ, as described in Ref. [75]. Finally, we compute

$$l(\boldsymbol{x}) = \boldsymbol{v} \cdot \left(\boldsymbol{R} \cdot (\boldsymbol{x}(t) - \vec{\xi}(t)) - \boldsymbol{\mu}\right) \tag{2.6}$$

By definition, $l(\boldsymbol{\mu}) = 0$.

To apply bias forces to this coordinate, we must be able to compute $\nabla l(\boldsymbol{x}(\mathrm{t}))$. Because of the inclusion of the optimal rotation process by SVD, it is non-trivial to compute this analytically, and we instead compute derivatives numerically.

2.2.4 Enhanced sampling with OPES-MetaD

Enhanced sampling simulations on LDA coordinates were performed using Well-tempered Metadynamics (WT-MetaD), and On the Fly Probability Enhanced Sampling-Metadynamics (OPES-MetaD) as implemented in `PLUMED` [25, 81–83].

WT-MetaD works by adding a bias formed from a history dependent sum of progressively shrinking Gaussian hills [24, 84]. The bias at time t for CV value Q_i is given by the expression

$$V(Q_i, t) = \sum_{\tau<t} h e^{-V(Q_i,\tau)/\Delta T} e^{-\frac{Q(\boldsymbol{x}(\tau))-Q_i)^2}{2\sigma^2}}, \tag{2.7}$$

where h is the initial hill height, σ sets the width of the Gaussians, and ΔT is an effective sampling temperature for the CVs. Rather than setting ΔT, one typically chooses the bias factor $\gamma = (T + \Delta T)/T$, which sets the smoothness of the sampled distribution [24, 84]. Asymptotically, a free energy surface (FES) can be estimated from the applied bias by $F(Q) = -\frac{\gamma}{\gamma-1}V(Q, t \to \infty)$ [84, 85] or using a reweighting scheme [84, 86].

In contrast to the use of sum of Gaussians in traditional MetaD, OPES-MetaD applies a bias that is based on a kernel density estimate of the probability distribution over the whole space, which is iteratively updated [25, 83]. The bias at time t for CV value Q_i is given by the expression

$$V(Q_i) = k_B T \left(\frac{\gamma - 1}{\gamma}\right) \log\left(\frac{P_t(Q_i)}{Z_t} + \epsilon\right). \tag{2.8}$$

Here in the prefactor, T is the temperature, k_B is Boltzmann's constant, and γ is the bias factor. $P_t(Q)$ is the current estimate of the probability distribution, Z_t is a normalization factor that comes from integrating over sampled Q space. Finally, $\epsilon = \exp\left(\frac{\Delta E}{k_B T}\frac{\gamma}{\gamma-1}\right)$ is a regularization constant that ensures the maximum bias that can be applied is ΔE. For one of our systems, we found that limiting the maximum bias using OPES-MetaD helped prevent unphysical exploration along our LDA coordinate (this is also possible using other approaches such as Metabasin Metadynamics [87]). Even with this limitation, we apply additional wall potentials to prevent exploration well beyond the LDA values for each of our two states. As in WT-MetaD, $F(Q)$ can be directly estimated from $V(Q)$ by $F(Q) \approx -\frac{\gamma}{\gamma-1}V(Q)$ or through a reweighting scheme [25]. Details of the sampling parameters used for each system are given in Section 2.5.

2.2.5 Implementation

Clustering and iterative alignment of trajectory frames prior to learning LDA vectors is performed using our `shapeGMMTorch` package, which is a high performance re-implementation of the methods from Ref. [75], implemented with `pyTorch` [88] for accelerated computation on

GPUs. `shapeGMMTorch` is available from https://github.com/mccullaghlab/shapeGMMTorch and can easily be installed in python using the command `pip install shapeGMMTorch`. We have also created a wrapper library for the training of LDA vectors directly from positional data, which is available from https://github.com/mccullaghlab/pLDA and which can be easily installed with `pip install posLDA` (although this wrapper was not used in the analysis performed in this paper as it was not yet available). Within posLDA, vectors are learned using the SVD implementation of the `scikit-learn` LinearDiscriminantAnalysis package [40].

In order to compute and bias these vectors on the fly within MD simulations, the optimal alignment and linear combination procedure has been implemented in the `PLUMED` open source library [81, 82]. All procedures, analysis for every case studied in this work, and `PLUMED` code are made available at https://github.com/hocky-research-group/posLDA_paper_2023, and the code for computing LDA coordinates and Mahalanobis distances on positions will be contributed as a module to `PLUMED` shortly.

2.3 Results and Discussion

2.3.1 LDA is a Good Reaction Coordinate for HP35 Folding

In previous work, we applied our shapeGMM clustering approach to a 305 μs trajectory of a 35-amino acid fast-folding folding mutant Villin headpiece domain (HP35), obtained from the D.E. Shaw Research Group [89]. From our data, we choose to study a six state representation of the data, whose states produce an interpretable representation of folding and unfolding, and which is found not to be overfit by a cross-validation approach. Details of the clustering and cross-validation are provided in Ref. [75]. The definition of this six state model, $\{\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}\}_{K=6}$ is trained from 25,000 frames out of $\sim$ 1.5 million, and then all frames are assigned to clusters based on which cluster center it is closest to in terms of Mahalanobis distance on positions.

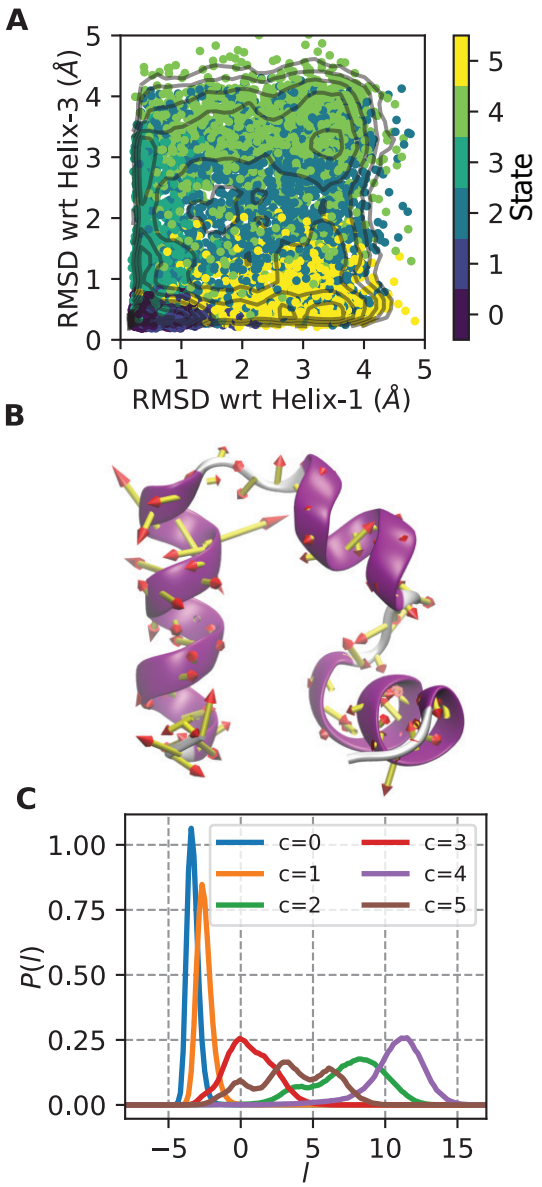

Figure 2.1: Folding/unfolding coordinate for HP35. (**A**) Points from HP35 trajectory are colored by state assignment and mapped into natural folding coordinates of the RMSD of residues in helix 1 or helix 3 to that in the folded state (which is a 3 helix bundle). State 0 is the most folded and state 4 is the most unfolded. Contours shown are every 0.5 kcal/mol in the range (0,6). (**B**) Porcupine plot showing the magnitude of the LDA coefficients trained only on states 0 and 4 from A, overlaid on the starting HP35 structure. (**C**) Histogram of LDA coordinate l for each separate state. l evenly separates all states, with state 0 and 4 at maximum separation.

A single folding/unfolding coordinate is constructed by performing LDA on frames assigned to the folded and unfolded states. The folded and unfolded states were assigned based on the

RMSD to folded helix 1 and RMSD to folded helix 2 2D map shown in Figure 2.1A for this long trajectory with points colored by the assigned states. From this figure, we can assign state 0 as the folded state because it is the state with lowest RMSDs (it also has the largest population) and state 4 as the most unfolded state because it is the state with the largest RMSDs. LDA is performed on these two states to produce a single LD vector, denoted l, after an iterative alignment of the amalgamated two-state trajectory to the global mean and covariance, as described above. The magnitude of the coefficients in this vector are illustrated as particle displacement vectors in the porcupine plot in Figure 2.1B. The histogram in Figure 2.1C shows the l values adopted in each state. We see from this data that this coordinate separates state 0 ($l \approx -3$) and state 4 ($l \approx 12$). To our surprise, this single coordinate, which was trained only on data from state 0 and state 4, separates the other four states as well, which suggests that it might be sufficient to produce transitions between folded and unfolded through physically meaningful configurations.

Figure 2.2A shows the variation of l versus time for this long trajectory, and exhibits many transitions between the folded ($l \approx -3$) and unfolded ($l \approx 12$) states (for comparison, Ref. [90] found that this long trajectory contains 61 folding transitions with their definition of folding). In order to assess the quality of this CV, we compute the committor of each frame in the trajectory $c(\boldsymbol{x_t})$ [30, 56, 91], which for time t is 1 if the system reaches a folded state before reaching an unfolded state in the times following t.

To assess the quality of a reaction coordinate, we can compute the committor probability for each value of l on a grid of size δl.

$$P_c(l_i) = \frac{1}{M_i} \sum_{t=1}^{N_{\text{frames}}} c(\boldsymbol{x_t})[l(\boldsymbol{x_t}) \in (l_i - \delta l, l_i + \delta l)] \tag{2.9}$$

$$M_i = \sum_{t=1}^{N_{\text{frames}}} [l(\boldsymbol{x_t}) \in (l_i - \delta l, l_i + \delta l)]. \tag{2.10}$$

In Figure 2.2B, we show the approximate FES along l computed as $F(l) = -k_B T \ln P(l)$ for the

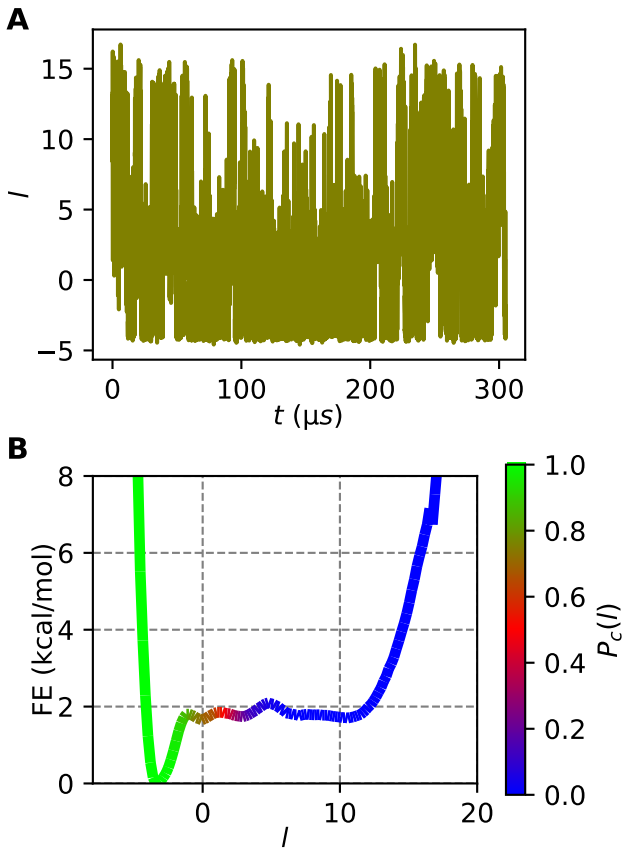

Figure 2.2: LDA results for the folding/unfolding of HP35 from unbiased MD. (**A**) LDA coordinate trained on states 0 and 4 vs. time for the full 305 μs HP35 trajectory shows many transitions between folded (~ -3) and unfolded (~ 12) states. (**B**) Free energy vs l for this data, colored by the committor probability in each bin, using 150 bins for the range -8 to 20. This result does not change when discretizing into 50 or more bins.

long unbiased trajectory, colored by the value of $P_c(l)$. The FES shows a stable well at a value of $l = -3$ corresponding to the highest population state, the folded one, and very shallow minima for each of the other states. The value of P_c varies continuously from 1 to 0 along this coordinate, reaching a value of 0.5 at $l = 1$, just outside the folded basin. By this metric, our very simple coordinate is a good CV for characterizing the transition between folded and unfolded states, although the lack of a high barrier separating the two states (due to the system being near its melting temperature) makes it more ambiguous how close the point of $P_c = 0.5$ is to a classic transition state. The coincidence of $P_c = 0.5$ with a clear barrier is observed in Figure 2.6 where

we train using all 6 states, but for this paper we chose to focus only on one dimensional LDA spaces. In Figure 2.7 we show the FES projected between the folded states and all other states, with each possible choice of alignment.

2.3.2 LDA is a Reasonable Sampling Coordinate for HP35 Folding

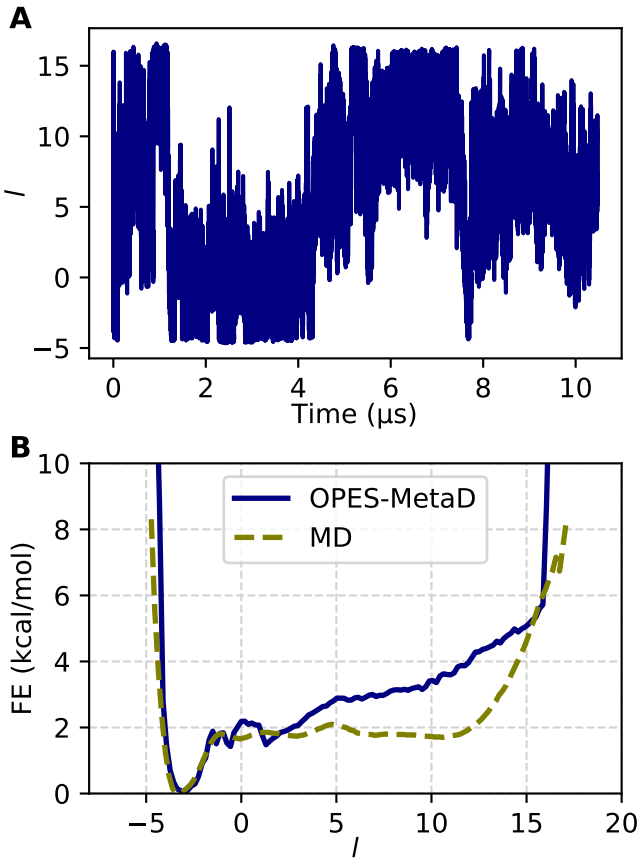

Figure 2.3: OPES-MetaD sampling on HP35 using the folding/unfolding LDA coordinate. (**A**) LDA coordinate vs. times for OPES-MetaD simulation. (**B**) Comparison of free energy estimated from unbiased MD and OPES-MetaD.

To assess the ability to sample along an LDA coordinate we perform OPES-MetaD to bias the system to explore l. For the MetaD parameters listed in Section 2.5, we find that transitions between the folded and unfolded state are accelerated. For these parameters, we are able to obtain several transitions in 10 μs, resulting in a estimated FES that is in fair agreement with

that obtained from the long unbiased trajectory considering it is obtained in only 3% of the MD time. This undersampling of the large unfolded region using only a single coordinate is likely a reflection of the usual problem of sampling slow orthogonal degrees of freedom. Despite this, when we look at the FES projected on natural folding coordinates in Figure 2.8, we see that our sampling does a good job capturing the main features of the long unbiased trajectory, including the presence of intermediates along the x- and y- axes, and the high energy unfolded state located in the upper right. As inferred from the 1d FES, the most unfolded regions are unexplored and the statistical weight of the central intermediate basin is incorrect. Shorter replicates of simulations starting from different initial structures (Figure 2.9) show the variance in FES estimates that could arise if one is not careful to converge sampling. On the whole, our results are evidience that our simple LDA coordinate is a promising first step for sampling between two states of a complex biomolecule.

2.3.3 Accurate Sampling Using LDA for a Bistable Helix

The LDA procedure can be applied to determine a reaction coordinate separating two states even without sampling the actual transition (analogous to Ref. [52]). To assess this behavior we investigate the right to left-handed helix transition of $(Aib)_9$, a nine residue peptide formed from the achiral α-aminoisobutyryl amino acid [92]. The helical states of achiral molecules must by symmetry have equal free energy, and we previously took advantage of this property in benchmarking sampling and clustering methods [75, 93]. The properties of $(Aib)_9$ have been characterized in simulation including recently as a tool to benchmark advanced methods for RC optimization [74, 94, 95].

We performed 20 ns simulations starting from the left and right-handed states of $(Aib)_9$ using inputs from Ref. [74] (see Section 2.5 for details). We did a three state clustering of the combined MD data (total 40 ns, sampled every ps) and verified that the two most populated clusters are the left and right hand states. The coordinates of backbone atoms only were used for the clus-

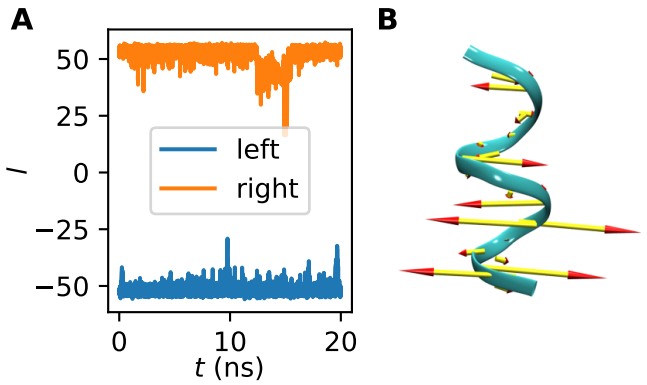

Figure 2.4: LDA coordinate for helical inversion of $(Aib)_9$. (**A**) LDA coordinate l vs. time for training data starting from left and right-handed helix. (**B**) Porcupine plot showing the magnitude of the LDA coefficients on the left-handed helical structure.

tering procedure. We then performed an iterative alignment of the combined data to compute a global $(\boldsymbol{\mu}, \Sigma)$, and then computed a single LDA vector between those frames coming from the left- and right- states, respectively from the globally aligned trajectory. Figure 2.4 shows that this coordinate separates the training data with $l \sim 50$ indicating a right-handed helix and $l \sim -50$ indicating a left-handed helix. The left-handed helix is the starting point for further runs.

Having trained l, we next performed conventional and WT-MetaD simulations starting from the structure in Figure 2.4A. Figure 2.5A shows that MetaD (right) substantially increases the rate of transition between the left and right-handed states as compared to conventional MD (left).

A more chemically motivated way of computing the helicity of $(Aib)_9$ is the parameter $\zeta' = -\sum_{n=3}^{7}\phi_n$, the negative sum over the central backbone ϕ dihedral angles [74]. This quantity takes on values of approximately 5 for right-handed and -5 for left-handed helices [74]. Figure 2.5B shows qualitatively similar behavior for ζ' as l.

Figure 2.5C shows the FES computed for these two quantities. The sampled l has a nearly perfectly symmetrical FES, and in particular the free energy difference between the left and right-handed states is negligible. For the FES of the non-biased ζ' computed by reweighting, the result is nearly as symmetrical, and the offset in free energy between the left and right-handed size

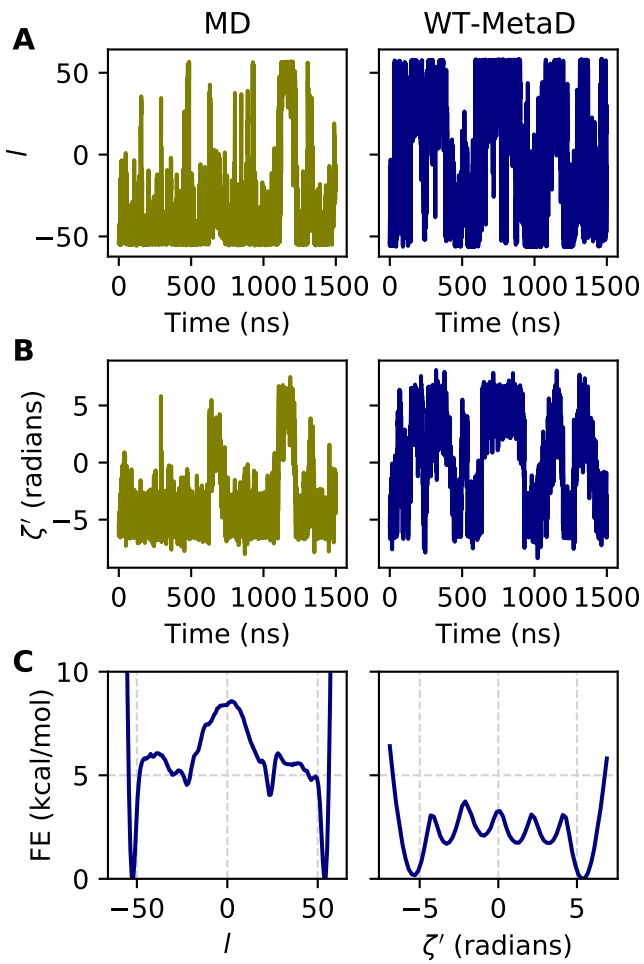

Figure 2.5: Metadynamics sampling results along the LDA coordinate for $(Aib)_9$. (**A**) LDA coordinate l vs. time for 1.5 μs of conventional MD and WT-MetaD. (**B**) Helical parameter ζ vs. time for the trajectories in A. (**C**) FES along l and ζ from WT-MetaD simulations.

is visible but minuscule. This result appears to be as good as that obtained in Ref. [74], which uses a very sophisticated iterative process and 900 ns of unbiased and biased simulation data to obtain an optimized sampling coordinate as compared to our 40 ns of input data; however, their optimized coordinate appears to perform better in terms of transitions per unit time generated with their choice of MetaD parameters. As detailed in Section 2.5, the parameters used in our WT-MetaD simulation are very gentile; their magnitude was limited by "crashing," which typically occurs due to inaccurate numerical integration. To check this, we demonstrate in Figure 2.10

that use of a 1fs integration timestep allows us to use much more aggressive MetaD parameters, which results in much more frequent transitions, as well as accelerated convergence enough to justify the use of a smaller timestep Figure 2.11. It is possible that implementation of analytical derivatives for our procedure may further mitigate this issue if they can be properly derived, and we will pursue this going forward.

2.4 Conclusions and Outlook

In this work, we demonstrated that LDA on positions computed from two states of a system produces a good reaction coordinate, both in terms of state transition kinetics and our ability to bias that coordinate to assess the FES along that coordinate. This was true for $(Aib)_9$ even though the RC was trained only using short simulations starting in each state, making this a promising approach even when only structures of endpoints of a process are available. In contrast to Ref. [52] where input features were internal coordinates, we were able to use standard LDA rather than HLDA in this case and achieve good performance.

We note that LDA on positions would not apply directly to problems such as molecular dissociation since the dissociated states cannot be aligned to a single average structure; however, we do think this coordinate would work well for apo-holo transitions of a biomolecule and could easily be combined with a ligand-distance coordinate to overcome sampling challenges e.g. as observed in Ref. [96]. There are, of course, difficulties in resolving structural states of globular proteins that could make application of shapeGMM and subsequent LDA challenging. Namely, structural states of globular proteins can differ in only a small fraction of the total degrees of freedom. We feel that the heterogeneous nature of allowed covariance in the Kronecker form of shapeGMM will allow us to resolve these states with adequate sampling. Once the clusters are resolved the LDA procedure described in the current manuscript will highlight the coordinates relevant to separate the clusters.

For HP35, multidimensional LDA by construction better separates all of the states of the molecule and may also provide an even better reaction coordinate for kinetics (Figure 2.6). It is not yet clear if this result is general or specific to the HP35 system. Regardless, the use of multidimensional LDA as a RC is intriguing and we are currently investigating the advantages and limitations of these coordinates. However, this is not an option when information about multiple states is unavailable a priori (such as in the case of $(Aib)_9$) which is why we did not include it here. For cases like that, it would be intriguing to first sample along the 1-dimensional reaction coordinate, then train a GMM with a higher number of states, and continue iterating this approach.

The use of states defined from our GMM clustering approach presents both an advantage and disadvantage as illustrated in the case of HP35. Our approach allowed us to explore the folding/unfolding process and most of the conformational landscape (Figure 2.8), but we were not able to fully sample the FES around the unfolded state. For sampling a broad and entropy dominated state, combining CV based sampling on position LDA coordinates with tempering or temperature accelerated methods should provide more accurate information in this region as in many past studies [97–101].

In both the case of HP35 and $(Aib)_9$, we were able to accelerate transitions between two states using MetaD or OPES-MetaD. In our hands, the biased simulations were sensitive to sampling protocol in terms of being able to run microseconds or longer without "crashing." HP35 was less sensitive to this issue using OPES-MetaD, while $(Aib)_9$ performed better with standard WT-MetaD. For this reason, we initially used small bias factors and hill heights/barrier heights, which resulted in fewer transitions and presumably worse convergence in fixed simulation time. We speculated that some of this sensitivity may come from our choice of the global trajectory mean and covariance as the reference state when computing our LDA vectors, however subsequent tests using alignment to left or right-handed helices for $(Aib)_9$ showed that these alignments were more sensitive to crashing and had worse convergence performance, supporting our initial

choice of global alignment (Figure 2.12, Figure 2.13). A compelling option is presented in the ATLAS method of Ref. [78], where bias is computed along vectors to multiple reference states, weighted by distance from that reference state, and we are beginning to assess that approach.

2.5 Simulation Details

All simulations were performed using GROMACS 2019.6 [102] with PLUMED 2.9.0-dev [81, 82]. GROMACS 'mdp' parameter files and PLUMED input files are available in our paper's github repository for complete details.

HP35 Simulations

A 305 μs all-atom simulation of Nle/Nle HP35 at $T = 360K$ from Piana et al.[89] was analyzed. The simulation was performed using the Amber ff99SB*-ILDN force field and TIP3P water model. In that simulation, protein configurations were saved every 200 ps, for a total of ~1.5M frames. For our simulations, we solvate and equilibrate a fresh system using the same forcefield at 40mM NaCl. Minimization and equilibration are performed using a standard protocol[1], at which point NPT simulations are initiated at $T = 360K$. mdp files for all steps of this procedure and the topology files are all available in the paper's github.

OPES-MetaD simulations are performed with $\gamma = 8$, $\Delta E = 10$ kcal/mol, pace of 500 steps, and a multiple time step [103] stride of 2. Quadratic walls are applied at $l = 5$ and $l = -15$ with a bias coefficient of 125 kcal/mol/Å^2.

$(Aib)_9$ Simulations

Equilibrated inputs for $(Aib)_9$ were provided by the authors of Ref. [74]. In brief, simulations using the CHARMM36m forcefield and TIP3P water [104]. MD simulations are performed in NPT

[1] http://www.mdtutorials.com/gmx/lysozyme/index.html

with a 2 fs timestep at $T = 400K$.

WT-MetaD simulations are performed with $h = 0.005$ kcal/mol, $\sigma = 0.43$, $\gamma = 2$ and a multiple time step [103] stride of 2. Quadratic walls are applied at $l = 70$ and $l = -60$ with a bias coefficient of 125 kcal/mol/Å^2. σ was chosen as the $\sigma_l/3$ where σ_l was the standard deviation in l over the 20 ns simulation starting from the left helical state used in the training of the CV.

2.6 Supplementary Figures

Two state vs. six state LDA

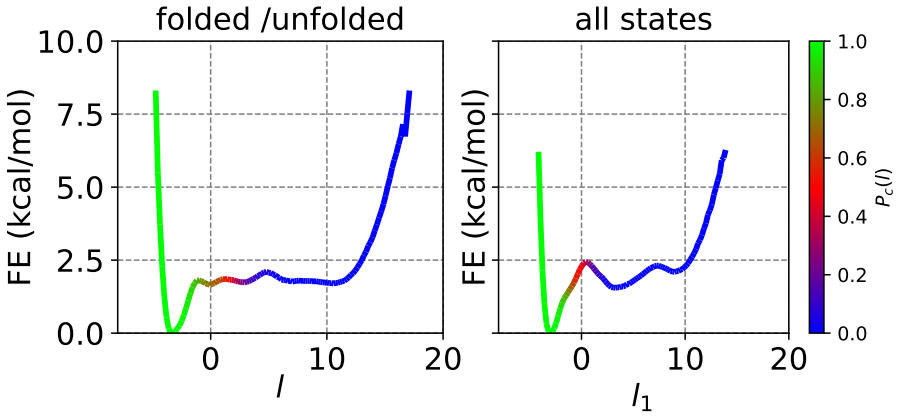

Figure 2.6: Unbiased estimate of free energy along 2-state and 6-state LD1 coordinates. The left shows the FES computed along l, the LDA coordinate from states 0 and 4 in our GMM model, and the right shows the FES computed along l_1, the first LDA coordinate from a model trained on all 6 states.

Comparison of different state pairs and alignment choices

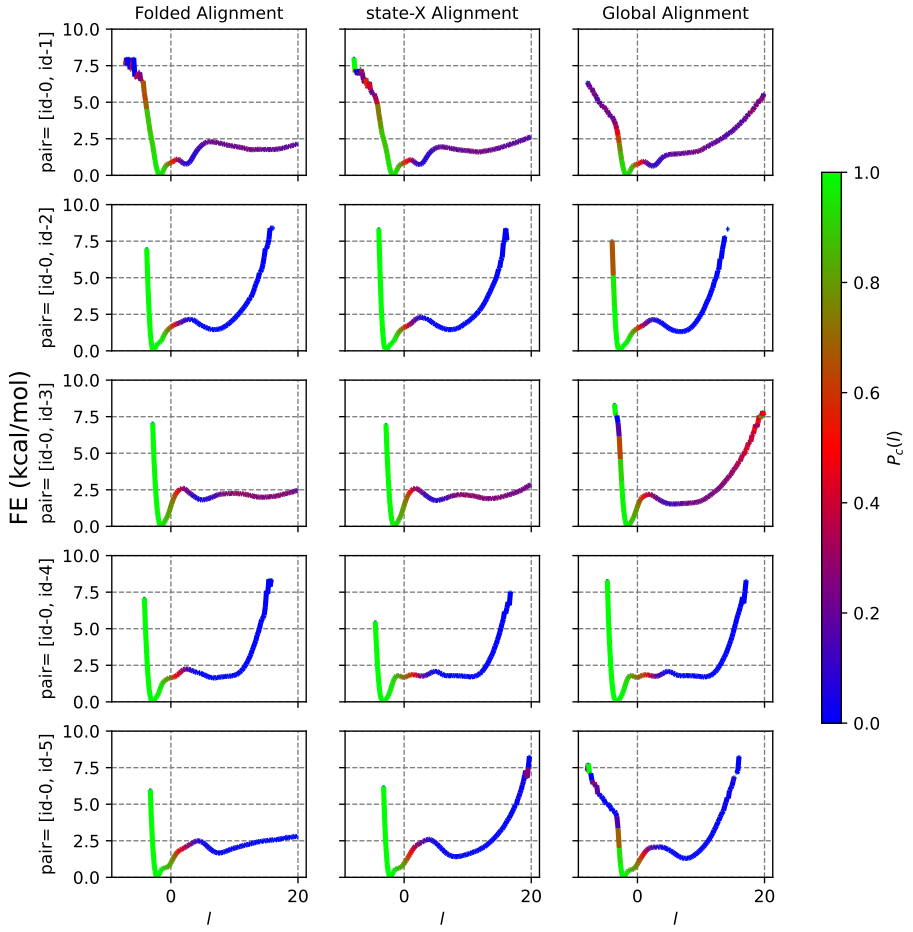

Figure 2.7: FE profiles along LD1 obtained from different state pairs and alignments. Each row represents total three FE profiles for a particular cluster pair using: (1) alignment to folded cluster, (3) alignment to cluster X, and (3) global alignment for that cluster pair respectively from left to right.

Comparing FES from unbiased and biased MD

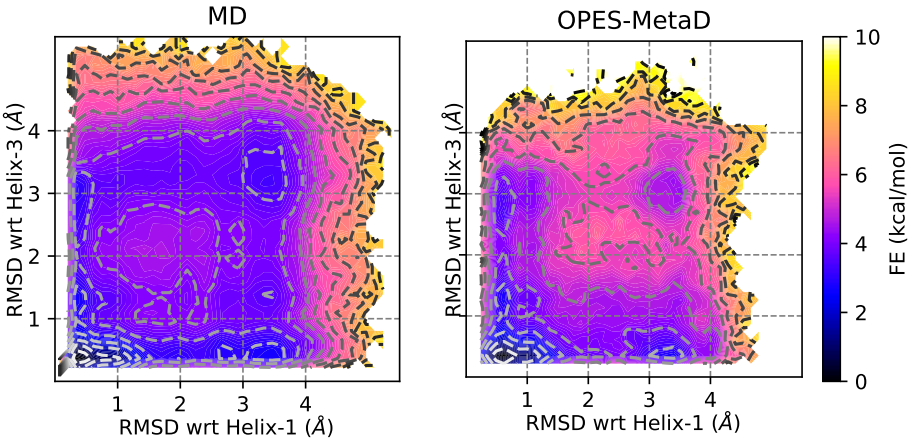

Figure 2.8: Comparison of FES projected along RMSD coordinates. FES from unbiased MD simulation and OPES-MetaD reweighted along coordinates measuring RMSD of the terminal two helices to a reference folded structure.

Comparison of independent runs with less sampling

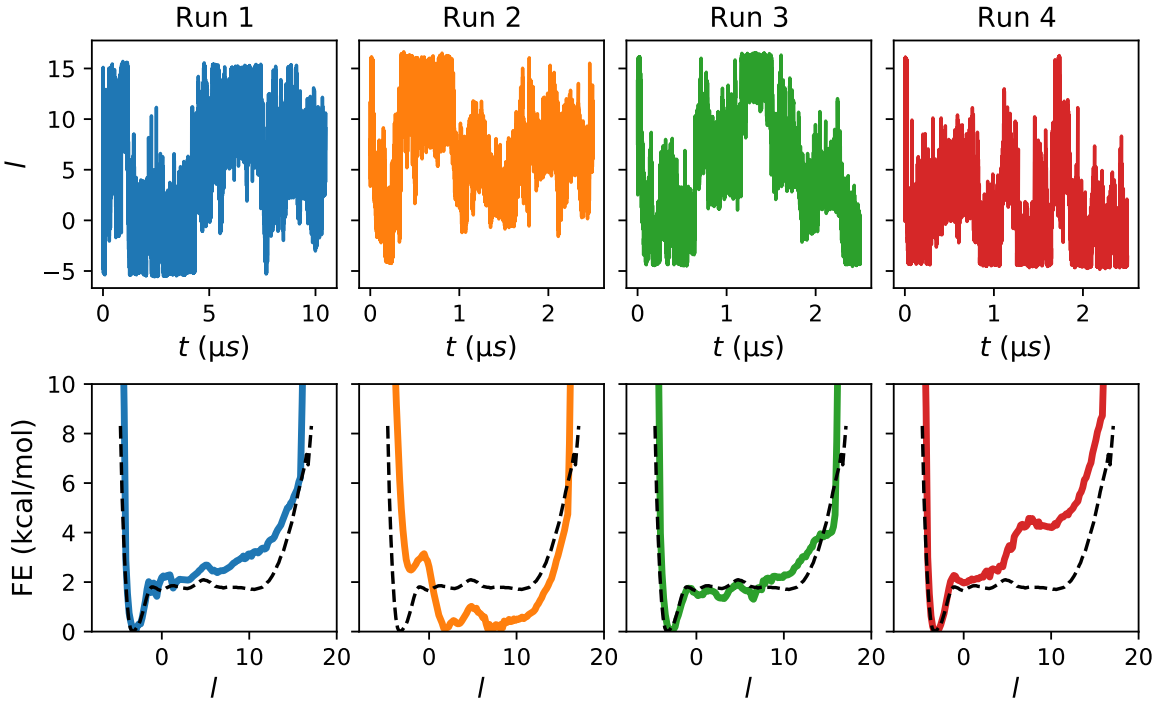

Figure 2.9: Comparing the convergence for independent runs. (top) OPES-MetaD replicate trajectories for HP35. Run 1 is the 10μs simulation studied in the main text. The other three runs are 2.5μs runs starting from other points obtained in the original trajectory, with separate equilibrium performed. Each of these three simulations has one to two transitions. (bottom) Comparison of FES obtained from OPES-MetaD for HP35, with a dashed line showing the FES obtained from unbiased MD. Run 3 producing a perfect FES by chance.

Comparing 1fs to 2fs for $(Aib)_9$

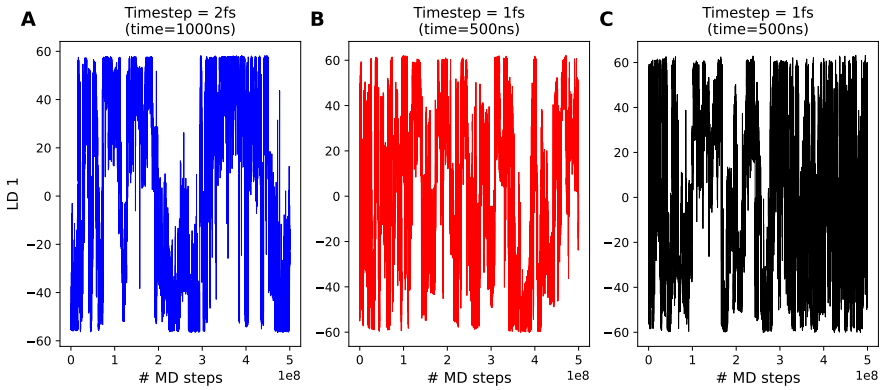

Figure 2.10: LD1 vs. time from three different simulations of $(Aib)_9$. All three coordinates are obtained from global alignment of input data. **A** was performed with h = 0.005 kcal/mol, σ = 0.43, γ = 2 and a multiple time step stride of 2. **B** was performed with h = 0.50 kcal/mol, σ = 0.50, γ = 8 and a multiple time step stride of 2. **C** was performed with h = 1.0 kcal/mol, σ = 1.20, γ = 8 and a multiple time step stride of 2.

Comparing sampling efficiency of different alignments in $(Aib)_9$

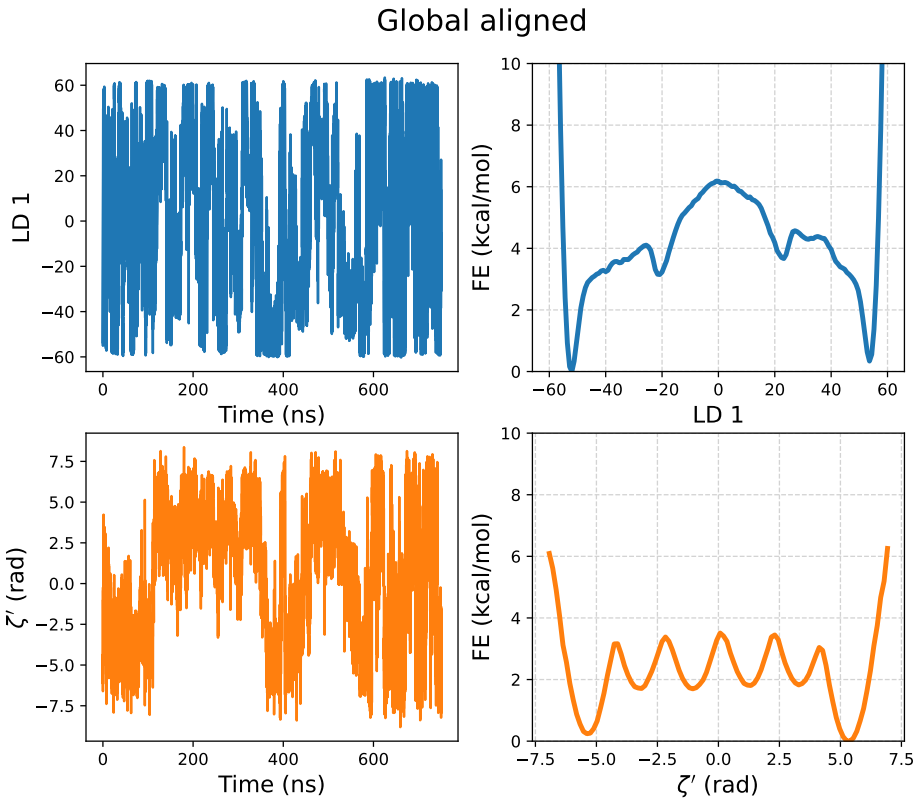

Figure 2.11: Global Alignment results. WT-MetaD simulation using 1fs time step, initiated from left handed state and the global alignment was used. Simulation was performed with h = 0.50 kcal/mol, σ = 0.50, γ = 8 and a multiple time step stride of 2. Top row shows fluctuation of LD1 with time and FE along it. Bottom row shows the same for ζ' coordinate.

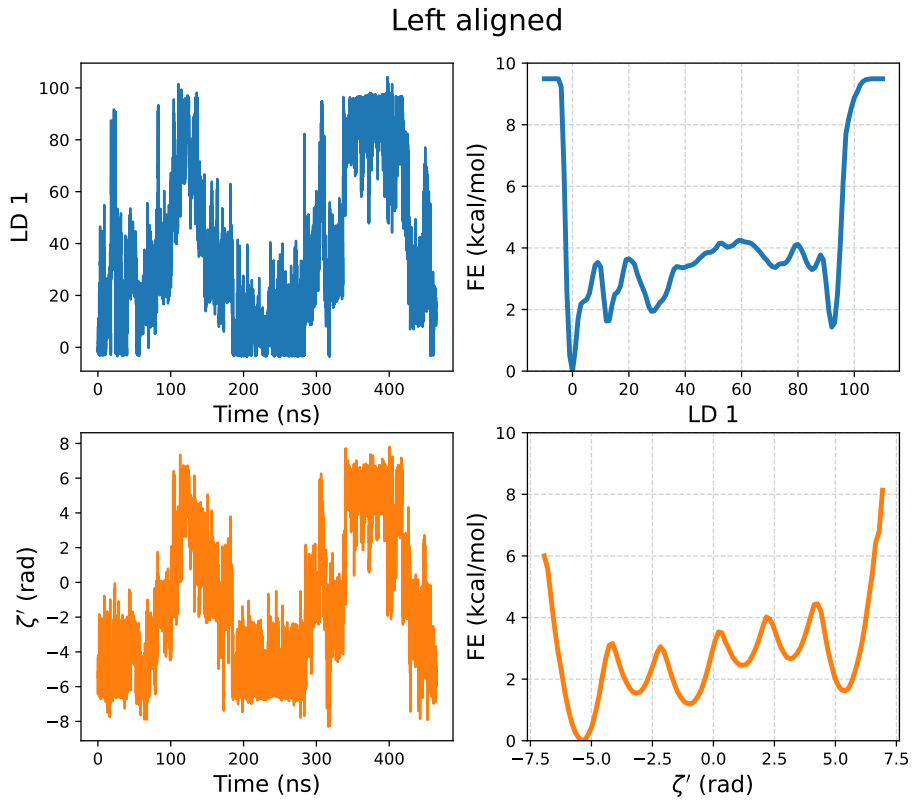

Figure 2.12: Left Alignment results. WT-MetaD simulation using 1fs time step, both initiated and aligned to the left handed helix. Simulation was performed with h = 0.005 kcal/mol, σ = 0.43, γ = 2 and a multiple time step stride of 2. Top row shows fluctuation of LD1 with time and FE along it. Bottom row shows the same for ζ' coordinate. Note that the initial value of LD1 is close to zero according to our definition of LDA coordinate.

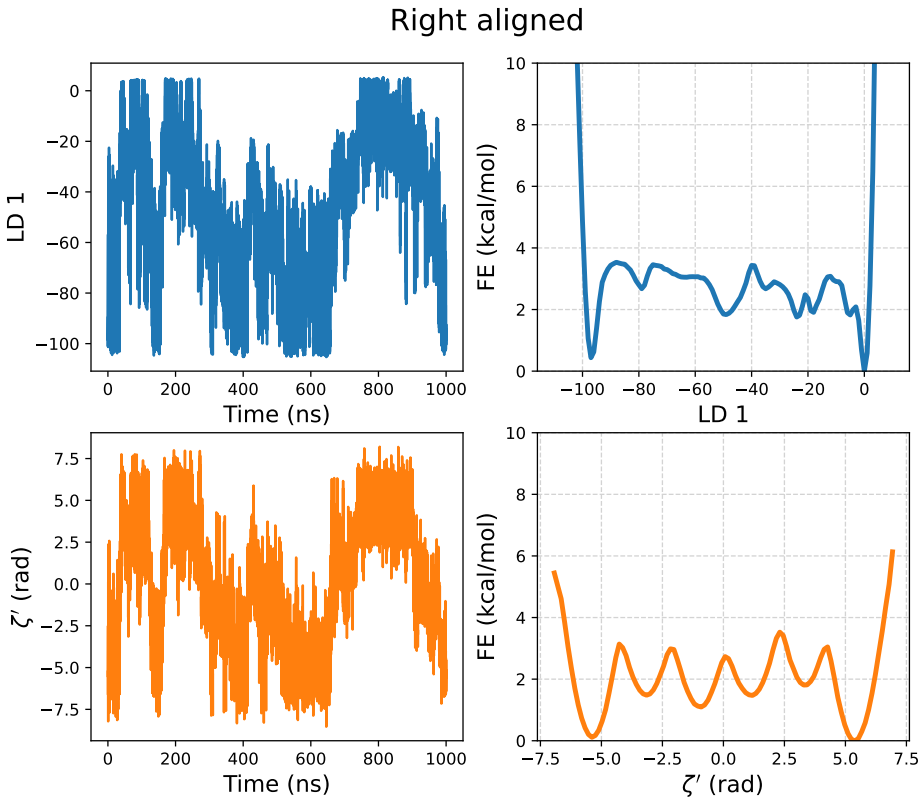

Figure 2.13: Right Alignment Results. WT-MetaD simulation using 1fs time step, initiated from left handed state and the system was aligned to the right handed helix. Simulation was performed with h = 0.10 kcal/mol, σ = 1.0, γ = 4 and a multiple time step stride of 2. Top row shows fluctuation of LD1 with time and FE along it. Bottom row shows the same for ζ' coordinate. Note that the initial value of LD1 is far away from zero unlike the case of left alignment.

3 Quantifying Unbiased Conformational Ensembles from Biased Simulations Using ShapeGMM

This chapter has been adapted from Ref. [105]

3.1 Introduction

Conformational ensembles of molecules dictate many of their thermodynamic properties. Conventional molecular dynamics (MD) simulations allow us to sample models of these ensembles but suffer from the so-called rare event problem. A variety of enhanced sampling techniques, such as Metadynamics (MetaD) [20, 24], Adaptive Biasing Force [106], Gaussian accelerated MD [107], and Temperature Accelerated MD/Driven Adiabatic Free Energy Dynamics [26, 27], have been developed to promote faster sampling by effectively heating some degrees of freedom [55]. Unfortunately, due to the biased sampling of many of these approaches, it is not obvious how to use the biased configurations in methods such as Markov State Models (MSMs) [37, 108] and/or structural clustering approaches that quantify the conformational ensemble. Here, we adapt shapeGMM [75], a probabilistic structural clustering method, to rigorously quantify the unbiased conformational ensembles generated from biased simulations. The result is a high dimensional Gaussian mixture model (GMM) characterizing the unbiased landscape that can be

used to extract important thermodynamic quantities and to give additional insight beyond the low dimensional projections often used to represent free energy landscapes.

Meaningful quantification of conformational ensembles from large molecular simulations requires the grouping of similar frames using a clustering algorithm. Clustering algorithms for molecular simulation can be grouped into two categories: temporal and structural. Temporal clustering, such as spectral clustering of the transition matrix [38, 39], has been successfully applied to MD trajectories to achieve kinetically stable clusters for use in objects like MSMs [109–111]. Enhanced sampling techniques, however, can distort the underlying kinetics of the system making temporal clustering difficult to apply properly in these circumstances. While there have been efforts to build MSMs from enhanced sampling data [112, 113] it still remains a challenge [114]. Additionally, building MSMs relies on an initial structural clustering step, making it critical to perform this step accurately even in the context of enhanced sampling. Structural clustering involves partitioning either frames or feature space into a finite number of elements. This can be achieved from enhanced sampling data but care must be taken to properly account for the non-uniform weights of the frames.

Previous efforts to use structural clustering algorithms on enhanced sampling simulations have focused on partitional, as opposed to model-based, algorithms. The main results of partitional algorithms are cluster populations that can be reweighted based on enhanced sampling frame weights to estimate the unbiased populations[112, 115]. Model-based clustering algorithms offer many advantages over partitional algorithms the most relevant being that the resulting probability density can be used predict clusterings on new data and estimate Thermodynamic properties of the underlying ensemble. Reweighting the cluster populations of model-based algorithms a posteriori is, however, not satisfactory for methods such as GMMs, as the frame weights will affect determination of additional model parameters. It is possible to use multiple copies of frames to approximately account for the frame weights but this can yield intractably large trajectories and inaccuracies due to rounding.

In this work, we present an adaptation to shapeGMM [75], a probabilistic structural clustering method on particle positions, to directly account for non-uniform frame weights. As opposed to introducing copies of input data and maintaining uniform weights, the current method directly accounts for non-uniform frame weights and is thus more efficient and scalable than the alternative. In the next section we briefly introduce the shapeGMM method and the adaptations necessary to account for non-uniform frame weights. This is followed by a demonstration of the method on three examples of increasing difficulty, specifically demonstrating that our intuitive choices of frame weights from Metadynamics simulations result in a reliable clustering procedure. We show in benchmark cases how this method can yield thermodynamic quantities directly, and use the complex case of actin flattening to show how a weighted shapeGMM can give physical insight into the conformations sampled, in a case where unbiased simulation would not be a practical option. In addition, frame-weighted shapeGMM is implemented in an easy-to-use python package (`pip install shapeGMMTorch`).

3.2 Theory and Methods

3.2.1 Overview of shapeGMM

In shapeGMM, a particular configuration of a macromolecule is represented by a particle position matrix, $\boldsymbol{x}_i$, of order $N \times 3$, where N is the number of particles being considered for clustering. To account for translational and rotational invariance, the proper feature for clustering purposes is an equivalence class,

$$[\boldsymbol{x}_i] = \{\boldsymbol{x}_i \boldsymbol{R}_i + \mathbf{1}_N \vec{\xi}_i^T : \vec{\xi}_i \in \mathbb{R}^3, \boldsymbol{R}_i \in \mathrm{SO}(3)\}, \tag{3.1}$$

where $\vec{\xi}_i$ is a translation in $\mathbb{R}^3$, $\boldsymbol{R}_i$ is a rotation $\mathbb{R}^3 \to \mathbb{R}^3$, and $\mathbf{1}_N$ is the $N \times 1$ vector of ones. $[\boldsymbol{x}_i]$ is thus the set of all rigid body transformations, or orbit, of $\boldsymbol{x}_i$.

The shapeGMM probability density is a Gaussian mixture given by

$$P(\boldsymbol{x}_i) = \sum_{j=1}^{K} \phi_j N(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \Sigma_j), \tag{3.2}$$

where the sum is over the K Gaussian mixture components, ϕ_j is the weight of component j, and $N(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \Sigma_j)$ is a normalized multivariate Gaussian given by

$$N(\boldsymbol{x}_i \mid \boldsymbol{\mu}, \Sigma) = \frac{\exp\left[-\frac{1}{2}(g_i^{-1}\boldsymbol{x}_i - \boldsymbol{\mu})^T \Sigma^{-1} (g_i^{-1}\boldsymbol{x}_i - \boldsymbol{\mu})\right]}{\sqrt{(2\pi)^{(3N)} \det \Sigma}}, \tag{3.3}$$

where $\boldsymbol{\mu}$ is the mean structure, Σ is the covariance, and $g_i^{-1}\boldsymbol{x}_i$ is the element of the equivalence class, $[\boldsymbol{x}_i]$, that minimizes the squared Mahalanbonis distance in the argument of the exponent. Determining the proper transformation, g_i, is achieved by translating all frames to the origin and then determining an optimal rotation matrix. Cartesian and quaternion-based algorithms for determining optimal rotation matrices are known for two forms of the covariance were considered $\Sigma \propto \boldsymbol{I}_{3N}$ [116, 117] or $\Sigma = \Sigma_N \otimes \boldsymbol{I}_3$ [118, 119], where Σ_N is the $N \times N$ covariance matrix and $\otimes$ denotes a Kronecker product. In this manuscript, we employ only the more general Kronecker product covariance.

3.2.2 Incorporating Non-uniform Frame Weights in shapeGMM

Previously, each frame in shapeGMM was considered to be equally weighted. Approximate weighting of frames could be taken into account by including frames multiple times in the training data to give them more importance, however this introduces the imprecision of rounding to the nearest integer and can be extremely computationally expensive due to the large increase in amount of training data. Here, we take non-uniform frame weights into account by performing weighted averages in the Expectation Maximization estimate of model parameters $\{\hat{\phi}_j, \hat{\mu}_j, \hat{\Sigma}_j\}$, consistent with other fixed-weight GMM procedures [120]. Considering a normalized set of frame

weights, $\{w_i\}$ where $\sum_{i=1}^{M} w_i = 1$ for M frames, their contribution to the probability can be accounted for by weighting the estimate of the posterior distribution of latent variables:

$$\gamma_{Z_i}(j) = w_i \frac{\hat{\phi}_j N(\boldsymbol{x_i} \mid \hat{\boldsymbol{\mu}}_j, \hat{\Sigma}_j)}{\sum_{j=1}^{K} \hat{\phi}_j N(\boldsymbol{x_i} \mid \hat{\boldsymbol{\mu}}_j, \hat{\Sigma}_j)}. \tag{3.4}$$

The frame weight will propagate to the estimate of component weights, means, and covariances in the Maximization step through $\gamma_{Z_i}(j)$:

$$\hat{\phi}_j = \sum_{i=1}^{M} \gamma_{Z_i}(j) \tag{3.5}$$

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_{i=1}^{M} \gamma_{Z_i}(j) g_{i,j}^{-1} \boldsymbol{x}_i}{\sum_{i=1}^{M} \gamma_{Z_i}(j)} \tag{3.6}$$

$$\hat{\Sigma}_j = \frac{\sum_{i=1}^{M} \gamma_{Z_i}(j) \langle \hat{\Sigma}_N \rangle_i}{\sum_{i=1}^{M} \gamma_{Z_i}(j)} \otimes \boldsymbol{I}_3 \tag{3.7}$$

Additionally, the log likelihood per frame is computed as a weighted average

$$\ln(L) = \sum_{i=1}^{M} w_i \ln \left(\sum_{j=1}^{K} \hat{\phi}_j N(\boldsymbol{x}_i \mid \hat{\boldsymbol{\mu}}_j, \hat{\Sigma}_j) \right). \tag{3.8}$$

3.2.3 Choosing Number of Clusters

Performing shapeGMM requires the user to choose a number of clusters, K. The "optimal" choice will be system and problem specific and potentially has no wrong answer. The choice is no different if you consider uniformly or non-uniformly weighted frames. We use a cluster scan with a combination of the elbow method and cross validation to assess if our choice of K is reasonable. A good choice of clusters based on this approach is to find the number of clusters where the increase in log-likelihood with K is decreasing fastest, which we can evaluate by choosing the minimum of the second derivative of $\ln(L)$ with respect to number of clusters. In practice, this works well for simple systems, but it may be hard to pick a "best" choice for more complex

systems, so we may seek a choice that is physically interpretable.

3.2.4 Assigning Frames to Clusters

After the model parameters have been fit using fuzzy assignments, individual frames are assigned to the cluster in which that frame has the largest likelihood (largest $\gamma_{Z_i}(j)$). This is the standard procedure for clustering from a GMM and is no different for the *frame-weighted* version.

3.2.5 Implementation

We have completely rewritten shapeGMM in PyTorch for computational efficiency and ability to use GPUs. The current implementation takes an array of frame weights as an optional argument to both the fit and predict functions (the code defaults to uniform weights). The PyTorch implementation is significantly faster than the original version and is available both on github (https://github.com/mccullaghlab/shapeGMMTorch) and PyPI (`pip install shapeGMMTorch`). Examples are also provided on that github, and all examples from this paper are provided in a second github page discussed below.

3.2.6 Choosing Training Sets

For non-uniformly weighted frames the choice of training set may be important. We have attempted a variety of training set sampling schemes and have found that, at least for the frame weight distributions that we have encountered, uniformly sampling the training data is at least as good as any importance sampling scheme. We discuss this further and show results for three different training set selection schemes for the beaded helix system in Section 3.6.

3.2.7 Biasing and weighting frames

If configuration $\boldsymbol{x}$ is generated from an MD simulation at constant T and V then $P(\boldsymbol{x}) \propto \exp(-H(\boldsymbol{x})/k_BT)$ where H is the system's Hamiltonian [121]. If $\boldsymbol{x}$ is generated from an MD simulation at a different state point (e.g. different T) or with a different Hamiltonian, it is sampled from a different distribution $Q(\boldsymbol{x})$. Samples from Q can be reweighted to P with weights [121]

$$w(\boldsymbol{x}) \propto \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})}, \tag{3.9}$$

from which averages over P can be estimated. This approach is only effective if Q and P are finite over the same domain. Nonetheless, (3.9) underlies many enhanced sampling approaches, for example, it is the basis of the original formulation of umbrella sampling [23]. By including weights in shapeGMM, we can predict the importance of clusters at nearby state-points or for similar systems.

3.2.8 Thermodynamic Quantities from ShapeGMM

Many Thermodynamic quantities can be computed from fit shapeGMM probability densities. One such quantity is the configurational entropy,

$$S_{\text{config}} = -\int P(\boldsymbol{x}) \ln P(\boldsymbol{x}) d\boldsymbol{x} = -\left\langle \ln P(\boldsymbol{x}) \right\rangle_P . \tag{3.10}$$

The configurational entropy has an analytic solution for a single multivariate Gaussian but for the general mixture of multivariate Gaussians we use sampling and Monte Carlo integration to approximate the integral.

To do so accurately requires that we generate points from the shapeGMM objects and not just use the trajectory on which the object was fit. We have introduced a `generate` function as an attribute to a fit shapeGMM object that produces configurations sampled from the underlying

trained distribution.

The second Thermodynamic quantity we consider is the free energy cost to move from one distribution to another. This is also known as the relative entropy or Kullback-Leibler divergence and the cost to go from distribution Q to distribution P is given by

$$D_{KL}(P||Q) = \int P(\boldsymbol{x}) \ln \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})} d\boldsymbol{x} = \left\langle \ln \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})} \right\rangle_P . \tag{3.11}$$

Here, again, we generate points from distribution P and average the difference in log likelihoods of these points in P and Q to assess this value. It should be noted that this is a non-equilibrium free energy and is thus not necessarily symmetric [122, 123]. The quantity can prove useful in applications, for example measuring the free energy cost to shift a distribution from an apo to a ligand-bound state, for example [124, 125].

A symmetric metric is useful when comparing the similarity of two distributions. Here we opt for the Jensen-Shannon divergence (JSD) [126] given by

$$JSD(P||Q) = \frac{1}{2 \ln 2} \left(D_{KL}(P||M) + D_{KL}(Q||M) \right), \tag{3.12}$$

where $M = \frac{1}{2}(P+Q)$ is the midpoint distribution between P and Q. JSD is restricted to be between 0 and 1.

All three of these measures have been implemented in the `similarities` library of the shapeGMM code. They use point generation and Monte Carlo sampling to assess the integrals and thus return both the mean value and the standard error.

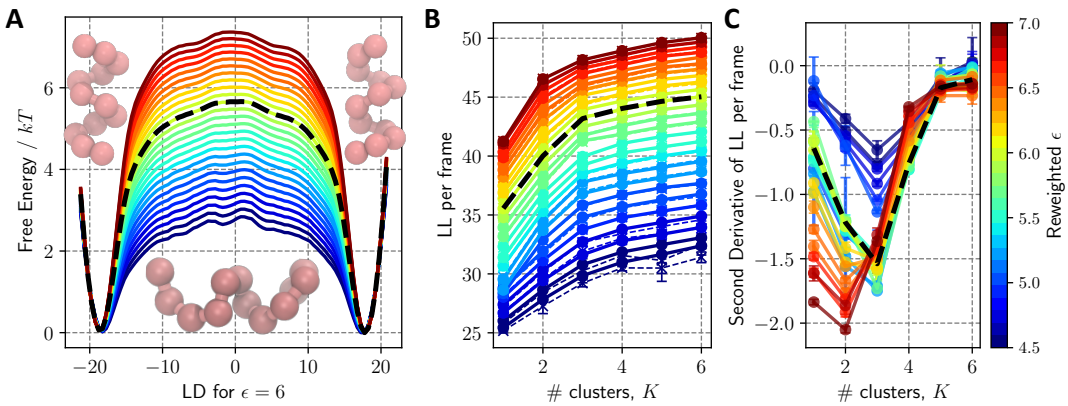

Figure 3.1: Beaded helix ϵ reweighting. Trajectory data for a 12 bead polymer having $i, i+4$ interactions with strength $\epsilon = 6$ was reweighted to predict the ensemble for ϵ values ranging from 4.5 to 7 in increments of 0.1. (**A**) The corresponding free energies as a function of the linear discriminant (LD) between the two helices are plotted with ϵ values denoted by the color bar on the right-hand. The weights per frame were fed in to shapeGMM to perform a cluster scan. (B) The resulting log likelihood per frame as a function of number of clusters from the cluster scan. (C) Second derivative of the curves from B. Error bars in (**B**) and (**C**) are estimated as the standard deviation from three different training sets. The true curve for $\epsilon = 6$ is given in black dashed lines in all three panels.

3.3 Results and Discussion

3.3.1 Proof of Concept: Reweighting the Beaded Helix

To demonstrate the accuracy of the frame-weighted shapeGMM process we perform Hamiltonian reweighting of a non-harmonic beaded helix previously studied in Refs. [75, 93]. The system is composed of 12 beads connected in sequential fashion by stiff harmonic bonds. Every fifth pairwise interaction is given by an attractive Lennard-Jones potential with well depth ϵ. The value of ϵ relative to kT dictates the stability of an alpha-helix-like structure as compared to a completely disordered state. Additionally, because of the symmetry of the model both the left- and right-handed helices have equal probability no matter the value of ϵ. A value of $\epsilon = 6$ in reduced units forms stable helices while allowing transitions between the two folded states; here we performed a long unbiased trajectory to sample both left and right states, as well as possibly intermediates (see Section 3.5 for details).

ShapeGMM suggest three clusters is a good choice for a simulation of the beaded helix with $\epsilon = 6$. Shown in blacked dash line in Figure 3.1A is the unbiased free energy for this system computed as $F(s) = -\ln P(s)$ for a linear discriminant (LD) reaction coordinate [54]. By performing a scan over number of clusters on 100k frames from an unbiased trajectory, we identify three clusters as the optimal number by observing a definite kink in the curves in Figure 3.1B and the presence of a minimum in the second derivative in Figure 3.1C. These clusters correspond to the left- and right- helical states as well as a partially unfolded intermediate cluster, examples shown in Figure 3.1A.

Reweighted clustering of the beaded helix system predicts that the prevalence of the partially unfolded intermediate will disappear by $\epsilon = 6.5$. To demonstrate this, we performed *frame-weighted* shapeGMM cluster scans of our trajectory at $\epsilon = 6$ with weights corresponding to ϵ values ranging from 4.5 to 7.0 in increments of 0.1. Given that the samples come from a Boltzmann distribution, the weights for each frame given by (3.9) are $w_i(\epsilon) = e^{(U(\boldsymbol{x}_i|\epsilon=6)-U(\boldsymbol{x}_i|\epsilon))/(k_\mathrm{B}T)}$. The log likelihood of the shapeGMM fits as a function of number of clusters are shown in Figure 3.1B,C with ϵ values indicated in the color bar on the right. We see that as ϵ increases from 6, the minimum in the second derivative moves from 3 clusters to 2 cluster. The transition happens between $\epsilon = 6.4$ and $\epsilon = 6.5$. This suggests that a simulation run at ϵ values of greater than 6.4 (in reduced units) will not exhibit the partially unfolded third cluster. These results are consistent with the increasing free energy barrier height as a function of ϵ depicted in Figure 3.1A.

The reweighting of ϵ for the beaded helix example also predicts that only one cluster will be present for small ϵ. In Figure 3.1B, the elbow at 3 clusters is evident for ϵ values as low as $\epsilon = 5$ and becomes less pronounced below this threshold. While a minimum at 3 clusters is still observed in the second derivative plot for $\epsilon = 4.5$, the trend is clear that as ϵ becomes small the choice of anything other than 1 cluster is less well supported by the elbow heuristic. This is an expected result, and consistent with the reduced free-energy barriers observed for small ϵ in Figure 3.1A, as ϵ approach thermal energy the prevalence of anything other than an unfolded

K	Q ϵ_R	$JSD(GT\|\|Q)$	$\Delta S_{\text{config}}/R$
3	6.0	0.401(2)	7.22(3)
3	6.0/8.0†	0.357(2)	4.30(2)
2	8.0	0.0071(3)	0.00(2)

† Only the cluster populations are reweighted to $\epsilon = 8$ in this probability density.

Table 3.1: Similarity measures between three beaded helix probability densities. ShapeGMM fit from a simulation with $\epsilon = 6$, Q, and the "ground-truth" (GT) probability density fit to a simulation at $\epsilon = 8$. The reweighted probability densities are denoted by the number of clusters, K, and the value of ϵ used in reweighting, ϵ_R. The three Qs are: $K = 3$ clusters and weighted to $\epsilon_R = 6.0$, $K = 3$ clusters from $\epsilon_R = 6.0$ with only the cluster populations reweighted to $\epsilon = 8$, and $K = 2$ clusters completely reweighted to $\epsilon_R = 8$. The similarity measures are the Jensen-Shannon divergence (JSD) and the difference in configurational entropy $\Delta S_{\text{config}} = S^Q_{\text{config}} - S^{GT}_{\text{config}}$. Error in the last digit is included in parentheses and are estimated as Monte Carlo sampling errors in estimating the integrals.

state is entropically unfavorable.

ShapeGMM reweighted clustering also produces *quantitatively accurate* probability densities for the the beaded helix. To demonstrate this, we compute a reweighted shapeGMM object ($\epsilon = 6 \longrightarrow 8$) to a shapeGMM object trained on an unbiased trajectory at $\epsilon = 8$, which we refer to as ground truth (GT). Because, as predicted, transitions at $\epsilon = 8$ are very unlikely, this object is trained on simulations, each with 100k frames, initiated from left and right helices and concatenated. Two controls are included that are fit to the $\epsilon = 6$ trajectory without reweighting: the predicted 3 cluster object and that same object with only the cluster populations reweighted to $\epsilon = 8$. To quantitatively compare between two probability densities we use two similarity metrics, both described above in more detail and introduced as (3.10), (3.11): Jensen-Shannon divergence (JSD) and change in configurational entropy S_{config}. These similarity metrics between the GT and the three different shapeGMM objects are tabulated in Table 3.1. JSD is a symmetric metric bounded between 0 and 1 where 0 indicates no divergence and 1 indicates complete divergence between the two probabilities. The reweighted shapeGMM object demonstrates a very small JSD (0.0071 ± 0.0003) to the GT as compared to either of the $\epsilon = 6$ objects (0.357 ± 0.002 and 0.401 ± 0.002). This trend holds true when comparing relative S_{config}'s with the difference in S_{config}

between the reweighted and GT $\epsilon = 8$ shapeGMM probabilities being within error of 0. These results indicate that the $\epsilon = 8$ reweighted shapeGMM probability density is nearly identical to the GT.

3.3.2 Conformational States of Alanine Dipeptide from Metadynamics Simulations

Alanine Dipeptide (ADP) in vacuum is a common benchmark system for methods designed to sample and quantify conformational ensembles. In this work, we demonstrate that ADP MetaD simulations can be used directly to achieve equilibrium clustering using various estimates of the frame weights. In Well-tempered MetaD (WT-MetaD), a history dependent bias is generated by adding Gaussian hills to a grid at the current position in collective variable (CV) space [24, 84] such that the bias at time t for CV value position $\boldsymbol{s}_i$ is given by

$$V(\boldsymbol{s}_i, t) = \sum_{\tau<t} h e^{-V(\boldsymbol{s}_i,\tau)/\Delta T} e^{-\frac{(Q(\boldsymbol{x}(\tau))-\boldsymbol{s}_i)^2}{2\sigma^2}}, \tag{3.13}$$

where h is Gaussian height, and σ is the width, and $T + \Delta T$ is an effective sampling temperature for the CVs. Rather than setting ΔT, one typically chooses the bias factor $\gamma = (T + \Delta T)/T$, which sets the smoothness of the sampled distribution [24, 84]. Asymptotically, a free energy surface (FES) can be estimated from the applied bias by $F(\boldsymbol{s}) = -\frac{\gamma}{\gamma-1} V(\boldsymbol{s}, t \to \infty)$ [84, 85] or using a reweighting scheme [84, 86]. In MetaD, frames are generated from a time dependent Hamiltonian so the choice of frame weights for clustering is not obvious. Reweighting of MetaD trajectories to compute free energy surfaces has been accomplished through several different schemes.

For a static bias V added to the initial Hamiltonian, the weight of a frame given by (3.9) would be $w_i = e^{V(\boldsymbol{s}_i)/k_\mathrm{B}T}$. Our first choice of frame weights (termed 'bias') corresponds to using this formula even though the bias is time-dependent. A second choice that removes some of the time-dependence is to use $w_i = e^{(V(\boldsymbol{s}_i(t_i))-c(t))/k_\mathrm{B}T}$, where $c(t) = -k_\mathrm{B}T \ln \left\langle e^{-V(\boldsymbol{s}(t))/k_\mathrm{B}T} \right\rangle$ is the bias

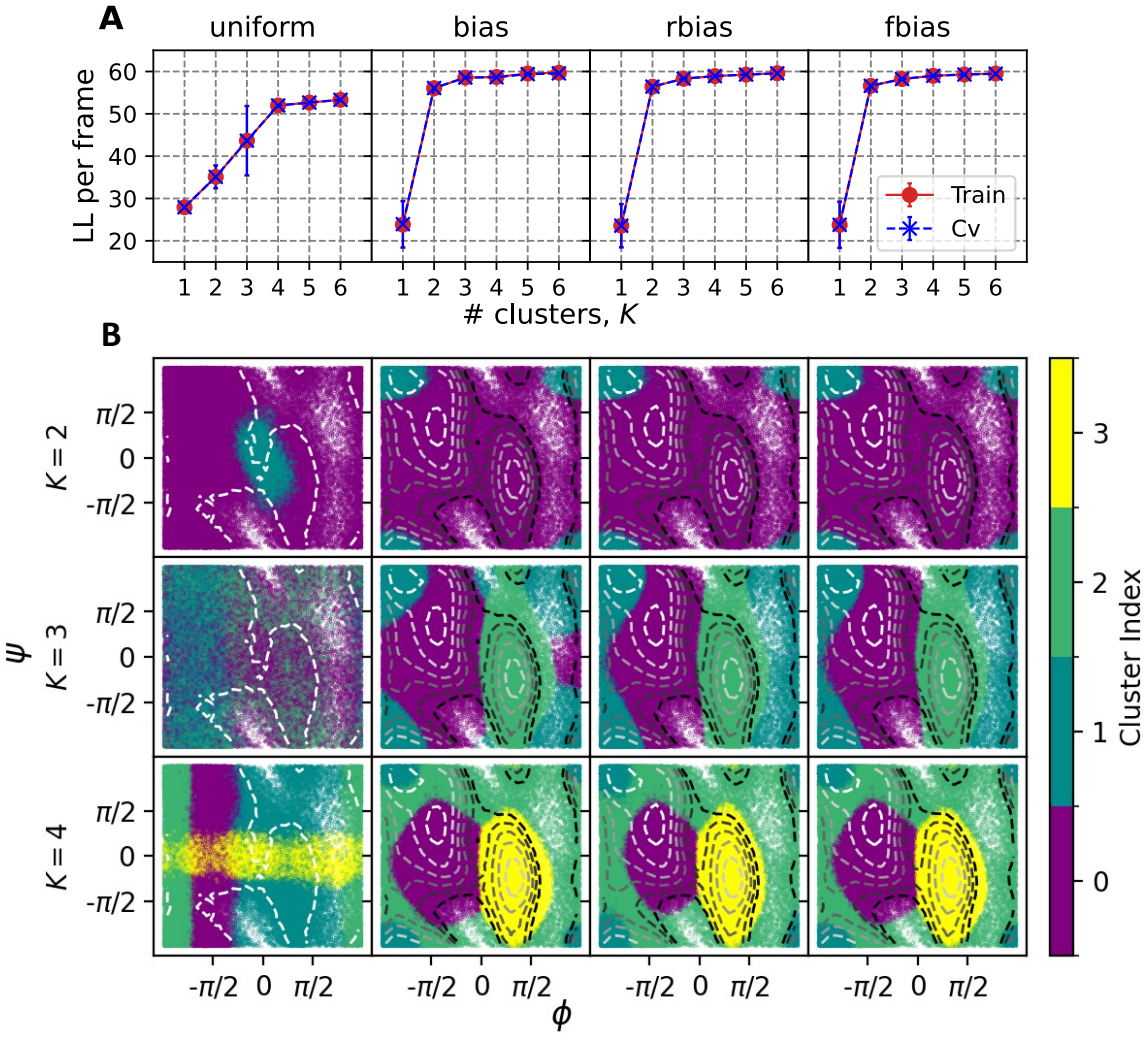

Figure 3.2: WT-MetaD simulation for ADP with BF 10. Each column represents a particular choice of weights been used in *frame-weighted* SGMM. (**A**) Cluster scans for each choice of frame weights using 50k frames, 4 training sets and 10 attempts for each case. (**B**) Clusterings performed for $K = 2, 3, 4$ are shown by coloring each of 100K sampled points by their cluster assignment. Contour lines indicate the underlying free energy surface as computed from the WT-MetaD simulation via reweighting with the different choice of weights. Contours indicate free energy levels above the minimum from 1 to 11 kcal/mol with a spacing of 2 kcal/mol.

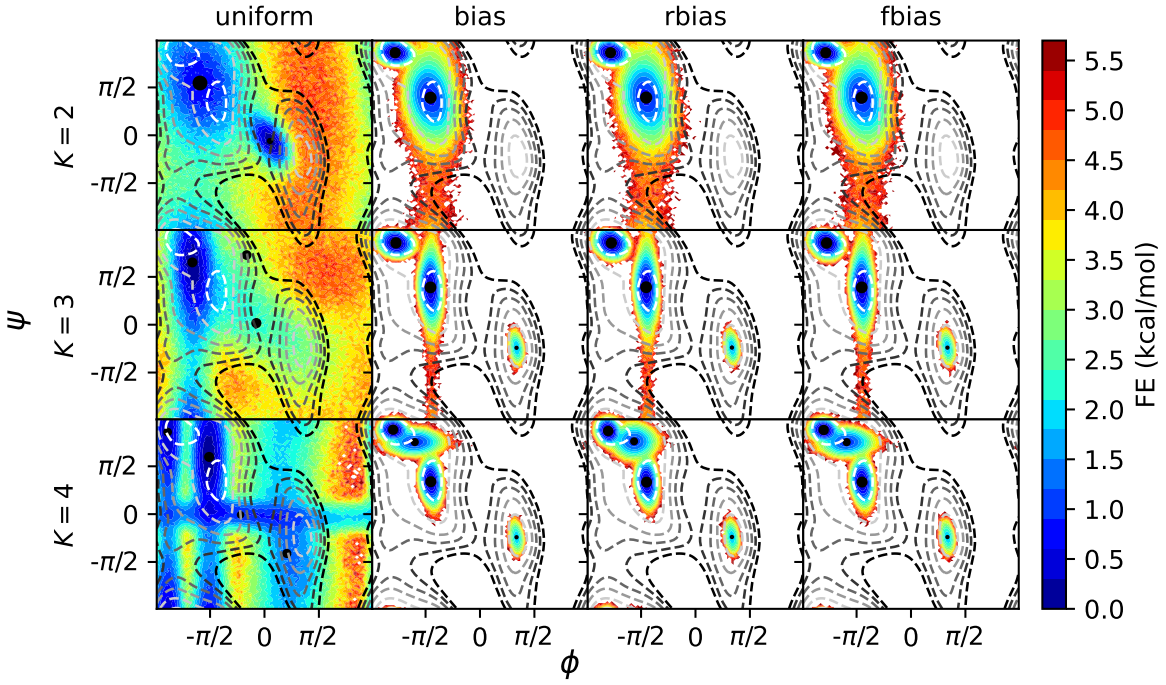

Figure 3.3: FE profiles obtained from GMM objects trained on BF=10 Metadynamics data. Each column corresponds to a different choice of bias and each row corresponds to a different number of clusters used. These are computed as unweighted histograms from 1M samples obtained from each GMM object. Black circles placed on the FEs are the centers calculated from the reference structures corresponding to different clusters, with the size indicating their relative population. Contour lines indicate the underlying free energy surface as computed from the WT-MetaD simulation, positioned at 1.0 to 11.0 kcal/mol with a spacing of 2 kcal/mol above the global minimum.

averaged over the CV grid at a fixed time. The quantity $V(\boldsymbol{s}_i(t_i)) - c(t)$ is called the "reweighting bias" and can be computed automatically in PLUMED [82], hence we term clustering using this scheme 'rbias.' Finally, we evaluate another commonly used approach to compute Boltzmann weights of each frame post-facto [127], which in the case of WT-MetaD would correspond to $w_i = e^{-F_{\text{final}}(\boldsymbol{s}(\boldsymbol{x}_i))/k_{\text{B}}T} = e^{\frac{\gamma}{\gamma-1}V_{\text{final}}(\boldsymbol{s}(\boldsymbol{x}_i))/k_{\text{B}}T}$; we label these weights 'fbias'. Other more sophisticated reweighting schemes have also been proposed, e.g. in Refs. [127, 128], but we did not test these here because, as will be seen, the bias, rbias, and fbias approaches all worked well for our test system. However, shapeGMM as implemented is capable of using any choice of frame weights. We include 'uniform' weights as a control.

For assessing the best choice of weights, we performed a 100 ns WT-MetaD simulation on ADP biasing backbone dihedral angles ϕ and ψ using bias factor 10, saving every 1 ps to generate 100k frames (see Section 3.5 for full details). The five atoms involved in the ϕ and ψ dihedral angles were chosen for shapeGMM clustering. The coordinates of these atoms and the frame weights from the four different schemes were fed into shapeGMM. The log likelihood per frame of the resulting fits as a function of number of clusters is shown in Figure 3.2A. In general, the three non-uniformly weighted clustering objects result in significantly higher log likelihoods than the uniform weights for equivalent numbers of clusters $K > 2$, indicating a better fit to the underlying data. The significant kink in the cluster scans for the non-uniformly weighted objects at 2 clusters indicate that at least 2 clusters are necessary for a good fit to the data; there is still substantial increase going from 2 to 3 clusters, however, indicating that there may be additional insight gained at $K = 3$ and above, as we shall see.

Non-uniform *frame-weighted* shapeGMM produces physically relevant clusterings. Figure 3.2B indicates how sampled points in ϕ and ψ space are assigned to two, three, or four clusters when using each of the choices of frame weights, with the underlying free energy landscape computed from a weighted histogram using the same choice of weights as used for the clustering indicated by contour lines. Clustering with uniform weights has little correlation with the underlying free energy landscape, whereas performance is much better when using any of the non-uniform weighting schemes. Weighted clustering with $K = 2$ tends to split the landscape into one cluster covering the most extended upper-left "C5" basin near (-2,2), while using a second cluster to cover the rest of the landscape (see Ref. [129] for a naming convention). However, higher number of clusters allows for separating the upper left basin into its two constituent states, C5 and "C7eq" at (-2,1), while also revealing the presence of the minor "C7ax" state at (1,-1). Slight differences in contour FES correspond with slight differences in the weighted cluster assignments; for example, in the $K = 3$ case the upper left and bottom left parts of the axial basin are disconnected at $\Psi = 0$ for bias weights but connected for rbias and fbias weights.

Non-uniform *frame-weighted* shapeGMM also works for standard (untempered) MetaD [20, 84] with $\Delta T \to \infty$. For untempered MetaD, we favor rbias weights, because the final bias is not static and the instantaneous bias diverges, meaning that initial frames receive no weight. In Figure 3.6, we show that shapeGMM clustering with rbias weights performs much better than equally weighted frames, and results are comparable to our study with WT-MetaD, indicating that *frame-weighted* shapeGMM can be extended to this method as well.

Non-uniform *frame-weighted* shapeGMM probability densities quantitatively capture the correct free energy basins. Because we know that the free energy in dihedral space is a good proxy for the configuration space of ADP, we here quantify the accuracy of our GMM fits (which are 15-dimensional objects) by predicting this FE landscape directly from the GMMs. To do so, we generate 1M samples in Cartesian space from each GMM object and compute the FES from an unweighted histogram of the backbone dihedral angles. Figure 3.3 shows a comparison of these predicted FES with the reference FES computed directly from the WT-MetaD bias as described above. Here we see that uniform weights produces FES that span all of dihedral space but whose minima are not centered on the true minima.

In contrast, the FESs generated from the non-uniform weighting schemes demonstrates that the clustering above captures the nature of the underlying FES as well as could be expected given a limited number of clusters. FES for $K = 2$ capture the primary C7 equatorial global minimum and C5 metastable state, while going to three or more clusters also allows resolution of the minor C7 axial basin. As should be expected, the GMM objects only resolve the configurational landscape of our system around the minima, and cannot resolve (non-convex) high free energy regions. Importantly, we note that the results reflect an intrinsic error due to the fact that we are fitting an anharmonic landscape to a locally harmonic model, resulting in an over-estimate of the FES away from the minima. We can also compute a FES that covers the entire energy landscape using a Monte Carlo procedure described in Section 3.6, resulting in FES shown in Figure 3.8 that are qualitatively correct but which also reflect the inherent overestimation of the Gaussian model.

The comparison of FESs can be further quantified by difference metrics which also provide an alternative metric to choose the best method or best number of clusters. In Figure 3.9 we show both the root-mean-squared error (RMSE) for the sampled region and the JSD as compared to the reference FES. While the uniform weights perform poorly, we see that all other weights do comparably well for 3 or more clusters. Using RMSE as a metric, rbias weights are the most accurate by a small margin, and a five state clustering is the best within the range $K = 2$ to $K = 6$. Additionally, we compute the change in configurational entropy between all shapeGMM objects and the metaD ground truth (ΔS_{config} in Table 3.2). The trend is similar to the other metrics in that the weighted objects all have smaller magnitude ΔS_{config} compared to the uniform weights. We also include a modified uniform weight shapeGMM object (uniform_{modf} in Table 3.2) in which we reweight only the cluster populations (ϕ_j) after the shapeGMM fit using final bias weights. ΔS_{config} values for these objects are almost identical to the unmodified uniform object indicating that simply reweighting cluster populations is unsatisfactory for shapeGMM.

3.3.3 Elucidating conformational states of the actin monomer

Up to this point, we have established that we can accurately train a GMM with data weighted from MetaD or Hamiltonian reweighting for small systems. In this section, we demonstrate that this approach can provide insight into data for a complex biochemical problem. The actin cytoskeleton, composed of filaments of actin, plays major roles in a wide range of active biological processes, including cell motility and division [131–133]. Actin filaments are non-covalent polymers that form from head-to-tail assembly of globular actin (G-actin), which is a 375-amino acid protein consisting of four primary subdomains (Figure 3.4A). Each actin monomer contains a bound nucleotide that is in the form of ATP in G-actin and is eventually hydrolyzed to ADP as filaments "age" [132, 134]. The polymerization from G-actin to filamentous actin (F-actin) results in a flattening of the protein which is characterized by a reduction of the ϕ dihedral angle shown in Figure 3.4A [132]. An open question in the field is whether the flat state is metastable in so-

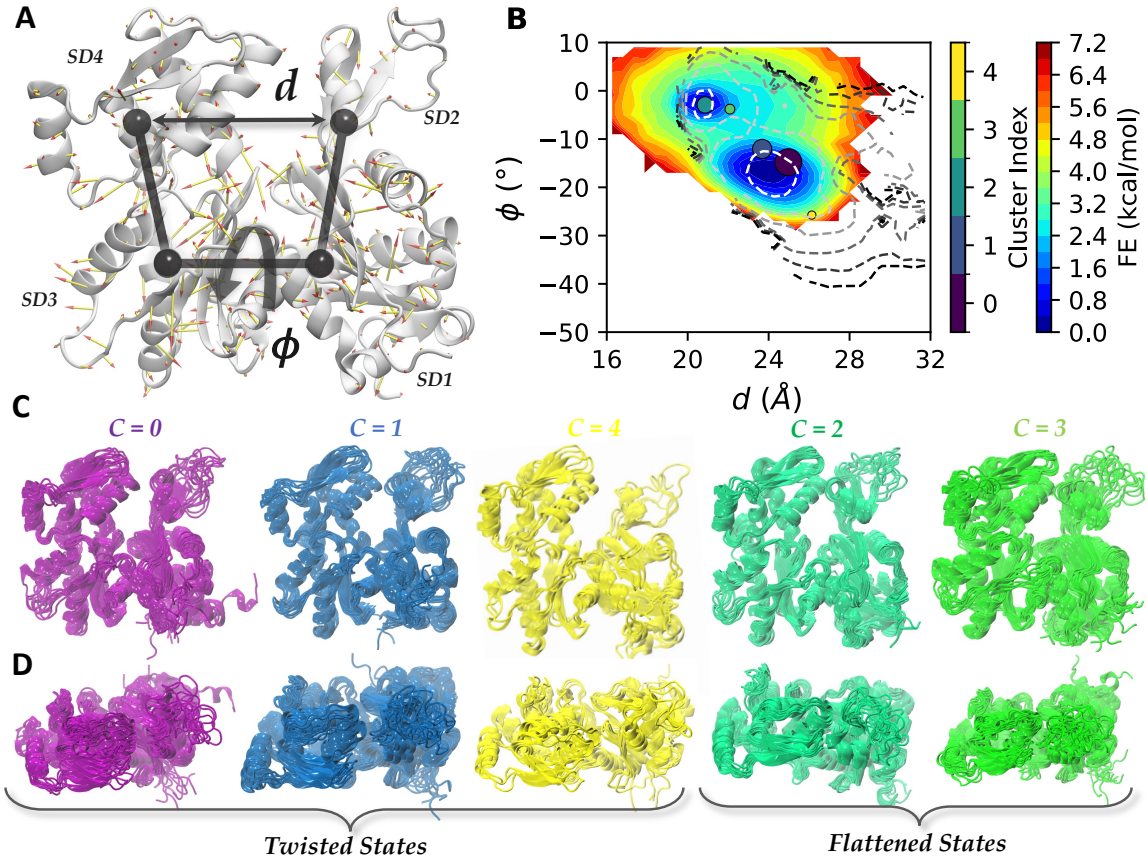

Figure 3.4: Conformational states of Actin monomer and 2D FES obtained from OPES-MetaD.(**A**) Cartoon representation of Actin monomer. The arrows representing the magnitude and directions of the LD vector acting on 375 C_α atoms. SD1 to SD4 are four subdomains defined for the monomer[130]. d is the distance between center of masses (COMs) of subdomains SD2 and SD4. ϕ is the dihedral angle defined using COMs of SD2-SD1-SD3-SD4 respectively (**B**) FES calculated by performing an unweighted histogram of ~1M samples generated from GMM. Contour lines represent the reweighted FE obtained from restarted OPES-MetaD trajectory using fbias frame weights. Contours are positioned at 1 to 11 kcal/mol with a spacing of 2 kcal/mol above the global minimum. Colored circles are the locations for different cluster centers weighted by relative population. (**C**) Snapshots of frames belonging to different clusters (front view) (**D**) Top view for the same.

lution, or whether it is only stabilized when contacting the end of a filament [135]. Additionally, the structural intermediates along the flattening pathway remain elusive.

Previous efforts to directly sample the flattening of G-actin have proven difficult. These efforts employed umbrella sampling or MetaD on two experimentally defined coordinates ϕ and d and demonstrate the difficulty in sampling the conformational landscape of actin, either because restraining those coordinates traps you in the starting state, or because a MetaD bias can quickly push you into unphysical regions of configuration space [87, 130]. Other related efforts have investigated the role of flattening on ATP hydrolysis catalyzed by actin, and analogous transitions in the homologous proteins Arp2 and Arp3 [87, 134, 136–139]. None of these previous studies have been able to identified intermediate structures that might occur during flattening.

Here, we report for the first time biased MD simulations that sample reversibly the flat to twisted transition of actin by using our method to produce a position linear discriminant analysis (posLDA) [54] coordinate separating the two states. To determine the LDA reaction coordinate, we performed two short MD simulations starting from each of these states and used 10 ns from the twisted and 5ns from the flat state (shorter because it eventually flattens [136]; see Section 3.5 for full details). We then performed iterative alignment of all frames in both states (using positions of all 375 C_α atoms) to the global mean and covariance as described in [54]. LDA on the resulting aligned trajectory yielded a single posLDA coordinate that separates the twisted and flat states. The coefficients for the posLDA coordinate separating the two states is illustrated using a porcupine plot in Figure 3.4A. We then performed the OPES variant of WT-MetaD [25, 83] along this reaction coordinate as described in Section 3.5.

Frame-weighted shapeGMM trained on an OPES MetaD trajectory indicates that five distinct structural states can be occupied during a twisted to flat transition of actin. The trajectory generated contains two full round trip trajectories between flat and twisted states as measured by changes in ϕ (Figure 3.12), which provides sufficient sampling to investigate the observed conformations and approximate relative free energies. The FES estimated from this approach is shown

in Figure 3.12. To increase the number of samples available for clustering purposes, we initiated new simulations using a fixed bias taken from the end of the simulation as described in Section 3.5. A cluster scan using these additional frames (see Figure 3.7) shows small kinks at $K = 3$ and $K = 5$, and in Figure 3.4B,C we show results for $K = 5$ in more detail. Reasonable agreement between the training set and the cross validation set in Figure 3.7 demonstrates a lack of overfitting on this data set.

The FES computed from the shapeGMM probability density ($K = 5$) agrees well with the MetaD free energy. Figure 3.4B shows the FESs computed from the shapeGMM probability density (in the colormap) and the MetaD (in the contours). The FESs are shown in the space of the ϕ and d coordinates illustrated in Figure 3.4A which have been used to describe the G- to F-actin transition, for better comparison with earlier MD studies [87, 130]. The MetaD simulation was performed in ϕ and the LD coordinate so was reweighted into these coordinates using the same weights fed into shapeGMM. There is impressively good quantitative agreement between the surfaces up to 3 kcal/mol ($\sim 5\ k_{\mathrm{B}}T$) considering the very high dimensionality of the GMM. The agreement around the energy minima in this space indicate that the shapeGMM probability density is a good representation of the MetaD simulation results for these regions.

The five states predicted by shapeGMM are in stark contrast to the two that would be predicted just by looking at a 2D free energy projection. Overlain on the FES depicted in Figure 3.4B are circles indicating the average ϕ and d for the structures assigned to each cluster, with the size indicating their relative population. The five state clustering detected two clusters in the flat F-actin like basin ($\phi \sim -3$) and three states in or around the twisted basin ($\phi < -10$). The 2D FESs either in d and ϕ (Figure 3.4B) or in the sampled ϕ and LD (Figure 3.12) space have two basins. Clustering in this space would thus likely yield two states. The five-state shapeGMM probability density, however, quantitatively matches the 2D FES thus demonstrating the potential oversimplification achieved in lower dimensional clusterings.

Figure 3.4C,D show representative snapshots from the frames assigned to each cluster in two

different orientations. To give some interpretation to these three different states, we have computed the average root-mean-squared deviation (RMSD) to several published crystal or CryoEM structures of actin alone (twisted), in a filament (flattened), or in complex with an actin binding protein for the C_α atoms available in all crystal structures (numbers 7-38, 53-365 out of a total 375). The twisted states ($C = 0, 1, 4$) all have lower RMSD to twisted than flat actin subunits, while the converse is true for the flat states ($C = 2, 3$). State $C = 4$, which is the most twisted, has the lowest RMSD to the starting structure 1NWK [140] (1.67 Å) and ADP-bound actin 1J6Z [141] (1.73 Å) than do clusters 0 and 1 (2.59 Å, 2.48 Å). It is expected based on earlier work that our simulations would produce a more flat equilibrium state for ATP-bound actin than what is seen in the crystal structure (which was solved with a non-hydrolyzable ATP analog [140]). What is interesting is that the clustering algorithm still picks up on this more twisted state as a possible structure, despite the fact that early frames in the trajectory have relatively low weight (since they have little bias applied at that point).

Interestingly, states $C = 0$ and $C = 1$ have equally low RMSD to actin structures in complex with another protein as to the twisted structures considered, for example 2.59 Å and 2.48 Å RMSD to the twisted starting structure 1NWK, but 2.28 Å and 2.09 Å to the structure of actin complexed with the protein profilin (3UB5 [142]), which is how a large fraction of actin monomers are found in cells. This suggests that our weighted GMM models may be able to point us towards biologically relevant configurations within a conformational ensemble.

Within the flat states, the most noteworthy difference appears to be in the disordered D-loop (upper right), with cluster 3 having a significantly higher variance than cluster 2. This difference is also evident if we look at the Root-Mean-Squared-Fluctuations of the D-Loop residues shown in Figure 3.10. This lower RMSF state ($C = 2$) could correspond to one of the intermediates previously probed through MetaD simulations along a disordered-folded pathway for the D-loop, which were metastable for the ATP-bound actin used in our study, but would be expected to become more stabilized after conversion to ADP [143]. Meanwhile, on close inspection ($C = 3$)

seems to contain some more disordered structures and some partially folded structures, meaning that the higher variance could be a result of combining two sub-populations into one single state. As it stands, both flattened states have higher RMSF than all twisted states, suggesting a coupling between D-loop structure and twisting that was previously ascribed to nucleotide state (ATP vs ADP), as opposed to the conformational transition which results in ATP hydrolysis, and this would be an interesting question to consider in the future.

3.4 Conclusions

In this work, we present a probabilistic structural clustering protocol that can rigorously account for non-uniform frame weights. This ability allows shapeGMM to be applied, directly, to reweighted or enhanced sampling simulation data to achieve a clustering of in the underlying Hamiltonian of interest. Additionally, we demonstrate that the resulting shapeGMM probability density is a good approximation to the underlying unbiased probability and can thus be used to calculate important thermodynamic quantities such as relative free energies and configurational entropies. To do so, we took advantage of our ability to generate biomolecular configurations from the trained clustering model; this is a unique and powerful advantage of a using a probabilistic clustering model that operates directly in position space which has not been previously exploited to our knowledge.

By applying our method to the flattening of G-actin, we have shown that this approach is capable of picking out physically meaningful structural clusters even for highly complex systems, and illustrates how structural clustering on biased data can provide additional insights that would be difficult to obtain only by looking at the free-energy projected into low dimensional coordinates. In sum, our work represents a significant advance in our ability to quantify biomolecular ensembles. In the future, we envision this approach to be useful in quantifying important biophysical processes such as ligand binding and allosteric regulation.

3.5 Simulation Details

Input files, shapeGMM objects, and analysis codes used to generate all figures are available from a github repository for this article: https://github.com/hocky-research-group/weighted-SGMM-paper. The simulation input files and plumed parameter files are also included in a PLUMED-NEST repository under plumID:24.009.

Beaded Helix

A 12-bead model designed to have two equi-energetic ground states as left- and right-handed helices [93] was simulated in LAMMPS [144]. 11 harmonic bonds between beads having rest length length 1.0 and spring constant 100 form a polymer backbone. Lennard-Jones (LJ) interactions between every $i, i+4$ pair of beads with $\sigma = 1.5$ and a cutoff length of 3.0 give rise to the helical shape. The ϵ value of this interaction dictates the stability of the helices and was the focus of our reweighting. Simulations were performed with $\epsilon = 6$ as the baseline and with $\epsilon = 8$ and $\epsilon = 4.5$ to assess the accuracy of the reweighting scheme. All non-bonded $i, i+2$ and farther also have a repulsive WCA interaction with $\epsilon = 3.0$ and $\sigma = 3.0$ added to prevent overlap, with the ϵ for $i, i+2$ reduced by 50%. Simulations at temperature 1.0 were performed using 'fix nvt' using a simulation timestep of 0.005 and a thermostat timestep of 0.5. A folding/unfolding trajectory of length 50000000 steps was generated and analyzed as above. Here, all parameters are in reduced (LJ) units.

Alanine dipeptide in vacuum

Alanine dipeptide simulations were performed using GROMACS 2019.6 with PLUMED 2.9.0-dev. GROMACS mdp parameter and topology files are obtained from previous PLUMED Tutorials (Belfast-7: Replica Exchange I). AMBER99SB-ILDN force field is used with a time step of 2 fs.

NPT ensemble is sampled using velocity rescaling thermostat and Berendsen barostat with a temperature of 300K and pressure 1 bar. For METAD simulations we used PACE=500, SIGMA=0.3 (for both ϕ and ψ) and HEIGHT=1.2 kcal/mol. PLUMED input files are available in our paper's github repository for complete details.

Actin monomer

Actin simulations were also performed using GROMACS 2019.6 with PLUMED 2.9.0-dev. G-actin with a bound ATP was built and equilibrated at 310 K as described previously [136]. The structure of the twisted, ATP-bound actin is derived from the crystal structure with PDB ID 1NWK [140], while that in the flat state is taken from PDB ID 2ZWH [145], with the nucleotide, magnesium ion, and surrounding water replaced with ATP as described previously. MD simulation for ~5ns was performed to relax the starting structure. NPT simulation was performed with 2 fs time step. Parrinello-Rahman barostat is used along with velocity rescaling thermostat with a temperature of 310K and pressure 1 bar. For OPES we used PACE=500, BIASFACTOR=12, BARRIER=15.0 kcal/mol and a multiple time step stride of 2. Two UPPER_WALLS were employed $\sim$ -1 $^\circ$ and 31 Å, for ϕ and d respectively. We also used one UPPER_WALLS at +40.0 and one LOWER_WALLS at -40.0 for the posLDA coordinate. All the walls used were quadratic with a spring constant of KAPPA=500 kcal/mol/nm^2. PLUMED input files are available in our paper's github repository.

We performed $\sim 1\mu$s of sampling along this LD coordinate and dihedral angle ϕ using the On the Fly Probability Enhanced Sampling variant of Metadynamics (OPES-MetaD) [25, 83]. This method uses a kernel density estimate of the probability distribution over the whole space for biasing rather than building this bias through a sum of Gaussians. The bias at time t for CV value $\boldsymbol{s}_i$ is given by the expression

$$V(\boldsymbol{s}_i) = k_B T \left(\frac{\gamma - 1}{\gamma}\right) \log\left(\frac{P_t(\boldsymbol{s}_i)}{Z_t} + \epsilon\right). \tag{3.14}$$

Here, $P_t(\boldsymbol{s})$ is the current estimate of the probability distribution, Z_t is a normalization factor. Finally, $\epsilon = \exp(\frac{\Delta E}{k_B T}\frac{\gamma}{\gamma-1})$ is a regularization constant that ensures the maximum bias that can be applied is ΔE. OPES-MetaD data can be reweighted similarly to standard WT-MetaD, using the exponential of the bias (which is similar to rbias for MetaD) or using the estimated free energy of each frame from the final bias [25].

We chose the OPES variant of MetaD because (a) literature precedent suggests that it converges more quickly than standard WT-MetaD, and (b) it allows us to set an free-energy cutoff above which bias is not applied (in this case 15 kcal/mol) which limits the amount of unphysical exploration, in a similar manner to Metabasin-MetaD that we previously showed was desirable for this problem [87]. Even with this energy cutoff, we needed to include upper and lower walls to prevent over-flattening or over-twisting observed here and in prior attempts by us [136].

A cluster scan on our OPES trajectory (Figure 3.7) showed a large difference between training and cross-validation curves. Hence we decided to generate to generate additional training frames. We did this by taking the bias accumulated after 900 ns of OPES simulation, and started four 1 ns simulations with random velocities from each of 191 initial configurations from the initial trajectory (separated by 5 ns each), saving every 5 ps; this resulted in ~153k frames available for clustering. The resulting training and cross-validation curves are in much better agreement as discussed in the main text, hence these data were used for clustering and analysis.

3.6 Supplementary Figures

Choosing Training Data

When fitting a shapeGMM, we split our data into a training set and a cross validation set. The Gaussian mixture components are fit on the training data and their ability to model the cross validation set is assessed by comparing the log likelihood per frame on both sets. Overfitting will

lead to a lower log likelihood on the cross validation set than on the training set. Both training and prediction routines now have built in frame weight arguments.

Training sets were chosen uniformly randomly for the original implementation of shapeGMM. For non-uniform frame weights, however, there are a variety of other methods one could consider to best choose a training set. We assessed a number of these including simple ranking, Poisson sampling, and a Metropolis Monte Carlo method using log frame weights as energies. It was found the the uniform sampling of frame weights worked as well as other methods especially when training sets are sufficiently large.

A uniform sampling of the training set performs at least as well as importance sampling of the training set for the beaded helix example. To assess this we compared shapeGMM objects fit using various training set sampling schemes. These include: a uniform sampling, a Monte Carlo sampling in which frames are replaced based on the Metropolis criteria using frame weights, and a Poisson sampling scheme in which frames are sampled from the frame weight distribution. The Poisson sampling method differs from the other two in that frames are equally weighted in the training set but can appear multiple times depending on their relative weights. The Jensen-Shannon divergence (JSD) between distributions fit using these methods to an $\epsilon = 6$ trajectory with $\epsilon = 8$ weights and distributions fit to an $\epsilon = 8$ simulation directly (the ground-truth; GT) as a function of training set size are depicted in Figure 3.5. The JSD between the GT and all fitted distributions is large (~ 0.3) for small training set sizes and tends to zero as training sets increase. This indicates that all methods are accurately reproducing the GT distribution for large enough training set. We find that the uniform sampling approach does as well or better than either importance sampling approach for all training set sizes. We note that this result will depend on the specific distribution of weights. We expect this behavior to hold for relatively uniform distributions of weights which occur in reweighting to Hamiltonians that don't deviate much from the original. It may be important, especially for small training set sizes, in cases in which the Hamiltonians are significantly different to consider choosing training sets using an importance

sampling approach. We use a uniform sampling approach for all other applications in this paper.

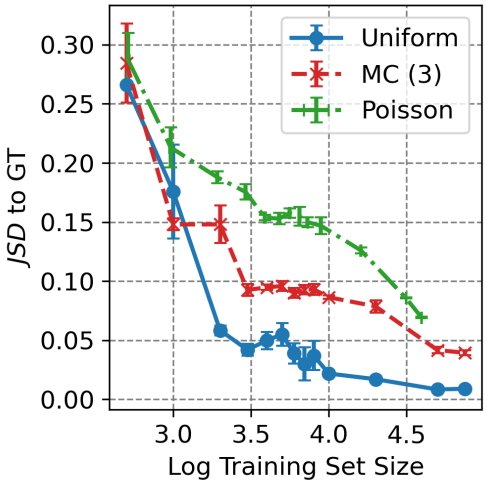

Figure 3.5: Accuracy of beaded helix reweighted cluster as a function of training set size. The Jensen-Shannon divergence (JSD) between shapeGMM distribution fit using reweighting to $\epsilon = 8$ and the ground-truth fit to a simulation run at $\epsilon = 8$ as a function of training set size. Three taining set selection schemes are compared: a uniform sampling of frames, a three-step Monte Carlo importance sampling method, and a Poisson sampling method.

Clustering untempered metadynamics

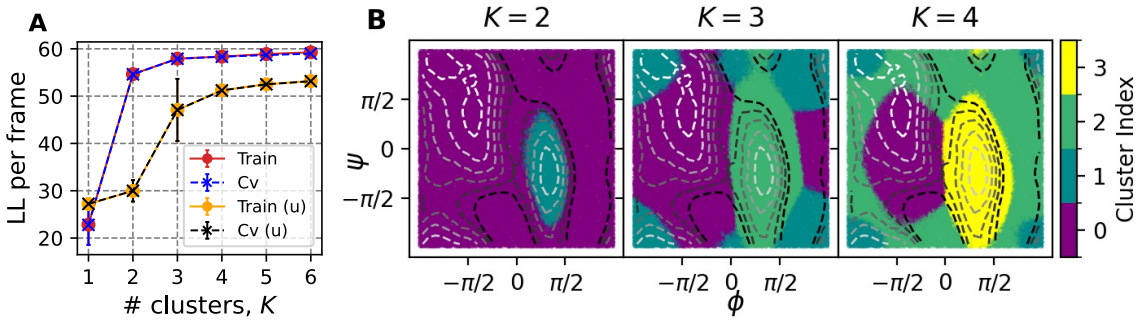

Figure 3.6: Untempered MetaD simulation of ADP. (**A**) Cluster scans obtained with 50K frames, 4 training sets and 10 attempts using rbias frame weights or with uniform weights (labeled 'u'). Training $\ln(L)$ curve is substantially higher with rbias weights, and matches CV curve. (**B**) Clusterings performed for $K = 2-4$ shown by coloring each of 100K sampled points by their cluster assignment. Contour lines indicate the underlying free energy surface as computed from the MetaD simulation via reweighting with rbias frame weights. Contours indicate free energy levels above the minimum from 1 to 11 kcal/mol with a spacing of 2 kcal/mol.

Cluster Scans

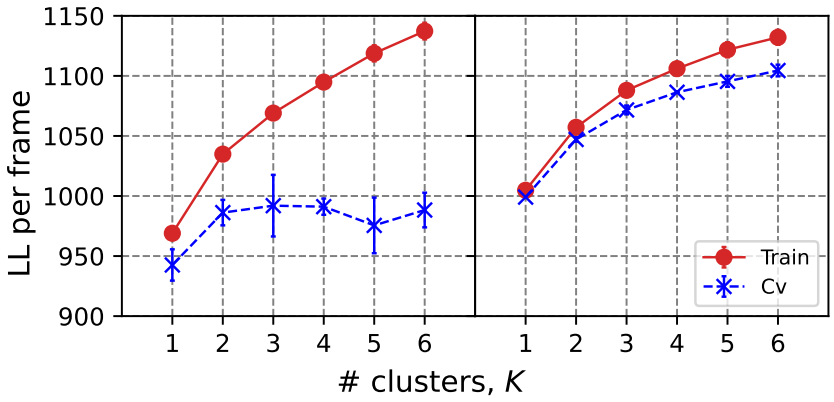

Figure 3.7: Cluster scans using Actin OPES-MetaD data. Log likelihood as a function of number of clusters K for the original $\sim 1\mu$s OPES-MetaD trajectory (~21K frames), and using a new set of frames generated by restarting as described in Section 3.5 (~ 153K frames).

ADP FES computed by evaluating GMM on WT-MetaD samples

In Figure 3.8 we assess an alternative approach to estimate an unbiased FES from a GMM object. In this case, we presume that the WT-MetaD simulation produced physically reasonable configurations spanning the configurational landscape of the molecule of interest. To estimate the FES for ADP, we compute a weighted histogram of (ϕ and ψ) where we give as weights the probability of each frame predicted by the GMM, $P(\boldsymbol{x}_i)$ given by (3.2). In practice, $P(\boldsymbol{x}_i)$ is computed from exponentiating the log-likelihood of frames within the GMM . We normalize the resulting histogram by samples in each bin, which accounts for the fact that frames were not generated uniformly by WT-MetaD, resulting in a new distribution $\tilde{P}(\phi, \psi)$. The FES is then computed as $F(\phi, \psi) = -k_{\mathrm{B}} \ln \tilde{P}(\phi, \psi)$.

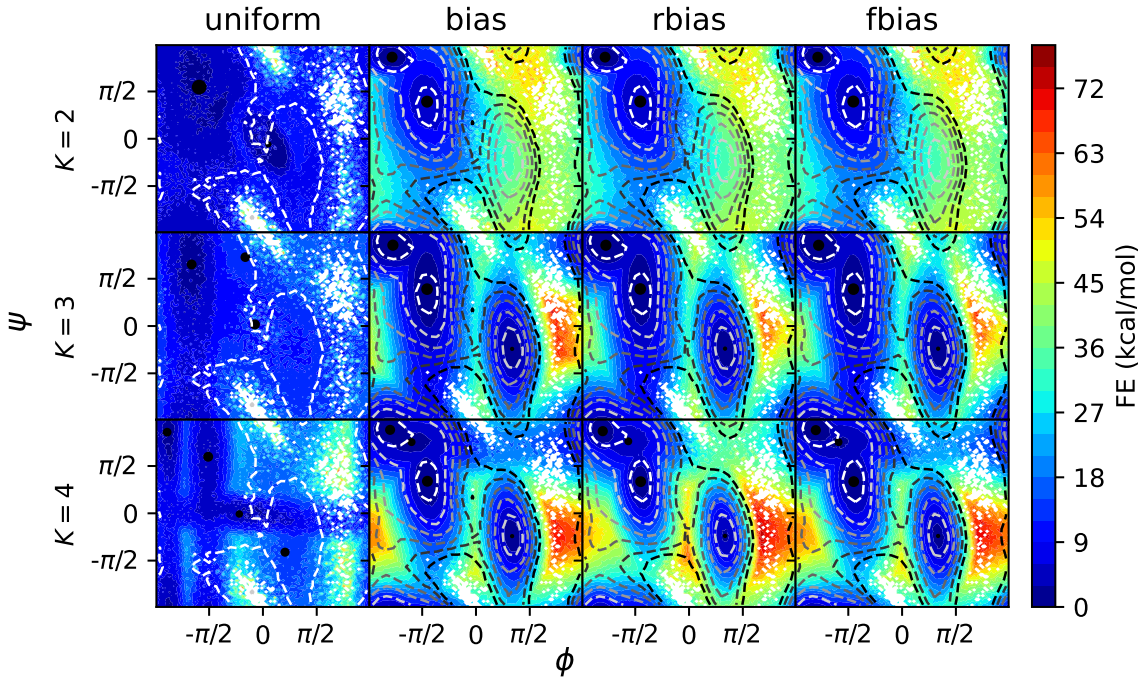

Figure 3.8: FE profiles obtained from GMM objects trained on BF=10 WT-MetaD data using Monte Carlo procedure. Each column corresponds to a different choice of bias and each row corresponds to a different number of clusters (K) used. Black circles placed on the FEs are the centers calculated from the reference structures corresponding to different clusters, with the size indicating their relative population. Contour lines indicate the underlying free energy surface as computed from the WT-MetaD simulation, positioned at 1.0 to 11.0 kcal/mol with a spacing of 2 kcal/mol above the global minimum.

Error analysis for GMM Free energies

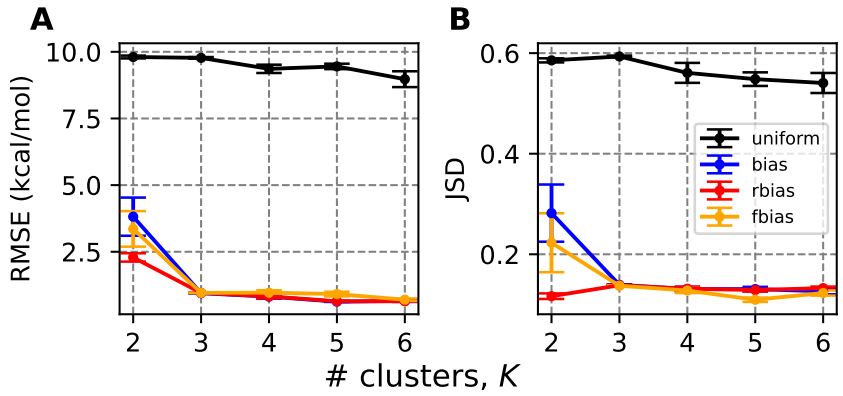

Figure 3.9: Error estimation for free energies. (**A**) Root mean-squared error for the free energy of ADP GMMs computed for different number of clusters and four different weighting schemes. Error bars are computed from five independent simulations which are fit to separate GMM objects, which are then used to compute free energy surfaces. The reference free energy surface is that computed by summing the Gaussian hills from the WT-MetaD simulation. (**B**) Same as A, except the Jenson-Shannon distance is computed between the distributions corresponding to $P(\phi, \psi) \propto \exp(-F(\phi, \psi)/(k_B T))$, where $F(\phi, \psi)$ corresponds to either the reference free energy or that computed from the GMM objects.

Variance of D-loop

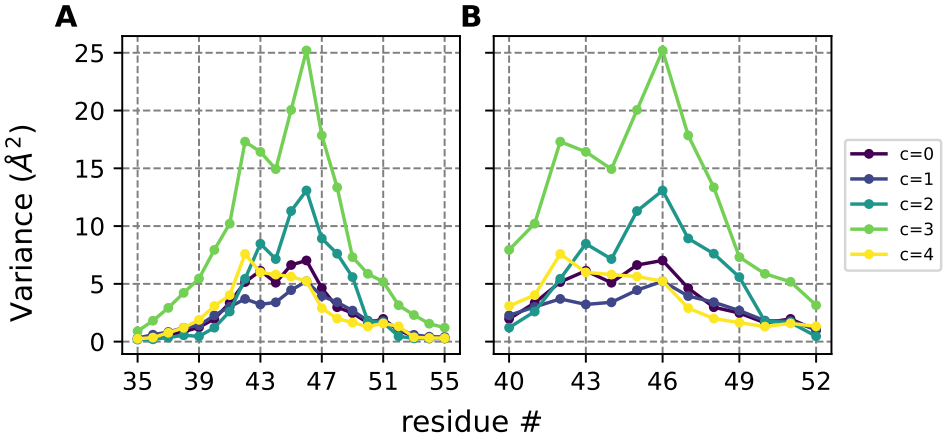

Figure 3.10: Variance of D-loop in Actin clusters. (**A**) RMSF for residues 35 to 55 within Actin's subdomain 2 including the D-loop. These are extracted from the diagonal of Σ_N for each of five clusters shown in Figure 3.4. (**B**) The same quantity for residues 40 to 52 which represent the core of the D-loop.

FEs from GMM for cluster size 5 and 6

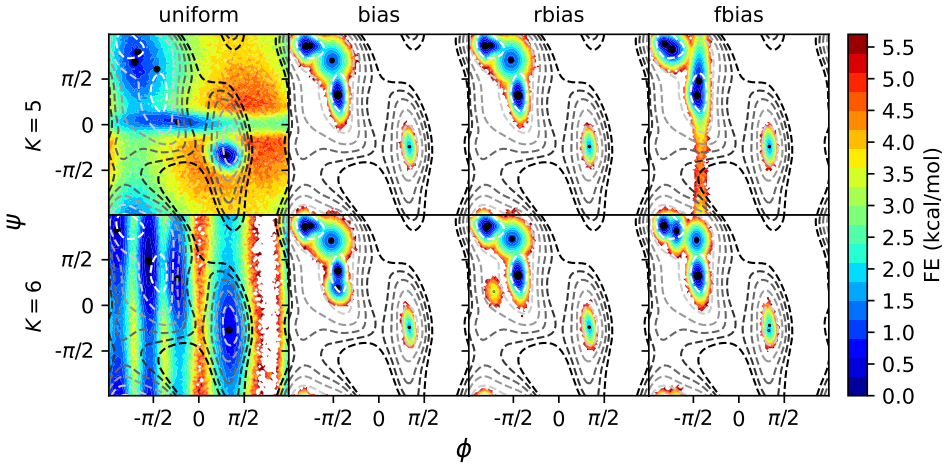

Figure 3.11: FEs from GMM for cluster size 5 and 6. Free energies are computed using BF=10 WT-MetaD data. Each column corresponds to a different choice of bias and each row corresponds to a different number of clusters used. These are computed as unweighted histograms from 1M samples obtained from each GMM object. Black circles placed on the FEs are the centers calculated from the reference structures corresponding to different clusters, with the size indicating their relative population. Contour lines indicate the underlying free energy surface as computed from the WT-MetaD simulation, positioned at 1.0 to 11.0 kcal/mol with a spacing of 2 kcal/mol above the global minimum.

OPES-MetaD simulation of Actin (~ 1μs)

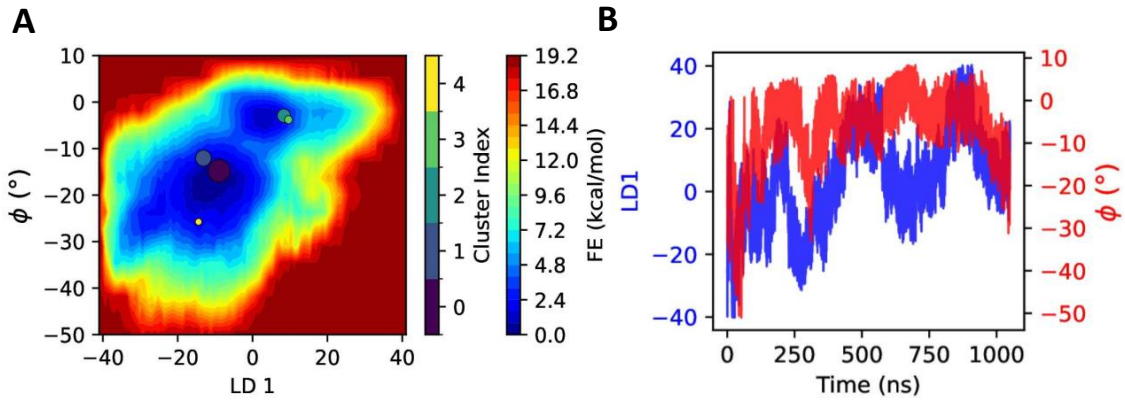

Figure 3.12: Results from OPES-MetaD simulation of Actin. (**A**) The 2D FES obtained from ~1us OPES-MetaD simulation. Colored circles are the locations for cluster centers weighted according to their relative population. (**B**) Time series of LD1 and Dihedral CVs from the same data.

Configurational Entropies from GMMs

$\Delta S_{\text{config}}/k$					
# Clusters, K	choice of weight, C				
	uniform	bias	rbias	fbias	uniform$_{modf}$
2	2.42	-0.34	-0.31	-0.34	2.42
3	2.18	-0.80	-0.79	-0.80	2.18
4	1.48	-0.82	-0.82	-0.82	1.48
5	2.23	-0.80	-0.78	-0.69	2.24
6	1.52	-0.74	-0.83	-0.80	1.52

Table 3.2: Configurational Entropies. Difference in two configurational entropies computed from probability distributions in dihedral space, comparing all shapeGMM objects with metadynamics taken as ground truth (GT). $\Delta S^{K,C}_{\text{config}} = S^{K,C}_{\text{config}} - S^{GT}_{\text{config}}$, where K= # Clusters, C= choice of weight. To compute $S^{K,C}_{\text{config}}$, 1M samples are generated from the shapeGMM object and a 2D normalized probability distribution is calculated in dihedral space with generated data. All S_{config} values are calculated using (3.10). uniform$_{modf}$ represents uniform weight shapeGMM objects where the cluster populations are reweighted using final bias weights after the shapeGMM fit. To reweigh, we update the weights for each cluster in a given shapeGMM object with the sum of normalized fbias weights for all frames assigned to that cluster in the uniform scheme. ΔS_{config} is always less for the weighted objects compared to uniform weights irrespective of cluster sizes.

4 Improved data-driven collective variables for biased sampling through iteration on biased data

This chapter has been adapted from Ref. [146]

4.1 Introduction

Molecular dynamics (MD) is a powerful approach for studying complex biochemical processes [147]. However, many critical events, such as protein folding and allosteric regulation of enzymes, occur on timescales that are often inaccessible to conventional MD due to the so-called rare event problem [121, 147]. In these cases, the system becomes trapped in an initial metastable state, unable to overcome high free energy barriers that separate different regions of the free energy landscape. This limitation is particularly pronounced in large systems with many degrees of freedom, where fully sampling all relevant states is nearly impossible, even with very long MD simulations.

Over the years, numerous enhanced sampling techniques have been developed to alleviate this challenge by facilitating more frequent transitions between different states of a system [55, 148]. One prominent class of these methods relies on collective variables (CVs), where an external bias is applied as a function of carefully chosen CVs. An ideal CV is thought to capture the

slowest modes of motion responsible for significant conformational changes in macromolecules. By applying a bias to enhance fluctuations in one or several CVs, these methods encourage the system to explore low-probability regions of the free energy surface. However, the effectiveness of CV-based enhanced sampling techniques depends heavily on the choice of CV, which can be particularly challenging for complex systems.

There are numerous methods designed to identify “optimal" CVs for a given system, each with its own strengths and limitations. Some approaches employ simple linear dimensionality reduction techniques, while others leverage machine learning (ML) and deep learning algorithms to construct sophisticated nonlinear coordinates [49–52, 54, 62, 149, 150]. Interestingly, most of these methods rely on training with initial, under-sampled MD simulation data, which often lacks sufficient information about different metastable states and their transitions. The effectiveness of these approaches is inherently dependent on the quality of the sampling used for training. The resulting CV, obtained from this limited dataset, serves as a fixed reaction coordinate that is subsequently biased in enhanced sampling simulations to achieve a well-converged free energy surface (FES).

Recent studies have introduced a different strategy for identifying optimal reaction coordinates. These approaches employ an iterative scheme that refines the initially defined CV on-the-fly by leveraging reweighted data from successive biased simulations [58, 59, 63, 65, 67, 73, 74, 78, 151, 152]. Unlike traditional methods where the CV remains fixed, this adaptive process continuously improves the coordinate as it is trained on progressively better-sampled data from each iteration of the biased simulations. This iterative refinement enhances the accuracy and efficiency of the CV, leading to a more reliable exploration of the free energy landscape. While these methods are highly efficient, they are computationally expensive. The resulting coordinates often lack clear physical interpretability and are sensitive to hyperparameters and neural network architecture, requiring careful tuning for optimal performance.

Here, we present an iterative scheme to improve our previously reported posLDA approach

[54]. The iterative process starts by creating an initial CV using data from short, unbiased MD simulations. Enhanced sampling is then performed along this coordinate, and the free energy surface is assessed for convergence. If convergence is not achieved, biased samples are reweighted and clustered using our frame-weighted ShapeGMM [105] method to further refine the coordinate. This process repeats until the free energy surface or another relevant observable converges, providing an optimal reaction coordinate for efficient sampling. We have applied this protocol on two systems of increasing complexity- a nine residue peptide $(Aib)_9$ and a 35-amino acid fast-folding Nle/Nle mutant Villin headpiece (also known as HP35). In both cases, iteration improves both the stability (meaning more aggressive biasing parameters can be used) and the sampling ability of the CV. This approach is implemented in tools that we have made available within an updated ShapeGMMTorch python package [153], with biased simulations available in a number of MD simulation packages via our `sizeshape` PLUMED module [54, 82].

4.2 Theory and Methods

4.2.1 Iteration Process

The iterative scheme employed in this work combines our previous three procedures [54, 75, 105] in a straightforward yet effective approach, as illustrated in the flowchart (Figure 4.1). Our focus is on identifying a reaction coordinate that connects two specific states of a given system.

The process begins with two short unbiased MD simulations initiated from each state (or data from a long unbiased MD trajectory can be used if both states of interest are sufficiently well represented). From the resulting labeled MD simulation data, an initial linear discriminant analysis (LDA) coordinate is constructed. In the next step, an enhanced sampling technique such as Metadynamics (MetaD) or the On-the-fly Probability Enhanced Sampling variant (OPES-MetaD) [83, 84] is used, applying a bias along the LDA coordinate within an MD simulation (see Section 4.2.4).

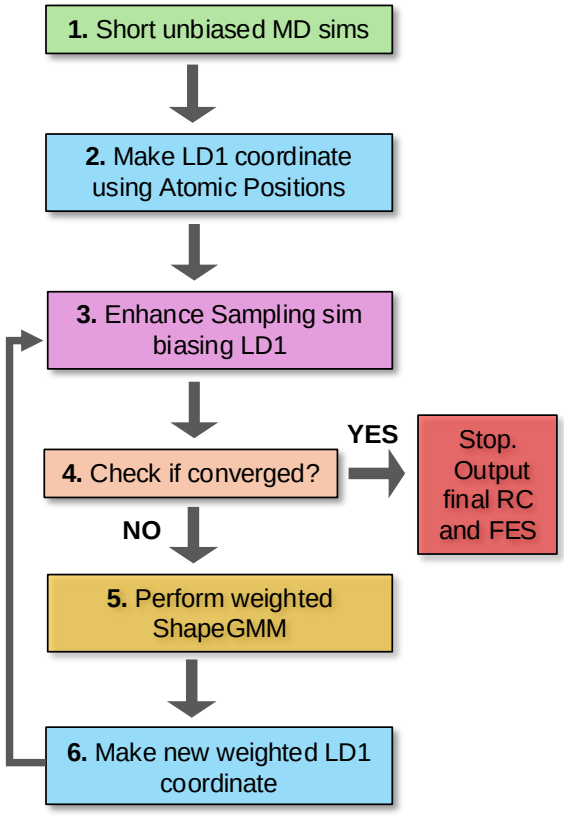

Figure 4.1: Workflow for iteration scheme. There are 6 steps: **1**. Short unbiased MD simulations are performed starting from two states of interest; **2**. use LDA method to make LD1 coordinate; **3**. perform enhanced sampling simulation, biasing LD1 coordinate; **4**. check the convergence of FES within a given threshold; if not converged, **5**. apply frame-weighted ShapeGMM on biased data to identify new states; **6**. apply frame-weighted LDA between two newly identified states to generate a new weighted LD1 coordinate. Repeat steps 3 to 6 until the simulation converges.

Following this, the convergence of the free energy (FE) is assessed. If the FE surface has not yet converged, the workflow proceeds to the next stage, where the frame-weighted ShapeGMM method is used to cluster the biased samples. This method accounts for the non-uniform weights associated with each biased sample, effectively reweighting them to provide an unbiased estimate of the clusters. Subsequently, the newly identified clusters are analyzed to determine which two new clusters best correspond to the original state definitions

A new, weighted LDA coordinate is then computed using reweighted samples from these clusters. The process iterates through steps 3 to 6 until the free energy surface (FES) or another

relevant physical observable converges within a predefined threshold. The reaction coordinate obtained at the end of this iterative process serves as an optimized bias coordinate, enhancing the efficiency of FES sampling when employed in enhanced sampling simulations. In practice, it may be difficult to assess convergence of the FES given finite time of sampling available, and so here we also focus on the efficiency of exploration, namely how frequently the biased CV and other physically intuitive CVs of interest transit between the values for the two target states.

4.2.2 Weighted ShapeGMM

In shapeGMM, a particular configuration of a macromolecule is represented by a particle position matrix, $\boldsymbol{x}_i$, of order $N \times 3$, where N is the number of particles being considered for clustering. To account for translational and rotational invariance, the proper feature for clustering purposes is an equivalence class,

$$[\boldsymbol{x}_i] = \{\boldsymbol{x}_i \boldsymbol{R}_i + \mathbf{1}_N \vec{\xi}_i^T : \vec{\xi}_i \in \mathbb{R}^3, \boldsymbol{R}_i \in \mathrm{SO}(3)\}, \tag{4.1}$$

where $\vec{\xi}_i$ is a translation in $\mathbb{R}^3$, $\boldsymbol{R}_i$ is a rotation $\mathbb{R}^3 \to \mathbb{R}^3$, and $\mathbf{1}_N$ is the $N \times 1$ vector of ones. $[\boldsymbol{x}_i]$ is thus the set of all rigid body transformations, or orbit, of $\boldsymbol{x}_i$.

The shapeGMM probability density is a Gaussian mixture given by

$$P(\boldsymbol{x}_i) = \sum_{j=1}^{K} \phi_j N(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \Sigma_j), \tag{4.2}$$

where the sum is over the K Gaussian mixture components, ϕ_j is the weight of component j, and $N(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \Sigma_j)$ is a normalized multivariate Gaussian given by

$$N(\boldsymbol{x}_i \mid \boldsymbol{\mu}, \Sigma) = \frac{\exp\left[-\frac{1}{2}(g_i^{-1}\boldsymbol{x}_i - \boldsymbol{\mu})^T \Sigma^{-1} (g_i^{-1}\boldsymbol{x}_i - \boldsymbol{\mu})\right]}{\sqrt{(2\pi)^{(3N)} \det \Sigma}}, \tag{4.3}$$

where $\boldsymbol{\mu}$ is the mean structure, Σ is the covariance, and $g_i^{-1}\boldsymbol{x}_i$ is the element of the equivalence class, $[\boldsymbol{x}_i]$, that minimizes the squared Mahalanbonis distance in the argument of the exponent. Determining the proper transformation, g_i, is achieved by translating all frames to the origin and then determining an optimal rotation matrix. Cartesian and quaternion-based algorithms for determining optimal rotation matrices are known for two forms of the covariance were considered $\Sigma \propto \boldsymbol{I}_{3N}$ [116, 117] or $\Sigma = \Sigma_N \otimes \boldsymbol{I}_3$ [118, 119], where Σ_N is the $N \times N$ covariance matrix and $\otimes$ denotes a Kronecker product. In this manuscript, we employ only the more general Kronecker product covariance.

While using input data from an enhanced sampling simulation, we take non-uniform frame weights into account by performing weighted averages in the Expectation Maximization estimate of model parameters $\{\hat{\phi}_j, \hat{\mu}_j, \hat{\Sigma}_j\}$. Considering a normalized set of frame weights, $\{w_i\}$ where $\sum_{i=1}^{M} w_i = 1$ for M frames, their contribution to the probability can be accounted for by weighting the estimate of the posterior distribution of latent variables:

$$\gamma_{Z_i}(j) = w_i \frac{\hat{\phi}_j N(\boldsymbol{x_i} \mid \hat{\boldsymbol{\mu}}_j, \hat{\Sigma}_j)}{\sum_{j=1}^{K} \hat{\phi}_j N(\boldsymbol{x_i} \mid \hat{\boldsymbol{\mu}}_j, \hat{\Sigma}_j)}. \tag{4.4}$$

The frame weight will propagate to the estimate of component weights, means, and covariances in the Maximization step through $\gamma_{Z_i}(j)$:

$$\hat{\phi}_j = \sum_{i=1}^{M} \gamma_{Z_i}(j) \tag{4.5}$$

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_{i=1}^{M} \gamma_{Z_i}(j) g_{i,j}^{-1} \boldsymbol{x}_i}{\sum_{i=1}^{M} \gamma_{Z_i}(j)} \tag{4.6}$$

$$\hat{\Sigma}_j = \frac{\sum_{i=1}^{M} \gamma_{Z_i}(j) \langle \hat{\Sigma}_N \rangle_i}{\sum_{i=1}^{M} \gamma_{Z_i}(j)} \otimes \boldsymbol{I}_3 \tag{4.7}$$

Additionally, the log likelihood per frame is computed as a weighted average

$$\ln(L) = \sum_{i=1}^{M} w_i \ln\left(\sum_{j=1}^{K} \hat{\phi}_j N(\boldsymbol{x}_i \mid \hat{\boldsymbol{\mu}}_j, \hat{\Sigma}_j)\right). \tag{4.8}$$

4.2.3 Frame-weighted LDA (wLDA)

LDA is a supervised classification technique that reduces dimensionality of the data by means of projection into a lower dimensional space [31]. It does so by simultaneously maximizing the between class scatter matrix and minimizing the within class scatter matrix. In our prior work, we have demonstrated that application of LDA on aligned particle positions produce a good one dimensional reaction coordinate that best separates two states. In Frame-weighted LDA approach, one can use input data obtained from enhanced sampling by incorporating nonuniform weights of the samples to account for relative probabilities of different classes. To do so, we must include weights corresponding to each configuration in the while computing the scatter matrices. For K different clusters, this is achieved by first computing the weighted within-cluster scatter matrix,

$$\boldsymbol{S}_W^w = \sum_{i=1}^{K} \sum_{j \in N_i} w_j (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T, \tag{4.9}$$

and the between-cluster scatter matrix,

$$\boldsymbol{S}_B^w = \sum_{i=1}^{K} W_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T, \tag{4.10}$$

where $\boldsymbol{\mu}_i = \frac{\sum_{j \in N_i} w_j \boldsymbol{x}_j}{\sum_{j \in N_i} w_j}$ is the weighted mean of cluster i, $\{w_j\}$ are the normalized weights of individual samples such that $\sum_{j=1}^{M} w_j = 1$ and N_i is the number of samples that belong to cluster i. $\boldsymbol{\mu} = \frac{\sum_{i=1}^{K} \sum_{j \in N_i} w_j \boldsymbol{x}_j}{\sum_{i=1}^{K} \sum_{j \in N_i} w_j}$ is the overall weighted global mean and $W_i = \sum_{j \in N_i} w_j$ is the total weight of samples in cluster i. The simultaneous minimization of within-cluster scatter and maximization of between cluster scatter can be achieved by finding the transformation G that maximizes the

quantity

$$\mathrm{Tr}\left((G^T \boldsymbol{S}_W^w G)^{-1} G^T \boldsymbol{S}_B^w G\right). \tag{4.11}$$

This maximization can be achieved through an eigenvalue/eigenvector decomposition but such a procedure is only applicable when $\boldsymbol{S}_W^w$ is non-singular. The LDA method was reformulated in terms of the generalized singular value decomposition (SVD) [80] extending the applicability of the method to singular $\boldsymbol{S}_W^w$ matrices such as those encountered when using particle positions.

We have implemented this modified approach in a WeightedLDA python package [154]. The result of an wLDA procedure on two labeled states will be a vector, $\boldsymbol{v}$, of coefficients that best separate the two states. These coefficients can be used directly for sampling within our PLUMED `sizeshape` module.

4.2.4 Enhanced sampling with LDA coordinates

The LDA coordinate used here is a dot product of the vector $\boldsymbol{v}$ with the atomic coordinates $\boldsymbol{x} - \boldsymbol{\mu}$ ans it is given by [54],

$$l(\boldsymbol{x}) = \boldsymbol{v} \cdot \left(\boldsymbol{R} \cdot (\boldsymbol{x}(t) - \vec{\xi}(t)) - \boldsymbol{\mu}\right) \tag{4.12}$$

To compute the value of the LDA coordinate l on the fly, we first translate $\boldsymbol{x(t)}$ by $\vec{\xi}(t) = \frac{1}{N}\sum_{i=1}^{N} \vec{x}_i(t) - \frac{1}{N}\sum_{i=1}^{N} \vec{\mu}_i(t)$, the difference in the geometric mean of the current frame and that of the reference configuration. Then, we compute $\boldsymbol{R}(t)$, the rotation matrix which minimizes the Mahalanobis difference between $\boldsymbol{x(t)} - \vec{\xi}$ and $\boldsymbol{\mu}$, for a given Σ, as described in Ref. [75].

Enhanced sampling simulations on LDA coordinates were performed using Well-tempered Metadynamics (WT-MetaD), and On the Fly Probability Enhanced Sampling-Metadynamics (OPES-MetaD) as implemented in `PLUMED` [25, 81–83].

WT-MetaD works by adding a bias formed from a history dependent sum of progressively

shrinking Gaussian hills [24, 84]. The bias at time t for CV value s_i is given by the expression

$$V(s_i, t) = \sum_{\tau<t} h e^{-V(s_i,\tau)/\Delta T} e^{-\frac{s(\boldsymbol{x}(\tau))-s_i)^2}{2\sigma^2}}, \tag{4.13}$$

where h is the initial hill height, σ sets the width of the Gaussians, and ΔT is an effective sampling temperature for the CVs. Rather than setting ΔT, one typically chooses the bias factor $\gamma = (T + \Delta T)/T$, which sets the smoothness of the sampled distribution [24, 84]. Asymptotically, a free energy surface (FES) can be estimated from the applied bias by $F(s) = -\frac{\gamma}{\gamma-1} V(s, t \to \infty)$ [84, 85] or using a reweighting scheme [84, 86].

OPES-MetaD applies a bias that is based on a kernel density estimate of the probability distribution over the whole space, which is iteratively updated [25, 83]. The bias at time t for CV value s_i is given by the expression

$$V(s_i) = k_B T \left(\frac{\gamma - 1}{\gamma}\right) \log\left(\frac{P_t(s_i)}{Z_t} + \epsilon\right). \tag{4.14}$$

Here in the prefactor, T is the temperature, k_B is Boltzmann's constant, and γ is the bias factor. $P_t(s)$ is the current estimate of the probability distribution, Z_t is a normalization factor that comes from integrating over sampled s space. Finally, $\epsilon = \exp\left(\frac{\Delta E}{k_B T}\frac{\gamma}{\gamma-1}\right)$ is a regularization constant that ensures the maximum bias that can be applied is ΔE.

As in WT-MetaD, $F(s)$ can be directly estimated from $V(s)$ by $F(s) \approx -\frac{\gamma}{\gamma-1} V(s)$ or through a reweighting scheme [25]. Details of the sampling parameters used for each system are given in Section 4.5.

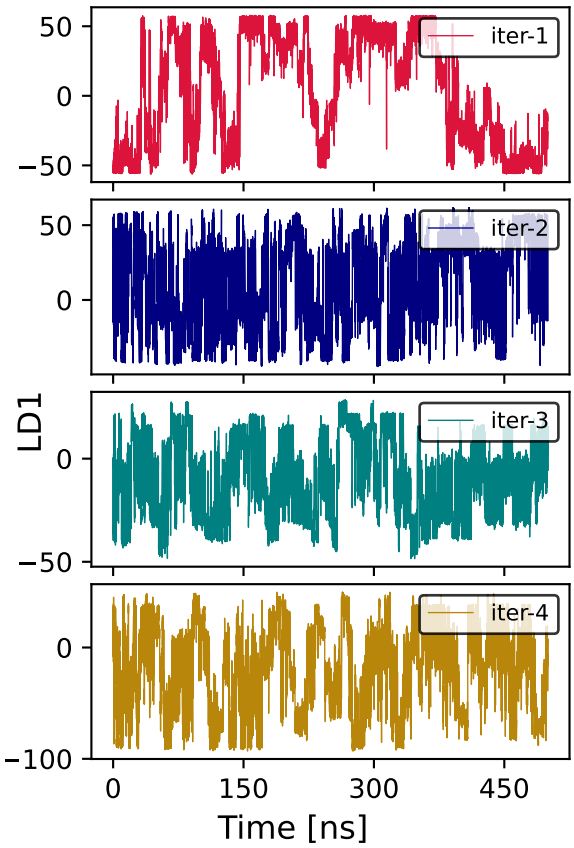

Figure 4.2: Time dependence of LD1 coordinates for $(Aib)_9$ iterations. 500 ns of data were used for clustering and training of the next LD coordinate.

4.3 Results and Discussion

4.3.1 Performing iterations on $(Aib)_9$

$(Aib)_9$ is a nine residue peptide formed from the achiral α-aminoisobutyryl that exhibits two well defined metastable states: left- and right-handed α-helices. Due to the symmetry inherent in a helix made of achiral building blocks, both states must have equal statistical likelihood. In work by us and others [54, 74, 93], this symmetry was leveraged to benchmark sampling and clustering methods. Here, we applied the proposed iterative scheme on this system to assess the improvement of the reaction coordinate over successive iterations.

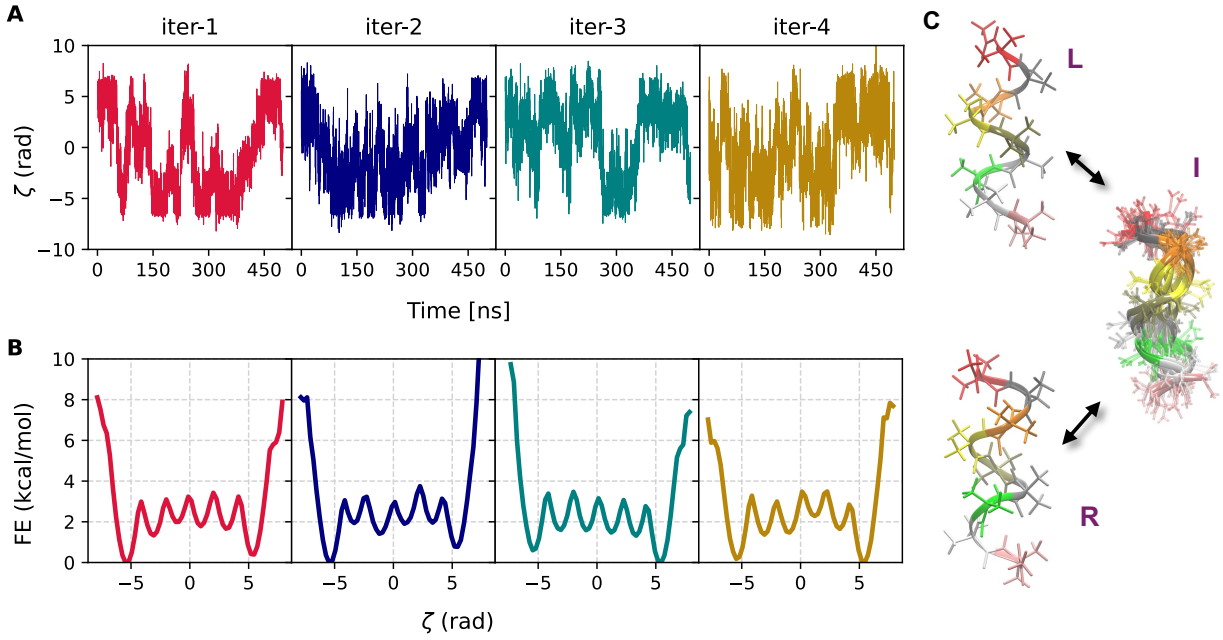

Figure 4.3: $(Aib)_9$ iteration results. (**A**) Trajectory of the physically motivated ζ with time for successive iterations, (**B**) Free energy vs. ζ for successive iterations estimated after 500 ns of MD, and that same data was used to perform the iteration. Each iteration shows transitions between left and right handed states, but iteration four shows the most transitions per unit time and the most symmetric free energy profile from the limited sampling, (**C**) Representative structures from iteration 1, showing transitions from left- (L) to right- (R) helical states of of $(Aib)_9$. Intermediate (I) shows representative snapshots of configurations having $\zeta \approx 0$.

The iterative process begins with an initial linear discriminant (LD1) coordinate derived from two short molecular dynamics (MD) simulations starting from the left- and right-handed states. This is followed by a WT-MetaD simulation biasing this coordinate (for details, see Ref. [54]). The LD coordinate exhibited several transitions between extreme values representing the left and right helical states (Figure 4.2, top) within 500 ns. We subsequently performed three more iterations, with each WT-MetaD simulation also running for 500ns (Figure 4.2).

To perform iterations, we scan over possible numbers of clusters, and compute the log-likelihood for each clustering as shown in Figure 4.7. These data were used to pick the "best" number of clusters as described in Ref. [75]. Figure 4.8 displays the calculated Bhattacharyya distances for all

newly identified clusters relative to the initial states, for the last three iterations (see Section 4.7 for details). The two states nearest to either the left or right side are selected to construct a new weighted LD1 coordinate for the subsequent iteration. The magnitudes of particle displacement vectors acting on individual atoms for all LD1 coordinates are depicted in Figure 4.9.

From these data, we are also able to compute FE profiles along each LD coordinate, as shown in Figure 4.10. Because the coordinate is changing each time, and we do not have a long unbiased reference data set to check convergence, we therefore focus on a previously defined helicity coordinate, ζ, that is the sum of the five central ϕ dihedral angles [74]. Values of $\zeta \sim -5$ and $\zeta \sim +5$ correspond to the right- and left-handed helices, respectively. This CV serves as a consistent reference coordinate to track state transitions and assess convergence of the FES along it (Figure 4.3).

Figure 4.3 illustrates the transitions along ζ during WT-MetaD simulations across all four iterations, along with the corresponding one-dimensional reweighted free energy profiles. The first coordinate is highly sensitive to the application of bias forces, as previously discussed in Ref. [54], meaning that gentle biasing had to be applied to prevent "crashing" due to rapid changes in forces. This resulted in relatively slow sampling of the configurational space. In contrast, here we observed that the CVs obtained in subsequent iterations are substantially more stable and effective in facilitating extensive sampling of the free energy surface (FES) when employed with higher hill heights and bias factors. For the specific values of the MetaD parameters used, refer to Section 4.5. Notably, state transitions increase significantly from the second to the fourth iteration with consistent MetaD parameters.

To test whether free energy profiles along each coordinate would eventually converge, we extended each simulation to 1.5μs (Figure 4.11). Figure 4.12 displays ζ fluctuations over time and show that in this time period, all CVs exhibit a FE profile with left and right states having equal free energy minima within 0.5 kcal/mol.

As a final experiment on this system, we explored the effect of enforcing equal contributions

from the two states when constructing the weighted LD1 coordinate (i.e., assigning each state a combined sample weight of one). This approach was tested for iteration 2 using biased data from the first enhanced sampling simulation. The resulting coordinate performs comparably to the original, as shown in Figure 4.13. These findings demonstrate that iterative refinement, utilizing enhanced sampling data from each step, systematically improves reaction coordinates for $(Aib)_9$, enhancing the exploration of its conformational landscape.

4.3.2 Performing iterations on HP35

We previously applied our shapeGMM clustering approach on a 305 μs long MD trajectory of the fast-folding Nle/Nle mutant of HP35, obtained from the D.E. Shaw Research Group. For our analysis, we selected a six-state representation of the system, which provides an interpretable depiction of the folding and unfolding process. Details of the clustering methodology and cross-validation are discussed in Ref. [75]. The six-state model was trained using 25,000 frames sampled from a dataset of approximately 1.5 million frames. In our subsequent study, we demonstrated that a single folding/unfolding coordinate could be derived by performing LDA on frames assigned to the folded and unfolded states from this six-state representation [54]. Remarkably, this coordinate—trained exclusively on two states—was sufficient to characterize transitions between the folded and unfolded states through physically meaningful configurations. Moreover, it proved to be an effective sampling coordinate when biased in OPES-MetaD simulations [54].

Here, we have implemented the proposed iteration scheme on the system to assess the effectiveness of our approach. The first iteration aligns with the methodology employed in our prior work. Following the procedure outlined in Section 4.2.1, we conducted a total of three iterations. In the second and third iterations only 2.5 μs, 1.5 μs of biased data from the previous runs, respectively, were used to train the new wLDA coordinates. The resulting cluster scan for each training iteration are illustrated in Figure 4.14. For the second iteration, the training process utilized 44,000 samples, with an additional ~5,000 samples reserved for cross-validation. In

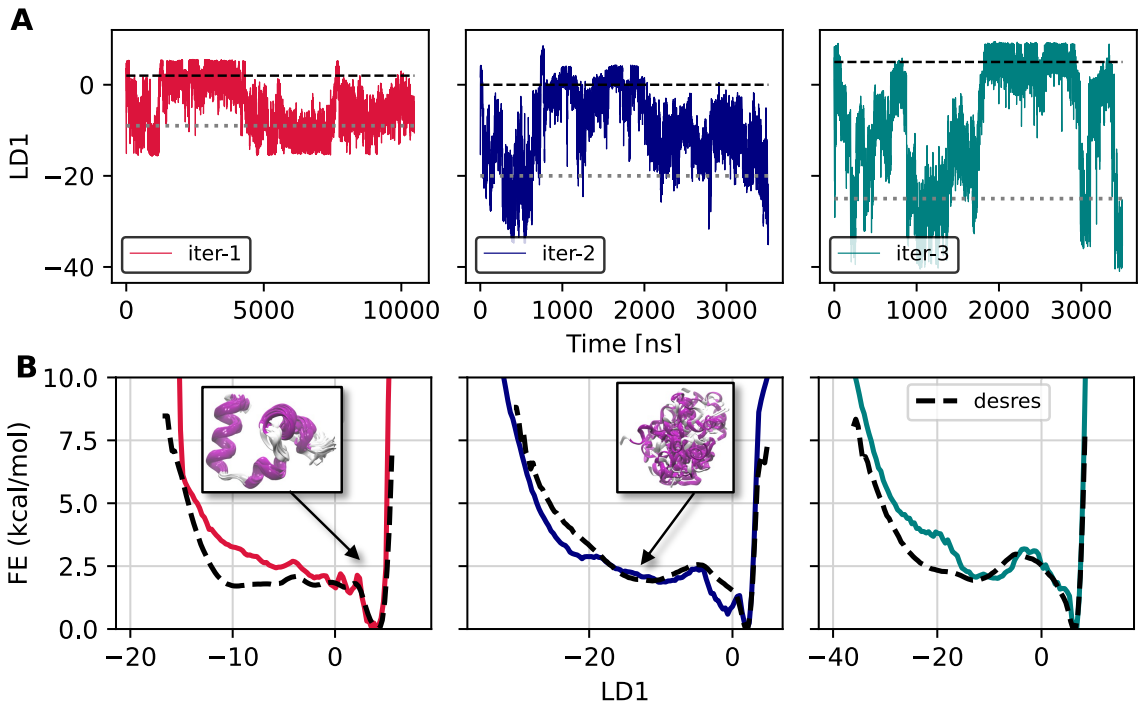

Figure 4.4: HP35 iteration results. (**A**) Fluctuations of LD coordinates with time from extended OPES-MetaD runs for HP35. Horizontal black dashed and grey dotted lines in each case indicate the approximate locations for the folded and unfolded states respectively. The simulations for iteration 1 ran for 10 μs and remaining two ran for approximately 3.5 μs. (**B**) FE as a function of LD1 for successive iterations. The black dashed line represents the unbiased free energy estimate derived from D.E. Shaw Research data [89]. The improved alignment in iteration 2 and 3 indicates better convergence and enhanced sampling efficiency. The insets highlight representative structures of the folded and unfolded clusters (both taken from iteration 2), illustrating the conformational changes during the transition. Protein conformations are colored according to their secondary structure.

the final iteration, the dataset was expanded to 90,000 training samples, supplemented by 10,000 samples for cross-validation. In each iteration, we computed the Bhattacharyya distance between the newly generated clusters and our predefined folded and unfolded states (see Section 4.7). The resulting D_B data, presented in Figure 4.15, quantifies the similarity between the two clusters.

The clusters most closely resembling either the folded or unfolded states are selected. The weighted LDA coordinates derived between the new states at each iteration differ from one another, and the coefficients of the LD1 vectors are illustrated in Figure 4.16. The variation of the LD1 coordinates from OPES-MetaD simulations, performed for every iteration and the corre-

sponding free energy (FE) profiles computed along them are displayed in Figure 4.17. Additionally, Figure 4.18 shows the reweighted 2D free energy surface (FES) projected onto the RMSD space, calculated using those biased simulations.

To assess free energy convergence, each OPES-MetaD simulation was extended, and Figure 4.4 presents the time dependence of LD1 coordinates and 1D FE profiles obtained along them from the extended simulations for all iterations. While the first simulation ran for approximately 10 μs, we achieved significant convergence to the reference free energy (FE) in the last two iterations within just 3.5 μs. This suggests a notable improvement in the effectiveness of the new coordinates used in iterations 2 and 3. To further evaluate this improvement, we computed the 2D free energy surfaces (FES) projected along the RMSD coordinates relative to Helices 1 and 3 for both simulations, as shown in Figure 4.5. The results clearly demonstrate that the new wLDA coordinates served as a more efficient sampling coordinate. The system was able to explore a broader region of the free energy landscape, showing strong agreement with the reference FES derived from D.E. Shaw Research data (depicted as contour lines).

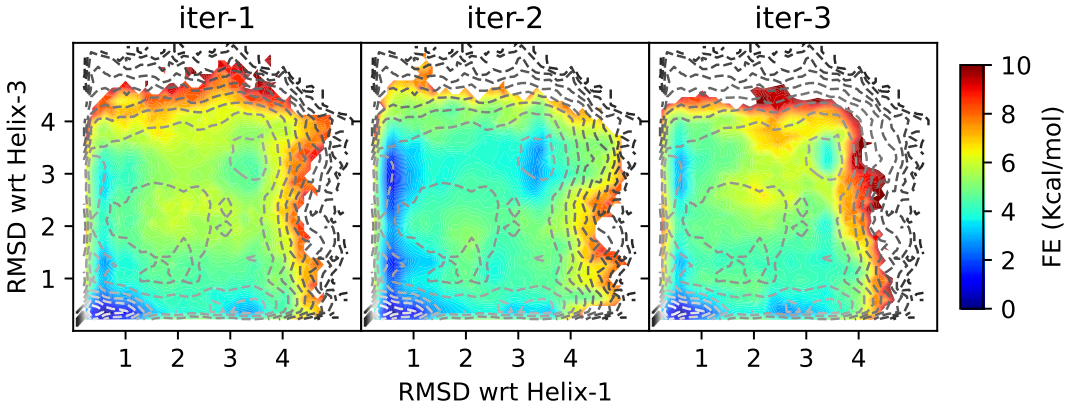

Figure 4.5: Convergence of FES across multiple iterations of HP35. 2D FES from three iterations, projected along the RMSD with respect to Helices 1 and 3. The color scale represents free energy in kcal/mol. Contour lines indicate the reference free energy estimate derived from D.E. Shaw Research data [89]. The FES from iteration 2 and 3 demonstrates improved sampling, capturing a broader region of the free energy landscape with better agreement to the reference.

4.4 Conclusions and Outlook

Our results demonstrate that LDA coordinates based on atomic positions can be iteratively improved using data generated by Metadynamics-like sampling. Iteration improves both the stability and sampling efficiency of the resulting CV. We note that these two effects are coupled, in that improved stability of the CV also allows us to bias more quickly, which improves sampling efficiency. Yet in our studies, we also find this is not the only effect, and the CV also allows better exploration even when using a similarly gentle MetaD bias to that from earlier iterations.

We speculate that this results from a better estimate of the positional covariance matrix around each metastable state, which produces a more smooth transition between the two targeted states, however this has been difficult to prove so far and requires further investigation.

Although our results are promising, some challenges and questions remain. One question in any such approach is: how long should each stage of the iteration be? If a stage of the iteration were run exhaustively to convergence, then there would be no need to iterate to produce a more efficient coordinate. Our results show for these examples that it is possible to improve the coordinate by using enough sampling time to have one to two round trip visits to each state. However, it remains to be investigated whether that generalizes to more challenging systems, and also whether it produces better results than running using the first CV for as long as all of the iterations combined. We believe that there is an actual improvement in CV quality that results in better estimates, for example in Figure 4.5, where the target unfolded state at the top right is barely populated even after 10 microseconds of sampling for iteration one, but is properly given weight in a much shorter simulation on the second iteration.

Going forward, we would like to build on this approach for sampling more challenging systems. When going to a larger system, we believe based on our experience that a single linear coordinate will not be sufficient to capture all the slow degrees of freedom when transitioning between two states. We therefore would like to investigate combining iterative improvement of

one posLDA CV that separates the two states of interest, with other CVs that promote exploration of other large domain motions. We also would like to investigate cases that include metastable intermediates, to see whether one single posLDA between the end states is a good CV for sampling, or whether we should in fact combine multiple posLDA CVs defined pairwise between metastable states, which would allow us to better sample from e.g. a starting state to an intermediate and then from the intermediate to a final state via a more physically realistic route.

4.5 Simulation Details

All simulations were performed using GROMACS 2020.4 [102] with PLUMED 2.9.0-dev [81, 82]. All analysis scripts, jupyter-notebooks and PLUMED input files used in the study are currently available in our paper's GitHub repository https://github.com/hocky-research-group/Sasmal_posLDA_iteration, and will also be available on Zenodo and PLUMED-Nest [82] on publication.

$(Aib)_9$ Simulations

Equilibrated inputs for $(Aib)_9$ were provided by the authors of Ref. [74]. In brief, simulations using the CHARMM36m forcefield and TIP3P water [104]. MD simulations are performed in NPT with a 2 fs timestep at $T = 400K$. The MetaD parameters used for first iteration were HEIGHT=0.005, BIASFACTOR=2, SIGMA=0.43, PACE=500. For all three remaining iterations we used, HEIGHT=0.70, BIASFACTOR=8, SIGMA=0.55 and PACE=500 and a multiple time step STRIDE for biasing of 2 [103]. Quadratic upper and lower walls were applied $\sim \pm 10.0$ of maximum and minimum value for each LD1 coordinate respectively, with a bias coefficient of 125 kcal/mol/Å^2. Complete details are provided in `PLUMED` input files on GitHub.

HP35 Simulations

A 305 μs all-atom simulation of Nle/Nle HP35 at $T = 360K$ from Piana et al.[89] was analyzed. The simulation was performed using the Amber ff99SB*-ILDN force field and TIP3P water model. In that simulation, protein configurations were saved every 200 ps, for a total of ~1.5M frames. For our simulations, we solvate and equilibrate a fresh system using the same forcefield at 40mM NaCl. Minimization and equilibration are performed using a standard protocol[1], at which point NPT simulations are initiated at $T = 360K$. mdp files for all steps of this procedure and the topology files are all available in the GitHub of our previous work [54]. All the OPES-MetaD simulations are performed with $\gamma = 8$, $\Delta E = 10$ kcal/mol, pace of 500 steps, and a biasing multiple time step [103] stride of 2. Quadratic walls were applied for each LD1 coordinate, specific to its range between upper and lower limits, with a bias coefficient of 125 kcal/mol/Å^2.

4.6 From Local Contributions to Global Bias

In this section, we describe an enhanced sampling strategy that integrates ShapeGMM clustering in size-and-shape space with a dynamic biasing approach. The method constructs a global bias potential along selected CVs by a nonlinear combination of local contributions from different metastable states, weighted by the posterior probability of the current state [78]. First, we perform ShapeGMM clustering to obtain the conformational ensembles embedded in high dimensional size-and-shape space. The probability of configurational space is approximated as -

$$P(\boldsymbol{x}) = \phi_0 + \sum_{j=1}^{K} \phi_j N(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \Sigma_j), \tag{4.15}$$

where the sum is over the K Gaussian mixture components, ϕ_j is the weight of component j, and $N(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \Sigma_j)$ is a normalized multivariate Gaussian given by (4.3). The additional term ϕ_0 in

[1] http://www.mdtutorials.com/gmx/lysozyme/index.html

(4.15), is a regularization parameter representing probability of not belonging to any metastable states. The optimal value of ϕ_0 is selected following the procedure described in Ref. [78]. The posterior probability is given as,

$$\gamma_j(\boldsymbol{x}) = \frac{\hat{\phi}_j N(\boldsymbol{x} \mid \hat{\boldsymbol{\mu}}_j, \hat{\Sigma}_j)}{\phi_0 + \sum_{j=1}^{K} \hat{\phi}_j N(\boldsymbol{x} \mid \hat{\boldsymbol{\mu}}_j, \hat{\Sigma}_j)} \tag{4.16}$$

To further reduce the dimensionality and characterize the states in a lower dimensional space, we use Mahalanobis distances relative to each cluster's reference structure or Principal Component Analysis (PCA). A well-tempered metadynamics (WT-MetaD)-like bias is then constructed in this reduced CV space by nonlinearly combining local bias contributions from each cluster. The total bias potential as a function of CVs, $\boldsymbol{s}(\boldsymbol{x})$ at time t is given as [78],

$$V(\boldsymbol{s}, t) = v_0(\boldsymbol{s}, t)\ \gamma_0(\boldsymbol{s}) + \sum_{j=1}^{K} v_j(\boldsymbol{s}, t)\ \gamma_j(\boldsymbol{s}) \tag{4.17}$$

$v_j(\boldsymbol{s}, t)$ is the local bias contribution from the j^{th} cluster,

$$v_j(\boldsymbol{s}, t) = h \sum_{t'<t} e^{-V(\boldsymbol{s}(t'),t')/\Delta T} e^{-\frac{(\boldsymbol{s}(\boldsymbol{x})-s(t'))^2}{2\sigma^2}} \times \frac{\gamma_j(\boldsymbol{s})}{\sum_{j=0}^{K} \gamma_j(\boldsymbol{s})^2}. \tag{4.18}$$

Here h is the initial hill height, σ sets the width of the gaussians, and ΔT is an effective sampling temperature for the CVs, related to bias factor $\gamma = (T + \Delta T)/T$, which sets the smoothness of the sampled distribution. $\gamma_0(\boldsymbol{s}) = \frac{\phi_0}{\phi_0 + \sum_{j=1}^{K} \phi_j N(\boldsymbol{x}|\boldsymbol{\mu}_j, \Sigma_j)}$ refers to the probability of being in background basin (not associated with any states), while $v_0(\boldsymbol{s}, t) = h \sum_{t'<t} e^{-V(\boldsymbol{s}(t'),t')/\Delta T} \times \frac{\gamma_0(\boldsymbol{s})}{\sum_{j=0}^{K} \gamma_j(\boldsymbol{s})^2}$ corresponds to the bias potential experienced in this region. Since not all physically meaningful clusters may be identified initially, the algorithm can be run iteratively, using reweighted samples from biased simulations at each step withframe-weighted ShapeGMM to identify find new clusters and refine the existing ones. The free energy landscape can be estimated using iterative trajectory reweighting scheme (ITRE) method[128].

We tested this approach on alanine dipeptide in vacuum without iterative refinement. Two short 20ns MD simulations were initiated from $C_{7\text{eq}}$ and $C_{7\text{ax}}$ states, followed by 2-state ShapeGMM clustering on the combined trajectory. PCA was performed separately for each cluster using

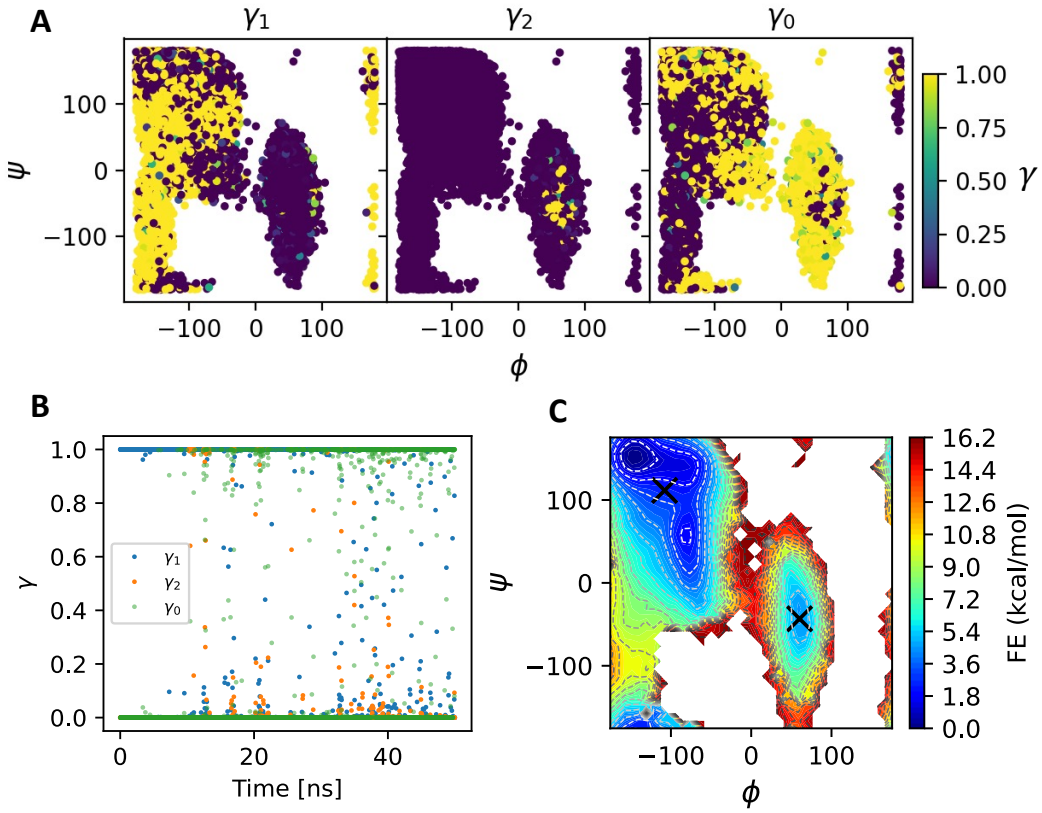

Figure 4.6: Alanine Dipeptide: Results from nonlinear combination of local biases. (**A**) 2D scatter plot of configurations projected onto dihedral angles colored by posterior probabilities γ_j. Values of $\gamma_j \approx 1$ indicate the high probability of belonging to $C_{7\text{eq}}$ or $C_{7\text{ax}}$ states, while $\gamma_0 \approx 1$ corresponds to transitions regions. (**B**) Time evolution of values of γ_j showing transitions between states via intermediate regions of high energy. (**C**) Reweighted FES along dihedral angles computed using the ITRE method [128]. Crosses mark the centers of the two metastable states.

aligned atomic coordinates, with the first two principal components used as CVs:

$$s_{cj}(\boldsymbol{x}, t) = \boldsymbol{v}_c.(\boldsymbol{R}.(\boldsymbol{x}(t) - \vec{\zeta}(t)) - \boldsymbol{\mu}_j); \quad c = 1, 2 \tag{4.19}$$

where $\boldsymbol{\mu}_j$ is the mean of j^{th} cluster, $\boldsymbol{R}$ is the optimal rotation matrix that minimizes separation

between current configuration and cluster mean and $\vec{\zeta}(t)$ is the translational alignment. v_c is a list of normalized coefficients for the particular principal component. This yielded a total of $j \times 2$ CVs. A 50 ns simulation was then performed using the bias potential defined in Eqs. (4.17) and (4.18). In order to do this, we introduced a new bias type in PLUMED, which was implemented by Prof. Gareth Tribello in his "hack-the-tree" branch of PLUMED. The codes for computing this bias, are available upon request[2]. The results, shown in Figure 4.6, demonstrate efficient sampling between metastable states. Figure 4.6(A) illustrates the sampled configurations colored by their γ values, where $\gamma_j \approx 1$ indicates high probability of belonging to the j^{th} cluster and $\gamma_0 \approx 1$ corresponds to transition regions. Figure 4.6(B) tracks the time evolution of γ values as the system transitions between states. Figure 4.6(C) displays the reweighted FES along dihedral angles, confirming efficient sampling of rare transitions.

While this method shows promise, challenges remain in selecting an optimal ϕ_0, particularly due to the asymmetric shapes of clusters in high-dimensional space, which can lead to configurations being incorrectly assigned to the background region. Additionally, future work should explore couple of directions- iterative refinement to dynamically update clusters during sampling, alternative descriptors beyond PCA for improved state discrimination and optimized bias protocols to enhance sampling efficiency.

4.7 Supplementary Figures

Bhattachrayya Distance

The Bhattacharayya distance is a statistical measure used to quantify the similarity between two probability distributions. It is derived from the Bhattacharyya coefficient which measures the amount of overlap between two distributions. For any two given continuous distributions

[2] https://github.com/SasmalSubarna/maha_atlas_project

$p(x)$ and $q(x)$, the coefficient is defines as -

$$BC(p,q) = \int dx p(x) q(x) \tag{4.20}$$

and the distance D_B is given by,

$$D_B(p,q) = -\ln\left(BC(p,q)\right) \tag{4.21}$$

D_B is a symmetric quantity that means $D_B(p,q) = D_B(q,p)$. It goes to zero for identical distributions and goes to infinity for entirely dissimilar distributions. It is assumed that the compared distributions are well defined and normalized. If the distributions are too different from each other, the distance can be very large.

If $p(x)$, $q(x)$ are multivariate normal distributions such as $p(x) \sim \mathcal{N}(\mu_p, \Sigma_p)$ and $q(x) \sim \mathcal{N}(\mu_q, \Sigma_q)$, then it can be derived to show that $D_B(p,q)$ is given as,

$$\begin{aligned} D_B(p,q) = \frac{1}{8}\left[(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)^T \Sigma^{-1} (\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)\right] \\ + \frac{1}{2}\ln\left[\frac{\det(\Sigma)}{\sqrt{\det(\Sigma_p)\det(\Sigma_q)}}\right] \end{aligned} \tag{4.22}$$

$\boldsymbol{\mu}_p$, $\boldsymbol{\mu}_q$ are the mean vectors corresponding to distributions $p(x)$ and $q(x)$ with covariances Σ_p, Σ_q respectively. And $\Sigma = \frac{\Sigma_p + \Sigma_q}{2}$, is the mean of two covariances. The first term in the Eq.4.22 is the Mahalanobis distance between two distributions that quantifies the difference in their locations. The second term accounts for the difference in the shapes of the two distributions and is a measure of the divergence due to differences in the spreads and orientations.

For this work, we implemented the Bhattacharayya distance metric in the `similarities` module of the `shapeGMMTorch` package [153].

Training curves for $(Aib)_9$ iterations

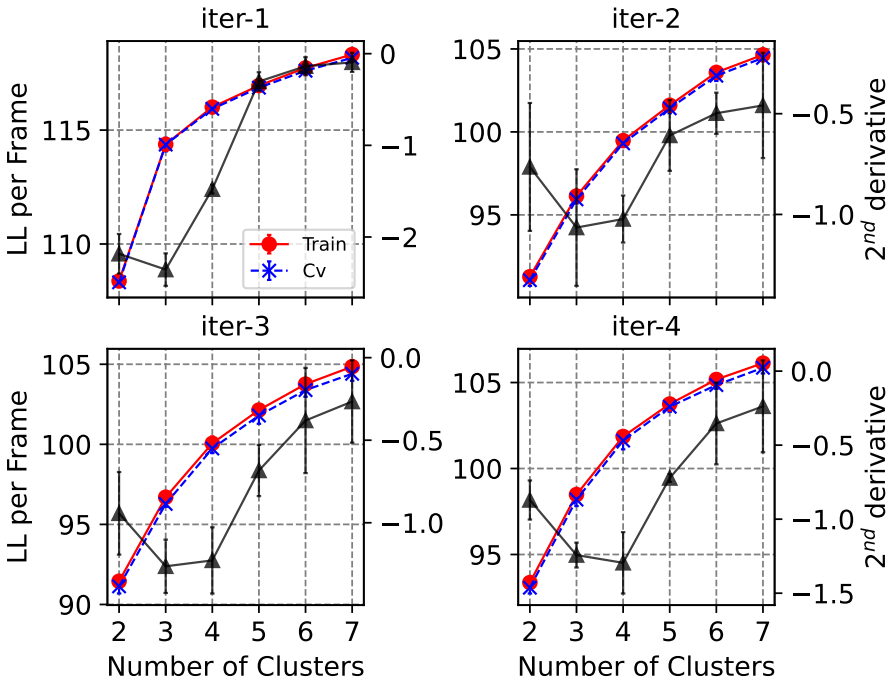

Figure 4.7: Cluster scans from four successive iterations of $(Aib)_9$. Iteration 2,3 and 4 were performed with data from biased simulations, using 90k training samples and 10k samples for cross validation. First iteration was performed with combined data from two short 20ns long MD simulations initiated form both left and right states. In first iteration, we used 20k frames for training along with 20k for cross validation. Black curves represent 2^{nd} derivatives(with error bars) of log likelihood with respect to number of clusters and minimum value indicates an optimal choice for number of clusters.

Bhattacharyya Distances for $(Aib)_9$

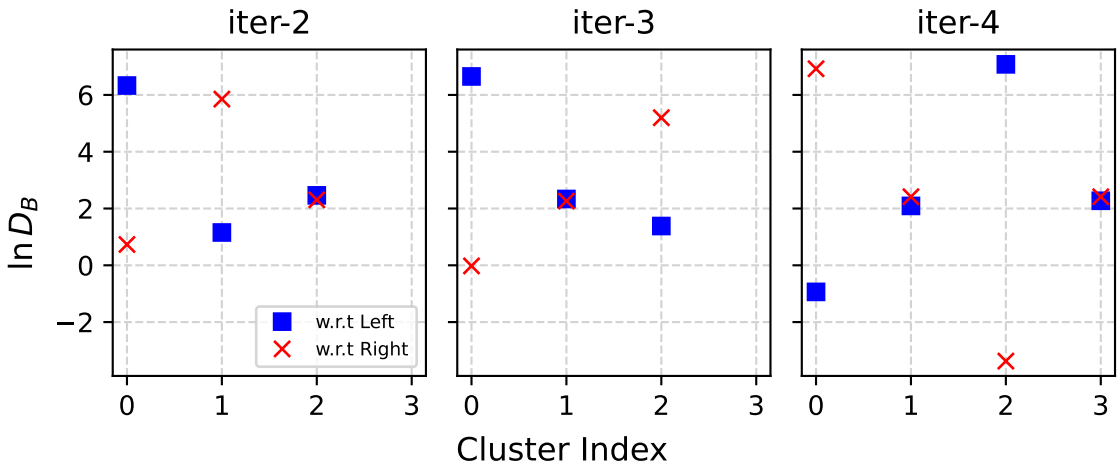

Figure 4.8: Bhattacharyya Distances for $(Aib)_9$. Logarithm of distances are computed for all clusters with respect to initial definitions of left and right helical states at every iteration. It gives a measure of similarity between two multivariate normal distributions that represent a cluster. Any two clusters with lower values of $\ln D_B$ are close to each other and those with higher values are far away from each other. It provides a consistent way of defining new left and right states at every iteration in accordance with initial definitions.

Coefficients of LD coordinates from $(Aib)_9$ iterations

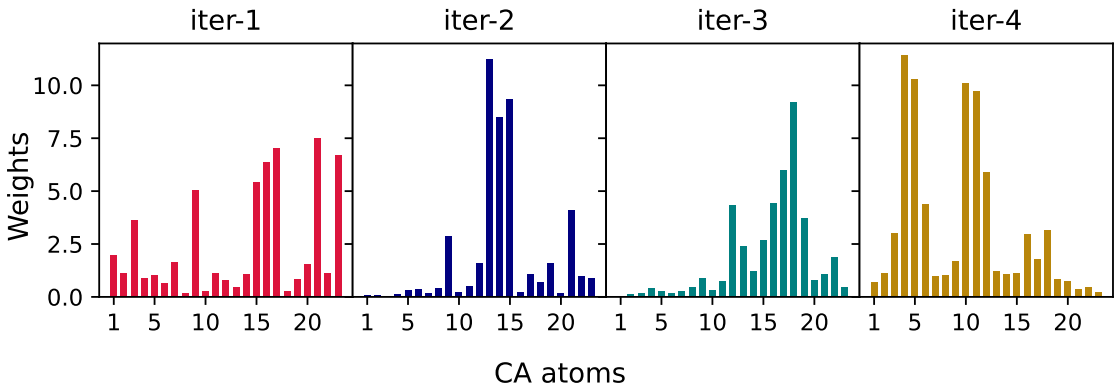

Figure 4.9: Comparing coefficients of LDA coordinates for $(Aib)_9$. Weights shown are the magnitudes of particle displacement vectors acting on each atom from LD1 after each iteration. In case of $(Aib)_9$, cartesian coordinates of total 23 backbone atoms are used to define LD1 coordinate which is a linear combination of $23 \times 3 = 69$ features with 69 real coefficients. Hence, each particle has a displacement vector of 3 components associated with it. In this figure, it shows the magnitude of those vectors. Weights are considered as contributions of different atoms in making the coordinate. The atoms with larger weights have a larger effect when biasing while those with smaller weights contribute less.

FEs vs. LD1 for $(Aib)_9$ iterations

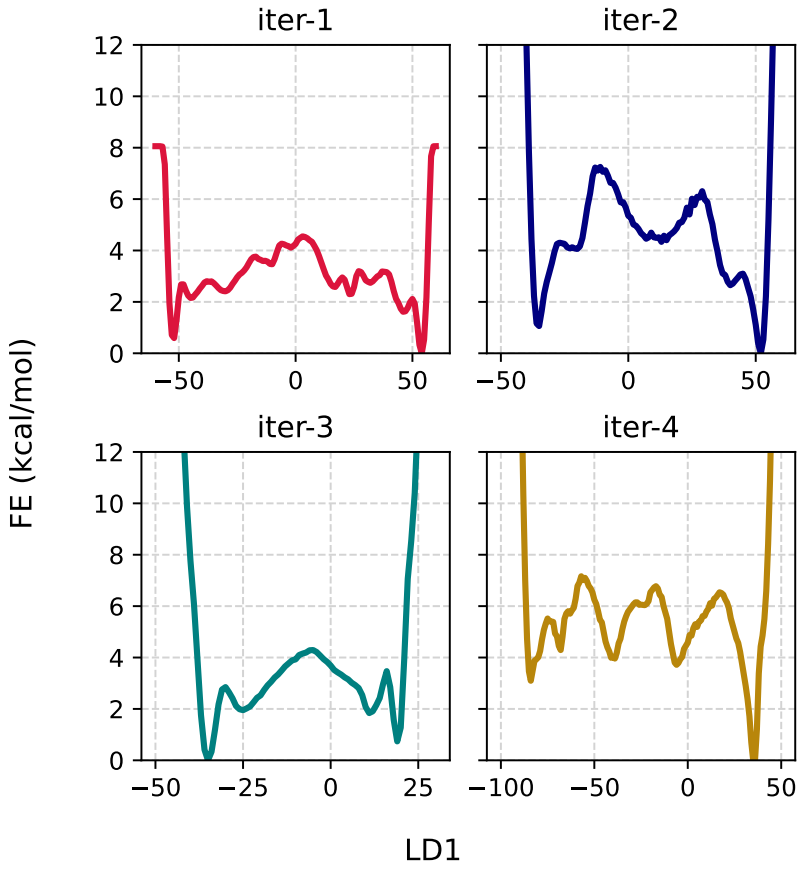

Figure 4.10: FE vs. LD1 obtained from $(Aib)_9$ iterations. Free energies are computed from 500ns long WT-MetaD simulations at each iteration.

FEs and time dependence of LD coordinates from $(Aib)_9$ 1.5μs simulations

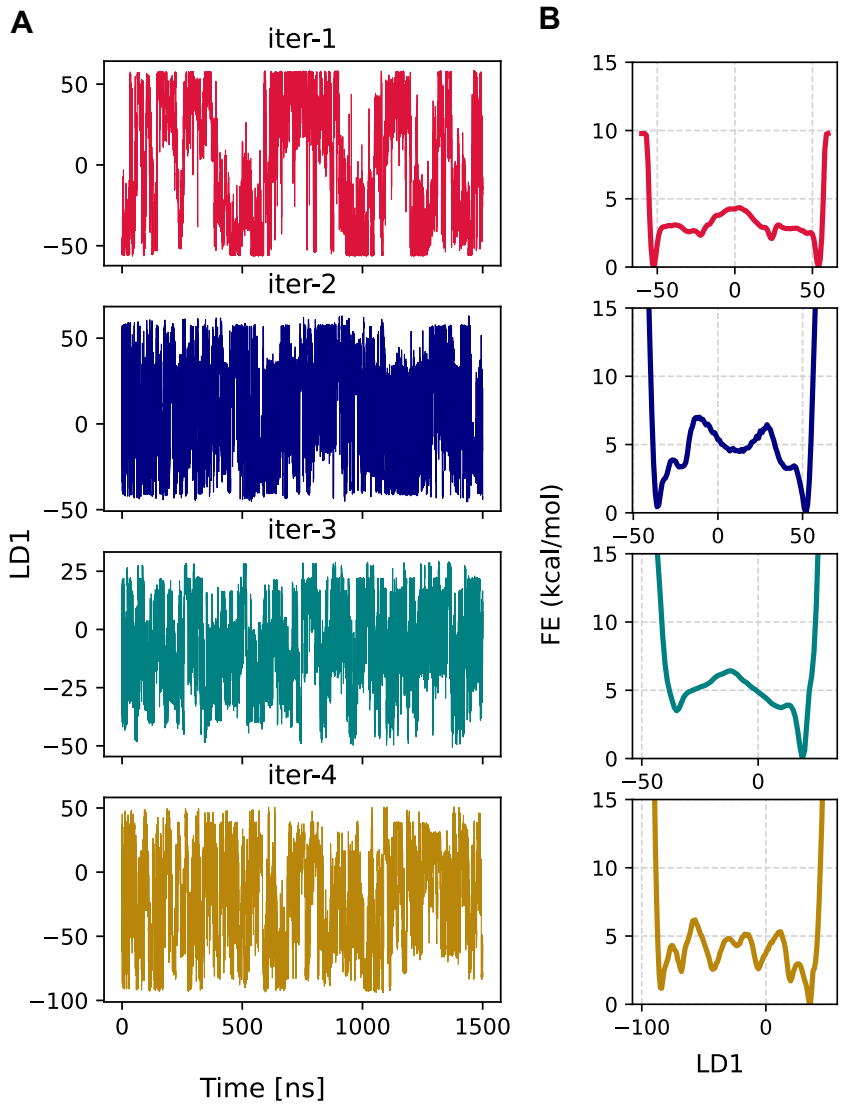

Figure 4.11: Results from 1.5μs simulations of $(Aib)_9$, for LD1 coordinates. Each WT-MetaD simulation from successive iterations were further extended upto 1.5 μs. (**A**) Fluctuations of LD1 with time and (**B**) FE profiles computed along LD1 in each case by summing all Gaussian hills deposited over the course of the simulations.

FEs and time dependence of ζ from $(Aib)_9$ 1.5μs simulations

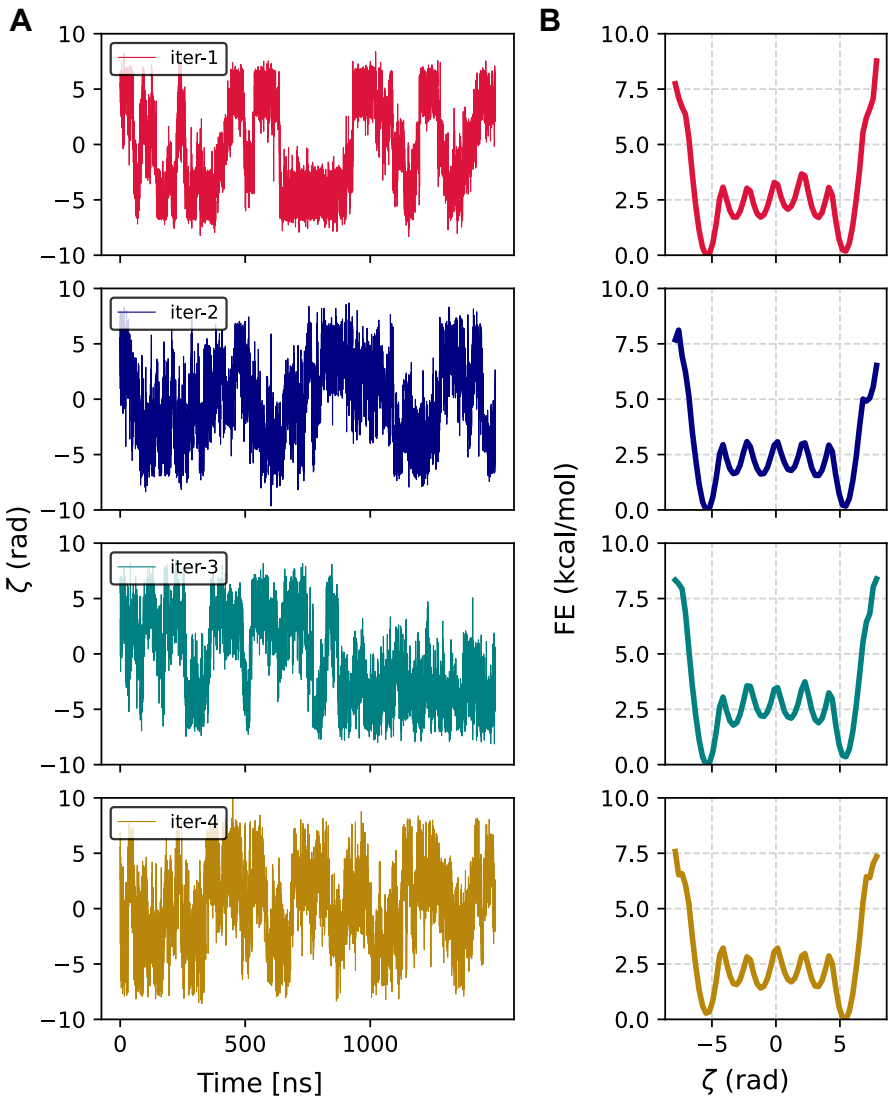

Figure 4.12: Results from 1.5μs simulations of $(Aib)_9$, for ζ coordinate. (**A**) fluctuations of ζ with time and (**B**) converged reweighted free energy profiles computed from 1.5μs long WT-Metad simulations. Efficient sampling between left and right states is observed in all cases. All free energy profiles along ζ are converged and symmetric.

Implementing Equal Weights for left and right helix

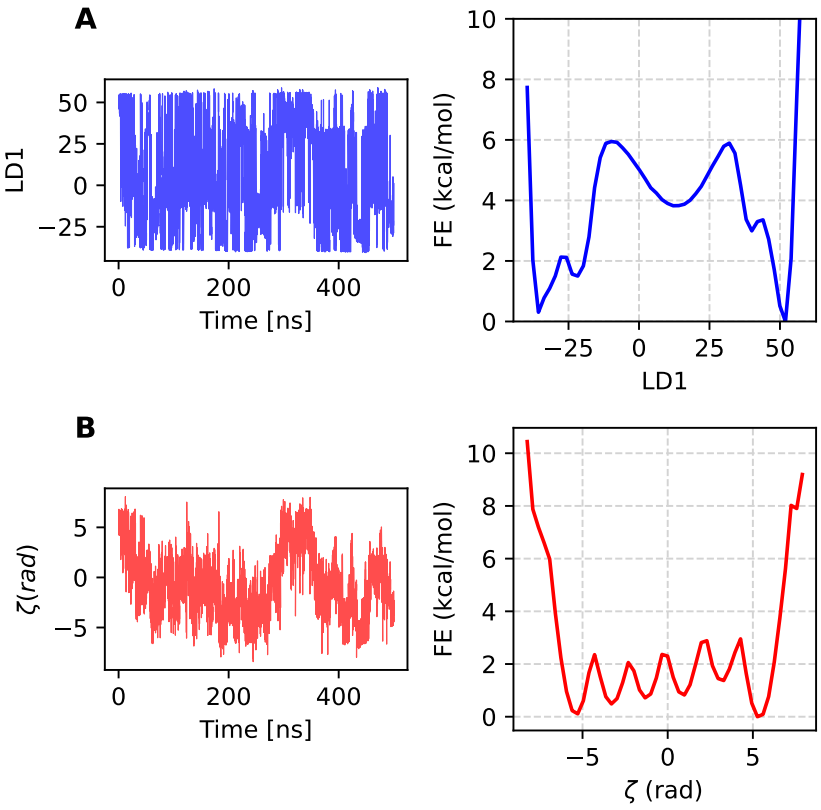

Figure 4.13: Equal weights for left and right helices of $(Aib)_9$. Results from a 500ns long WT-MetaD simulation by biasing LD1 coordinate which is obtained by implementing equal total probability for samples belonging to left and right states. Equal total probability means that sum of weights for all the samples from either left or right state is equal to 1. To test this we used WT-MetaD data from first iteration. After computing the correct weights for samples from biased data, here we normalized the weights separately for left and right states before feeding it into LDA algorithm, so that each state contributes equally to the coordinate. (**A**) LD1 vs. time and free energy profile calculated along LD1 and (**B**) ζ vs. time along with FES along ζ. The MetaD parameters used for this simulation are, HEIGHT=0.01, BF=8, PACE=2000, SIGMA=0.55 and STRIDE=2. Two quadratic walls were applied at LD1=+60.0 and LD1=-60.0 with force constant of 125.0 kcal/mol/Å^2.

Training curves for HP35 iterations

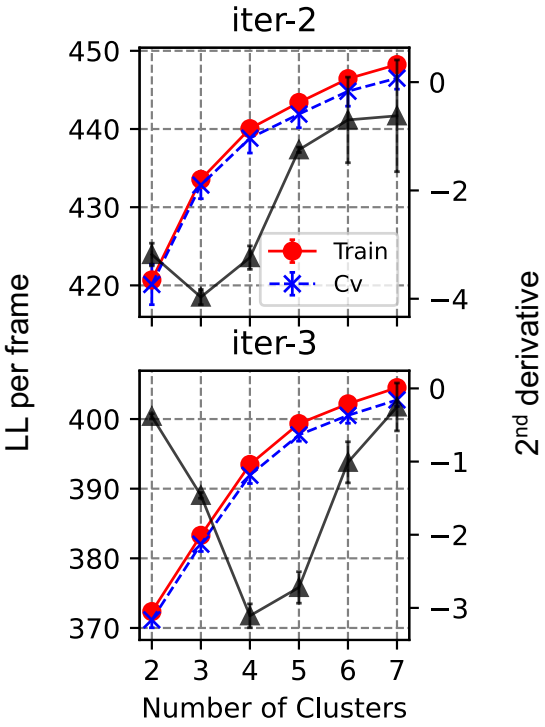

Figure 4.14: Cluster scans for last two iterations of HP35. The first scan (reported in Ref. [75], not shown here) was performed with 305μs long MD simulation trajectory of HP35 provided by D. E. Shaw Research [89]. The second scan was performed with 2.5μs long OPES-MetaD simulation data with 44k frames for training along with ~5k frames for cross validation. The third scan was performed with 90k samples for training and 10k samples for cross validation, using the biased data from previous 1.5μs long OPES-MetaD simulations. Training curves with error bars are shown in red and cross validations curves with error bars are shown in blue. Black curves represent 2^{nd} derivatives(with error bars) of log likelihood with respect to number of clusters and minimum value indicates an optimal choice for number of clusters.

Bhattacharyya Distances for HP35

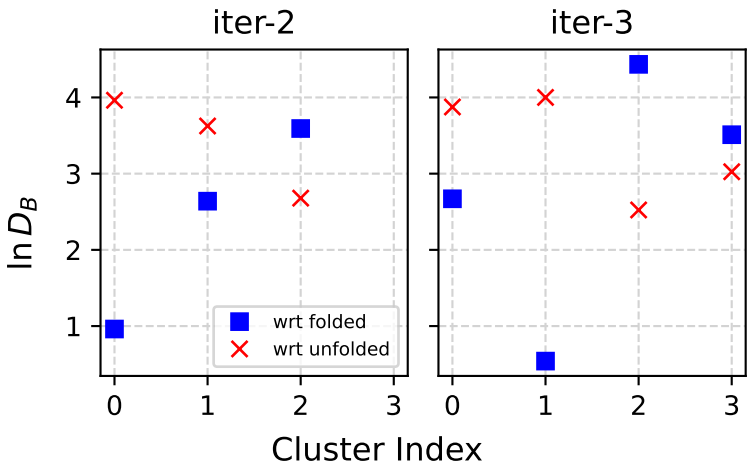

Figure 4.15: Bhattacharyya Distances for HP35. Logarithm of Bhattacharyya distance for all clusters (see Figure 4.14) in our HP35 iterations with respect to initial definitions of folded and unfolded clusters.

Coefficients of LD coordinates from HP35 iterations

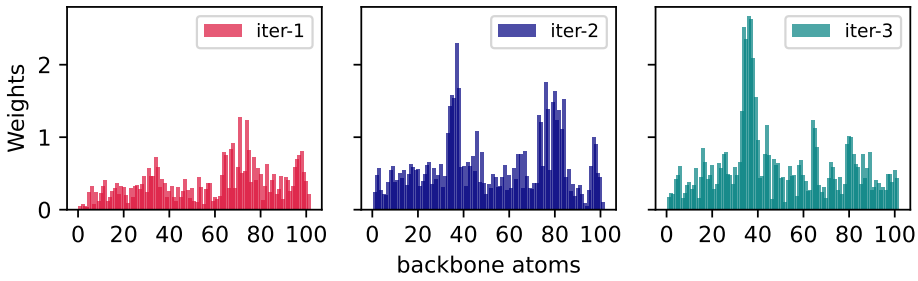

Figure 4.16: LDA weights at each iteration for HP35. Here, input cartesian coordinates consist of 101 backbone atoms, which is a linear combination of $101 \times 3 = 303$ features with 303 real coefficients.

LD1 time dependence and FE vs LD1 for HP35 iterations

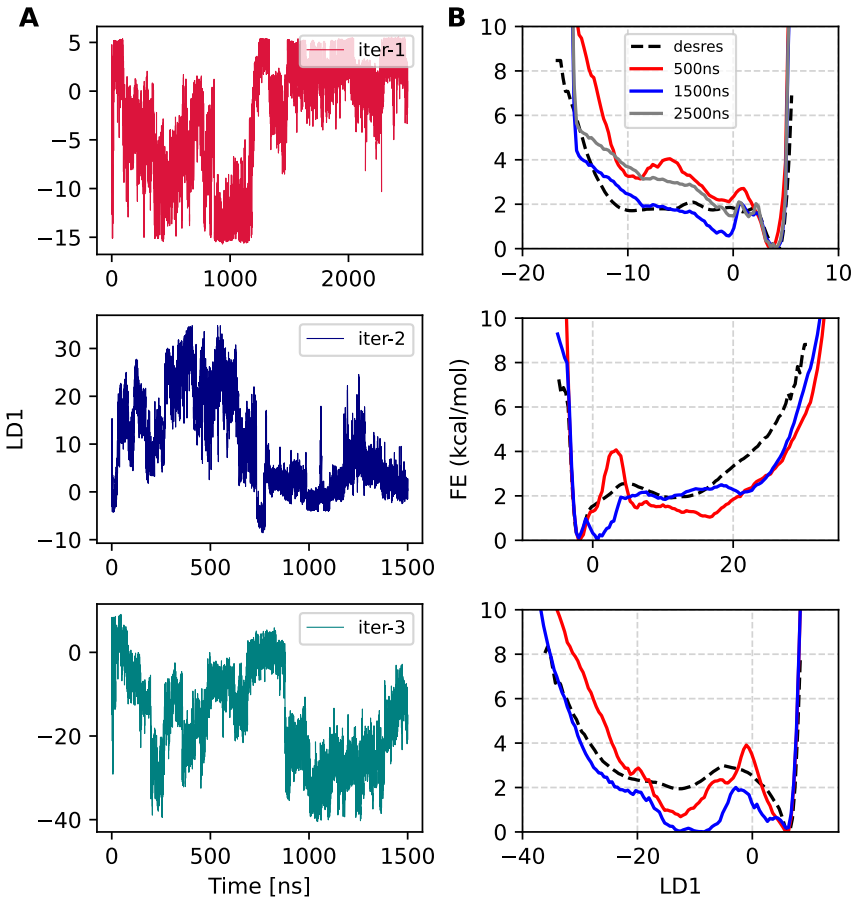

Figure 4.17: Results from HP35 iterations. (**A**) Trajectory of LD1 obtained from OPES-MetaD simulations in three successive iterations of HP35. The first simulation is 2.5μs and the remaining two are 1.5μs long. Note that the coordinate obtained at each iteration is different than others. (**B**) FE profiles computed in each oteration. FE profiles calculated using 500ns, 1500ns and 2500ns long data are shown in red, blue, grey colors respectively. In each case for comparison, we also computed a reference FE using 305μs long unbiased MD simulation of villin, provided by D. E. Shaw Research [89]. The reference FE profiles are shown in black dashed lines.

2D FES on RMSD space from HP35 iterations

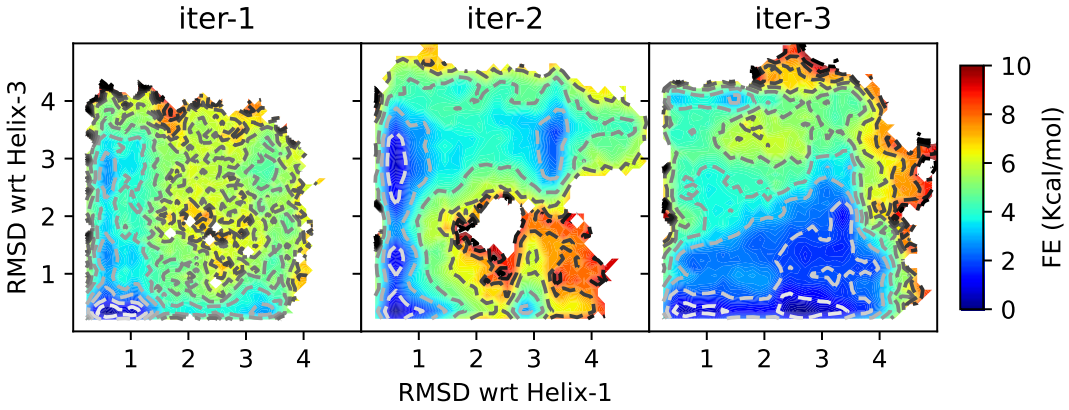

Figure 4.18: Free energy landscapes obtained from HP35 simulations. 2D reweighted FES projected along RMSDs (computed using only backbone atoms) with respect to helix-1 and helix-3. To compute the free energy profiles, OPES-MetaD simulation data generated at each iteration is used. The first one is 2.5μs long and the later two are 1.5μs long only (see Figure 4.17). This also illustrates the input data which is used in an iteration to generate the next wLDA coordinate.

5 Conclusion

In this thesis, we have demonstrated three key contributions. First, we illustrated that applying Linear Discriminant Analysis to atomic positions of two metastable states of a biomolecular system, produces a good collective variable for use in enhanced sampling methods, enabling rapid convergence of the free energy profile. While, the resulting collective variable has proven to be efficient, there is still room for improvements that could be explored, such as using a multi-state LDA approach or combining multiple LDA coordinates to improve sampling efficiency. Another promising direction is iteratively refining the LDA coordinate by incorporating reweighted data from biased simulations to train an improved weighted LDA coordinate. Another important aspect is the use of pairwise LDA coordinates defined for intermediate states of a system, to explore the complete free energy path connecting two end states of the system.

Second, we have introduced the frame-weighted ShapeGMM method, which enables the extraction of unbiased conformational ensembles directly from biased simulations. This is particularly useful when long unbiased MD simulations, capturing multiple transitions between metastable states are unavailable and one has to rely on biased simulations to explore the full conformational landscape. The method produces an estimate of unbiased high-dimensional probability distribution that serves as a generative model, allowing us to draw new samples from metastable basins and compute thermodynamic properties such as configurational entropy and free energy differences between states. Several open questions still remain, including how effectively we can generate physically realistic samples that mimic MD-derived configurations,

whether the quality of the generative model can be improved using sample covariance and uniform alignment instead of model covariance with Kronecker alignment, and whether this method can be extended to compute the binding free energy difference of small molecules in enzyme active sites. We have already begun exploring some of these directions.

Third, we have successfully implemented an iterative approach for optimizing collective variables, demonstrating that iteration enhances sampling efficiency and accelerates convergence. While this method has proven effective overall, several key questions remain unanswered. For instance, its performance in larger systems with numerous degrees of freedom requires further investigation. Additionally, combining the iterative approach with supplementary coordinates to better capture large scale conformational changes could further improve sampling. As an example, to study a drug binding process, a single LDA coordinate can be biased along with a center of mass distance between the active site of the protein and drug molecule to facilitate the binding process and compute binding free energy profiles. Another important aspect is a quantitative analysis of the increased sampling efficiency of the coordinates, which could involve examining reference structures and positional covariances that represent the underlying metastable states. Moreover, our current study primarily focuses on coordinates derived between two states, leaving room for future exploration of multi-state coordinates and their potential advantages. Moving forward, we hope to pursue these open directions and develop improved versions of the methods introduced in this thesis.

Bibliography

[1] Martin Karplus and J.Andrew McCammon. "Molecular dynamics simulations of biomolecules". In: *Nature Structutal Biology* 9 (2002), pp. 646–652. DOI: https://doi.org/10.1038/nsb0902-646.

[2] Hans Frauenfelder, Stephen G. Sligar, and Peter G. Wolynes. "The Energy Landscapes and Motions of Proteins". In: *Science* 254.5038 (1991), pp. 1598–1603. DOI: 10.1126/science.1749933.

[3] Katherine Henzler-Wildman and Dorothee Kern. "Dynamic personalities of proteins". In: *Nature* 450 (2007), pp. 964–972. DOI: https://doi.org/10.1038/nature06522.

[4] Ron O. Dror et al. "Biomolecular Simulation: A Computational Microscope for Molecular Biology". In: *Annual Review of Biophysics* 41.Volume 41, 2012 (2012), pp. 429–452. ISSN: 1936-1238. DOI: https://doi.org/10.1146/annurev-biophys-042910-155245.

[5] Ron O. Dror et al. "Pathway and mechanism of drug binding to G-protein-coupled receptors". In: *Proceedings of the National Academy of Sciences* 108.32 (2011), pp. 13118–13123. DOI: 10.1073/pnas.1104614108.

[6] Outi M. H. Salo-Ahen et al. "Molecular Dynamics Simulations in Drug Discovery and Pharmaceutical Development". In: *Processes* 9.1 (2021). ISSN: 2227-9717. DOI: 10.3390/pr9010071.

[7] Jacob D Durrant and J Andrew McCammon. “Molecular dynamics simulations and drug discovery”. In: *BMC Biology* 9 (2011), p. 71. DOI: https://doi.org/10.1186/1741-7007-9-71.

[8] Marco De Vivo et al. “Role of Molecular Dynamics and Related Methods in Drug Discovery”. In: *Journal of Medicinal Chemistry* 59.9 (2016). PMID: 26807648, pp. 4035–4061. DOI: 10.1021/acs.jmedchem.5b01684.

[9] Paola Gallo et al. “Water: A Tale of Two Liquids”. In: *Chemical Reviews* 116.13 (2016). PMID: 27380438, pp. 7463–7500. DOI: 10.1021/acs.chemrev.5b00750.

[10] Carlos Vega and Jose L. F. Abascal. “Simulating water with rigid non-polarizable models: a general perspective”. In: *Phys. Chem. Chem. Phys.* 13 (44 2011), pp. 19663–19688. DOI: 10.1039/C1CP22168J.

[11] Hiroshi Watanabe, Nobuyasu Ito, and Chin-Kun Hu. “Phase diagram and universality of the Lennard-Jones gas-liquid system”. In: *The Journal of Chemical Physics* 136.20 (May 2012), p. 204102. ISSN: 0021-9606. DOI: 10.1063/1.4720089.

[12] Jing Li Hesam N. Motlagh James O. Wrabl and Vincent J. Hilser. “The ensemble nature of allostery”. In: *Nature* 508 (2014), pp. 331–339. DOI: 10.1038/nature13001.

[13] Jingjing Guo and Huan-Xiang Zhou. “Protein Allostery and Conformational Dynamics”. In: *Chemical Reviews* 116.11 (2016). PMID: 26876046, pp. 6503–6515. DOI: 10.1021/acs.chemrev.5b00590.

[14] K. Gunasekaran, Buyong Ma, and Ruth Nussinov. “Is allostery an intrinsic property of all dynamic proteins?” In: *Proteins: Structure, Function, and Bioinformatics* 57.3 (2004), pp. 433–443. DOI: https://doi.org/10.1002/prot.20232.

[15] Ruth Nussinov and Chung-Jung Tsai. “Allostery in Disease and in Drug Discovery”. In: *Cell* 153.2 (2013), pp. 293–305. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2013.03.034.

[16] Kresten Lindorff-Larsen et al. “How Fast-Folding Proteins Fold”. In: *Science* 334.6055 (2011), pp. 517–520. DOI: 10.1126/science.1208351.

[17] Vijay S. Pande and Daniel S. Rokhsar. “Molecular dynamics simulations of unfolding and refolding of a -hairpin fragment of protein G”. In: *Proceedings of the National Academy of Sciences* 96.16 (1999), pp. 9062–9067. DOI: 10.1073/pnas.96.16.9062.

[18] Ken A. Dill and Justin L. MacCallum. “The Protein-Folding Problem, 50 Years On”. In: *Science* 338.6110 (2012), pp. 1042–1046. DOI: 10.1126/science.1219021.

[19] David E. Shaw et al. “Atomic-Level Characterization of the Structural Dynamics of Proteins”. In: *Science* 330.6002 (2010), pp. 341–346. DOI: 10.1126/science.1187409.

[20] Alessandro Laio and Michele Parrinello. “Escaping free-energy minima”. In: *Proc. Natl. Acad. Sci. U.S.A.* 99.20 (2002), pp. 12562–12566.

[21] Rafael C. Bernardi, Marcelo C.R. Melo, and Klaus Schulten. “Enhanced sampling techniques in molecular dynamics simulations of biological systems”. In: *Biochimica et Biophysica Acta (BBA) - General Subjects* 1850.5 (2015). Recent developments of molecular dynamics, pp. 872–877. ISSN: 0304-4165. DOI: https://doi.org/10.1016/j.bbagen.2014.10.019.

[22] Omar Valsson, Pratyush Tiwary, and Michele Parrinello. “Enhancing Important Fluctuations: Rare Events and Metadynamics from a Conceptual Viewpoint”. In: *Annual Review of Physical Chemistry* 67.Volume 67, 2016 (2016), pp. 159–184. ISSN: 1545-1593. DOI: https://doi.org/10.1146/annurev-physchem-040215-112229.

[23] Glenn M Torrie and John P Valleau. “Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling”. In: *J. Comp. Phys.* 23.2 (1977), pp. 187–199.

[24] Alessandro Barducci, Giovanni Bussi, and Michele Parrinello. “Well-tempered metadynamics: a smoothly converging and tunable free-energy method”. In: *Phys. Rev. Lett.* 100.2 (2008), p. 020603.

[25] Michele Invernizzi and Michele Parrinello. “Rethinking metadynamics: from bias potentials to probability distributions”. In: *J. Phys. Chem. Lett.* 11.7 (2020), pp. 2731–2736.

[26] Jerry B Abrams and Mark E Tuckerman. “Efficient and direct generation of multidimensional free energy surfaces via adiabatic dynamics without coordinate transformations”. In: *J. Phys. Chem. B* 112.49 (2008), pp. 15742–15757.

[27] Luca Maragliano and Eric Vanden-Eijnden. “A temperature accelerated method for sampling free energy and determining reaction pathways in rare events simulations”. In: *Chem. Phys. Lett.* 426.1-3 (2006), pp. 168–175.

[28] Yuji Sugita and Yuko Okamoto. “Replica-exchange molecular dynamics method for protein folding”. In: *Chemical Physics Letters* 314.1 (1999), pp. 141–151. ISSN: 0009-2614. DOI: https://doi.org/10.1016/S0009-2614(99)01123-9.

[29] Pu Liu et al. “Replica exchange with solute tempering: A method for sampling biological systems in explicit water”. In: *Proceedings of the National Academy of Sciences* 102.39 (2005), pp. 13749–13754. DOI: 10.1073/pnas.0506346102.

[30] Peter G Bolhuis et al. “Transition Path Sampling: Throwing Ropes”. In: *Annu. Rev. Phys. Chem* 53 (2002), pp. 291–318.

[31] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[32] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Vol. 344. John Wiley & Sons, 1991.

[33] Matt Bonakdarpour and Matt Stephens. *Introduction to EM: Gaussian Mixture Models*. https://stephens999.github.io/fiveMinuteStats/intro_to_em.html.

[34] Chuong B Do and Serafim Batzoglou. “What is the expectation maximization algorithm?” In: *Nat. Biotech.* 26 (2008), pp. 897–899.

[35] M Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise”. In: AAAI Press, Menlo Park, CA (United States), Dec. 1996.

[36] Ricardo J. G. B. Campello et al. “Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection”. In: *ACM Trans. Knowl. Discov. Data* 10.1 (July 2015). ISSN: 1556-4681. DOI: 10.1145/2733381.

[37] Nina Singhal, Christopher D. Snow, and Vijay S. Pande. “Using path sampling to build better Markovian state models: Predicting the folding rate and mechanism of a tryptophan zipper beta hairpin”. In: *J. Chem. Phys.* 121 (2004), pp. 415–425.

[38] P. Deuflhard et al. “Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains”. In: *Linear Algebra Appl.* 315.1-3 (2000), pp. 39–59. ISSN: 00243795. DOI: 10.1016/S0024-3795(00)00095-1.

[39] Peter Deuflhard and Marcus Weber. “Robust Perron cluster analysis in conformation dynamics”. In: *Linear Algebra Appl.* 398.1-3 (2005), pp. 161–184. ISSN: 00243795. DOI: 10.1016/j.laa.2004.10.026.

[40] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *J. Mach. Learn. Res.* 12 (2011), pp. 2825–2830.

[41] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2019. DOI: https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data.

[42] Peter J. Rousseeuw. “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. ISSN: 0377-0427. DOI: https://doi.org/10.1016/0377-0427(87)90125-7.

[43] L. Molgedey and H. G. Schuster. “Separation of a mixture of independent signals using time delayed correlations”. In: *Phys. Rev. Lett.* 72 (23 June 1994), pp. 3634–3637. DOI: 10.1103/PhysRevLett.72.3634.

[44] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605.

[45] Leland McInnes et al. "UMAP: Uniform Manifold Approximation and Projection". In: *Journal of Open Source Software* 3.29 (2018), p. 861. DOI: 10.21105/joss.00861.

[46] R. R. Coifman et al. "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps". In: *Proceedings of the National Academy of Sciences* 102.21 (2005), pp. 7426–7431. DOI: 10.1073/pnas.0500334102.

[47] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. "A Global Geometric Framework for Nonlinear Dimensionality Reduction". In: *Science* 290.5500 (2000), pp. 2319–2323. DOI: 10.1126/science.290.5500.2319.

[48] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. "Nonlinear Component Analysis as a Kernel Eigenvalue Problem". In: *Neural Computation* 10.5 (July 1998), pp. 1299–1319. ISSN: 0899-7667. DOI: 10.1162/089976698300017467.

[49] Luigi Bonati, Valerio Rizzi, and Michele Parrinello. "Data-driven collective variables for enhanced sampling". In: *J. Chem. Phys. Lett.* 11.8 (2020), pp. 2998–3004.

[50] Luigi Bonati, GiovanniMaria Piccini, and Michele Parrinello. "Deep learning the slow modes for rare events sampling". In: *Proceedings of the National Academy of Sciences* 118.44 (2021), e2113533118.

[51] Dan Mendels et al. "Folding a small protein using harmonic linear discriminant analysis". In: *J. Chem. Phys.* 149.19 (2018), p. 194113.

[52] Dan Mendels, GiovanniMaria Piccini, and Michele Parrinello. "Collective variables from local fluctuations". In: *J. Phys. Chem. Lett.* 9.11 (2018), pp. 2776–2781.

[53] Yann LeCun and Corinna Cortes. "The mnist database of handwritten digits". In: 2005.

[54] Subarna Sasmal, Martin McCullagh, and Glen M. Hocky. "Reaction Coordinates for Conformational Transitions Using Linear Discriminant Analysis on Positions". In: *Journal of Chemical Theory and Computation* 19.14 (2023). PMID: 37130367, pp. 4427–4435. DOI: 10.1021/acs.jctc.3c00051.

[55] Jérôme Hénin et al. "Enhanced Sampling Methods for Molecular Dynamics Simulations [Article v1. 0]". In: *LiveCoMS* 4.1 (2022), pp. 1583–1583.

[56] Ao Ma and Aaron R Dinner. "Automatic method for identifying reaction coordinates in complex systems". In: *J. Phys. Chem. B* 109.14 (2005), pp. 6769–6779.

[57] Behrooz Hashemian, Daniel Millán, and Marino Arroyo. "Modeling and enhanced sampling of molecular systems with smooth and nonlinear data-driven collective variables". In: *J. Chem. Phys.* 139 (2013), p. 214101.

[58] Pratyush Tiwary and BJ Berne. "Spectral gap optimization of order parameters for sampling complex molecular systems". In: *Proc. Natl. Acad. Sci.* 113.11 (2016), pp. 2839–2844.

[59] Wei Chen and Andrew L Ferguson. "Molecular enhanced sampling with autoencoders: On-the-fly collective variable discovery and accelerated free energy landscape exploration". In: *J. Comp. Chem.* 39.25 (2018), pp. 2079–2102.

[60] Christoph Wehmeyer and Frank Noé. "Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics". In: *J. Chem. Phys.* 148.24 (2018), p. 241703.

[61] GiovanniMaria Piccini, Dan Mendels, and Michele Parrinello. "Metadynamics with discriminants: A tool for understanding chemistry". In: *J. Chem. Theory Comput.* 14.10 (2018), pp. 5040–5044.

[62] Mohammad M Sultan and Vijay S Pande. "Automated design of collective variables using supervised machine learning". In: *J. Chem. Phys.* 149.9 (2018), p. 094106.

[63] João Marcelo Lamim Ribeiro et al. “Reweighted autoencoded variational Bayes for enhanced sampling (RAVE)”. In: *J. Chem. Phys.* 149.7 (2018), p. 072301.

[64] Yue-Yu Zhang et al. “Improving collective variables: The case of crystallization”. In: *J. Chem. Phys.* 150.9 (2019), p. 094509.

[65] Yihang Wang, Joao Marcelo Lamim Ribeiro, and Pratyush Tiwary. “Machine learning approaches for analyzing and enhancing molecular dynamics simulations”. In: *Curr. Opin. Struct. Biol.* 61 (2020), pp. 139–145.

[66] Frank Noé et al. “Machine learning for molecular simulation”. In: *Annu. Rev. Phys. Chem.* 71 (2020), pp. 361–390.

[67] Hythem Sidky, Wei Chen, and Andrew L Ferguson. “Machine learning for collective variable discovery and enhanced sampling in biomolecular simulation”. In: *Mol. Phys.* 118.5 (2020), e1737742.

[68] Tarak Karmakar et al. “Collective variables for the study of crystallisation”. In: *Mol. Phys.* 119.19-20 (2021), e1893848.

[69] Sun-Ting Tsai, Zachary Smith, and Pratyush Tiwary. “Sgoop-d: Estimating kinetic distances and reaction coordinate dimensionality for rare event systems from biased/unbiased simulations”. In: *J. Chem. Theory Comput.* 17.11 (2021), pp. 6757–6765.

[70] Ferry Hooft, Alberto Perez de Alba Ortiz, and Bernd Ensing. “Discovering collective variables of molecular transitions via genetic algorithms and neural networks”. In: *J. Chem. Theory Comput.* 17.4 (2021), pp. 2294–2306.

[71] Lixin Sun et al. “Multitask Machine Learning of Collective Variables for Enhanced Sampling of Rare Events”. In: *J. Chem. Theory Comput.* 18.4 (2022), pp. 2341–2353.

[72] Jakub Rydzewski et al. “Reweighted Manifold Learning of Collective Variables from Enhanced Sampling Simulations”. In: *J. Chem. Theory Comput.* 18.12 (2022), pp. 7179–7192.

[73] Wei Chen, Hythem Sidky, and Andrew L Ferguson. "Nonlinear discovery of slow molecular modes using state-free reversible VAMPnets". In: *J. Chem. Phys.* 150.21 (2019), p. 214114.

[74] Shams Mehdi et al. "Accelerating all-atom simulations and gaining mechanistic understanding of biophysical systems through state predictive information bottleneck". In: *J. Chem. Theory Comput.* 18.5 (2022), pp. 3231–3238.

[75] Heidi Klem, Glen M. Hocky, and Martin McCullagh. "Size-and-Shape Space Gaussian Mixture Models for Structural Clustering of Molecular Dynamics Trajectories". In: *J. Chem. Theory Comput.* 18.5 (2022). PMID: 35483073, pp. 3218–3230. DOI: 10.1021/acs.jctc.1c01290.

[76] Gareth A Tribello, Michele Ceriotti, and Michele Parrinello. "A self-learning algorithm for biased molecular dynamics". In: *Proc. Natl. Acad. Sci.* 107.41 (2010), pp. 17509–17514.

[77] Annie M. Westerlund and Lucie Delemotte. "InfleCS: Clustering Free Energy Landscapes with Gaussian Mixtures". In: *J. Chem. Theory Comput.* 15.12 (2019), pp. 6752–6759.

[78] F Giberti, GA Tribello, and M Ceriotti. "Global free-energy landscapes as a smoothly joined collection of local maps". In: *J. Chem. Theory Comput.* 17.6 (2021), pp. 3292–3308.

[79] Ian L. Dryden and Kanti V. Mardia. *Statistical Shape Analysis*. Chichester: John Wiley & Sons, 1998.

[80] Peg Howland, Moongu Jeon, and Haesun Park. "Structure Preserving Dimension Reduction for Clustered Text Data Based on the Generalized Singular Value Decomposition". In: *SIAM J. Matrix Anal. Appl.* 25.1 (2003), pp. 165–179. DOI: 10.1137/S0895479801393666.

[81] Gareth A Tribello et al. "PLUMED 2: New feathers for an old bird". In: *Comp. Phys. Comm.* 185.2 (2014), pp. 604–613.

[82] Massimiliano Bonomi et al. "Promoting transparency and reproducibility in enhanced molecular simulations". In: *Nat. Methods* 16.8 (2019), pp. 670–673.

[83] Michele Invernizzi, Pablo M Piaggi, and Michele Parrinello. "Unified approach to enhanced sampling". In: *Phys. Rev. X* 10.4 (2020), p. 041034.

[84] Giovanni Bussi and Alessandro Laio. "Using metadynamics to explore complex free-energy landscapes". In: *Nat. Rev. Phys.* 2.4 (2020), pp. 200–212.

[85] James F Dama, Michele Parrinello, and Gregory A Voth. "Well-tempered metadynamics converges asymptotically". In: *Phys. Rev. Lett.* 112.24 (2014), p. 240602.

[86] Pratyush Tiwary and Michele Parrinello. "A time-independent free energy estimator for metadynamics". In: *J. Phys. Chem. B* 119.3 (2015), pp. 736–742.

[87] James F Dama et al. "Exploring valleys without climbing every peak: more efficient and forgiving metabasin metadynamics via robust on-the-fly bias domain restriction". In: *J. Chem. Theory Comput.* 11.12 (2015), pp. 5638–5650.

[88] Adam Paszke et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Adv. Neur. Inf. Proc. Syst.* 32 (2019).

[89] Stefano Piana, Kresten Lindorff-Larsen, and David E. Shaw. "Protein folding kinetics and thermodynamics from atomistic simulation". In: *Proc. Natl. Acad. Sci. U. S. A.* 109.44 (2012), pp. 17845–17850. ISSN: 00278424. DOI: 10.1073/pnas.1201811109.

[90] Stefano Piana, Kresten Lindorff-Larsen, and David E. Shaw. "How robust are protein folding simulations with respect to force field parameterization?" In: *Biophys. J.* 100.9 (2011), pp. L47–L49. ISSN: 15420086. DOI: 10.1016/j.bpj.2011.03.051.

[91] Rose Du et al. "On the transition coordinate for protein folding". In: *J. Chem. Phys.* 108.1 (1998), pp. 334–350.

[92] Isabella L Karle and Padmanabhan Balaram. "Structural characteristics of. alpha.-helical peptide molecules containing Aib residues". In: *Biochem.* 29.29 (1990), pp. 6747–6756.

[93] Michael J. Hartmann et al. “Infinite switch simulated tempering in force (FISST)”. In: *J. Chem. Phys.* 152.24 (June 2020), p. 244120. ISSN: 10897690. DOI: 10.1063/5.0009280.

[94] Sebastian Buchenberg, Norbert Schaudinnus, and Gerhard Stock. “Hierarchical biomolecular dynamics: Picosecond hydrogen bonding regulates microsecond conformational transitions”. In: *J. Chem. Theory Comput.* 11.3 (2015), pp. 1330–1336.

[95] Mithun Biswas, Benjamin Lickert, and Gerhard Stock. “Metadynamics enhanced Markov modeling of protein dynamics”. In: *J. Phys. Chem. B* 122.21 (2018), pp. 5508–5514.

[96] Willmor J Peña Ccoa and Glen M Hocky. “Assessing models of force-dependent unbinding rates via infrequent metadynamics”. In: *J. Chem. Phys.* 156.12 (2022), p. 125102.

[97] Giovanni Bussi et al. “Free-energy landscape for β hairpin folding from combined parallel tempering and metadynamics”. In: *J. Am. Chem. Soc.* 128.41 (2006), pp. 13435–13441.

[98] Carlo Camilloni et al. “Exploring the protein G helix free-energy surface by solute tempering metadynamics”. In: *Proteins* 71.4 (2008), pp. 1647–1654.

[99] Cameron Abrams and Giovanni Bussi. “Enhanced sampling in molecular dynamics using metadynamics, replica-exchange, and temperature-acceleration”. In: *Entropy* 16.1 (2013), pp. 163–199.

[100] Alejandro Gil-Ley and Giovanni Bussi. “Enhanced conformational sampling using replica exchange with collective-variable tempering”. In: *J. Am. Chem. Soc.* 11.3 (2015), pp. 1077–1085.

[101] Shalini Awasthi and Nisanth N Nair. “Exploring high dimensional free energy landscapes: Temperature accelerated sliced sampling”. In: *J. Chem. Phys.* 146.9 (2017), p. 094108.

[102] Mark James Abraham et al. “GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers”. In: *SoftwareX* 1 (2015), pp. 19–25.

[103] Marco Jacopo Ferrarotti et al. “Accurate multiple time step in biased molecular simulations”. In: *J. Chem. Theory Comput.* 11.1 (2015), pp. 139–146.

[104] Jing Huang et al. “CHARMM36m: an improved force field for folded and intrinsically disordered proteins”. In: *Nat. Methods* 14.1 (2017), pp. 71–73.

[105] Subarna Sasmal et al. “Quantifying Unbiased Conformational Ensembles from Biased Simulations Using ShapeGMM”. In: *Journal of Chemical Theory and Computation* 20.9 (2024). PMID: 38662196, pp. 3492–3502. DOI: 10.1021/acs.jctc.4c00223.

[106] Eric Darve and Andrew Pohorille. “Calculating free energies using average force”. In: *J. Chem. Phys.* 115.20 (2001), pp. 9169–9183.

[107] Yinglong Miao, Victoria A Feher, and J Andrew McCammon. “Gaussian accelerated molecular dynamics: Unconstrained enhanced sampling and free energy calculation”. In: *J. Chem. Theory Comput.* 11 (2015), pp. 3584–3595.

[108] Peter Kasson et al. “Ensemble molecular dynamics yields submillisecond kinetics and intermediates of membrane fusion”. In: *Proc. Natl. Acad. Sci. U.S.A.* 103.32 (2006), pp. 11916–11921.

[109] Bettina Keller, Xavier Daura, and Wilfred F. Van Gunsteren. “Comparing geometric and kinetic cluster algorithms for molecular simulation data”. In: *J. Chem. Phys.* 132.7 (2010). ISSN: 00219606. DOI: 10.1063/1.3301140.

[110] Jun Hui Peng et al. “Clustering algorithms to analyze molecular dynamics simulation trajectories for complex chemical and biological systems”. In: *Chinese J. Chem. Phys.* 31.4 (2018), pp. 404–420. ISSN: 23272244. DOI: 10.1063/1674-0068/31/cjcp1806147.

[111] Aldo Glielmo et al. “Unsupervised Learning Methods for Molecular Simulation Data”. In: *Chem. Rev.* 121.16 (2021), pp. 9722–9758. ISSN: 15206890. DOI: 10.1021/acs.chemrev.0c01195.

[112] Fabrizio Marinelli et al. “A Kinetic Model of Trp-Cage Folding from Multiple Biased Molecular Dynamics Simulations”. In: *PLOS Computational Biology* 5.8 (Aug. 2009), pp. 1–18. DOI: 10.1371/journal.pcbi.1000452.

[113] Pratyush Tiwary and Michele Parrinello. “From metadynamics to dynamics”. In: *Phys. Rev. Lett.* 111.23 (2013), pp. 1–5. ISSN: 00319007. DOI: 10.1103/PhysRevLett.111.230602.

[114] Dhiman Ray and Michele Parrinello. “Kinetics from Metadynamics: Principles, Applications, and Outlook”. In: *J. Chem. Theory Comput.* 19.17 (2023), pp. 5649–5670. ISSN: 15499626. DOI: 10.1021/acs.jctc.3c00660.

[115] M Bonomi, A. Barducci, and M. Parrinello. “Reconstructing the Equilibrium Boltzmann Distribution from Well-Tempered Metadynamics”. In: *J. Comput. Chem.* 30.11 (2009), pp. 1615–1621.

[116] W. Kabsch. “A discussion of the solution for the best rotation to relate two sets of vectors”. In: *Acta Crystallogr. Sect. A* 34.5 (1976), pp. 827–828. ISSN: 16005724. DOI: 10.1107/S0567739478001680.

[117] Berthold K P Horn. “Closed-form solution of absolute orientation using unit quaternions”. In: *J. Opt. Soc. Am. A* 4.4 (1987), p. 629. ISSN: 1084-7529. DOI: 10.1364/josaa.4.000629.

[118] Colin Goodall. *Procrustes Methods in the Statistical Analysis of Shape*. Tech. rep. 2. 1991, pp. 285–339.

[119] Douglas L Theobald and Deborah S Wuttke. “Empirical bayes hierarchical models for regularizing maximum likelihood estimation in the matrix Gaussian procrustes problem”. In: *Proc. Natl. Acad. Sci. U. S. A.* 103.49 (2006), pp. 18521–18527. ISSN: 00278424. DOI: 10.1073/pnas.0508445103.

[120] Israel Dejene Gebru et al. “EM Algorithms for Weighted-Data Clustering with Application to Audio-Visual Scene Analysis”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (12 Dec. 2016), pp. 2402–2415. ISSN: 01628828. DOI: 10.1109/TPAMI.2016.2522425.

[121] Mark E Tuckerman. *Statistical mechanics: theory and molecular simulation*. Oxford university press, 2023.

[122] Hong Qian. "Relative entropy: Free energy associated with equilibrium fluctuations and nonequilibrium deviations". In: *Phys. Rev. E* 63 (4 2001), p. 042103. ISSN: 1063651X. DOI: 10.1103/PhysRevE.63.042103.

[123] Kreseten Lindorff-Larsen and Jesper Ferkinghoff-Borg. "Similarity Measures for Protein Ensembles". In: *PLoS ONE* 4.1 (2009), e4203.

[124] Dengming Ming and Michael E. Wall. "Quantifying allosteric effects in proteins". In: *Proteins* 59 (4 2005), pp. 697–707. ISSN: 08873585. DOI: 10.1002/prot.20440.

[125] Michael E. Wall. "Ligand binding, protein fluctuations, and allosteric free energy". In: *AIP Conf. Proc.* 851 (August 2006), pp. 16–33. ISSN: 0094243X. DOI: 10.1063/1.2345620.

[126] Jianhua Lin. "Divergence Measures Based on the Shannon Entropy". In: *IEEE Trans. Inf. Theory* 37 (1991), pp. 145–151.

[127] Timo M Schäfer and Giovanni Settanni. "Data reweighting in metadynamics simulations". In: *J. Chem. Theory Comput.* 16.4 (2020), pp. 2042–2052.

[128] Federico Giberti et al. "Iterative unbiasing of quasi-equilibrium sampling". In: *J. Chem. Theory Comput.* 16.1 (2019), pp. 100–107.

[129] Douglas J Tobias and Charles L Brooks III. "Conformational equilibrium in the alanine dipeptide in the gas phase and aqueous solution: A comparison of theoretical results". In: *J. Phys. Chem.* 96.9 (1992), pp. 3864–3870.

[130] Marissa G Saunders et al. "Nucleotide regulation of the structure and dynamics of G-actin". In: *Biophys. J* 106.8 (2014), pp. 1710–1720.

[131] Thomas D. Pollard and John A. Cooper. "Actin, a Central Player in Cell Shape and Movement". In: *Science* 326.5957 (Nov. 2009), pp. 1208–1212. DOI: 10.1126/science.1175862.

[132] Roberto Dominguez and Kenneth C. Holmes. “Actin Structure and Function”. In: *Annu. Rev. Biophys.* 40.1 (2011), pp. 169–186. DOI: 10.1146/annurev-biophys-042910-155359.

[133] Thomas D. Pollard. “Actin and Actin-Binding Proteins”. en. In: *Cold Spring Harb. Perspect. Biol.* 8.8 (Aug. 2016), a018226. ISSN: , 1943-0264. DOI: 10.1101/cshperspect.a018226.

[134] Martin McCullagh, Marissa G Saunders, and Gregory A Voth. “Unraveling the mystery of ATP hydrolysis in actin filaments”. In: *J. Am. Chem. Soc.* 136.37 (2014), pp. 13053–13058.

[135] Vilmos Zsolnay et al. “Structural basis for polarized elongation of actin filaments”. In: *Proc. Natl. Acad. Sci. U.S.A.* 117.48 (2020), pp. 30458–30464.

[136] Glen M Hocky, Thomas Dannenhoffer-Lafage, and Gregory A Voth. “Coarse-grained directed simulation”. In: *J. Chem. Theory Comput.* 13.9 (2017), pp. 4593–4603.

[137] Glen M Hocky et al. “Structural basis of fast-and slow-severing actin–cofilactin boundaries”. In: *J. Biol. Chem.* 296 (2021).

[138] Yuvraj Singh, Glen M Hocky, and Brad J Nolen. “Molecular dynamics simulations support a multistep pathway for activation of branched actin filament nucleation by Arp2/3 complex”. In: *J. Biol. Chem.* 299.9 (2023), p. 105169.

[139] Fatemah Mukadum, Willmor J. Peña Ccoa, and Glen M. Hocky. “Molecular simulation approaches to probing the effects of mechanical forces in the actin cytoskeleton”. In: *Cytoskeleton* Early View (2024). DOI: 10.1002/cm.21837.

[140] Philip Graceffa and Roberto Dominguez. “Crystal structure of monomeric actin in the ATP state: structural basis of nucleotide-dependent actin dynamics”. In: *J. Biol. Chem.* 278.36 (2003), pp. 34172–34180.

[141] Ludovic R Otterbein, Philip Graceffa, and Roberto Dominguez. “The crystal structure of uncomplexed actin in the ADP state”. In: *Science* 293.5530 (2001), pp. 708–711.

[142] Jason C Porta and Gloria EO Borgstahl. “Structural basis for profilin-mediated actin nucleotide exchange”. In: *J. Mol. Biol.* 418.1-2 (2012), pp. 103–116.

[143] Jim Pfaendtner et al. “Nucleotide-dependent conformational states of actin”. In: *Proc. Natl. Acad. Sci. U.S.A.* 106.31 (2009), pp. 12723–12728.

[144] Aidan P Thompson et al. “LAMMPS-a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales”. In: *Comp. Phys. Comm.* 271 (2022), p. 108171.

[145] Toshiro Oda et al. “The nature of the globular-to fibrous-actin transition”. In: *Nature* 457.7228 (2009), pp. 441–445.

[146] Subarna Sasmal, Martin McCullagh, and Glen M. Hocky. “Improved data-driven collective variables for biased sampling through iteration on biased data”. In: *bioRxiv* (2025). DOI: 10.1101/2025.03.25.644418.

[147] Tamar Schlick et al. “Biomolecular modeling and simulation: a prospering multidisciplinary field”. In: *Annu. Rev. Biophys.* 50.1 (2021), pp. 267–301.

[148] Haochuan Chen and Christophe Chipot. “Enhancing sampling with free-energy calculations”. In: *Curr. Opin. Struct. Biol.* 77 (2022), p. 102497.

[149] Enrico Trizio and Michele Parrinello. “From Enhanced Sampling to Reaction Profiles”. In: *The Journal of Physical Chemistry Letters* 12.35 (2021). PMID: 34469175, pp. 8621–8626. DOI: 10.1021/acs.jpclett.1c02317.

[150] Dhiman Ray, Enrico Trizio, and Michele Parrinello. “Deep learning collective variables from transition path ensemble”. In: *The Journal of Chemical Physics* 158.20 (May 2023), p. 204102. ISSN: 0021-9606. DOI: 10.1063/5.0148872.

[151] Dedi Wang and Pratyush Tiwary. “State predictive information bottleneck”. In: *The Journal of Chemical Physics* 154.13 (Apr. 2021), p. 134111. ISSN: 0021-9606. DOI: 10.1063/5.0038198.

[152] Jordane Preto and Cecilia Clementi. “Fast recovery of free energy landscapes via diffusion-map-directed molecular dynamics”. In: *Phys. Chem. Chem. Phys.* 16 (36 2014), pp. 19181–19191. DOI: 10.1039/C3CP54520B.

[153] Martin McCullagh. *shapeGMMTorch: Gaussian Mixture Model clustering in size-and-shape space using PyTorch*. Version 1.7.1. 2024.

[154] Peter T. Lake and Martin McCullagh. *WeightedLDA: Linear discriminant analysis with weights associated with each observation*. Version 0.0.1. 2024.