Characterizing Protein Conformational Space Using Robotics-Based Search and Topological Data Analysis

Fatemeh Afrasiabi*, Ramin Dehghanpoor*, Nurit Haspel†

Department of Computer Science, University of Massachusetts Boston, Boston, MA 02125, USA

**Denotes equal contribution †E-mail: Nurit.Haspel@umb.edu*

Investigating the conformational space of proteins is essential in order to associate their structures with their fundamental functions. Nonetheless, it is a challenging task, both experimentally and computationally. Because of the transient nature of these conformational changes and the fact that they are impermanent, empirical methods have fallen short to capture them. In silico methods, on the other hand, have shown great promise in exploring these conformational pathways. In this article, we provide an extensive evaluation of our previously introduced, robotics inspired conformational search algorithm (RRT* with Monte Carlo). We then identify what intermediate conformations appear the most in our generated conformational pathways using TDA Mapper, a topological data analysis algorithm, and examine how close these intermediate conformations are to existing experimental data.

Keywords: Monte Carlo Method; Conformational Space of Proteins; Rapidly Exploring Random Tree Search; Topological Data Analysis; Principal Component Analysis; Mapper Algorithm.

1. Introduction

Proteins perform vital functions, such as catalysing metabolic reactions, regulating cell signaling, DNA replication, and so forth. They often go through large-scale conformational transitions to perform these functionalities. To understand and interpret proteins functionalities, we need to understand their structures, their conformational spaces and the pathways they take when going through these transitions. Having said that, the conformational space of proteins is complex and high dimensional, and hence studying it becomes a challenging task. Several attempts have been made during the past decades to characterize proteins conformational pathways and find intermediate structures created during these transitions. Experimental methods such as X-ray crystallography,[1] NMR,[2] and cryo-Electron Microscopy[3] have been used, but because of the rapid dynamics of proteins these changes happen in microseconds or less, and experimental methods cannot capture all of them. For this reason, there has been increasing interest in computational methods to simulate and explore these pathways. Computationally speaking, it is still a challenging problem to simulate these conformational changes because these simulations are usually physics-based and take considerable time. Molecular Dynamics (MD) simulations can simulate these transitions at the microsecond level, but they are still not fast enough to capture transitions that take place in larger timescales.[4] Therefore, researches

© Preprint of an article submitted for consideration in Pacific Symposium on Biocomputing 2021 World Scientific Publishing Co., Singapore, http://psb.stanford.edu/

have come up with variants of MD simulations; Steered MD,[5] Replica Exchange MD,[6] and Targeted MD,[7] to name a few. In addition to MD simulations, a number of methods attempt to explore the geometrically constrained conformational space by using the degrees of freedom (dihedral angles) to perturb the protein structure. A few instances of these search methods are Robotic Motion Planning,[8,9] elastic network model,[10,11] and normal mode analysis.[12,13] The advantage of these search methods is that they run much faster than MD simulations, but their downside is the fact that they may result in pathways that are not physically probable because of the element of randomness. We will review this issue more in section 2.4 and 2.5. It is of paramount importance to do an accurate sampling of the conformational space to identify intermediate conformations that correspond to local energy minima and are close to actual intermediates appearing in nature. Thus, there is a need to extract biologically relevant conformations and filter out implausible ones. To do so, biological data can be incorporated to steer the search algorithm in order to produce meaningful conformations. Rigidity and flexibility information about the protein molecules, for example, can provide useful knowledge about the rigid parts of the molecule that are generally preserved in nature so flexible parts can be manipulated. This information can be gathered experimentally, from coevolution data, from computational analysis, or by using rigidity analysis tools such as KINARI.[14,15] In our previous work,[16] we integrated rigidity analysis information into our RRT*-inspired algorithm to reduce randomness and increase the accuracy of produced conformations. In section 2.4, we provide an overview of our previous work, and in this article, in section 3, we evaluate how close our generated intermediate conformations are to experimentally known intermediates.

2. Overview of RRT* Search and TDA

2.1. *RRT* Search*

The Rapidly exploring Random Tree (RRT) algorithm[17] is used to rapidly explore constrainted spaces. It builds a tree rooted at the start configuration and grows by using random samples from the exploration space and connecting them to the closest vertex in the tree. We used this algorithm in our previous efforts[16,18] to explore the conformational space of proteins.

Here is a quick review of the basic steps in the RRT algorithm. At every iteration, the algorithm generates a random conformation q_{rand}, finds the closest node to q_{rand} in the tree, and tries to create a path from it to the q_{rand}. If the new random conformation has high energy, it repeats the same steps of generating the random conformations as long as the conformations are below a given energy threshold. The entire process is repeated many times until the new conformation is close enough to the target or the maximum number of iterations is reached.

RRT explores the search space rapidly. But given that it randomly expands to everywhere in the search space, it may not converge fast enough for complex spaces like protein conformations. For this reason we bias the search towards the goal in some percent of the runs (1/3 in our implementation). Additionally, as a result of the randomness in RRT, the produced paths could be uneven and rugged. To tackle these issues, the RRT* algorithm was introduced[19] to produce smoother paths. The optimization of paths in RRT* is done by proposing a heuristic cost function to find less costly (smoother) parents for every new node that is added to the tree and changing parents if needed.

2.2. *Rigidity Analysis*

Proteins can be modeled as a mechanical structure made up of rigid and flexible parts.[20] Rigid parts denote the group of atoms that are more likely to stay together during conformational changes because destabilizing and deforming them requires more energy than the flexible residues. Taking the rigidity analysis information into account helps us gain insights into potential motions of proteins, and select non-rigid (flexible) parts for manipulation.

We used the Kinari software[15] in our previous work to incorporate rigidity analysis information to better guide our search algorithm. It implements a faster and more robust variation of the pebble game algorithm to extract rigidity information about the molecules without having to run extensive MD simulations.

2.3. *TDA and the Mapper Algorithm*

There has been a surge in the employment of Topological Data Analysis (TDA) in recent years to analyze complex high-dimensional datasets and extract valuable information and insights.[21] TDA has been extensively used in biomedical research.[22–25] This is due to the fact that we now have massive amounts of data, but our ability to analyze and visualize the data still lags behind. By using TDA methods, researches have found a way to understand the structures and the underlying shapes of high-dimensional datasets. The three key ideas of TDA are: studying shapes in a coordinate-free way, studying the properties of shapes that are invariant under "small" deformations, and create compressed representations of shapes. These key applications have made it possible to study intricate data. Other methods have been introduced before, such as Principal Component Analysis (PCA), Multidimensional Scaling (MDS), and so forth. However, these methods may not catch all large and small-scale patterns existing in the data as well as TDA.

The Mapper algorithm[26] is an unsupervised learning TDA algorithm that utilizes a combination of dimensionality reduction, clustering, and graph networking methods to provide a graph (called the Mapper graph) that summarizes the structure of a dataset. The general idea of the Mapper algorithm is as follows:

Suppose that we can divide our data into groups (determined based on the data) and represent each group by a node. Next, we represent the relationships between two groups through edge connections. Now we can visualize our massive data in a simple graph. Figure 1 shows an example of the Mapper algorithm on the data points of a 2D torus (A).

To divide the data into groups, Mapper uses any dimensionality reduction algorithm (PCA for instance) to reduce the data into a low-dimensional network where local relationships are preserved. This process is done using a *filter function* – which can be any function that can divide the data into bins. In Figure 1, for example, the x values of the data points is used as the filter function, where bins are shown in different colors (B). The Mapper then creates a cover of the projected space as a set of partially overlapping bins (intervals) (C). Note that orange data points for instance, in Figure 1 (C), are the data points that fall into both red and yellow bins. Then clustering is done on each bin. The clustering method is usually single-linkage. Finally, Mapper constructs the simple graph representing the data by using the set of clusters as vertices and adds an edge between any two clusters if they have a data point in

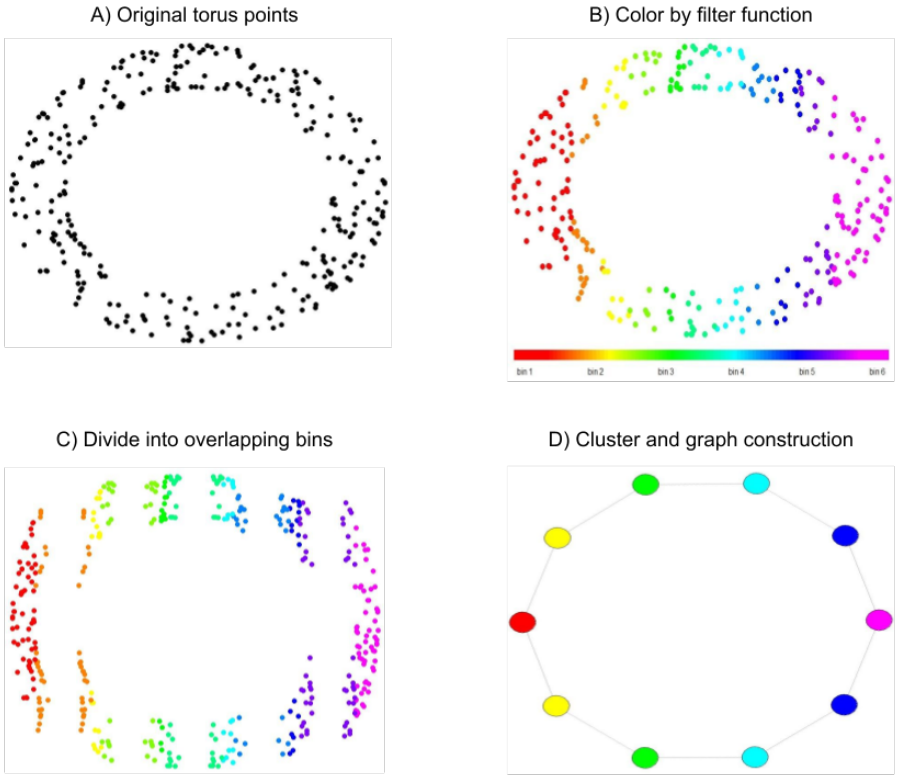

Fig. 1. An example of the Mapper algorithm run on the data points of a 2D torus.

common (D).

2.4. *Previous Work*

In our previous work, we focused specifically on rigidity analysis information and consequently extracted the flexible residues. We used the flexible residues to lead the search by using flexible hinges to manipulate. This helped the algorithm to choose dihedral angles to rotate in order to perform the conformational transition. Since the rigidity properties of proteins may change when they go through large-scale conformational changes, we re-calculate the rigidity during the run. To balance efficiency and accuracy, we found that optimal point to redo the rigidity analysis calculations is every time there is a considerable change ($2\mathring{A}$) in a produced conformations with respect to the previous calculation.

2.5. *This Contribution*

In this study, we do an evaluation of our previous work using topological data analysis to detect intermediate conformations. Using the TDA Mapper algorithm we construct a graph of our high dimensional dataset. We use about 200 runs of one of our proteins from our test cases. We used Adenylate Kinase (AdK), as input to the Mapper algorithm since it has experimentally known intermediate structures. Afterward, we use PCA to reduce the dimensionality of our data. The first **three** principal components are used in the Mapper algorithm as the projection

of the data. We run the mapper algorithm and extract the biggest clusters that show the sets of most populted regions that can correspond to intermediate conformations. We hypothesized these conformations would be close to known conformations of AdK that exist experimentally.

3. Materials and Methods

3.1. *Data Collection and Preprocessing*

In this contribution, our primary goal is to evaluate whether our RRT*-inspired algorithm produces conformations that are likely to represent intermediate conformations that exist in nature. To achieve this goal, we use the Mapper algorithm to extract conformations that our algorithm generates frequently and then compare them with known conformations of AdK. Intermediate conformations are not available for many, proteins so we chose AdK to do the evaluation. Because of the randomness in the nature of robotic motion planning algorithms, we ran the algorithm many times to make sure we have a reliable coverage of the conformational paths produced by the algorithm. We ran the algorithm on each test case at least 150 times (150 for bigger molecules, and about 600 for small molecules), and for AdK we did about 450 times (both $1AKE \rightarrow 4AKE$ and $4AKE \rightarrow 1AKE$). In our RRT-inspired algorithm, iterative sampling of new conformations continues up to the point that the score of the newly generated conformation ($lRMSD$ regarding the target conformation) is below a preset threshold or the minimum score of achieved by generated conformations does not decrease for 500 iterations. The preset $lRMSD$ threshold is between $1.7\mathring{A}$ and $4\mathring{A}$, based on the protein size and pathway difficulty. In order to find the best threshold that would be a tradeoff between time and how close our conformations are to the target conformation, we tried different thresholds for each of our test cases. Figure 2 shows the result for adenylate kinase (AdK), calmodulin (CaM), DNA, lysine/arginine/ornithine-binding (LAO), cyanovirin-N (CVN), and ribose-binding protein (RBP). This plot shows a realistic lower bound for our test case proteins. Realistically, $lRMSD$ values below 1.5-2Å are hard to achieve due to sampling errors and variations in proteins. Each protein was represented using the backbone atoms (N, C-alpha, C, O) and C-beta.

For our test case protein (AdK), we use runs of one direction; $4AKE \rightarrow 1AKE$ (closed conformation to open conformation) for AdK For AdK, each path created an average of ten conformations going from the start conformation ($4AKE$) to the end conformation ($1AKE$). This gave us a total of 1,415 intermediate conformations (excluding the start conformation and outliers) that form our high-dimensional dataset. Each path is stored as a numeric matrix of x, y, z coordinates with one conformation per row and a Cartesian coordinate per column. That means for instance, for a path of 20 intermediate conformations going from 1AKE to 4AKE we get a matrix of 20 rows, and $3*1050$ columns (since AdK has 1,050 backbone atoms). Adding all the paths gives us a matrix of 1415×3150.

3.2. *Using Mapper for Evaluation*

We used the Mapper algorithm[26] implemented by Paul Pearson et al.,[27] accessed on 2 August 2021), which is an **R** package for using discrete Morse theory to analyze a data set using the Mapper algorithm.

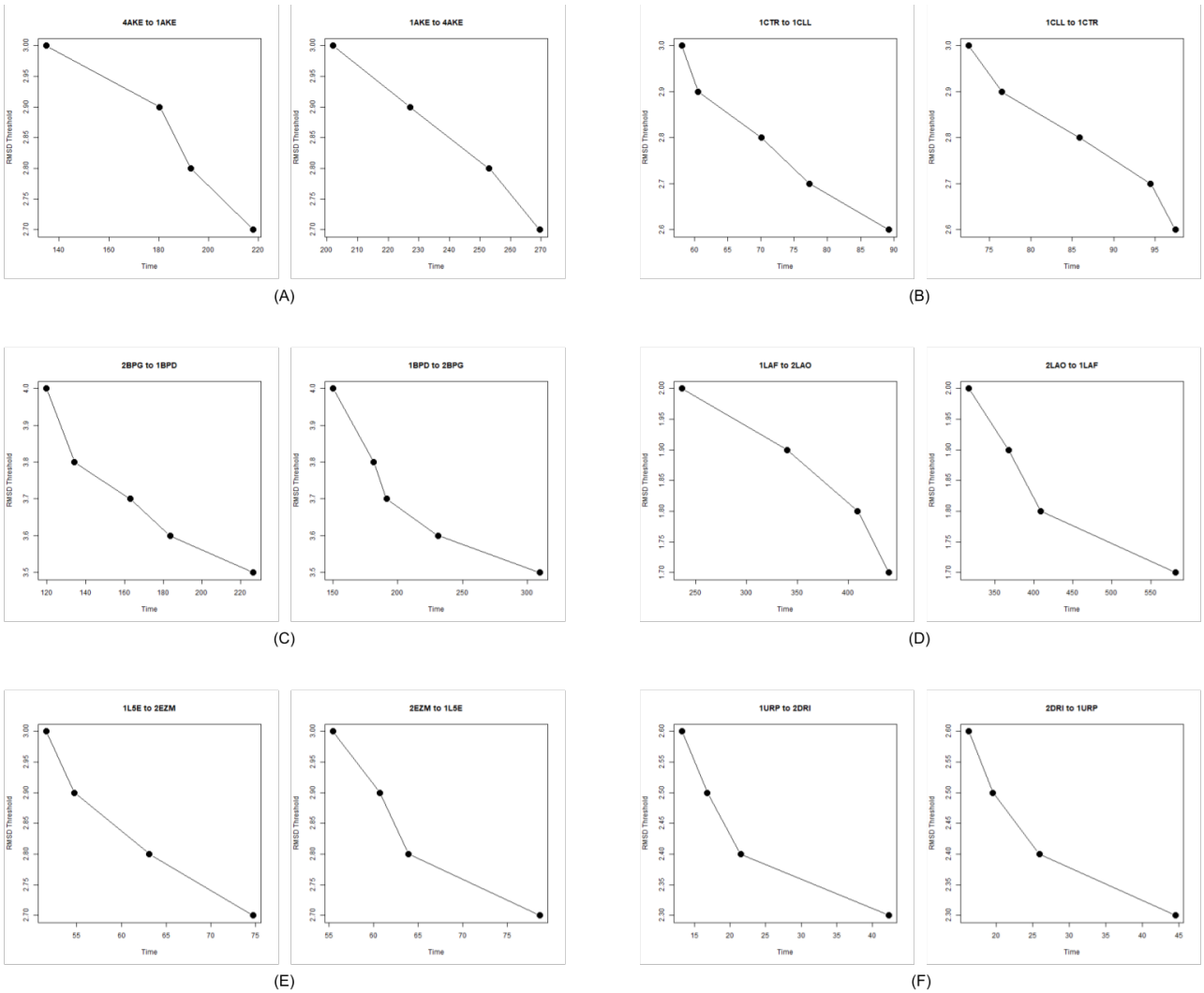

Fig. 2. RMSD threshold versus runtime average for (A) AdK, (B) CaM, (C) DNA, (D) LAO, (E) CVN, and (F) RBP.

By running Mapper, our 3-dimensional data (Refer to 3.1) is divided into partially overlapping bins as described above. We use the default clustering method, single linkage, to cluster the data points in each bin. At last, a graph is created from the set of clusters (graph vertices) and whether they have data points in common (graph edges). The choice of parameters is discussed in section 4.1. The resulted graph is shown in Figure 3. The size of each vertex in the figure is proportional to the number of data points in the cluster presented by that vertex. We use different strategies to color the vertices of the graph to gain useful information. In Figure 3 (A) we color the graph according to average energy values of conformations that reside in the same cluster (A). To get to at least90% of the total variance, we use principal component four (PC4) to color the vertices. For each cluster that is represented by a vertex, we calculate the average value of PC4 and color that vertex accordingly (B); In (C) we colored every vertex by their RMSD from the end conformation (C). Each Mapper graph in Figure 3

is discussed extensively in section 4.1.

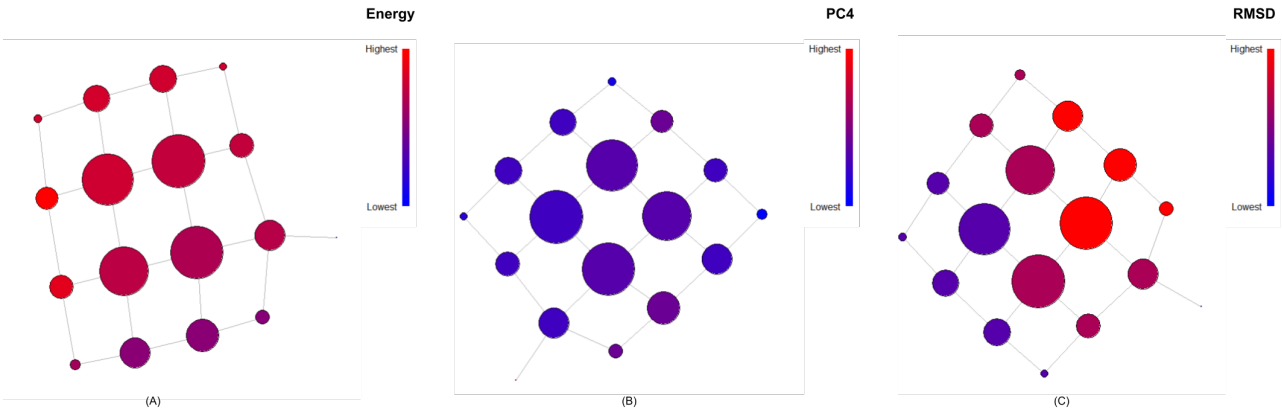

Fig. 3. Mapper graph for AdK. Vertices colored by average values of (A) energy, (B) PC4, and (C) RMSD to the target conformation

4. Results

4.1. *Adenylate Kinase (AdK)*

AdK is a monomeric phosphotransferase enzyme that catalyzes the reversible transfers of a phosphoryl group from ATP to AMP. The structure of AdK, which contains 214 amino acids, is formed of the three main domains, the CORE, the ATP binding domain called the LID, and the NMP binding domain. It is an interesting protein to study as far as conformational changes because it's a signal-transducing protein, and there should be a balance between its conformations for it to regulate protein activity.

We used the following parameters to create a simple graph representing the structure of the data (Refer to section 2.3 for an overview of Mapper):

- distance-matrix: The matrix of distances between points. We used the standard Euclidean distance measure.
- filter-value: The vector of the values of the filter at the points of the data set. We generated this vector using PCA. We used PC1, PC2, and PC3.
- num-intervals: Number of intervals in the co-domain of the filter function (regular grid over the filter range).
- percent-overlap: Percentage of overlap between the intervals.
- num-bins-when-clustering: A parameter that controls the number of clusters for clustering each inverse image (number of intervals of the histogram of edge lengths in the single linkage algorithm)

Since mapper is an unsupervised learning algorithm, and in unsupervised learning algorithms the user usually selects the parameters, we applied a grid search to find the best parameters. For our grid search we used the following parameter ranges:

- $\{3, 4, 5, 6\}$ for the number of intervals,

- $\{35, 40, 45, 50\}$ for the overlap percentage, and
- $\{10, 11, 12, 13, 14, 15\}$ for number of bins when clustering.

For these ranges, we found that for AdK, 4 for the number of intervals, 35 for the percentage of overlap, and 13 for the number of bins work the best. We chose these parameters this way to avoid either having all the data points (conformations) be clustered into one big cluster or having a large number of clusters with only a few data points in each of them. In Figure 3, the resulting graph has 17 vertices showing the clusters that contain all of our data points. Some points may be in more than one cluster (because of the overlap) and where that happens there is an edge connecting the two clusters. The size of each vertex is proportional of the number of points that reside in that cluster. That means we have four clusters that contain most of our data points, where the largest contains 445 conformations. These are the conformations that have been generated the most and are more expected to be close to highly populated regions. Small clusters means that the conformations in those clusters are not likely to occur – for example due to having high energy, and should probably be disregarded. Figure 3 (A) shows the graph colored by the energy value. To calculate the energy value of our conformations we used the following function:[28]

$$E_{total} = E_{HB} + E_{burial} + E_{water} + E_{bond} + E_{angle} + E_{VdW} \tag{1}$$

The first three terms represent hydrogen bonds, burial, and water-mediated interaction terms taken from Ref. 28. The other three terms E_{VdW}, E_{bond}, and E_{angle} came from the $AMBER$ $ff03$ force field.[29] The E_{VdW} is the Lennard Jones potential, slightly adapted to tolerate soft collisions between atoms, as a result of the lower resolution. These terms were used to correct for clashes and deformities in the structure. The cluster with the highest energy value has an average energy of $4251kJ/mol$, and the one with the lowest energy value has an average of $-365kJ/mol$. Many of the conformations have energy values between these numbers, which may correspond to local energy minima. Notice that energy values may be high due to the coarse-grained representation of the molecule.

In Figure 3 (C), we can see the colors relative to their RMSD values from the end conformation (the closed conformation, PDB: $1AKE$). These colors are relative to each other and it does not mean that the highest RMSD is far from the end conformation. The highest RMSD to the end conformation for the biggest clusters is less than $5.5\mathring{A}$ (the start-goal RMSD is approximately 7Å). The reason is that RRT is faster in the first steps and it gets slower by getting closer to the end conformation (the high negative slope at the beginning of the plot in Figure 4 can be seen). From the Mapper graph, we obtained the nine biggest clusters (since the average number of conformations in the simulated paths is ten and the first one is the start conformation), calculated the average coordinates of all the conformations in each cluster, and then calculated their RSMD with respect to the end conformation. The result is shown by the blue line which can represent a more realistic path. We can see most of the bigger clusters span the range of RMSD values. Because of randomness in our algorithm, and also excessive number of runs, we can see there are a lot of outliers in gray lines far from the blue line, which appears smoother. Figure 4 shows that we can remove those outliers by using the Mapper algorithm. In this plot, for each simulated path of AdK, the RMSD values of all

the conformations with respect to the end conformation are shown (gray lines). The red line is not an actual path from our algorithm. It's just a numerical average of all the simulations (i.e. gray lines in the plot).

Using the first three principal components when doing PCA on our data covered 78% of the total variance and by using PC4, we get close to 90%, but it's not possible to use 4 vectors for the TDA mapper. Therefore, we used PC4 to color the vertices (Figure 3 (C)).

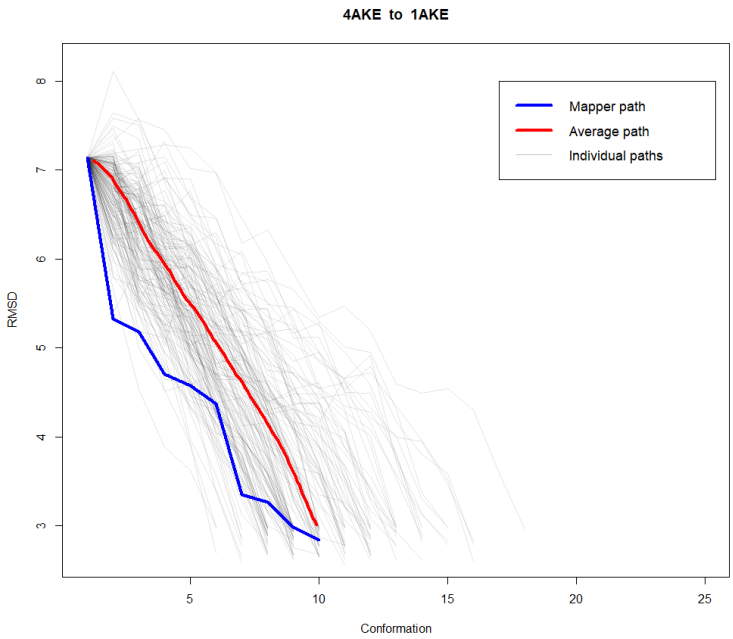

Fig. 4. RMSD values to the end conformation (1AKE) for simulated paths of RRT* algorithm (gray lines), average of simulated paths (red line), and conformations found by Mapper (blue line).

Figure 5 shows the intermediate conformations for the path from 4AKE to 1AKE that were extracted by Mapper. We used the biggest clusters, averaged the coordinates of the conformations in each cluster, and used these coordinates to create the intermediate conformations shown in the Figure. We can see how the conformations follow a hinge motion.

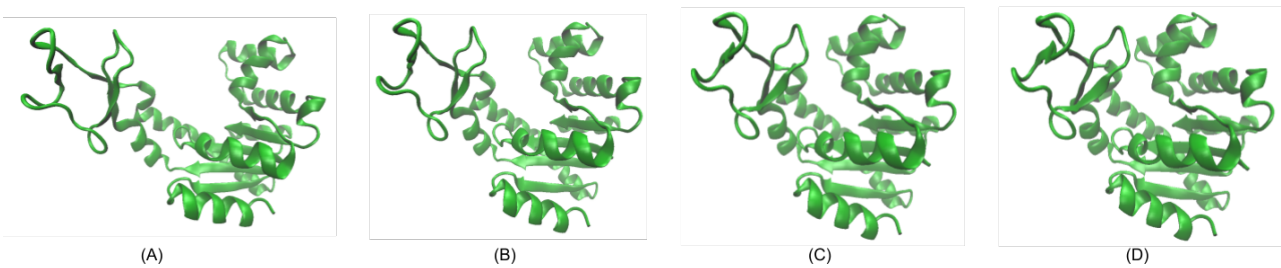

Fig. 5. Conformational path of AdK - 4AKE (A) → 1AKE (D) - using intermediate conformations produced by Mapper)

4.2. *Comparison to Experimental Results*

As mentioned in section 1, experimentally known intermediate conformations are hard to find. Because of that, experimental validation of computational methods is hard to do. We used AdK, for which there exist several intermediate experimental structures.[30] We used our highly populated conformations, created by taking the average coordinates of conformations in each of the nine biggest clusters, and calculated their RSMDs with respect to the end conformation, 1AKE. Table 1 shows that we could find intermediate conformations that are close to the experimental ones (as close as 2.6Å).

Table 1. Comparison of Mapper-generated intermediates to known intermediates

Protein	PDB ID	RMSD to closest conformer
AdK	2AK2	3.9
	1AK2	4
	1DVR	3.2
	1E4Y	2.6
	1E4V	2.8
	2ECK	2.9
	1ANK	2.9

4.3. *Discussion*

Exploring the conformational space of proteins is of primary importance for understanding their functions. Many proteins go through large-scale conformational changes in order to perform their functionalities. Therefore, it is considerably important to study these conformational pathways to find intermediate conformations that may correspond to local energy minima. It is experimentally hard to do, however, because these conformations are transitory and are hard to capture experimentally. To address this problem, computational methods are used to simulate these pathways. We need to characterize the sampling space of these methods to get conformations that are more probable to happen in nature. We developed a robotics-inspired algorithm to generate random conformations by rotating dihedral angles and added rigidity analysis information to make the run more efficient and converge better. In this paper we present an evaluation of our method using the Topological Data Analysis algorithm called Mapper, to find the intermediate conformations that are frequently produced by our algorithm and compare them to experimentally available ones. Considering the results in Table 1, we believe using TDA Mapper is highly beneficial to complement computational methods, and that our RRT*-inspired algorithm is capable of generating conformations that are close to the highly populated intermediate structures. Current and future work includes using Mapper in order to evaluate multiple other proteins (where experimental data about intermediates is available) as well. More future work includes developing other dimensional-

ity reduction methods (using Auto Encoders and deep learning techniques, for instance) and clustering techniques for the Mapper algorithm, and improving our algorithm to find conformations closer to experimental intermediates. We also plan to incorporate more examples in our analysis to get an understanding of conformational spaces of various protein molecules.

Acknowledgments

This research was funded in part by NSF grant IIS:2031260. The tests were run on the supercomputing facilities managed by the Research Computing Department at the University of Massachusetts Boston, Chimera which is a heterogeneous distributed memory high performance compute cluster (AMD Opteron 6128 processors, 2.0 Ghz).

References

1. M. Levantino, B. A. Yorke, D. C. Monteiro, M. Cammarata and A. R. Pearson, Using synchrotrons and xfels for time-resolved x-ray crystallography and solution scattering experiments on biomolecules, *Current Opinion in Structural Biology* **35**, p. 4148 (Dec 2015).
2. S. J. Opella and F. M. Marassi, Applications of nmr to membrane proteins, *Archives of Biochemistry and Biophysics* **628**, p. 92101 (Aug 2017).
3. E. Nogales and S. H. Scheres, Cryo-em: A unique tool for the visualization of macromolecular complexity, *Molecular Cell* **58**, p. 677689 (May 2015).
4. S. A. Adcock and J. A. McCammon, Molecular dynamics: Survey of methods for simulating the activity of proteins, *Chemical Reviews* **106**, p. 15891615 (May 2006).
5. *Computational Molecular Dynamics: Challenges, Methods, Ideas* (Springer Berlin Heidelberg, 1999).
6. H. Fukunishi, O. Watanabe and S. Takada, On the hamiltonian replica exchange method for efficient sampling of biomolecular systems: Application to protein structure prediction, *The Journal of Chemical Physics* **116**, p. 90589067 (May 2002).
7. S. Wolf and G. Stock, Targeted molecular dynamics calculations of free energy profiles using a nonequilibrium friction correction, *Journal of Chemical Theory and Computation* **14**, p. 61756182 (Nov 2018).
8. J. Cortes, T. Simeon, V. Ruiz de Angulo, D. Guieysse, M. Remaud-Simeon and V. Tran, A path planning approach for computing large-amplitude motions of flexible molecules, *Bioinformatics* **21**, p. i116i125 (Jun 2005).
9. S. Liu and P. Liu, Robot motion planning benchmarking and optimization using motion planning pipeline (Apr 2021).
10. G. F. Schrder, A. T. Brunger and M. Levitt, Combining efficient conformational sampling with a deformable elastic network model facilitates structure refinement at low resolution, *Structure* **15**, p. 16301641 (Dec 2007).
11. G. Hu, L. Di Paola, Z. Liang and A. Giuliani, Comparative study of elastic network model and protein contact network for protein complexes: The hemoglobin case, *BioMed Research International* **2017**, p. 115 (2017).
12. J. A. Bauer, J. Pavlovi and V. Bauerov-Hlinkov, Normal mode analysis as a routine part of a structural investigation, *Molecules* **24**, p. 3293 (Sep 2019).
13. H. Wako and S. Endo, Normal mode analysis as a method to derive protein dynamics information from the protein data bank, *Biophysical Reviews* **9**, p. 877893 (Nov 2017).
14. R. Dehghanpoor, E. Ricks, K. Hursh, S. Gunderson, R. Farhoodi, N. Haspel, B. Hutchinson and F. Jagodzinski, Predicting the effect of single and multiple mutations on protein structural stability, *Molecules* **23**, p. 251 (Jan 2018).

15. N. Fox, F. Jagodzinski, Y. Li and I. Streinu, Kinari-web: a server for protein rigidity analysis, *Nucleic Acids Research* **39**, p. W177W183 (Jun 2011).
16. F. Afrasiabi, R. Dehghanpoor and N. Haspel, Integrating rigidity analysis into the exploration of protein conformational pathways using rrt* and mc, *Molecules* **26**, p. 2329 (Apr 2021).
17. S. M. LaValle and J. Kuffner, James J., Randomized kinodynamic planning, *The International Journal of Robotics Research* **20**, p. 378400 (May 2001).
18. F. Afrasiabi and N. Haspel, Efficient exploration of protein conformational pathways using rrt* and mc, in *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, (ACM, Sep 2020).
19. S. Karaman and E. Frazzoli, Sampling-based algorithms for optimal motion planning, *The International Journal of Robotics Research* **30**, p. 846894 (Jun 2011).
20. F. Jagodzinski, P. Clark, J. Grant, T. Liu, S. Monastra and I. Streinu, Rigidity analysis of protein biological assemblies and periodic crystal structures, *BMC Bioinformatics* **14**, p. S2 (2013).
21. Topological data analysis, *Inverse Problems* **27**, p. 120201 (nov 2011).
22. R. Rabadn, Y. Mohamedi, U. Rubin, T. Chu, A. N. Alghalith, O. Elliott, L. Arns, S. Cal, . J. Obaya, A. J. Levine and et al., Identification of relevant genetic alterations in cancer using topological data analysis, *Nature Communications* **11** (Jul 2020).
23. M. Nicolau, A. J. Levine and G. Carlsson, Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival, *Proceedings of the National Academy of Sciences* **108**, p. 72657270 (Apr 2011).
24. K. Sasaki, D. Bruder and E. A. Hernandez-Vargas, Topological data analysis to model the shape of immune responses during co-infections, *Communications in Nonlinear Science and Numerical Simulation* **85**, p. 105228 (Jun 2020).
25. J. A. Pike, A. O. Khan, C. Pallini, S. G. Thomas, M. Mund, J. Ries, N. S. Poulter and I. B. Styles, Topological data analysis quantifies biological nano-structure from single molecule localization microscopy, *Bioinformatics* (Oct 2019).
26. G. Singh, F. Memoli and G. Carlsson, Topological methods for the analysis of high dimensional data sets and 3d object recognition, in *Eurographics Symposium on Point-Based Graphics*, eds. M. Botsch, R. Pajarola, B. Chen and M. Zwicker (The Eurographics Association, 2007).
27. P. Pearson, D. Muellner and G. Singh, *TDAmapper: Analyze High-Dimensional Data Using Discrete Morse Theory*, (2015). R package version 1.0.
28. G. A. Papoian, J. Ulander, M. P. Eastwood, Z. Luthey-Schulten and P. G. Wolynes, Water in protein structure prediction, *Proceedings of the National Academy of Sciences* **101**, 3352 (2004).
29. Y. Duan, C. Wu, S. Chowdhury, M. C. Lee, G. Xiong, W. Zhang, R. Yang, P. Cieplak, R. Luo, T. Lee *et al.*, A point-charge force field for molecular mechanics simulations of proteins based on condensed-phase quantum mechanical calculations, *Journal of computational chemistry* **24**, 1999 (2003).
30. P. Maragakis and M. Karplus, Large amplitude conformational change in proteins explored with a plastic network model: Adenylate kinase, *Journal of Molecular Biology* **352**, p. 807822 (Sep 2005).