Clustering of Protein Conformations Using Parallelized Dimensionality Reduction

Arpita Joshi and Nurit Haspel
Dept. of Computer Science, University of Massachusetts, Boston, USA
Email: {arpita.joshi001, nurit.haspel}@umb.edu

***Abstract*—Ascertaining the conformational landscape of a macromolecule, like protein is indispensable to understanding its characteristics and functions. In this work, an amassment of these techniques is presented, that would be an aid in sampling of these conformations better and faster. The datasets that represent these conformational dynamics of proteins are complex and high dimensional. Therefore, there arises a need for dimensionality reduction methods that best conserve the variance and further the analysis of the data. We present a parallelized version of a well-known dimensionality reduction method, Isomap. Isomap has been shown to produce better results than linear dimensionality reduction in approximating the complex landscape of protein folding. However, the algorithm is compute-intensive for large proteins or a large number of samples, used to model a path that a protein undergoes. We present an algorithm, parallelized using OpenMP, with a speed-up of approximately twice. The results are in agreement with the ones obtained using sequential Isomap.**

***Index Terms*—dimensionality reduction, Isomap, OpenMP**

I. Introduction

In order to understand how proteins function, it is essential to characterize their conformational space. Understanding the connection between protein structure, dynamics and function can contribute to our understanding of cellular processes. The question of how the structure and dynamics of proteins relate to their function has challenged scientists for decades but has still remains open. Methods that explore the conformational landscape of proteins include Molecular Dynamics (MD) [1], Monte Carlo sampling [2] geometric-based sampling [3]-[6], Elastic Network Modeling [7], [8], normal mode analysis [9], [10], morphing [11] and several other methods. The complex, high dimensional nature of the protein conformational space requires the generation of tens of thousands, if not more, conformations per trajectory. An average protein contains several hundreds to several thousands of atoms. Therefore, this is an enormous amount of data which requires a large amount of time and space to process, store and analyze. The data is presented in a way that each row represents a different conformation. If a protein has N atoms, then every row that represents the atomic coordinates of the protein has thrice of N variables. All the numbers in the columns together make up for also the orientation that the specific conformation would have. However, due to the mutual constraints between atoms in the protein, the "real" dimensionality of the conformational space is much lower than that. Therefore, one of our preliminary tasks is to find a more efficient way to represent the data. Dimensionality reduction techniques are often used to form a lower-dimensional representation of high dimensional data.

Linear dimensionality reduction like Principal Component Analysis (PCA) and its variants may not capture the complex, non-linear nature of protein conformational landscape. Dimensionality reduction techniques are broadly classified based on the solution space they generate, as convex and non-convex [12]. Techniques described in [13] give explicit details of the various well established non-convex methods. These methods are further sub-divided into Full Spectral Techniques, the ones that perform the Eigen decomposition of a full matrix and Sparse Spectral Techniques, the ones that do the same for a sparse matrix. The latter ones have better time-complexity but these approaches are *local*. They attempt at retaining only the local structure that the sparse portions of the dataset present. On the other hand, Full Spectral Techniques, capture the covariance between all the data instances and form a more thorough representation of the structure as a whole. The Isomap algorithm [14] is a non-linear dimensionality reduction method that falls into the Convex Full Spectral category. It takes as input the distances between points in a high-dimensional observation space, and outputs their coordinates in a low-dimensional embedding that best preserves their intrinsic geodesic distances. The algorithm operates in three modes, each of which differs in the kind of data they accept (see Methods). Despite its advantage in efficient representation of molecular data [15] and [16], Isomap is computationally expensive, especially with very large, multi-dimensional datasets. To overcome this, we have implemented our version of the Mode-III of Isomap. Improvements over Isomap are presented in [17] and [18]. A similar approach is adopted but in a way that is more suited for protein data. Our method uses a distance function to calculate the distance between the points that in turn measures the similarity between the conformations that each of these points represent. The algorithm runs twice as fast as the serial implementation with comparable results. The output is a lower-

Manuscript received April 20, 2019; revised September 26, 2019.

doi: 10.12720/jait.10.4.142-147

dimensional projection that can be used later for purposes of visualization and analysis. In particular, one of our goals is to detect intermediate structures by obtaining distinct clusters using Persistent Homology as in our previous work [19].

II. Methods

The parallelization of the dimensionality reduction algorithm used here, Isomap, begins with the very first step itself of Isomap,

- The search for k-nearest neighbors of every point in the dataset. This yields a neighborhood graph. If the input matrix is N times M, the neighborhood graph would be N times k which is made to be N times N substituting the value of distance for the k neighbors of each point and infinity for the rest (except for the point itself, because distance of a point with itself would be zero).
- Next comes the computation of the shortest path tree of this graph.
- Once this is done, Multi-dimensional scaling is performed. The dimension of the matrix still remains N times N.
- Next, the Eigen vectors and values are computed, and the matrix dimension after it, is reduced to N times *i*, where *i* is the number of dimensions desired for the final embedding.

Each of these algorithms, is elucidated in detail as under:

A. KNN

The dataset here is of the order of tens of thousands points. The word 'near' finds its meaning in quantization of the features for every point with respect to every other point. A distance function [14] has been formulated to achieve this. In contrast with the sequential Isomap, each of the data points can be operated on simultaneously. The number of neighbors to be sought for each point are so chosen, so as to obtain one connected component in the resultant neighborhood graph. We opt for least such number.

B. Floyd-Warshall

Floyd-Warshall's all pair shortest path algorithm can be used for construction of the shortest path tree in case the graph is directed and non-symmetric and preserves the sense of direction by negative distances. The algorithm has a triple nested loop, of which the middle one executes independently and hence has been parallelized [20]. This parallelization of the middle loop renders the run-time of the algorithm $O(N^2)$.

```
for(int a=0; a<N; a++)
 {
  #pragma omp parallel for
  for(int b=0; b<N; b++)
        for(int c=0; c<N; c++)
          if (...)
           ...
 }
```

C. Dijkstra's Algorithm

For the protein data in this work, we have an undirected, symmetric graph with positive distances. So, Dijkstra's algorithm with multiple sources is used. We chose this algorithm over Floyd Warshall's to exploit the symmetry of the neighborhood graph; which allows for having to save only a triangular matrix in the memory, reducing the storage requirements to half. Also, it allows for an early notification if the numbers of neighbors are not enough to get one connected component. In this scenario, the algorithm is allowed to halt at the first iteration itself, knowing that there is at least one point that isn't reachable from the first. At each iteration the number of nodes to be worked with is reduced by one. The first iteration of the loop establishes the shortest paths between the first source node and the rest of the nodes. Subsequent iterations do the same for the remaining nodes. So, by the time the last node of the graph is reached, there is just one node left to work with and that is the node in question itself and so the algorithm walks out of the loop.

D. Union Find

The number of connected components in the graph so produced can be found using the classic union-find operations [21]. This portion of code comes into play if the Floyd-Warshall is used as the shortest path tree finding algorithm. While using Dijkstra's algorithm, the first iteration itself tells whether the chosen number of neighbors are enough to find one connected component, if they aren't the program is designed to halt prompting for more neighbors. The largest component found first is chosen to embed. Since we aim at only reducing the dimensionality of the available data in this step, we find as many neighbors of each point as would be required to obtain one connected component. This gives a thorough coverage of the conformational space.

E. MDS

Next, the classical version of Multi-Dimensional Scaling is performed on the data thus far [14].

F. Eigenvectors and Eigenvalues

To obtain a rigid transformation, eigenvectors and values are to be found in as many dimensions as one wishes to observe. The power method is used to find the dominant eigenvalue and its corresponding eigenvector in a loop that stretches for as many iterations as the number of dimensions. The number of dimensions has been set to five to observe the residual variance, and the first three dimensions are embedded and written to a file.

III. Results and Discussion

The proteins we used here range between 9 and around 200 amino acids. Each data point here, represents a conformation of the molecule generated using Molecular Dynamics simulations. We typically use an average of twenty two thousand such conformations. Table I gives more details about the molecules used. As mentioned earlier, OpenMP has been employed to parallelize the

algorithms rewritten in C. Due to the need of computing all pair shortest paths, the time complexity of Isomap is $O(N^3)$. In our version, because we harness the capabilities of a modern multi-core CPU, it has been brought down to $O(N^2)$ when the Floyd Warshall method of shortest path evaluation is used and $O(N^2 \log N)$ when the Dijkstra's method is used for this purpose (see Methods).

In order to compare the performance of our method to that of the serial Isomap, we compared the residual variance in each sample molecule for a number of dimensions for each of the two embeddings. Residual variance is a measure of how different the generated embedding is with the original data. With increasing dimensionality, the number received gives an estimate of how much of the variance in the data is still unexplained. It is computed as the unit difference from the squared correlation coefficient at each dimension. The data matrix of N points is first subject to thorough analysis, by obtaining the shortest path tree that represents the connectivity of each of these points with every other point. The low-dimensional embedding obtained by Eigen-decomposition of this tree forms the matrix with which the correlation coefficient is calculated. The residual variance is supposed to decrease with increasing dimensionality, as each dimension explains a fraction of the variance. A sphere when observed in two dimensions would be a circle. So, more dimensions are to be taken into account to reveal the true structure of any object. More the dimensions, less is the unaccounted information. For instance, for Oxytocin, the residual variance for the first dimension in Sequential Isomap (refer Table II) is 0.448 which means 44.8 percent of the data is still unexplained, it decreases to 30 percent in the second dimension and it continues to decrease as more dimensions are added. The trend is reported for the first four dimensions for both serial and parallel versions. Similarly, in the Parallel Isomap (refer Table III), the corresponding values for Oxytocin are 66.7 percent, 54 percent and they continue decreasing as more dimensions are observed. The difference in these values can be alluded to the way Eigen-decomposition is performed in our version of the algorithm. A three dimensional projection of the embeddings generated by the serial and parallelized version of the algorithm is presented in Fig. 1 and Fig. 2 in the same order as they appear in Table I. The first three dimensions were embedded.

TABLE I. DETAILS ABOUT DATA SAMPLES

Molecule	Name	No. of atoms	Rows[1]	Columns[2]	k[3]	RMSD Error
4did	CDC42, complex-ed with GDP	1880	20000	534	3	432.98
Hgalanin	Human Galanin,a neurope-ptide	434	22750	363	18	532.76
Pgalanin	Porcine Galanin	442	22750	351	8	489.62
Neurome-dlin	a mammal-ian peptide	116	22750	123	7	312.45
Oxytocin	A hormone	135	22750	111	8	328.98
Vasopressin	A hormone	140	22500	111	6	516.23

1. no. of conformations used
2. no. of features used to represent one conformation of the molecule
3. no. of neighbors that produced one connected component

TABLE II. ISOMAP TRENDS IN RESIDUAL VARIANCE WITH INCREASING DIMENSIONS

Molecule	1	2	3	4	Time (min)
4did	0.012	0.006	0.005	0.003	195
Hgalanin	0.544	0.383	0.269	0.184	380
Pgalanin	0.234	0.142	0.138	0.095	362
Neuromedlin	0.665	0.544	0.428	0.349	245
Oxytocin	0.448	0.3	0.199	0.144	253
Vasopressin	0.533	0.246	0.184	0.144	282

TABLE III. PARALLEL-ISOMAP TRENDS IN RESIDUAL VARIANCE WITH INCREASING DIMENSIONS

Molecule	1	2	3	4	Time (min) 8 cores	Time (min) 4 cores	Time (min) 2 cores
4did	0.05	0.03	0.01	0.01	92	132	156
Hgalanin	0.75	0.69	0.62	0.49	175	255	306
Pgalanin	0.42	0.33	0.26	0.06	152	243	290
Neuromedlin	0.97	0.78	0.64	0.34	125	165	198
Oxytocin	0.67	0.54	0.38	0.25	110	170	205
Vasopressin	0.83	0.66	0.28	0.11	162	190	230

Another proof of quantitative validation comes with the least RMSD (Root Mean Square Deviation) computation for the two embeddings. This method first eliminates the translation component by shifting the center of mass of the two embeddings to the same place, and then it finds the optimal rotation between the two sets using Singular Value Decomposition. The difference between the two structures is reported in the form of an error. Let the two matrices being compared be, A and B with dimensions N times M. First, the centroid of the two is found, both the molecules are dragged to the origin by subtracting from each point the value of the centroid. The RMSD between the structures can then be calculated as under:

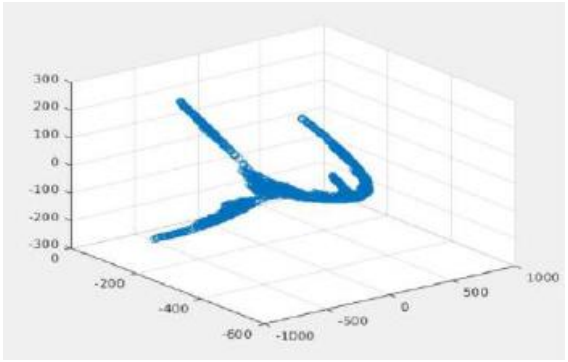
$$\sqrt{1/N} \sum_i \sum_j (a_{ij} - b_{ij})^2 \quad (1)$$

here, a_{ij} and b_{ij} are the corresponding elements of A and B respectively. The first summation goes from i=1 to N and the second from j=1 to M. The number returned by (1) indicates how similar the two structures are. It is enlisted for the various protein molecules in Table I in the column named reconstruction error. Our implementation offers a good coverage even in lower dimensions.

The last column of Table II shows the performance of the serialized code in Matlab. The last three columns of Table III do so for our version of the algorithm with scalability, depending on how much power of a CPU is harnessed. As is evident, the time consumption is much less for the parallelized code. The time consumption in the parallelized version depends on the number of cores of the CPU being employed for the computations. The results of Table III represent the time taken by a system with two quad-core processors, which means at a time eight threads are at work simultaneously for the parallelized portions of the code. The number of threads can be set for any program written in C using OpenMP [22]. The chunk of pseudo-code in Methods shows how to parallelize a simple for loop and how many threads one wishes to use can be set before executing the program, in the shell, using the command:

OMP_NUM_THREADS=8

Since the number of cores here are eight, the performance isn't affected much by using more threads for this configuration of a machine. The average speed-up drops to about 1.5 when only four threads are at work and about 1.248 for two threads.



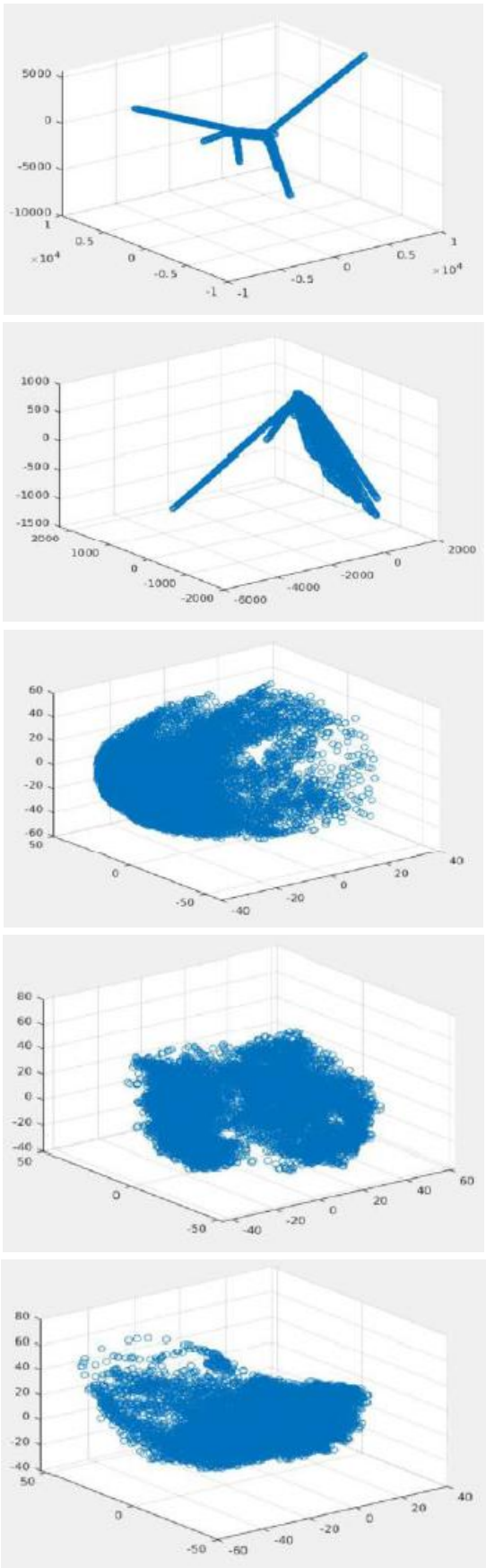
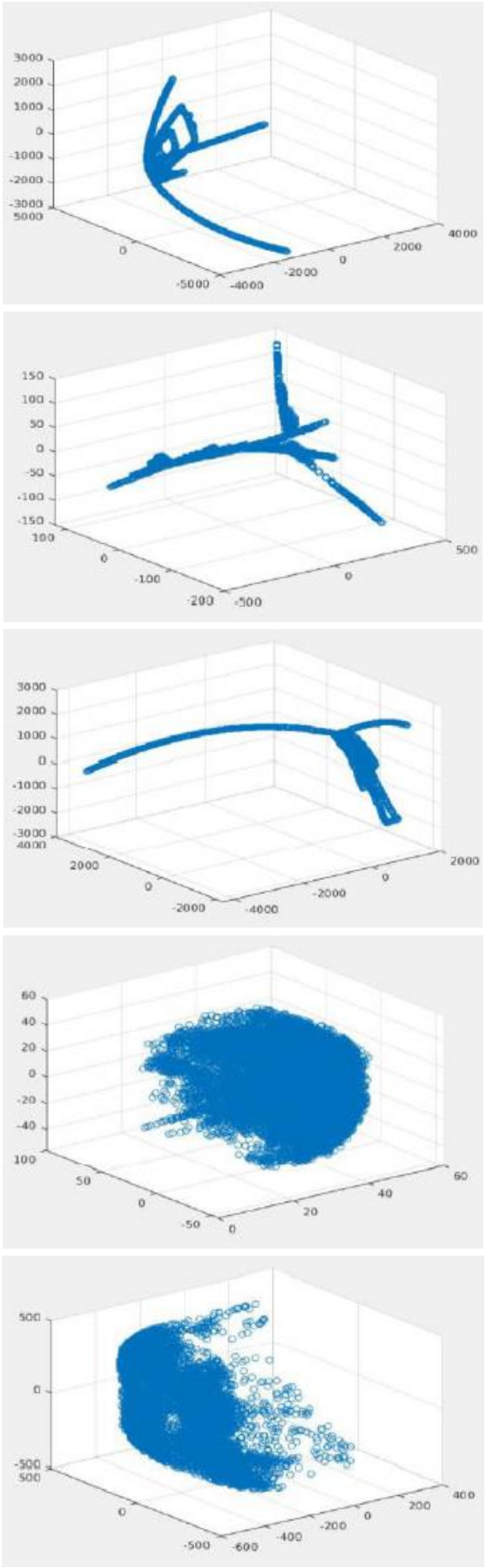


Figure 1. Isomap embeddings.



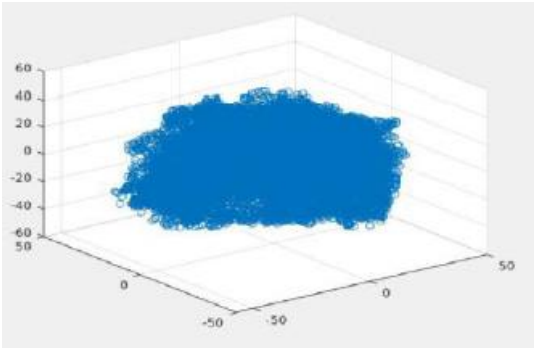


Figure 2. Parallel-Isomap embeddings.

IV. Conclusions and Future Work

Besides the improvement in average time complexity due to betterment in computation of the shortest path tree, the algorithm offers better speed at every stage, when each function is compared in isolation. In future, we plan on attempting to do this even faster. The algorithm in its current form, only reduces the features of the data. Coming up with a way to reduce the noisy and redundant points themselves, is an area of ongoing research. As mentioned earlier, the Isomap algorithm is compute intensive. It executes in a way that hogs the system's resources, leaving all the other applications running on the system starved. In comparison, our parallelized version, efficiently makes use of the available memory, without having to write anything on the disk and hence doesn't hamper any other processes running on the system. The amount of data to be operated on is so large here, that programming platforms for vectorized data like Matlab and R sometimes cannot allocate memory for it. The coordinate data here is about 80 MB and the intermediate data files can be about 4GB. In situations like these, our version of the algorithm becomes the only resort.

References

[1] D. Case, *et al.*, "The amber biomolecular simulation programs," *J. Computat. Chem.*, vol. 26, pp. 1668-1688, 2005.

[2] S. Kirkpatrick, C. D. G. Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, 1983.

[3] N. Haspel, M. Moll, M. Baker, W. Chiu, and L. E. Kavraki, "Tracing conformational changes in proteins," *BMC Structural Biology*, suppl. 1, p. S1, 2010.

[4] B. Raveh, A. Enosh, O. Furman-Schueler, and D. Halperin, "Rapid sampling of molecular motions with prior information constraints," *Plos Comp. Biol.*, vol. 5, no. 2, p. e1000295, 2009.

[5] A. Shehu and B. Olson, "Guiding the search for native-like protein conformations with an ab-initio tree-based exploration," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1106-1127, 2010.

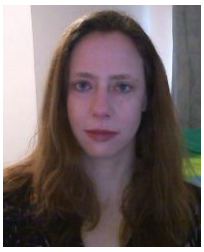
[6] I. Al-Bluwi, M. Vaisset, T. Siméon, and J. Cortés, "Modeling protein conformational transitions by a combination of coarse-grained normal mode analysis and robotics-inspired methods," *BMC Structural Biology*, vol. 13, suppl. 1, p. S2, 2013.

[7] W. Zheng and B. Brooks, "Identification of dynamical correlations within the myosin motor domain by the normal mode analysis of an elastic network model," *J. Mol. Biol.*, vol. 346, no. 3, pp. 745-759, 2005.

[8] L. Yang, G. Song, and R. L. Jernigan, "Protein elastic network models and the ranges of cooperativity," *National Academy of Sciences*, vol. 106, no. 30, pp. 12347-12352, 2009.

[9] G. Schroeder, A. T. Brunger, and M. Levitt, "Combining efficient conformational sampling with a deformable elastic network model facilitates structure refinement at low resolution," *Structure*, vol. 15, pp. 1630-1641, 2007.
[10] V. Frappier, M. Chartier, and R. J. Najmanovich, "Encom server: Exploring protein conformational space and the effect of mutations on protein function and stability," *Nucleic Acids Research*, vol. 43, pp. W395-W400, 2015.
[11] D. Weiss and M. Levitt, "Can morphing methods predict intermediate structures?" *J. Mol. Biol.*, vol. 385, pp. 665-674, 2009.
[12] P. Boyd and L. Vandenberghe, *Convex Optimization*, 1st ed., New York, NY, USA: Cambridge University Press, 2004.
[13] L. V. D. Maaten, E. Postma, and J. V. D. Herik, "Dimensionality reduction: A comparative review," *J. Mach. Learn. Res.*, vol. 10, pp. 66-71, 2009.
[14] J. Tenenbaum, V. D. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319-2323, 2000.
[15] P. Das, M. Moll, H. Stamati, L. Kavraki, and C. Clementi, "Low dimensional, free energy landscapes of protein folding reactions by nonlinear dimensionality reduction," *Proc. Nat. Acad. Sci.*, vol. 103, no. 26, pp. 9885-9890, 2006.
[16] A. Vajdi and N. Haspel, "A new DP algorithm for comparing gene expression data using geometric similarity," in *Proc. IEEE International Conference on Bioinformatics and Biomedicine*, Washington D.C., USA, 2016, pp. 1157- 1161.
[17] V. D. Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," *Advances in Neural Information Processing Systems*, 2003.
[18] T. Ameet, S. Kumar, and H. Rowley, "Large-scale manifold learning," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008.
[19] D. Luo, E. González, and N. Haspel, "Detecting intermediate protein conformations using algebraic topology," *BMC Bioinformatics*, vol. 18, suppl. 15, p. 502, 2017.
[20] Z. Yan and Q. Song, "An implementation of parallel Floyd-Warshall algorithm based on hybrid mpi and openmp," in *Proc. International Conference on Electronics, Communications and Control*, 2012, pp. 2461-2466.
[21] R. Sedgewick and K. Wayne, *Algorithms*, 4th ed., Addison-Wesley Professional, 2011.
[22] A. Grama, A. Gupta, G. Karypis, and V. Kumar, *Introduction to Parallel Computing*, 2nd ed., Addison-Wesley Professional, 2003.

Arpita Joshi is currently a PhD candidate in the Department of Computer Science at University of Massachusetts at Boston. She graduated with a masters in Computer Science from the same department in 2016. In addition, she has a Bachelor's in Electronics and Communication Engineering from Rajiv Gandhi Technical University, India. Her research interests are High Performance Computing, Algorithm Design, Bioinformatics, Compiler Design and Communication Systems.



Nurit Haspel is an associate professor in the Department of Computer Science at the University of Massachusetts, Boston. Before that she was a postdoctoral re-search associate with the Physical and Biological Computing group at Rice University. She graduated from the Department of Computer Science in Tel Aviv University, Israel, where she was a member of the structural bioinformatics group.