Protein Dynamics Reconstruction from Unordered Images

Christian Choe
Department of Chemical Engineering & Electrical Engineering
cachoe@stanford.edu

Min Cheol Kim
Department of Electrical Engineering
mincheol@stanford.edu

***Abstract*—In this project, we describe the steps to reconstruct a frame order from a series of images with the ultimate goal being the reconstruction of protein trajectories. We apply nonlinear dimensionality reduction to generate a graph of frames with edges representing distances. From the graph we reconstruct our frame order through various means such as finding the long path in the minimum spanning tree. We show promise of this approach through reconstructing the frame order of simple biological simulations.**

Keywords—protein dynamics; reconstruction; nonlinear dimensionality reduction

I. Introduction

When we think of videos, we almost always take the frame order for granted. After all, someone must have generated that video in some way. However, if one was given a bag of randomly ordered frames, how would one go about ordering them to be coherent?

This problem is inspired by the data generated by the Linac Coherent Light Source (LCLS) facility at the Stanford Linear Accelerator Center. This facility can take an "image" of a biological molecule, acquiring data such as the (x, y, z) coordinates of all the atoms and the angles between the important bonds present in the backbone. However, because the procedure destroys every molecule after each "snapshot," we end up with many images of the biological molecule in different states and conformations that are part of their dynamics trajectory. Here, the dynamics trajectory is analogous to the frame order of a video; at the LCLS facility, we get frames of the video without the information about their relative orders.

The motivation is to order these images so that we can learn something about the dynamics about the protein. For this project, we approached this problem step by step, first by applying our algorithm on very simple simulated datasets and then moving onto the reconstruction of shuffled biological molecule simulations.

II. Moitivation and Conceptual Framework

Before we go further in depth, this section will outline our conceptual framework that guided the project. Our method involved a type of dimensionality reduction and finding a trajectory within that reduced dimensional space. Consider a cube rotating through space, as represented in Figure 1. The cube starts out in the left middle position and rotates through the space to arrive at the right middle position. Let us say that these images are 100x100 pixels large. Then, we are talking about 10000-dimensional vectors for each frame.

However, now consider what is actually changing through series of the images. The only thing that is changing is θ, the angle that describes the progress of the rotation. If these images were all shuffled and we did not know the order, but if we performed the correct type of dimensionality reduction from the 10000-dimensional vector space to a single number, we could simply sort by the number we get (which would be representative of θ) and we would have our reconstruction of the ordering.

The project is based on the hypothesis that a similar approach may be successful in reconstructing a series of biological data: dimensionality reduction followed by a simple algorithm for the trajectory reconstruction.

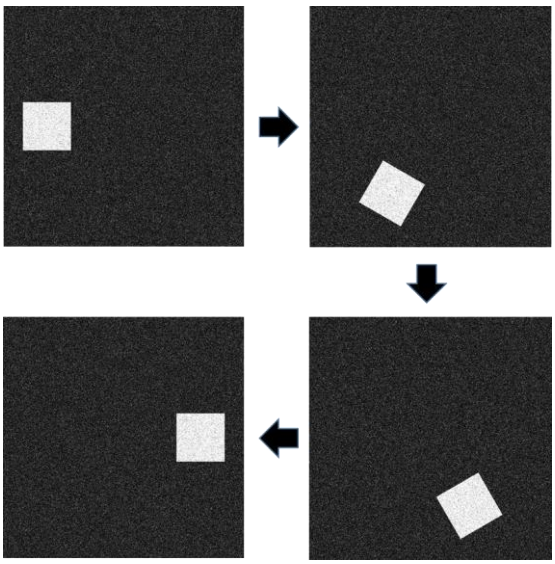

Fig. 1. Rotating Cube

III. Methods

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use

the scroll down window on the left of the MS Word Formatting toolbar.

A. Dimensionality Reduction

The method we utilized for nonlinear dimensionality reduction was Isomap [1]. We first vectorize all images into a 2-D matrix and apply Isomap which consists of the following procedure:

1) Determine the neighbors of each point
2) Construct neighborhood graph
3) Compute shortest path between nodes
4) Compute lower-dimensional embedding

For the Isomap parameters, n-neighbors and n-components, we discuss the optimization of the them below.

B. Trajectory reconstruction in the reduced dimension

Using the values generated by Isomap we create a fully connected graph between all the images where the edge weight between each image is the euclidean distance of their isomap values. In order to prune the graph we find the minimum spanning tree to minimize the distance between closely positioned frames.

From the minimum spanning tree there are many approaches to reconstruct the frame order. We used two different methods for reconstruction:

1) Greedy reconstruction

Find the two closest frames and connect them. Then continue to search for the closest frames to the ones connected and attach to the end. Iterate until all frames are used and the fine linear sequence is the reconstruction.

2) Longest path in minimum spanning tree

While many frames are lost be pruning the branches, the reconstruction tends to perform well. (Figure 2)

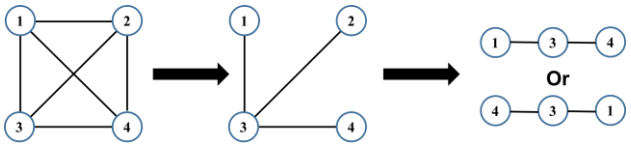

Fig. 2. Longest path in the minimum spanning tree

C. Trajectory reconstruction in the reduced dimension

We considered two metrics for evaluating our reconstructed trajectory, the RMSD error between frames and the frame alignment score.

1) RMSD error between frames

This metric was calculated by calculating the difference matrix, defined as the matrix whose rows represent the difference between each adjacent pairs of rows in the original matrix. If our original matrix represented 5 frames (rows) each with 3 components (columns), the difference matrix would contain 4 rows with 3 components where each row is the difference between adjacent rows in the original matrix. We then compute element-wise sum of squares of this matrix, divide by its dimensions (in this case 4 x 3), and square root the result. In other words, this number is the Frobenius norm of the difference matrix squared, divided by the number of elements in the matrix, and then square rooted. With this measure, larger numbers indicate worse reconstruction.

2) Frame alignment score

The alignment score is based on a similar problem in bioinformatics, DNA alignment. Both the Needleman-Wunsch and Smith-Waterman algorithm was implemented to given an alignment score between two frame orders where one order is the true order and the other one is the calculate one.[2][3] In addition, there are many parameters that can be fine-tuned when calculating the alignment score such as the score matrix, gap penalty, and extending gap penalty. In order to keep our model simple made our scoring matrix equivalent to an identity matrix. Every correct frame match is plus one while every mismatch is plus zero. We also made the gap penalty and extending gap penalty 0.1 for simplicity. Although both alignment algorithms were implemented, we focused on using Needleman-Wunsch for the alignment score since our matrix did not contain any negative numbers which is one requirement for using Smither-Waterman. With this measure, larger numbers indicate better reconstruction.

IV. Results: Proof of Concept: Square and Cube

As a proof of concept for the frame order reconstruction, we made a two simple simulations for testing. The first simulation consists of a square orbiting around a central point. The simulation was made by first creating an image with a square drawn on which was 500 by 500 pixels. The frames of the video were made by continually rotating the image around the central point.

Using the rotating square simulation, we validated the use of nonlinear dimensionality reduction by examining just the first principal value from the Isomap. By simply sorting with this value, we are able to reorder the whole sequence of images into its original frame order.

The second simulation we developed a 3D cube simulation to test our reconstruction. While the square simulation also lets us test reconstruction, due to the simplicity of the model and lack of similarity with protein data, it was not the best choice in order to validate our reconstruction methods.

Each frame of the simulation consists of a list of x, y, and z, coordinates where all the points make up a 3D grid of a cube. Coordinates were used since protein simulation data contains a list of coordinates for each atom. Using the cube simulation, after creating a totally connected graph we attempted various methods of reconstruction.

The first method we tried was greedy reconstruction, which failed to work in most situation. Unless we sampled the simulation with very small time steps, the algorithm failed to give reasonable reconstruction. As a result we decided to first obtain a minimum spanning tree since we want all the adjacent frames to be connected. Since most of the branches of the minimum spanning tree already contained correct sequences of frames, we decided to take the longest path in the minimum spanning tree as our reconstructed frame order.

V. Results: Small Peptide Simulation Data

To verify that our method is feasible with biological data before we get to true data from SLAC, we acquired a very simple simulation of an AK peptide starting in a linear, unfolded state to a helical structure. The AK peptide is a short sequence of amino acids that is relatively easy to study compared to other large biomolecules that may be studied at SLAC.

This simulation was acquired by molecular dynamics techniques, which we will not go into detail here. This simulation gives us the similar type of data that we would get from the LCLS facility at SLAC, namely the X, Y, Z coordinates of every single atom in each frame. The entire simulation is quite long, and for the purposes of this project, we focused on the first one thousand frames of the simulation.

In other words, the simulation we will be concerned with contains a thousand frames, each frame with information on the position of each atom and the important angles that define the protein backbone.

A. Using raw X, Y, Z coordinates from the simulation

Our very first attempt at this simply used the raw X, Y, Z coordinates from the shuffled AK peptide simulation. AK peptide contains 229 atoms, so our design matrix was of the shape n x p where n = 1000 and p = 229 * 3 = 687. We then used Isomap for dimensionality reduction and the longest path in minimum spanning tree for the frame order reconstruction. Our reconstructed trajectory was evaluated using both frame order score metric and the RMSD error metric.

We adjusted the hyperparameters and had seemingly the best results when n_components and n_neighbors for the Isomap algorithm was 30 and 40, respectively. We got 0.13 for RMSD error between frames and 11.1 for the frame alignment score.

B. Using phi, psi angles from simulation

After we had our initial results using the raw X, Y, and Z coordinates, we had a meeting with Dr. Lane. He pointed out an important red flag that our Isomap approach may be learning the slow "tumble" (a slow, steady, rotation that the entire molecule undergoes during the simulation) rather than the actual dynamics of the protein. This would not be very helpful, since the SLAC data does not have any information on the orientation of the protein. To remove this "tumbling" information from our dataset, we first attempted the same approach as above (Isomap, then reconstruction via longest path in minimum spanning tree) but by using phi/psi angles. These angles give information about the conformation of the protein backbone. Further reading can be found in [4].

We got worse results than we did from using the raw X, Y, Z coordinates, with the RMSD error of 0.297 and frame alignment score of 4.9 when we used n_components = 30 and n_neighbors = 40 for hyperparameters. The reconstructed video also was visually not as satisfactory as using the raw X, Y, and Z coordinates of the atoms.

C. Using X, Y, Z coordinates after aligning every frame to a reference frame

Another approach to removing the "tumbling" information from our dataset and making it more realistic was to align every frame to a single reference frame, so that every protein lies in the same orientation in every frame. This was done using the MDTraj package (same package for reading and writing these simulation data); after the frames were shuffled, all the frames were aligned to the very first frame in the shuffled dataset.

We then performed the identical procedure as the two sections above, and got results much better than using phi/psi angles. We got 0.142 for the RMSD error between frames and 7.8 for the frame alignment score. The resulting video was also visually satisfactory. We see that in Table 1 that both metrics indicate that using raw X, Y, Z coordinates performed the best, then aligned X, Y, Z, and then the phi/psi angles. This is very consistent with our visual results (Figure 3).

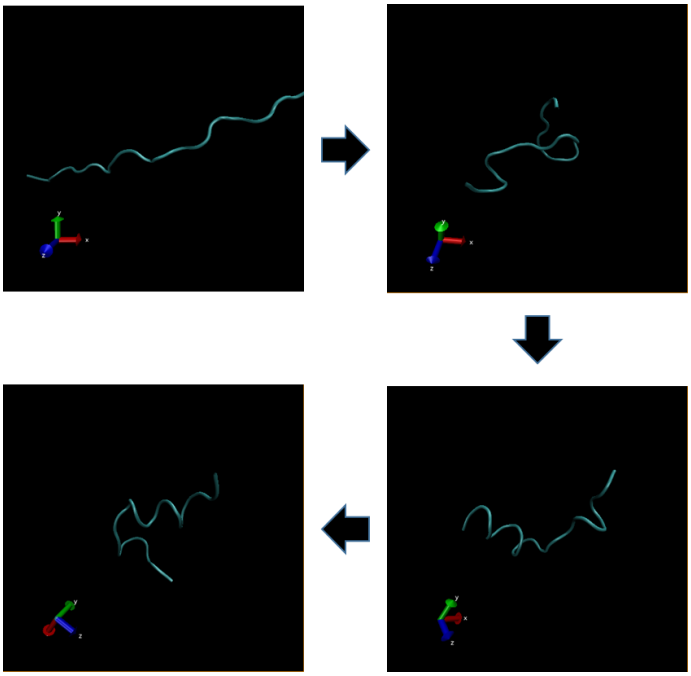

Fig. 3. Snapshots of trajectory reconstruction with aligned X, Y, Z coordinates

TABLE I. SUMMARY OF SCORES FOR THE THREE AK PEPTIDE APPROACHES

	RMSD error between frames	Frame alignment score
Perfect reconstruction	0.102	1000
Raw X, Y, Z	0.13	11.1
Phi/Psi angles	0.297	4.9
Aligned X, Y, Z	0.142	7.8

D. Hyperparameter Screening with 100 and 500 frames

After we acquired our results for the three approaches, we attempted to sweep a large number of ranges for the hyperparameters for the Isomap algorithm: n_components and n_neighbors. We were only able to sweep values for frame counts of 100 and 500 due to computational constraints. The performance at each value of n_components and n_neighbors was measured both with the RMSD error between frames and the frame alignment score. Some of the results are shown below. In general it was difficult to draw conclusions exactly what set of hyperparameters yielded the best results, but it seemed that for 500 frames, the best result with RMSD error between frames came from using n_components = 28 and n_neighbors = 23. With n_neighbors fixed at 23, we generated the following graph (Figure 2). With n_components fixed at 28, we generated another graph (Figure 3).

When using frame alignment score, for 500 frames, the best result came from using n_components = 11 and n_neighbors = 22. Similar graphs as with the RMSD error score follows.

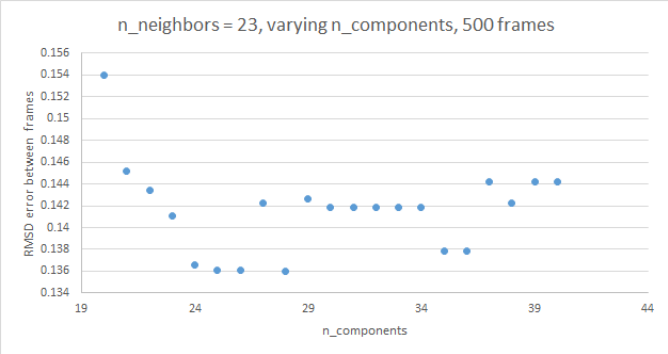

Fig 4. n_neighbors fixed, varying n_components, RMSD error metric

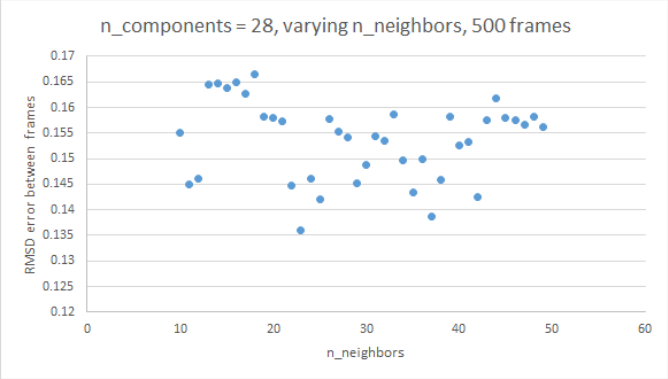

Fig 5. n_components fixed, varying n_neighbors, RMSD error metric

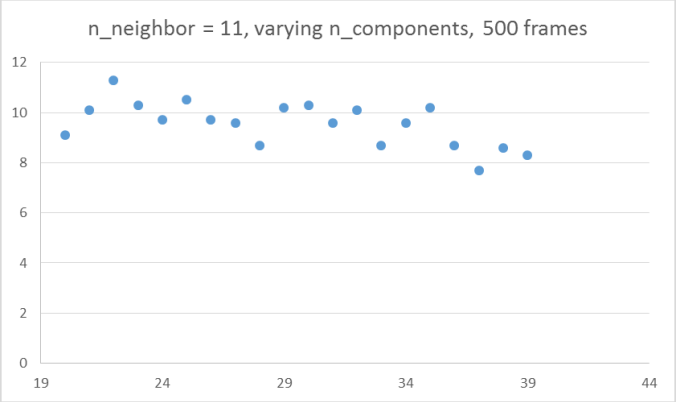

Fig 6. n_neighbors fixed, varying n_components, frame alignment metric

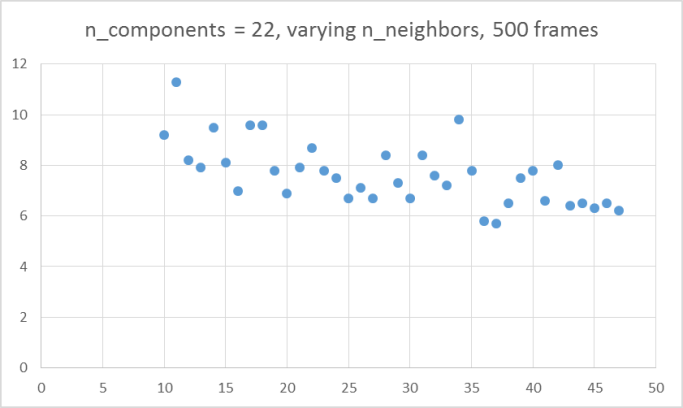

Fig 7. n_components fixed, varying n_neighbors, frame alignment metric

VI. CONCLUSION AND FUTURE WORKS

Nonlinear dimensionality reduction via Isomap then reconstruction via a graph algorithm seems like a promising approach to reconstructing the protein dynamics trajectory. Specifically, using X, Y, Z coordinates from aligned frames seem to perform much better than using phi/psi angle information - this could be due to the fact that phi/psi angles only give information on the conformation of the protein backbone, whereas X, Y, Z coordinates give information about the every single atom present in the frame.

More improvements can be made both on the dimensionality reduction and on the trajectory reconstruction, by considering different algorithms (e.g., rearranging the minimum spanning tree, t-NSE instead of Isomap). In addition, improvement can be made on the evaluation metric to more accurately measure the quality of reconstruction for optimal hyperparameter optimization.

Another feature that may be useful aside from X, Y, Z is the total energy of the frame. Biological molecules have a thermodynamic tendency to move to lower energy states, so this may be relevant information in reconstructing the protein trajectory.

NOTES

The code that reconstructs the molecular dynamics simulation data produces a .dcd trajectory file. The file can be opened in VMD along with the corresponding PDB file to view.

ACKNOWLEDGMENT

We would like to thank Dr. TJ Lane for providing us with the molecular dynamics simulation data as well as mentorship. We would also like to thank Professor Gordon Wetzstein for teach EE368 and providing a project opportunity.

REFERENCES

[1] Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. science, 290(5500), 2319-2323.

[2] Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology, 48(3), 443-453.

[3] Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. Journal of molecular biology, 147(1), 195-197.

[4] Kleywegt, Gerard J., and T. Alwyn Jones. "Phi/psi-chology: Ramachandran revisited." Structure 4.12 (1996): 1395-1400.

APPENDIX

Both Christian Choe and Min Cheol Kim worked on experimenting with and designing the algorithm we used to reconstruct the frame order. Christian then focused on building the cube simulations to ensure that our algorithm is robust, and Min Cheol focused more on applying the algorithm on the biological dataset.