Automation of Protein Trajectory Analysis and Critical Event Detection

Lin Kuang[1]

Abstract:

My DMP project this summer lies in the field of protein conformational trajectory analysis. It emphasizes the automation of the comparision of various protein conformational trajectories, as well as critical event detection. I develop a novel algorithm that makes use of the interatomic distance and contact information to detect critical intermediate protein conformations. The algorithm then can further identify characteristics of various conformational trajectories and categorize them.

1. Introduction:

A necessary issue to address in the process of drug design is the modeling of protein flexibility. This issue is computationally very expensive for meaningful macromolecules such as proteins. A lot of work and collaboration from many scientific communities goes into adding flexibility to proteins. Much computational effort is devoted to designing conformational search algorithms that can deal with the time and space complexity of protein flexibility. While results from both academic and industrial protein flexibility projects are accumulating every year, little work has gone into regulating and rating the results of conformational search techniques.

Justifying the success of conformational search techniques remains somewhat subjective. Therefore, it is very important to quantify the comparison between the protein conformations produced by different conformational search techniques. It is necessary to address the question of whether it is possible to extract some essential information from conformational trajectories that claim to be adequate samples of a protein's conformational space. How hard is it to define and detect critical events in a trajectory? How can one objectively rank/compare trajectories based on their respective essential information?

There is no satisfying answer to the above questions in the current scientific literature. An extensive exploration of fields such as computer graphics, video surveillance, statistics, information theory, machine learning, and physical chemistry in the topic of critical event detection in signal and motion did not reveal any relevant techniques for the problem of trajectory analysis. Therefore, developing novel methods for protein conformational trajectory analysis is necessary.

---

[1] Undergraduate student at the Department of Electrical Engineering and Computer Science Department at UC Berkeley. E-mail: klin7@berkeley.edu
My DMP Mentor: Professor Lydia Kavraki, at the Department of Computer Science and Department of Bioengineering at Rice University.

In this project, I develop an algorithm that automates critical event detection and trajectory comparison through interatomic distances [1] and contact information. I test the success of this algorithm on a simple protein model of 60 residues and correctly detect the formation of bends and loops across trajectories. My next aspiration is to extend the capabilities of the method to bigger proteins and larger trajectory ensembles.

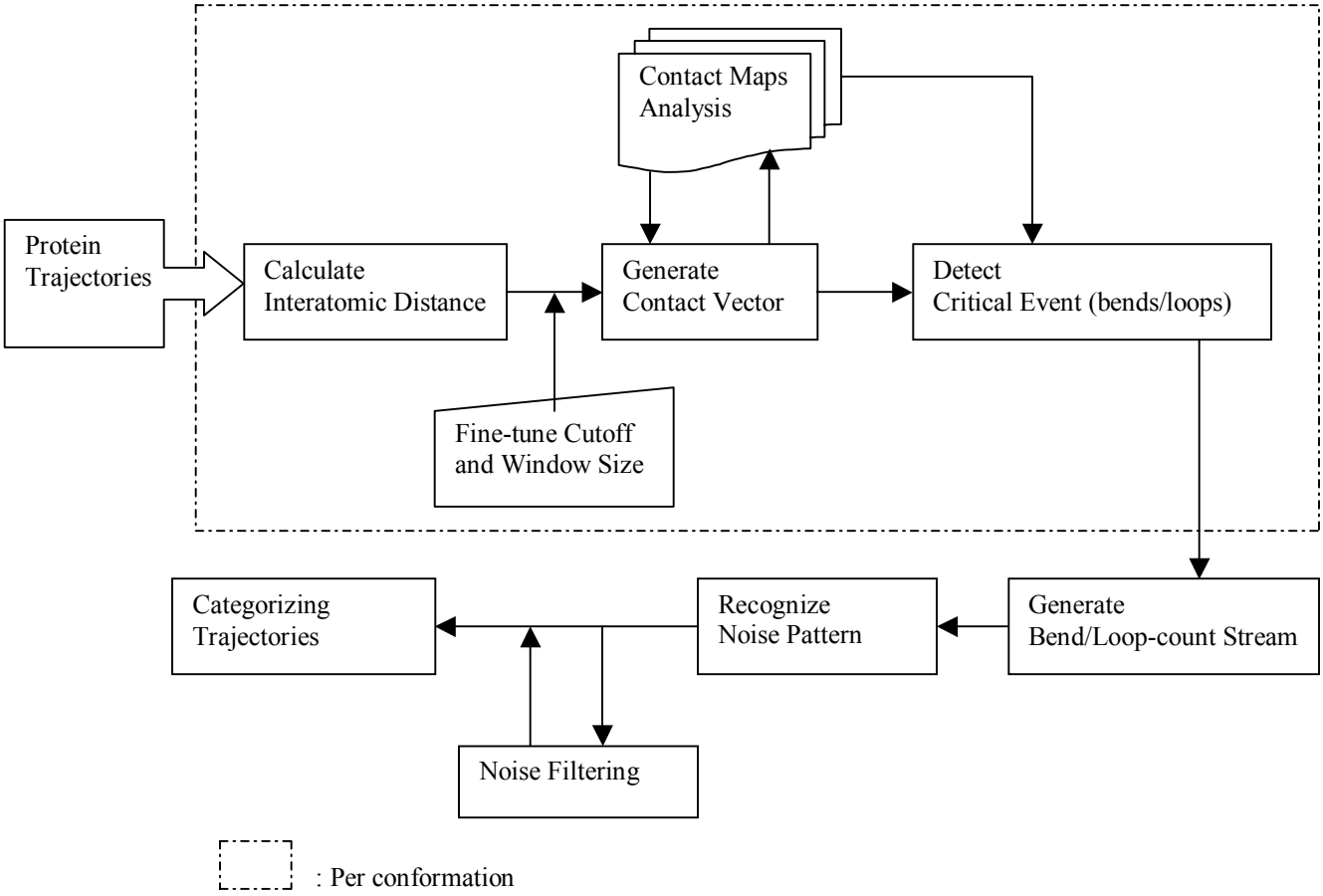
2. Problem Statement:

Detect presence and amount of bends and loops in trajectories.

3. Algorithm:

The algorithm I developed can be summarized into a flow chart as shown in Figure 3.0.1.

Figure 3.0.1 Algorithm flow diagram



The calculation and fine-tuning of the contact vectors is discussed in 3.1. While the binary and real contact map analysis and critical even detection will be discussed in 3.2.

Then the last section, section 3.3, will be devoted to extend the analysis from the conformation level to the trajectory level, and noise pattern recognition and noise filtering through the trajectory will also be discussed there.

3.1 Contact Vector Calculation and Fine-Tuning the Cutoff and Window Size

3.1.1 Interatomic Distance and Contact Vector Calculation

In this algorithm, interatomic distance is used as a key feature to distinguish different conformations for the protein throughout the trajectory. A trajectory is a continuous sequence of frames, where the protein under study has its corresponding conformation, and the interatomic distances are calculated pair-wise for all atoms in the protein.

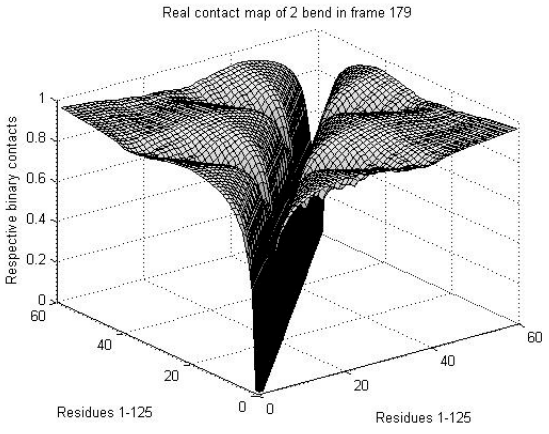
Based on the interatomic distances, a binary and a real contact vector can be generated using the following equations. The formula in Eqn 3.2 ensures that the real vector always lies in the range between zero and one.

if distance > cutoff and not adjacent within windowSize
contact_vec = 0.0
otherwise contact_vec= 1.0;
Eqn. 3.1 Binary vector calculation

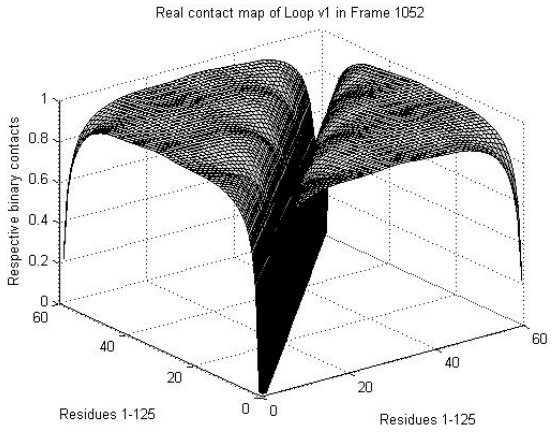
if (cutoff/distance) > 1 and not adjacent within windowSize
contact_vec= 1.0
otherwise contact_vec = cutoff/distance;
Eqn 3.2 Real vector calculation

Once the contact vector for a conformation is calculated, we generate a rough real or binary contact map for this protein conformation. Several such real and binary contact maps are show in Figure 3.1.1.

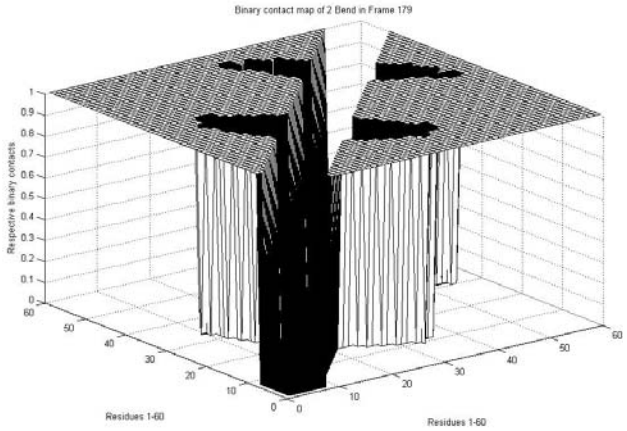
Figure 3.1.1 Real and binary contact map for 2bend, and loop (version 1)



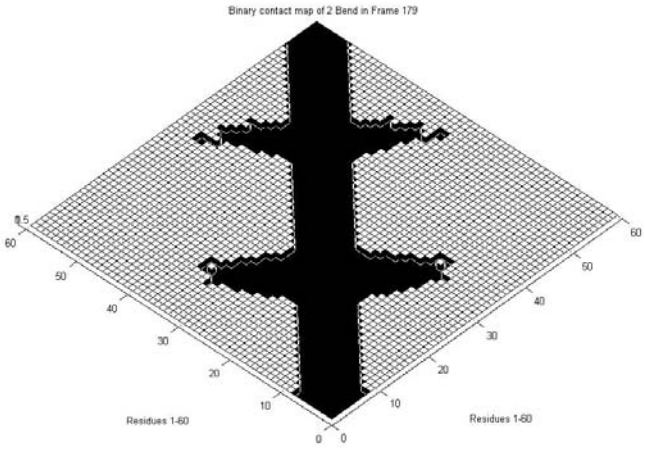
(1) 3D 2 bend real with 0.1 cutoff



(2) 3D loopv1 real with 0.1 cutoff



(3) 3D 2bend binary with 0.8 cutoff



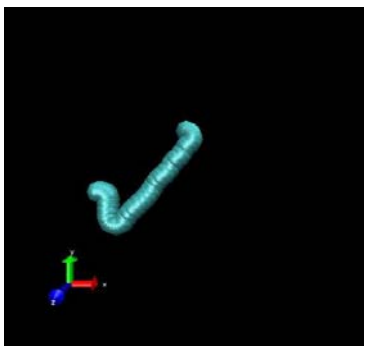
(4) 2D 2bend binary with 0.8 cutoff

3.1.2 Fine-tuning the Cutoff and Window Size

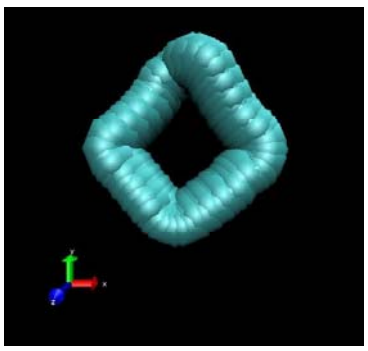
Binary and real contact maps are used as a visualization tool to help better generate and analyze the real and binary contact vectors. As seen in Figure 3.1.1, local structure that is not interesting and therefore classifies as noise exists commonly throughout all contact maps. In order to get rid of them, one approach is to fine-tune the cutoff and window size.

We observe that the window size and cutoff in each binary contact map are dynamically related. The ideal ratio of cutoff/windowSize for every binary map in any trajectory is approximately 0.1. This is as predicted because the interatomic distance of each of the neighboring atoms in the model can be treated as a constant, and the motion of atoms in a protein is similar to the motion of polymers linked together by a chain. The models under study include trajectories with critical conformations of symmetric one bend, two bends, three bends, loops, and asymmetric one bend. Three of the models are shown in Figure 3.1.2. Thus, intuitively, if a model whose interatomic distance varies, an additional parameter will be needed to add to the cutoff/windowSize formula to achieve a constant ratio.

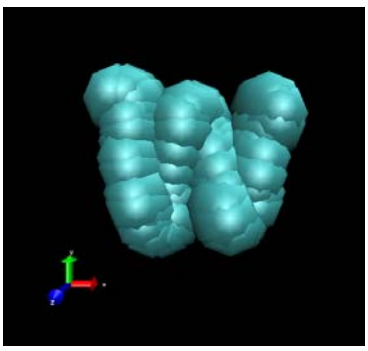
Figure 3.1.2. Snap shots of the models under study.



(1) Asymmetric 1 bend



(2) Loop (version 1)



(3) 3 bends

One of the biggest challenges now lies in finding one and only one proper cutoff and window size that will serve to identify all the conformations for the protein models under study. This is one of the key requirements to achieve the automation in protein conformation comparison. In order to do so, in addition to contact map analysis, we apply a numerical analysis technique to further study the contact vectors and fine-tune the cut-off and window size. By calculating the average of the real contact vectors, one of the observations is that the probability for atoms to be in contact in the conformations for loops and asymmetric one bend is lower than that for the other conformations. This observation is summarized in Table 3.1.2.

Table 3.1.2

Critical Conformation	Average Contact Vector
1 Bend version 1	0.158904
1 Bend version 2	0.154751
2 Bend	0.177857
3 Bend	0.189409
Loop version	0.149280
Loop2 version	0.150872
Asymmetric 1 Bend	0.134184

In this way, we find that the cutoff/window size ratio needs to be adjusted lower than 0.1. The fined-tuned cutoff and window size are 0.6, 7 for the binary vector and 0.28, 4 for the real contact vector.

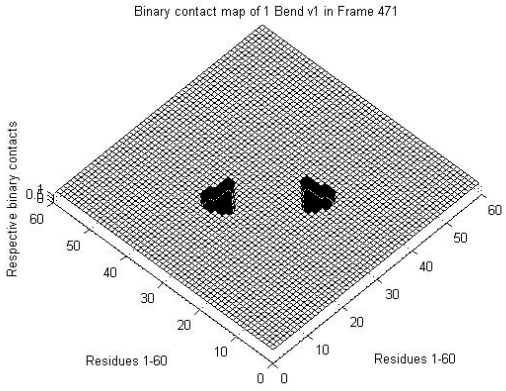
In the future, we might be able to extend these single values for cutoff or window size into intervals of acceptable values because the interatomic distance between atoms for most proteins usually falls in a certain range. This will loosen the strictness and therefore make it easier to find the cutoff and window size and enable the comparison to be more general.

3.2 Contact Vector Visualization and Contact Map and Analysis

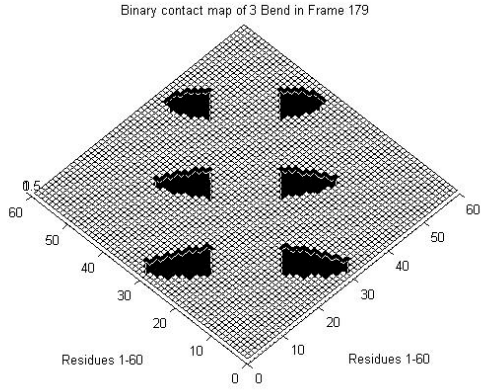
3.2.1 Binary Contact Map with the Proper Cutoff and Window Size

Once the contact vectors are properly calculated according to the fine-tuned cutoff and window size, new contact maps based on these new contact vectors can be generated. Some of them are shown in Figure 3.2.1.

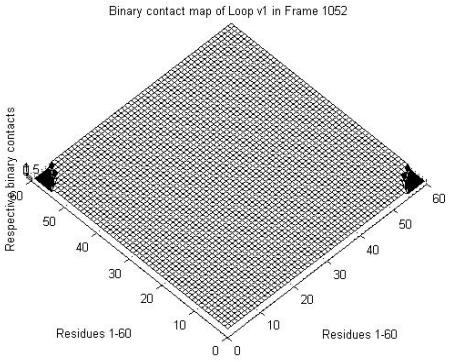
Figure 3.2.1 Binary maps with the proper cutoff and window size



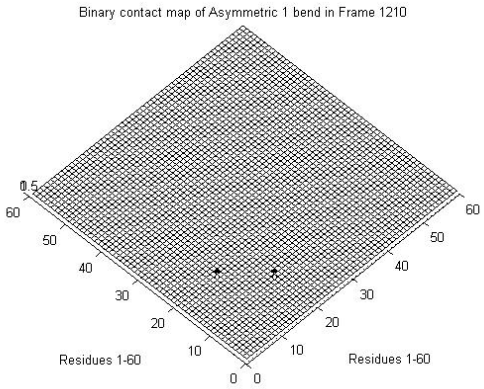
(1) 2D binary map for 1bend



(2) 2D binary map for 3bend



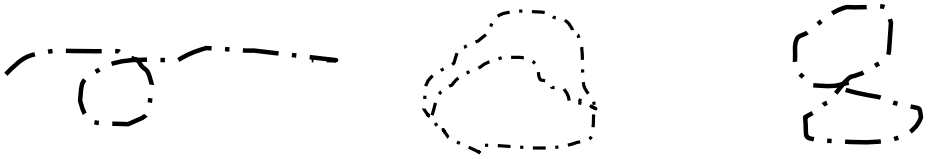
(3) 2D binary map for loop



(4) 3D binary map for asymmetric 1bend

In 2D, the regions marked black in the contact map are regions considered in contact. The difference between loop and regular bends is that for loop, the left and right hand corners are always in contact. There is an adjustable window size to detect a loop. By varying this window size, one can detect not only loops generated by the terminal residues coming in contact with each other, but also loops that occur anywhere in the protein structure, such as the ones shown in **Figure 3.2.2.**

Figure 3.2.2. Example of detectable loops

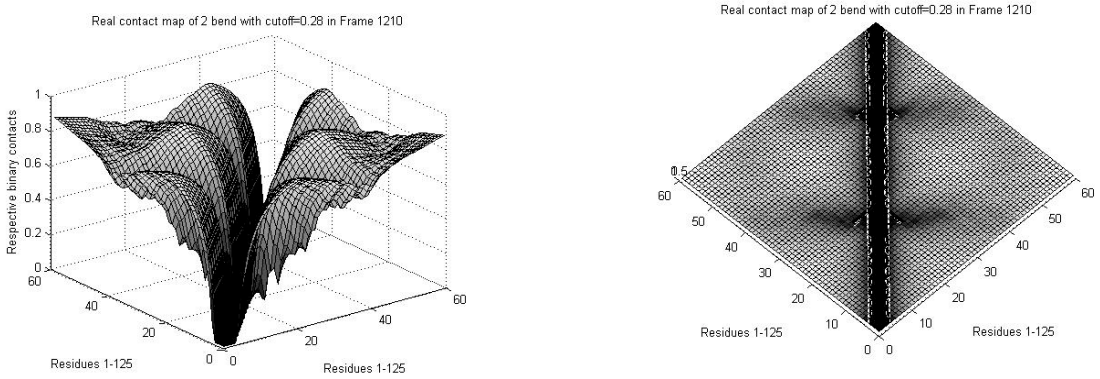


(1) in the middle (2) nested (3) sharing bottleneck

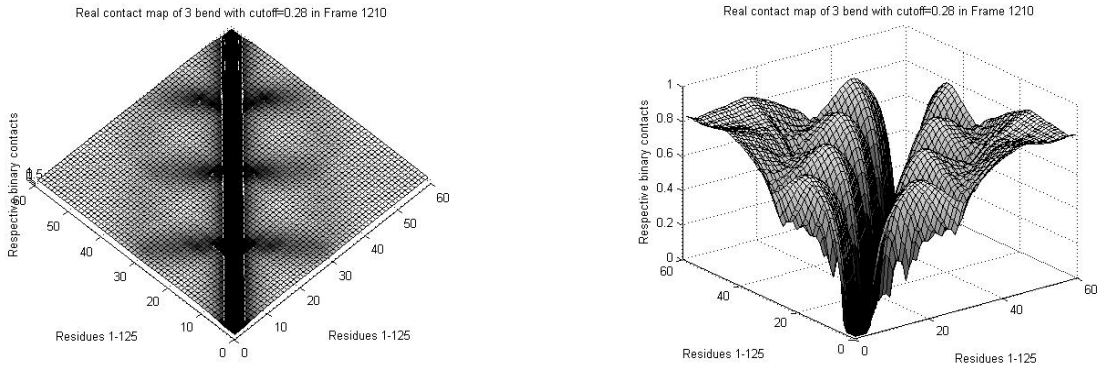
3.2.2 The Tradeoffs of the Real Contact map

While the binary contact map has crystal clear division between contact and non-contact regions, the real map is relatively smoother and the information it provides is more fine-grained. However, in these real maps, the line that separates the contact and non-contact regions is fuzzy and thus will add more complexity for further detection and atom clustering.

Figure 3.1.2.2 Real contact map with the proper cutoff and window size



(1) 3D 2bend with cutoff= 0.28 at frame 179 (2) 2D 2bend with cutoff=0.28 at frame 179



(3) 2D 3bend with cutoff=0.28 at frame 179 (4) 3D 2bend with cutoff=0.28 at frame 179

The fuzziness is due to the fact that the real contact vector can have any of the infinite different real numbers between zero and one. In the numerical analysis aspect, we need two cutoffs besides the window size. One is used to distinguish the non-contact and contact region (correspondingly, the white and black/gray region in the map), and the other used to filter out the obvious contacts. That is, contacts that exist because the atoms involved are neighbors in the residue chain. Finding a single window size in the real map to filter out the neighbors failed. Due to the degree of complexity involved in the real vector analysis and the promising result in the binary contact vector/map analysis, we decided the later state of the project to focus on the information contained in the binary contact vectors only.

3.2.2 Atom Clustering and Critical Even Detection

Seen from the contact maps, it is intuitive that if we are able to cluster the atoms into contact and non-contact groups, we will be able to detect the critical evens, say bends or loops. Our approach is to scan though the contact vectors and divide them into sub-vectors. For example, let (x1, x2) denote a contact where x1 and x2 are the indices of these in-contact atoms. A sample contact-stream is shown in Figure 3.2.2.1,

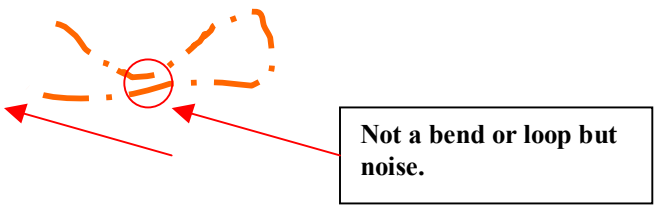
Figure 3.2.2.1 Portion of a Contact Stream

.... (18, 30) (18, 31) (19, 27) (19, 28) (19, 29) (19, 30) (19, 31) (20, 28) (20, 29) (20, 30) (21, 29) (30, 54) (30, 55) (31, 52) (31, 53) (31, 54) (31, 55) (32, 52) (33, 50) .....

The x1 value in elements in the contact-stream is piecewise continuous. Thus by setting up a minimum gap size and use it to detect the gap, we are able to distinguish different contact regions and further cluster all atoms in the protein. In the example of Figure 3.2.2.1, clearly there is a jump between the pair *(21, 29) and (30, 54)*. And thus, we can conclude that there are two bends in the parts shown.

For conformations that are more complicated, noise exists. Such an example is shown in Figure 3.2.2.2.

Figure 3.2.2.2 Example of noise in a conformation



3.3 Trajectory Analyses and Noise Filtering

3.3.1 Generating the Even-stream for a Trajectory

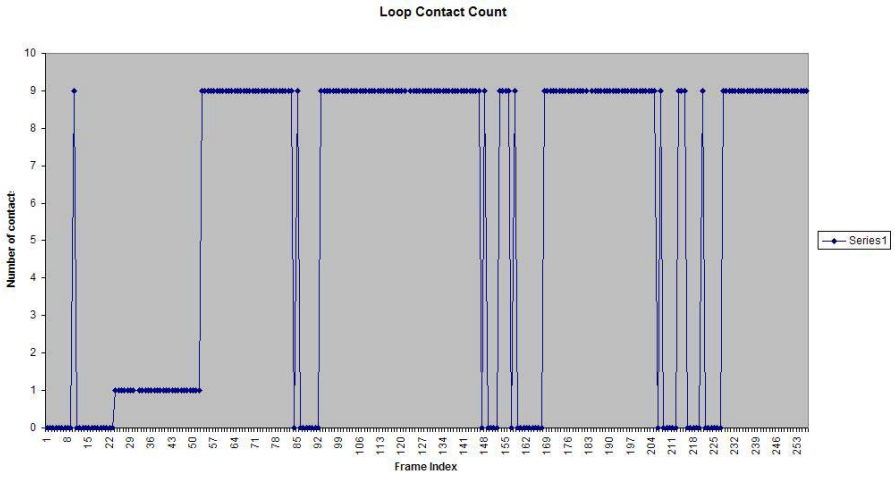
Since a trajectory is made up of different frames, where each contains a different conformation, once we detect the critical events in each conformation, we can chain this information as beats in a timely manner to produce a critical-characteristic stream for an entire trajectory. In this stream, each element represents the characteristic of one frame, for instance, the number of bends in the conformation in that frame. By recursively detecting the number of critical events in each frame, we generate an abstract event-stream for the entire trajectory.

Thus the trajectory analysis can be reduced to the event-stream analysis. Similar to the atom clustering in conformational analysis, our goal is to identify the critical event and recognize the noise pattern, if there is any, and filter them out. Finally, we can use the purified event-stream to determine the characteristic of the trajectory.

3.3.2 Noise Pattern Recognition and Noise Filtering

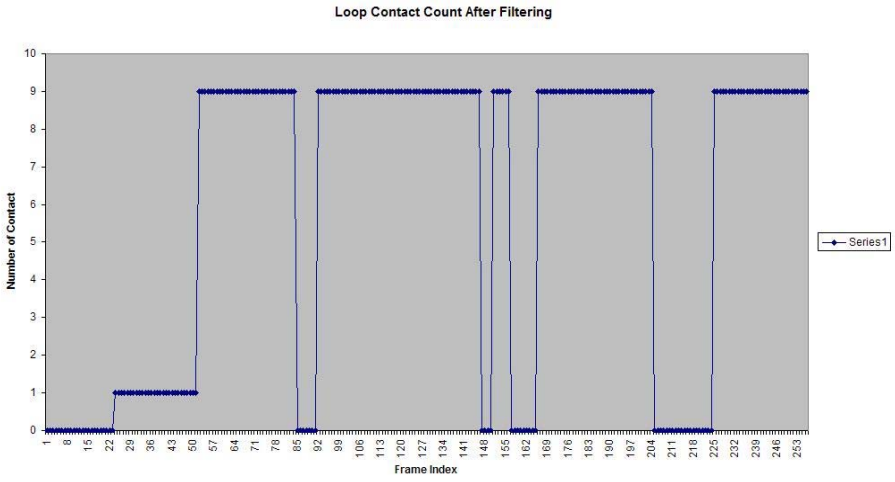
Based on the algorithm I used, symmetric one bend and two bends are noiseless. On the other hand, asymmetric 1 bend, 3 bends, and loops do have a small pool of interesting patterns of noise but the noise lasts no longer than a small number of consecutive frames. Noise Patterns are shown in Figure 3.3.2.1.

Figure 3.3.2.1 Loop Noise Pattern



By implementing a linear filter with the appropriate slicing window size, we are able to filter out the noise and identify the similarity and difference among different trajectories.

Figure 3.3.2.2 Loop after filtering



4. Conclusion:

The algorithm developed is able to detect the critical conformations and analyze the characteristics of different conformations in trajectories, and further distinguish different protein trajectories under study. Through this method, it is now possible to automate critical event detection and trajectory comparison.

5. Future Work:

We intend to apply our method to more complex trajectories of proteins, where there are more interesting critical events. We can cascade critical conformation detection to divide and compare the complex trajectories into smaller and simpler ones and so analyze them piecewise using the algorithm developed.

In addition, we will experiment more so we can extend the cutoff and window size from a single value to a range of acceptable values to simplify the automatic detection, and therefore make the algorithm applicable to proteins with more complex structures.

References:

[1] C. Best and H. Hege, "Visualizing and Identifying Conformational Ensembles in Molecular Dynamics Trajectories," Computing in Science & Engineering, Vol. 4, No. 3, May/June 2002, pp 68-75