GMDA Project: Mode-seeking for detecting metastable states in protein conformations

March 2018

Abstract

The aim of the project is to analyse protein conformations and their metastable states using mode-seeking techniques and find relevant clusters to classify them. But clustering is a difficult task due to really high dimension of one conformation state: It is why mode-seeking techniques (like ToMATo) are really useful to perform a relevant clustering.

Our dataset is composed of 14,207,380 different atom 3-d coordinates - 10 consecutive atoms forming a conformation. Thus there is in reality 1,420,738 different 30-dimension conformations because one conformation can be seen like ten lines which represent the position in the space of its ten atoms. In our project, we will compute the RMSD (Root Mean Square Deviation) of atomic positions between the different conformations, and then execute our mode-seeking algorithm, ToMATo to the resulting distance matrix.

1 INTRODUCTION

The general goal of the project is to detect metastable states and their proximity relations. In order to do that, we will apply mode-seeking techniques to protein conformations. A metastable state is a cluster of conformations: The conformations regroup themselves into these clusters; inside of them the probability of transition is high (low outside).

The complexity to retrieve the metastable states come from several issues:

- The number of clusters can be very large (of the order of hundreds or thousands)
- The data is not sampled along linear structures
- The clusters are non convex

The mode-seeking techniques can be an answer to this kind of problem, with what we call the "mean shift". It is a non-parametric feature-space analysis technique for locating the maxima of a density function. We will then use these maxima as the centers of the clusters.

2 FRAMEWORK OF THE METHOD USED

2.1 Data Collection

First we collected the set of alanine dipeptide conformations (aladip_implicit.xyz), then the set of conformations projected down to 2 dimensions for visualization (dihedral.xyz). This first dataset will be useful to do all the calculations; it is from this dataset that we calculate the distance matrix. The second one is used for data visualization:

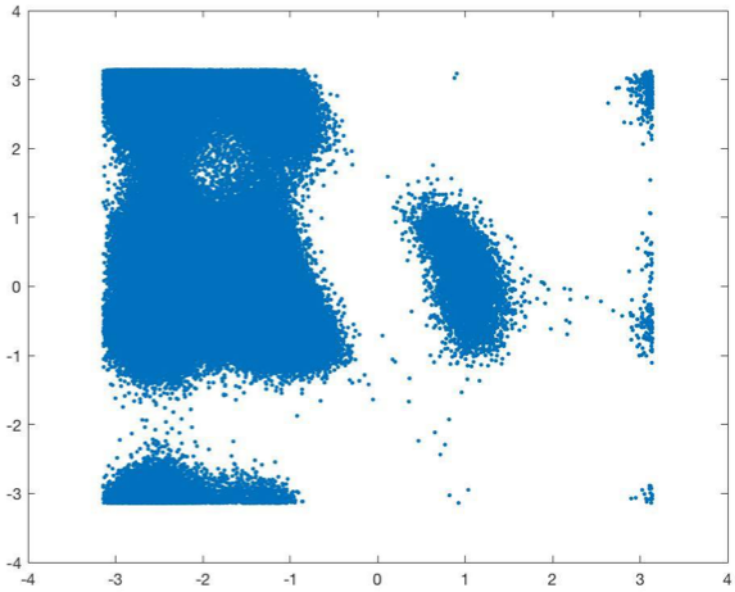

Figure 1: 2D Data representation on the dihedral axis

Note that the scale being in radians, the edges of the image appear to be far away but are actually right next to each other. We can visually identify 5 different clusters:

2.2 Computation of the Root Mean Squared Error distance matrix

The calculation of this distance matrix raises a real problem of calculation time: the estimated calculation times were first of 380 hours (approximately 15 days) with the "IRMSD" package, which purpose is fast structural RMSD computation. Finally, we managed to compute the matrix with the IRMSD package and Hadoop. The computation was faster and also it made it possible to put the code on several distributed machines if desired. The output RMSD matrix has m lines for each conformations and k columns which are the k random distances,

the use of only k values allows us to accelerate the time computation and limit the use of memory. We first wanted to sort the nearest distances, but we were not able to find a way (with limited storage use) to create a matrix with those nearest neighbors and distances. To that end, we had to create a dictionary but ToMATo requires a matrix and a matrix with "None" spaces was still too big for our machines.

1. Explanations about our script without MapReduce framework:

   - We run the script which computes RMSD distance between m conformations in k dimension (k nearest distance). As input, we have one atom in the format: key x y z; and as output we have RMSD matrix:

$$\begin{array}{c} key_{m1}, \text{rmsd(confo m1, confo k1)...rmsd(confo m1,confo kj...} \\ ... \\ key_{ml}, \text{rmsd(confo ml,confo k1...rmsd(confo ml,confo kj...} \end{array}$$

2. Explanations about our script with MapReduce framework:

   - First, we add a key for each conformation
   - Then we run the job 1: the code gathers each conformation into a single line. As input, we have one atom in the format: key x y z; and as output we have one conformation with the format:
     $key[x_1, y_1, z_1]...[x_{10}, y_{10}, z_{10}]$
   - As the reducer sorts the output, we apply a shuffling algorithm to be able to select only few of them without distorting the experiment. Here we kept 20,000 conformations out of the 1,420,738 we had, because of the computation time
   - Finally, we run the job 2 with two arguments, the number of sample m from the data set and the number of distance we keep k for each conformation. The output is a matrix $m \times k$ which, for each conformation, provides the k random distance to others and the key. So as input, we have one conformation with the previous format:
     $key[x_1, y_1, z_1]...[x_{10}, y_{10}, z_{10}]$
     and as output, we have the following matrix:

$$\begin{array}{c} key_{m1}, \text{rmsd(confo m1, confo k1)...rmsd(confo m1,confo kj...} \\ ... \\ key_{ml}, \text{rmsd(confo ml,confo k1...rmsd(confo ml,confo kj...} \end{array}$$

2.3 Application of ToMATo to the previously computed matrix

ToMATo: A Topological Mode Analysis Tool. This algorithm is a clustering engine and performs the mode-seeking technique we are interested in. The

difficulty in using ToMATo appears when we try to find the right parameters: the cluster rate and its radius. This is how we fined tuned the model:

- For the radius, we wanted to limit the number of local maxima. Each maximum being represented by a dot on the diagram, we increased the radius to decrease the number of maxima
- For τ (the cluster rate): if the visualization of the diagram was really useful, analytically we looked for different for which the number of clusters were the same (excluding 0 or 1).
- We were aiming for about 30 local maxima, and between 4 and 6 clusters, as suggested in article (2) *(Part 5: Experimental Results - 2. On the Alanine-Dipeptide Conformations)*.
- Parameter optimization: in order to obtain final results, using *main_w_density* we optimized the parameters finding a cluster that would create the right number of clusters, in our case 5. And then we had the issue of having points too far away to be in a cluster, we thus raised the number of nearest neighbors until all conformation were added to a cluster

3 CHALLENGES ENCOUNTERED

One of the challenges encountered was the optimization we had to set up in order to apply ToMATo. The data was huge and it was impossible to apply the algorithm to the whole RMSD distance matrix without considerable computation time or storage capacity. We tackled this issue by using the technique presented in the reference (2) *(Persistence-Based Clustering in Riemannian Manifolds)*: To avoid having to calculate all the distances per pair, for each conformation we calculate the distances only for the 15,000 conformations that are closest to it in the matrix. But even with this technics it was quite long to compute the code and so we finally selected only 20,000 conformations of the whole initial dataset to make our clustering. Thus it was difficult to obtain consistent results because of the poor number of our matrix distance compare to the initial dataset.

4 RESULTS ACHIEVED

On the final data, we obtained 5 clusters, and the representations really take into account the fact that it's a rotation (e.g. The cluster on the lower part is the same as the one on the upper part). To obtain those results, we used the parameters:

- number of nearest neighbors: 30
- radius: 0.4
- persistence threshold: 28

We thus obtained the following persistence diagram:

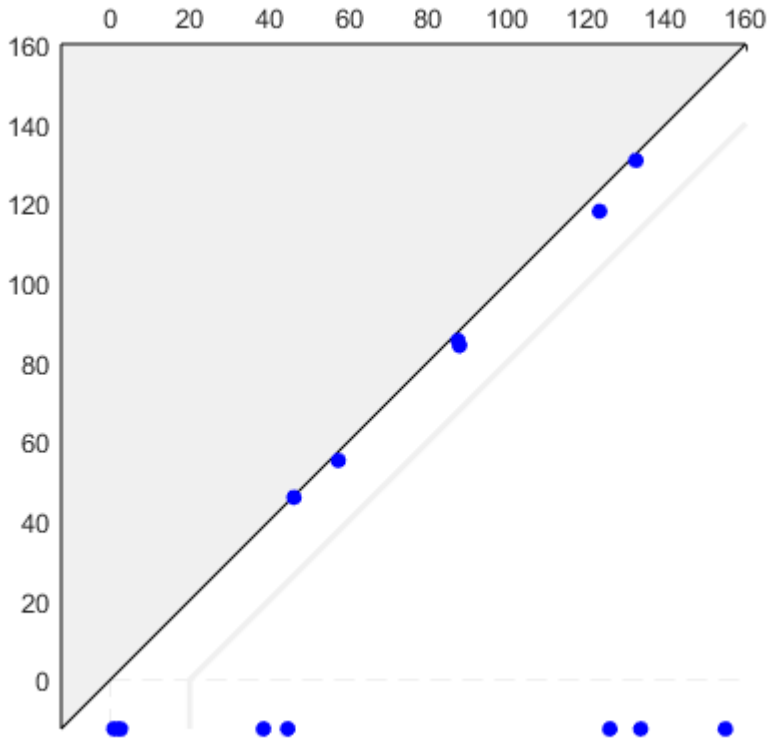

Figure 2: Persistence diagram after clustering

Finally we obtained the following clustering graph on our 20 000 conformations:

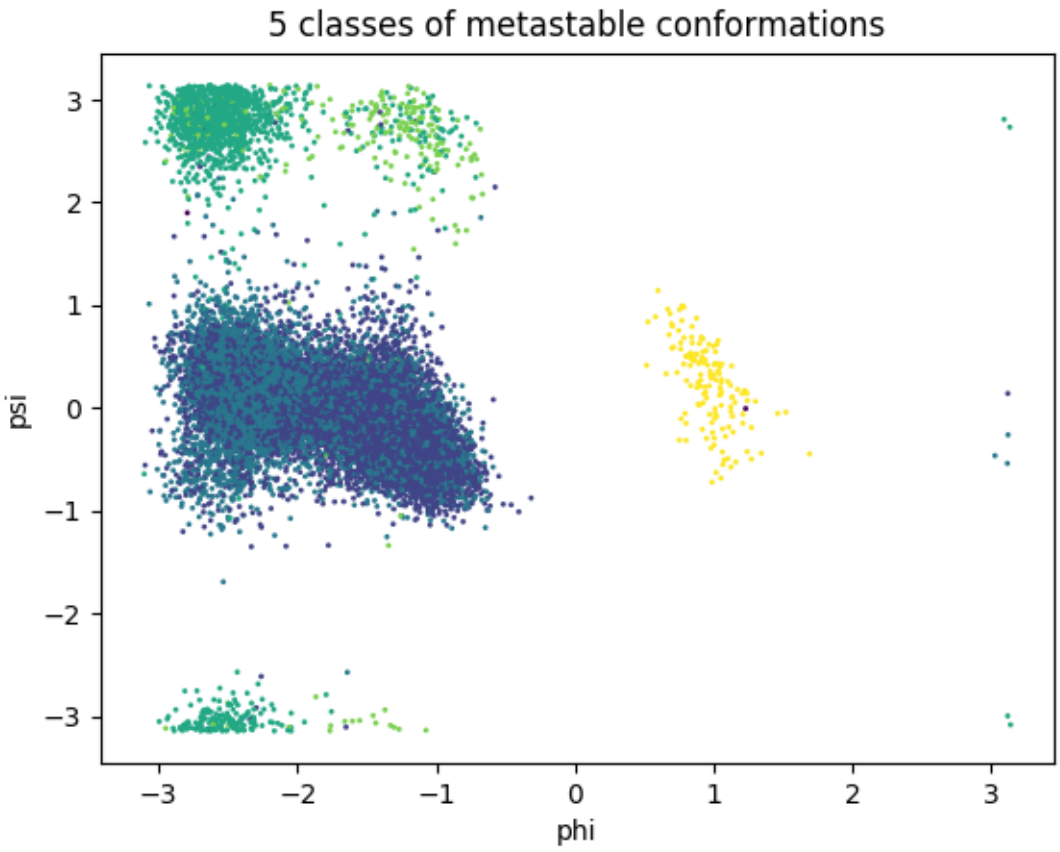

Figure 3: 2D Data representation of the clusters on the dihedral axis

In the end, it seems there are miss classification on the 2D representation graph, we see two possible factors to explain it:

- First, this consists only of a representation of the final graph (which is in 3D), which could explain the impression that the clustering was not efficient
- Second, maybe the number of points was not enough to obtain results (doesn't seem like a good explanation)

References

[1] J. CHODERA, W. SWOPE, J. PITERA, AND K. DILL, *Long-time protein folding dynamics from short-time molecular dynamics simulations*, 2006

[2] F. CHAZAL, L. J. GUIBAS, S. Y. OUDOT, P. SKRABA, *Persistence-Based Clustering in Riemannian Manifolds*, 2013