CentraleSupélec

CentraleSupélec
Université Paris Saclay

ToMATo for protein conformation

Link to the Github Repo

Project Supervisors: Frédéric CHAZAL
Mathieu CARRIERE

Anouar OUSSALAH, Laurane GORGUES

May 7, 2023

Contents

1 Introduction and problem definiton **2**

2 Methodology **2**
2.1 Data Collection 2
2.2 RMSD matrix computation 3
2.3 MDS embedding 4
2.4 ToMATo for clustering 5
2.4.1 ToMATo algorithm 5
2.4.2 Implementation 5

3 Experimental Results **6**
3.1 ToMATo Clustering 6

4 Conclusion and discussion **8**

Abstract

The main objective of this project is to examine protein conformations and their metastable states through mode-seeking techniques, with the ultimate goal of identifying relevant clusters for classification. However, due to the high dimensionality of each conformation state, clustering poses a significant challenge. To address this issue, we used ToMATo, a mode-seeking technique that has proven to be highly effective in facilitating accurate clustering. In order to accomplish this, we employed the ToMATo algorithm on a dataset comprising 14,207,380 distinct atom 3-d coordinates, where each conformation consisted of 10 consecutive atoms. Initially, we computed the RMSD matrix for these conformations before proceeding with the ToMATo clustering. Despite the rapid performance of ToMATo clustering, the bottleneck in the project was primarily associated with computing the RMSD matrix, which had an exponential temporal complexity.

1 Introduction and problem definiton

The aim of our project is to detect metastable states and their proximity relations in protein conformations using the ToMATo algorithm. Metastable states are defined as clusters of conformations with high probabilities of transition within the cluster and low probabilities outside the cluster. This analysis is important for understanding the dynamics of protein structures and their function.

However, detecting metastable states poses several challenges due to the large number of clusters that can be generated (in the order of hundreds or thousands), the non-linearity of the data sampling, and the non-convexity of the clusters.

To address these challenges, we applied mode-seeking techniques using the ToMATo algorithm. ToMATo uses a density-based approach to partition the data into clusters based on the similarity of their persistent homology. The algorithm first constructs a simplicial complex using a neighborhood graph, and then computes the persistent homology of the complex. Finally, it uses a Tanimoto coefficient to measure the similarity between two persistence diagrams and cluster the data points.

Our approach builds on previous works in protein conformation clustering and mode-seeking techniques. Indeed, this was actually already done in [1].

2 Methodology

2.1 Data Collection

The first step of this project was collecting the data and then converting it into a usable format. Thus, after downloading the two files in the xyz format we converted the content into lists thanks to the function `read_xyz` which uses regular expressions to extract the coordinates of the atoms. The resulting matrices were of size 3 x 14207380 for the *aladip_implicit.xyz* file and 2 x 1420738 for the *dihedral.xyz* file.
Then we grouped the atoms in the list from the *aladip_implicit.xyz* file 10 by 10 to get a list of conformations usable as input for the RMSD matrix computation step. At this step, the *dihedral.xyz* file allowed for a first observation of the dataset projected onto the space given by the two angles that parametrize the two relevant degrees of freedom.

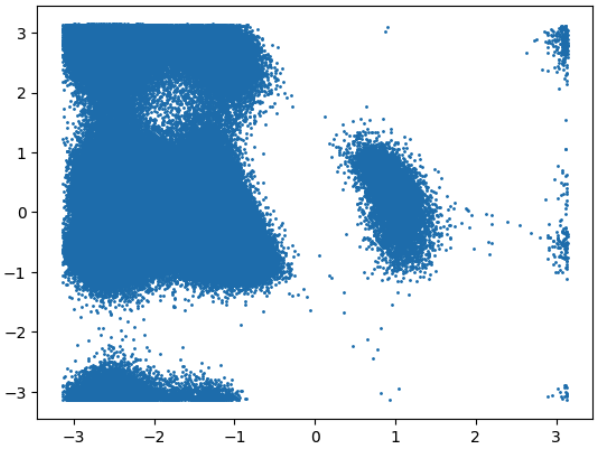

Figure 1: Data representation on the dihedral axis

Visually, and keeping in mind that the right and the left sides of the graph correspond as well as the bottom with the top of the graph, we can differentiate five clusters: one in the middle and four in the half on the left.

2.2 RMSD matrix computation

The first step of the RMSD matrix computation was to select a subsample of the initial data: indeed, computing the RMSD matrix for the full dataset turned out to be unfeasible in practice with our personal computers and the algorithm used (the computation time would have been hundreds of hours at the computation rate observed on the subsample). After some trials and error to try and get the best representativity of the original dataset while maintaining a reasonable computation time we chose to select one every seventy conformations which gave us the dataset visualised in the image below:

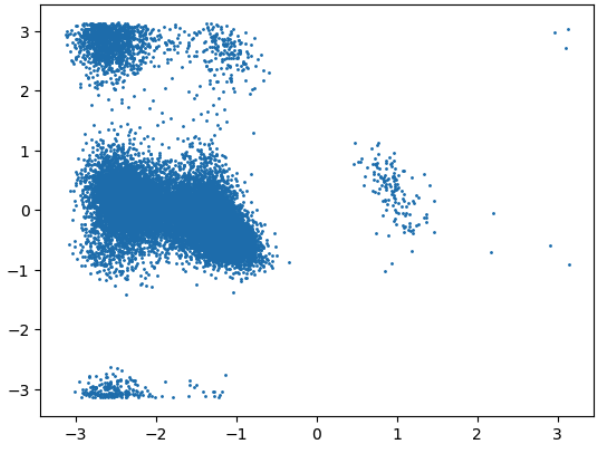

Figure 2: Data representation of the subsample on the dihedral axis

This dataset is composed of roughly 20000 points (we made attemps to compute more points - 30000 - but the kernel eventually restarted due to an excessive computation time and it was thus unusable). We can observe a similar repartition to the one of the full dataset but with a lower density (as expected).

After some reaserch, three different ways to compute the RMSD were identified (they only vary in their way to determine the rotation matrix that minimizes the distance):

- the one suggested in the assignement: minimizing the value with gradient descent
- using the quaternion method
- using the Kabsch algorithm

We implemented all of them to test their respective computation times (the last two methods were adapted from [2]) and the fastest one turned out to be the one based on the Kabsch algorithm so this is the one we used.
The Kabsch algorithm is a method for calculating the optimal rotation matrix that minimizes the RMSD between two sets of atoms (molecules) each represented by their coordinates. This is done in several steps:

- translate the molecules so that their centroids are in the center of the coordinate system
- compute the cross-covariance matrix between the two sets of coordinates (seen as matrices)
- compute the optimal rotation matrix R (with $R = (H^T H)^{\frac{1}{2}} H^{-1}$ with $H = P^T Q$ where P and Q are the matrices of the coordinates of each molecules) using singular value decomposition.

2.3 MDS embedding

After computing the RMSD distance matrix between the (around) 20000 conformations we used MDS to produce an embedding of the dataset in 2 dimensions and thus obtain another 2-dimensional visualisation.
To compute the MDS we used the built-in fuction **sklearn.manifold.MDS** which takes as input the distance matrix (parametrized with: n_components = 2, random_state = 123 and dissimilarity = 'precomputed') and outputs a 2D embedding.
Unfortunately the 2D embedding for the final subsample (around 20000 data points) couldn't be computed as running the MDS function on it made the kernel crash on both our machines. Thus we computed it for less data points in order to still visualise the results which are visible in Figure 3.

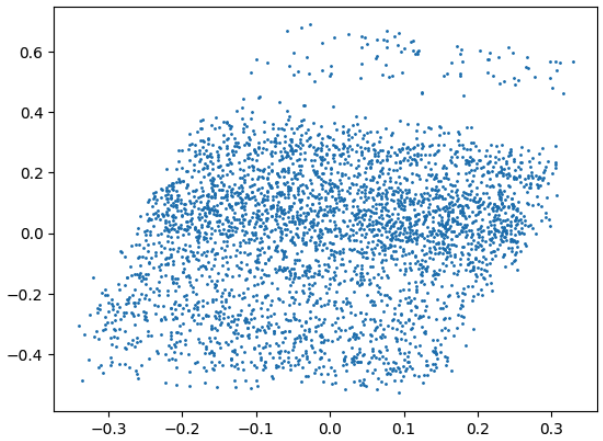

Figure 3: 2D embedding with MDS on a smaller subsample for visualisation purposes

Even if the clusters are less defined than on the provided file for visualisation (*dihedral.xyz*) we can still make out 4-5 clusters placed one above the other along the y-axis and all having different densities of points.

2.4 ToMATo for clustering

2.4.1 ToMATo algorithm

Our project proposes the use of ToMATo, a clustering algorithm that combines mode-seeking and cluster merging phases to detect relevant clusters in an efficient and accurate manner. The algorithm relies on a graph-based hill-climbing scheme to detect modes, and uses topological persistence theory to guide cluster merging, providing valuable feedback in the form of a persistence diagram. This allows us to easily determine parameter values that will yield a relevant clustering on subsequent runs, mainly the number of clusters).

ToMATo is highly generalizable and applicable to any arbitrary metric space, requiring only knowledge of approximate pairwise distances (here RMSD distances) and rough density estimates. Despite its efficiency, the algorithm provides theoretically sound clustering results, with a provably correct number of clusters. These guarantees hold under mild sampling conditions and even when data points are distributed along an unknown Riemannian manifold.

Compared to other clustering algorithms, ToMATo is able to handle non-convex clusters and is not sensitive to initialization, making it a powerful tool for protein conformation analysis. The feedback provided by the persistence diagram is particularly useful for identifying relevant parameter values and refining clustering results.

2.4.2 Implementation

We used the ToMATo algorithm for clustering protein conformations, which is implemented in the `gudhi` library. In order to achieve the best clustering results, we needed to fine-tune several hyperparameters.

- The first hyperparameter we tuned was the number of clusters (n_clusters). This parameter was actuallay tuned once we got the persistence diagram, looking at the prominent peaks.

- The second hyperparameter we adjusted was the graph type used for neighborhood selection, which can be either k-nearest neighbor or radius-based. We tested both options and chose the one that gave us the best results, which was actually kNN.
- The third hyperparameter we fine-tuned was the density type, which can be either DTM or logDTM. We experimented with both options and selected the one that produced the most meaningful clusters. Experimentally, we got better results we logDTM
- Finally, we also modified the merge threshold, which is the minimum prominence required for a cluster to avoid being merged with another cluster. However, we found that varying this parameter did not significantly affect our results.

It is worth noting that we did not find the parameters δ and τ mentioned in the original ToMATo paper, so we could not fine-tune these parameters.

In addition, in the absence of a clear metric to assess the goodness of a set of parameters, we relied on a qualitative assessment based on the persistence diagram and clusters created.

3 Experimental Results

3.1 ToMATo Clustering

Despite experimenting with various hyperparameters, we were unable to achieve clustering results as good as those presented in the paper.

Our ToMATo algorithm employed the following hyperparameters:

- density tpe : logDTM
- k = 5
- n_clusters = 4 to 6

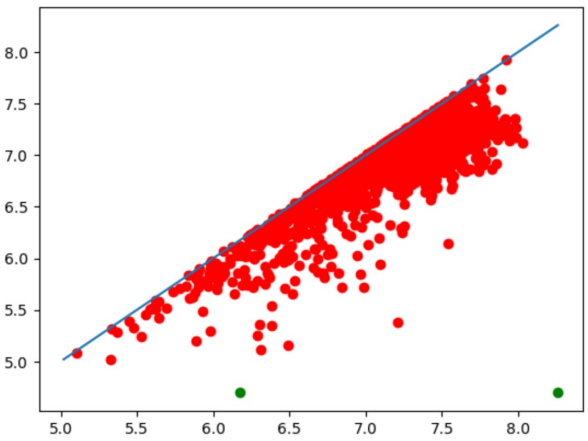

Figure 4: Persistence diagram on the 20 000 conformations

The persistence diagram in figure 4 reveals 4 to 6 prominent peaks, which is consistent with the original dataset. This diagram bears a resemblance to those seen in our lectures.

However, when we plotted the clusters on the original graph, we obtained the results shown in figure 5. This result may indicate that the ToMATo algorithm struggled to distinguish between the different clusters. Nevertheless, when we plotted the clusters using the MDS embedding, we obtained more promising results, as depicted in figure 6.

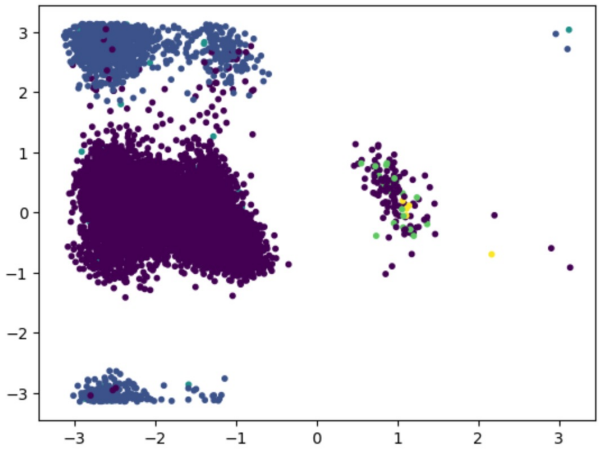

Figure 5: 2D Data representation of the clusters on the dihedral axis

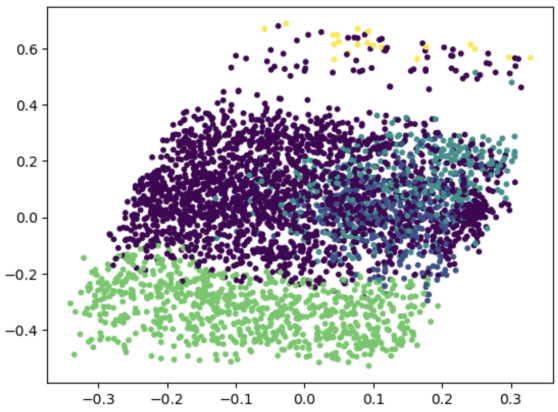

Figure 6: 2D Data representation of the clusters using the MDS embdding (on a sample of 3500 conformations)

We were unable to provide an explanation for the significant difference in results obtained, but we suspect that it may be due to inadequate parameterization of the representation on our part.

We still can interpret the results of the figure 6. Indeed, it is apparent that although some clusters are clearly distinguished (green vs other clusters), there is some overlap between others. We should keep in mind that the conformations are three-dimensional, and we are visualizing them in two dimensions. Thus, the apparent suboptimal clustering may be attributed to the dimensionality reduction, and the inherent limitations of visualizing high-dimensional data in 2D.

4 Conclusion and discussion

In this project, we attempted to cluster the conformations of a small protein using the Topology-based Methods for Approximate Tanimoto coefficients (ToMATo) algorithm. Our goal was to compare the clustering results of our implementation to the results obtained in a research paper that used ToMATo for the same task, with a different dataset.

We encountered several challenges in this project, including the time required to compute the root-mean-square deviation (RMSD) matrix, which is the basis for ToMATo's distance metric. We addressed this challenge by restricting ourselves to a sample of 20,000 points out of the 1,4 million points in the original dataset.

ToMATo algorithm showed promising results in the original paper, where it was demonstrated to be faster and more accurate than other clustering methods. Despite the algorithm's theoretical efficiency, our experimental results were not as satisfactory as those in the original paper. We tried different hyperparameters such as density type, the number of clusters, and the neighborhood graph type, but we couldn't achieve the same level of clustering as in the original paper. Even though we managed to identify some clusters with ToMATo, some clusters overlapped, and the clustering results were not visually appealing. The difference between our results and those in the original paper could be due to several reasons, such as different parameter settings, limited sample size, or poor representation of the conformational data.

Our project demonstrated the difficulty of clustering atomistic conformations of a protein. Conformations are typically high-dimensional and continuous, and finding meaningful clusters in such data is challenging. This task is further complicated by the fact that small changes in a protein's conformation can result in vastly different properties, such as binding affinity or catalytic activity. Hence, clustering protein conformations can be a valuable tool in understanding the protein's functional landscape, but it requires careful consideration of the problem's unique features.

References

[1] Frédéric Chazal, Leonidas J. Guibas, Steve Y. Oudot, Primoz Skraba, "Persistence-Based Clustering in Riemannian Manifolds", 2011, *INRIA*.

[2] Calculate Root-mean-square deviation (RMSD) of Two Molecules Using Rotation, *GitHub*, Github repo.