ToMATo for protein conformation*

Bhargav Ramudu Manam, Marina Diaz Uzquiano
M.Sc. Data Science and Artificial Intelligence

Université Côte d'Azur, 06410 Sophia Antipolis, France

1 Introduction

Goal:

The goal of this project is to analyze protein conformations using mode-seeking techniques, in order to detect metastable states and their proximity relations.

In this project, we have used Topological Mode Analysis Tool (ToMATo) (2) to cluster the confirmations. The code is provided along with this report. The following sections detail the steps involved in obtaining the clusters using ToMATo.

2 Data

The data consists of 1.4 million alanine dipeptide conformations: 3 coordinates per atom, 10 atoms per conformation. Hence, our confirmations lie in 30-D dimensional space.

The data also contains the set of conformations projected down to 2 dimensions for visualization. The visualization can be seen in the figure (1) below.

Remark 1. Due to limited computational resources and time constraints, only a portion of the data is used in our analysis.

3 RMSD Matrix

The (minimized) root mean square deviation (RMSD) between two protein confirmations A and B involves aligning the proteins, which is achieved by translation and rotation of the proteins, and mathematically it is given as:

$$RMSD(A,B) = minimum_{R_\theta} \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|\mathbf{A_i} - R_\theta \mathbf{B_i}\|_2^2} \tag{1}$$

Where $\forall i \in \{1, 2, \ldots, N\}$, A_i, B_i are the 3-D vectors of atoms containing the coordinates in x, y and z axes respectively. These are the translated (for all confirmations) such that the centroids of all confirmations are at origin. R_θ is the rotational matrix.

* This project is done to fulfill the requirements of **Geometric and topological methods in machine learning** course instructed by **Jean Daniel Boissonnat, Frederic Cazals, Mathieu Carriere** of Inria, Sophia Antipolis.

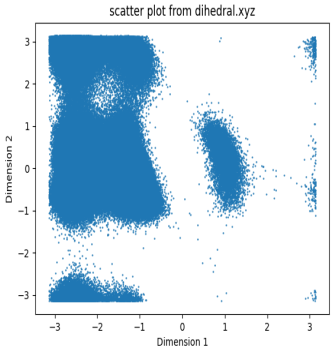


Fig. 1: Scatter plot of the 2-Dimensional input embeddings

3.1 Theobald Method for RMSD calculation

Firstly, we tried to use simple method like gradient descent to compute the optimal Rotational Matrix, R_θ, that minimizes the RMSD in the above equation but we found it to be computationally expensive.

We have implemented a fast algorithm called Theobald method (5) to calculate the RMSD between the protein confirmations and the procedure for computing the RMSD matrix is detailed in the following algorithm (2) found below.

Theobald method uses quaternion-based characteristic quartic polynomial method by Horn (3) for calculating RMSDs. In essence, Horn has shown that, finding the optimal R_θ is equivalent to finding the largest positive eigen value of the matrix K shown below:

$$K = \begin{bmatrix} S_{xx}+S_{yy}+S_{zz} & S_{yz}-S_{zy} & S_{zx}-S_{xz} & S_{xy}-S_{yx} \\ S_{yz}-S_{zy} & S_{xx}-S_{yy}-S_{zz} & S_{xy}+S_{yx} & S_{zx}+S_{xz} \\ S_{zx}-S_{xz} & S_{xy}+S_{yx} & -S_{xx}+S_{yy}-S_{zz} & S_{yz}+S_{zy} \\ S_{xy}-S_{yx} & S_{zx}+S_{xz} & S_{yz}+S_{zy} & -S_{xx}-S_{yy}+S_{zz} \end{bmatrix}$$

$$\text{Where}, \forall p, q \in \{x, y, z\}; S_{pq} = \sum_{i=1}^{n} p_{B,i} q_{A,i} \tag{2}$$

In fact, S_{pq}'s are the elements of the matrix $M = B^T A$.

Then, the RMSD can re-written as:

$$RMSD = \sqrt{\frac{(G_A + G_B - 2\lambda_{max})}{n}} \tag{3}$$

Here,

$$G_A = \text{tr}(A^T A) = \sum_{i=1}^{n} (x_{A,i}^2 + y_{A,i}^2 + z_{A,i}^2)$$

and

$$G_B = \mathrm{tr}(B^T B) = \sum_{i=1}^{n} (x_{B,i}^2 + y_{B,i}^2 + z_{B,i}^2)$$

.

Estimating λ_{max}: Since, the matrix K is symmetric the eigen values are obtained by solving for the roots of the characteristic polynomial equation given below:

$$C_4\lambda^4 + C_3\lambda^3 + C_2\lambda^2 + C_1\lambda + C_0 = 0 \tag{4}$$

Where,

$$\begin{aligned}
C_4 &= 1 \\
C_3 &= -\mathrm{tr}(K) = 0 \\
C_2 &= -2(S_{xx}^2 + S_{xy}^2 + S_{xz}^2 + S_{yx}^2 + S_{yy}^2 + S_{yz}^2 + S_{zx}^2 + S_{zy}^2 + S_{zz}^2) \\
C_1 &= -8(S_{xx}S_{yz}S_{zy} + S_{yy}S_{zx}S_{xz} + S_{zz}S_{xy}S_{yx}) \\
&\quad - 8(S_{xx}S_{yy}S_{zz} + S_{yz}S_{zx}S_{xy} + S_{zy}S_{yx}S_{xz}) \\
C_0 &= |K| = D + E + F + G + H + I
\end{aligned}$$

With,

$$\begin{aligned}
D &= \left(S_{xy}^2 + S_{xz}^2 - S_{yx}^2 - S_{zx}^2\right)^2 \\
E &= \left(-S_{xx}^2 + S_{yy}^2 + S_{zz}^2 + S_{yz}^2 + S_{zy}^2 - 2\left(S_{yy}S_{zz} - S_{yz}S_{zy}\right)\right) \\
&\quad * \left(-S_{xx}^2 + S_{yy}^2 + S_{zz}^2 + S_{yz}^2 + S_{zy}^2 + 2\left(S_{yy}S_{zz} - S_{yz}S_{zy}\right)\right) \\
F &= (-(S_{xz} + S_{zx})(S_{yz} - S_{zy}) + (S_{xy} - S_{yx})(S_{xx} - S_{yy} - S_{zz})) \\
&\quad * (-(S_{xz} - S_{zx})(S_{yz} + S_{zy}) + (S_{xy} - S_{yx})(S_{xx} - S_{yy} + S_{zz})) \\
G &= (-(S_{xz} + S_{zx})(S_{yz} + S_{zy}) - (S_{xy} + S_{yx})(S_{xx} + S_{yy} - S_{zz})) \\
&\quad * (-(S_{xz} - S_{zx})(S_{yz} - S_{zy}) - (S_{xy} + S_{yx})(S_{xx} + S_{yy} + S_{zz})) \\
H &= ((S_{xy} + S_{yx})(S_{yz} + S_{zy}) + (S_{xz} + S_{zx})(S_{xx} - S_{yy} + S_{zz})) \\
&\quad * (-(S_{xy} - S_{yx})(S_{yz} - S_{zy}) + (S_{xz} + S_{zx})(S_{xx} + S_{yy} + S_{zz})) \\
I &= ((S_{xy} + S_{yx})(S_{yz} - S_{zy}) + (S_{xz} - S_{zx})(S_{xx} - S_{yy} - S_{zz})) \\
&\quad * (-(S_{xy} - S_{yx})(S_{yz} + S_{zy}) + (S_{xz} - S_{zx})(S_{xx} + S_{yy} - S_{zz}))
\end{aligned}$$

After substituting the values of C_4 and C_3, the resultant characteristic polynomial equation in terms of λ is :

$$P(\lambda) = \lambda^4 + C_2\lambda^2 + C_1\lambda + C_0 = 0 \tag{5}$$

Its first order derivative is given by:

$$\frac{dP(\lambda)}{d\lambda} = 4\lambda^3 + 2C_2\lambda + C_1 \tag{6}$$

We have used Newton-Raphson QCP algorithm (6) (refer **Algorithm 1** below) for estimating the λ_{max}. The author in (5) opines that this algorithm converges on average within 5 iterations. We have used a tolerance of $10e^{-3}$.

Algorithm 1 Newton-Raphson QCP Algorithm

while ($|\lambda - \lambda_{\text{old}}| >$ tolerance) **do**
$\lambda_{old} = \lambda$
$\lambda = \lambda - \frac{P(\lambda)}{\frac{dP(\lambda)}{d\lambda}}$

Remark 2. All the quantities of K matrix, G_A, G_B and coefficients C_0, C_1, C_2 can be computed using the coordinates of A and B. So, only quantity we need to evaluate is λ_{max}.

Algorithm 2 RMSD Matrix Algorithm

Step 1:

1. **Translation:** Given a total of N proteins, center all the proteins such that all of them have their centroids at origin.
2. Calculate:
$$\forall j, G_j = tr({G_j}^T G_j); j \in \{1, 2, \ldots, N\}$$

Step 2:

1. For all the diagonal elements of the matrix, $RMSD = 0$
2. Consider only the elements in upper triangular of the matrix.
   For all the elements:
   (a) Calculate all the S_{pq}'s as per equation (2)
   (b) Calculate the coefficients: C_0, C_1, C_2
   (c) Evaluate the $P(\lambda)$ and $\frac{dP(\lambda)}{d\lambda}$ from equations (5) and (6), respectively.
   (d) Estimate the λ_{max} from the Newton-Raphson QCP Algorithm 1.
   (e) Obtain the RMSD of the element using equation (3).
3. Invert the upper triangular matrix to obtain the entire matrix.

Remark 3. We used the NEF cluster to compute the RMSD Matrix and we are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support.

4 MDS

Due to the time constraints, RMSD Matrix is only computed (using Algorithm 2) for $20,000$ protein confirmations (chosen at random). We believe $20,000$ protein confirmations are sufficient for analysis, as we tested with several quantities (less than $20,000$) and experienced no significant deviation in their respective 2-D embeddings.

The resultant RMSD Matrix is then used as an input to obtain the 2-D embeddings using MDS (1). The 2-D embeddings are shown in the figure (2)below.

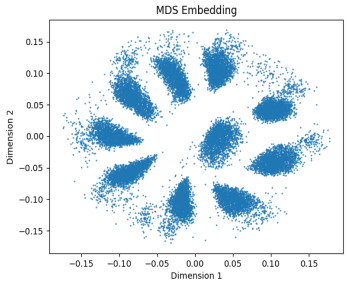


Fig. 2: 2-D embeddings of $20,000$ protein confirmations using MDS

Remark 4. Although the embeddings in figures (1) and (2) represent the same protein confirmations, it can be inferred that they look very different. This can be due to several reasons such as variability in the RMSD matrix, since the distances in figure (1) depends on 22-D space as per (2) whereas in our case, we have the input in 30-D space. Also, other reasons might be noise incurred due to estimation of λ_{max} or the fact that MDS only preserves the pairwise distances and not the orientation

5 ToMATo Clustering

ToMATo involves, given a point cloud or distance matrix, estimating the density of the point clouds using desity estimators like Distance to a measure (DTM) and Kernel Density Estimation (KDE). Then the point cloud is sorted by decreasing density values. A neighbourhood graph is built using either KNN or Rips filtrations and the order of the computed density values is passed through the edges of the graph by computing the upper-star filtration. Hierarchical clustering is achieved through mode-seeking by computing the 0-dimensional persistence diagram (PD) of this filtration to Union-find data structure. The robustness of the Stabililty theorem guarantees that the Union-find data structure results in basins of attraction.

Parameters:

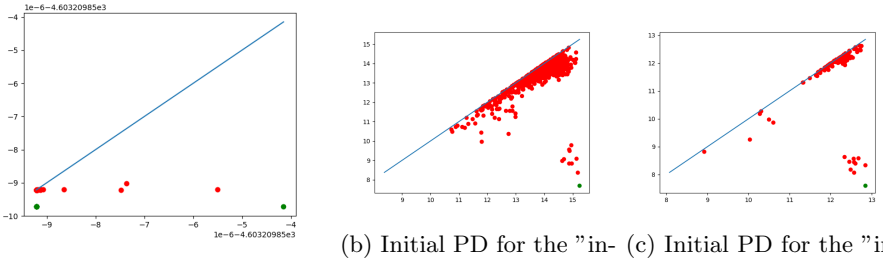
ToMATo mainly depend on the Prominence Threshold τ, which means that for a given value of τ the ToMATo results in clusters whose prominence (threshold) to not merge during mode seeking phase is at lease τ. Given a persistence diagram, one can estimate the optimal τ by inspection. It also depends on the Rips parameter, δ (if the neighbourhood graph is built through Rips-complexes), it is very challenging to find an appropriate value for δ and because of this in our analysis we only used the KNN for the neighbourhood graph. Both, KDE and KNN depend on another parameter k, denoting the number of nearest neighbours to

consider. Though, k does not have significant impact on the final clustering (because this is taken care by τ as there exists some prominence gap for a given τ with very high probability), but having higher k reduces the topological noise although this results in higher computational times and decreasing k gives the opposite result (refer to figure (3b and 3c).

5.1 Clustering Results

Clustering results obtained for the 1.4 million proteins ("input data", given along with the dataset) and 2-D embeddings, 20,000 proteins from our implementation, using Gudhi implementation of ToMATo (4), are discussed.

We did not consider giving RMSD Matrix as the direct input to the ToMATo because they resulted in inconsistent persistence diagrams (refer to figure (3a)). The value of the τ is negative suggesting background noise i.e., the true data does not actually lie in 30-D space but instead a much lower space. Hence, we used 2-D embeddings as input to the ToMATo algorithm inline with the practice followed by the scientific community (2).



(a) PD of 20,000 proteins using RMSD Matrix from 30-D space. Very negative values for τ (horizontal axis) indicates there is a lot of background noise.

(b) Initial PD for the "input data" embeddings using "logDTM" density estimator with $k = 10$, results in 966 clusters (high topological noise)

(c) Initial PD for the "input data" embeddings using "logDTM" density estimator with $k = 50$, results in only 120 clusters (lower topological noise)

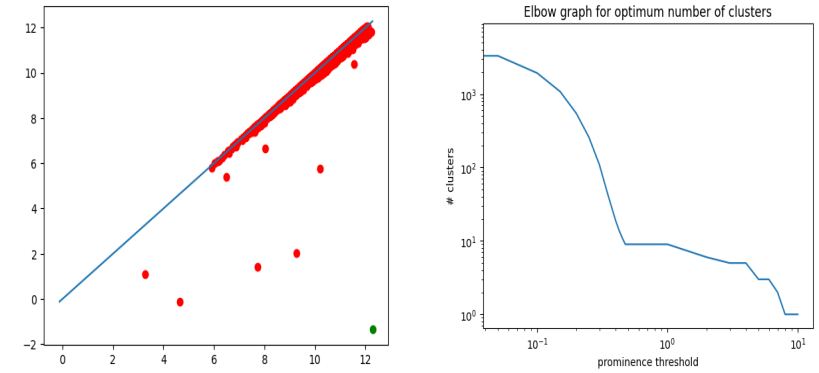
Fig. 3

For both "input data" and "2-D" embeddings of RMSD Matrix, we applied the ToMATo algorithm to the embeddings to obtain the PD as shown in the figure(4a, 5a) and then using the PD we performed hierarchical clustering by propagating the Prominence threshold τ. We plotted these observations in the form of a Elbow graph (4b, 5b) and also a table(1). The segmentation and density estimates are given in the figures (4c, 5c) and (4d, 5c) respectively. For these models, $logDTM$ is the Density estimator and $k = 100$ for "input data" and $k = 20$ for the "2-D" embeddings of the RMSD matrix.

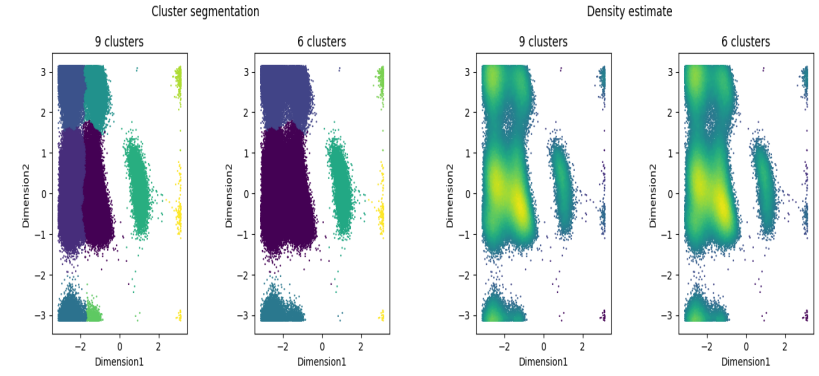
From figure (4) and table (1), we identified 9 and 6 as the possible number of clusters for the "input data". Though from the PD it appears that there are

6 clusters in the prominence peak area and the remaining 3 clusters are closer to the diagonal and hence can be seen as topological noise but the Elbow graph suggests that these 3 clusters are significant as well. Looking at the segmentation graphs it is reasonable for us to say that there are 9 clusters. Also, from figure (5) and table (1) we identified 10 as the possible number of clusters. In this case, it is clearer since all these clusters belong to the prominence peak area of the PD.

We are of the opinion that our analysis conform with the results presented in the reference article (2), given that even in the article the authors have expressed ambiguity in their final cluster selection. Though, we are still cautious about our result since we did not perform the same analysis as in, using Histogram or Bar code of PD's to come to our conclusions. Nevertheless, with the knowledge from the lectures and the Elbow diagram, persistence diagram we are confident in our assertions.

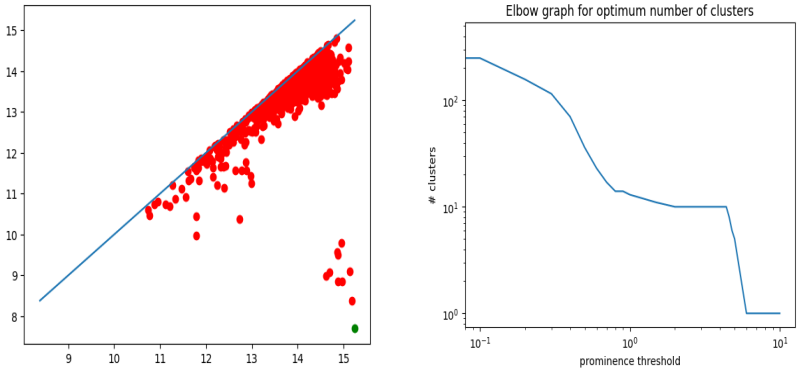


(a) Persistence Diagram with $k = 100$ (b)

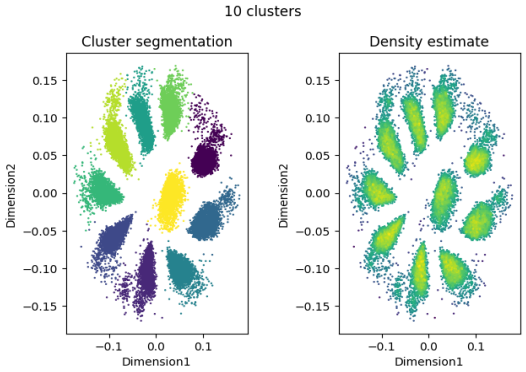


(c) (d)

Fig. 4: ToMATo applied on "input data"



(a) Persistence Diagram with $k = 20$ (b)



(c)

Fig. 5: ToMATo applied on "2-D" embeddings of RMSD Matrix

Remark 5. In our analysis we did not present the case where Gaussian Density estimator (Kernel Density Estimator- KDE) can be used, this is because we experimented and found that working with KDE has several issues. First, it is computationally very expensive and also raising memory issues during execution unlike DTM, which has fixed complexity for a given set of points. Secondly, the PD's from KDE are very unstable and has lot of background noise (refer to figure 6) and more often produces the diagrams similar to the ones in figure (3a)

Table 1

"input data" embeddings		"2-D" embeddings of RMSD Matrix	
Prominence Threshold (τ)	# Clusters	Prominence Threshold (τ)	# Clusters
10.000	1	10.0	1
8.000	1	6.0	1
7.000	2	5.0	5
6.000	3	4.8	6
5.000	3	4.6	8
4.000	5	4.4	10
3.000	5	4.2	10
2.000	6	4.0	10
1.000	9	3.0	10
0.500	9	2.0	10
0.475	9	1.5	11
0.450	11	1.0	13
0.425	14	0.9	14
0.400	19	0.8	14
0.35	42	0.7	17
0.3	110	0.6	23
0.25	257	0.5	36
0.2	545	0.4	70
0.15	1084	0.3	115
0.10	1935	0.2	157
0.05	3320	0.1	249
0.00	6011	0.0	402

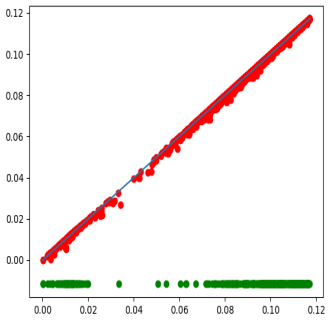


Fig. 6: Persistence Diagram for a sub sample (10 %) of "input data" with KDE estimator and $k = 5$

Bibliography

[1] Sklearn mds, https://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html

[2] Chazal, F., Guibas, L.J., Oudot, S.Y., Skraba, P.: Persistence-based clustering in riemannian manifolds. J. ACM (2013). https://doi.org/10.1145/2535927, https://doi.org/10.1145/2535927

[3] Horn, B.K.: Closed-form solution of absolute orientation using unit quaternions. Josa a **4**(4), 629–642 (1987)

[4] Inria: Tomato clustering using gudhi library, https://gudhi.inria.fr/python/latest/clustering.html

[5] Theobald, D.L.: Rapid calculation of rmsds using a quaternion-based characteristic polynomial. Acta crystallographica. Section A, Foundations of crystallography **61**(Pt 4), 478 – 480 (2005)

[6] Wikipedia: Newton raphson algorithm, https://en.wikipedia.org/wiki/Newton's$_m ethod$