UNIVERSITY OF PADUA

Structural Bioinformatics Project

Identification of structure clusters in Molecular Dynamics trajectories from Residue Interaction networks

Students
Eugenia Anello 1228000, Bianca Andreea Ciuche 1236661, Elisa Sartori 1244021

June, 2020 - Academic Year 2019/2020

Contents

1 Introduction **1**

1.1 Background 1

1.2 Problem Statement 2

2 Materials and Methods **4**

2.1 Manhattan distance 4

2.2 Clusters dendrogram 4

2.3 Affinity propagation clustering 5

3 Implementation **6**

3.1 Construction of the contact matrices for each snapshot of the trajectory of a protein 6

3.2 Construction of the distance matrix representing the distance between each snapshot 8

3.3 Development of clusters dendrogram 9

3.4 Development of the optimal clusterization 10

4 Results and conclusions **12**

Bibliography **16**

Chapter 1

Introduction

1.1 Background

In recent decades, thousands of 3D protein structures have been determined by X-ray crystallography and NMR spectroscopy, and many more have been computationally modeled [1]. Although 3D visualization still plays a vital role in the analysis of protein structure and function, complementary approaches based on 2D representations of residue interaction networks (RINs) have been proposed more recently [2].
Residue Interaction Networks (RINs) decreases the visual complexity of 3D protein structures and allow the user to focus on individual residues and their molecular interactions, that are usually difficult to characterize because they have a wide range of different energies and lengths [3]. While the energy contribution of a single interaction is almost negligible, together they determine the three-dimensional protein structure [4]. In essence, RINs are derived from protein structures estimating contacts from distance measures. Each RIN consider single **amino acid residues** as nodes and **physico–chemical interactions**, like covalent and non-covalent bonds, as edges. Exploration and analysis of the network of interacting residues can provide additional insights into the structural and functional role of residues.

RINs can be applied to study residue interactions in a number of relevant application scenarios, for instance, with regard to protein dynamics and engineering, structure–function relationships, protein and ligand binding, and the impact of amino acid substitutions. On the other hand, software generating RINs still has limitations due to the use of simplified interaction types. RINalyzer [5, 6] for example calculates only hydrogen bond, van der Waals (VDW) and generic contacts based on distance. This limitation can be explained by technical reasons, such as the computational cost of measuring the distance of all possible atom pairs in a protein, in particular for large biopolymers. Another problem is defining distance and angle constraints for certain interactions (e.g. involving -systems) in large molecules like proteins.

The **Residue Interaction Network Generator** (RING) has been presented to address these limitations [7]. It is a software, that is able to generate an interaction network in two steps. At first, it identifies a list of residue-residue pairs eligible for interaction based on all atom distance measurements. Contacts are then characterized by identifying specific interaction types. Considering that RING calculation is based on geometric criteria, every pair of residues can form multiple interactions. However, the software provides three options of cardinality [8]:

- One type of interaction
- Multiple interactions
- All types of interactions

Moreover the software is able to classify different types of contacts based on geometrical and physical-chemical properties of the amino acids. Type of contacts are:

1. Hydrogen Bonds (HBOND)
2. Van Der Waals interactions(WDW)
3. Disulfide bridges (SSBOND)
4. Salt bridges (IONIC)
5. pi-pi stacking (PIPISTACK)
6. pi-cation (PICATION)
7. Inter-Atomic Contact (IAC), generic contact simply based on distance

RING takes as input a PDB file and generates an edge file containing the contacts, in which each node is identified by a string like this “A:159:_:PRO”, in which the chain, residue index, insertion code and residue name are column (“:”) separated. An insertion code equal to “_” indicates there is no insertion code for that residue.

1.2 Problem Statement

The outputs of a **Molecular Dynamics** (MD) simulation are trajectory files which describes the change of atomic coordinates from an initial state to a final state (after a certain amount of time) when a forcefield is applied. The full trajectory can be described by a subset of intermediate snapshots.

In this project, we focus on performing the **optimal clustering** analysis for grouping MD **snapshots** of seven protein structures with high affinity in order to extract the most relevant information during the Molecular Dynamic simulations. In particular we want to estimate an "optimal" number of clusters. We compared the **contact maps**, calculated by the software RING, that saves them in files called edges. The reason behind the use of contact maps is because they provide more information about the type of contacts that each residue in the structure has.

The objectives of this study consist in finding:

- the residues, that are relevant to describe the transition between clusters
- the representative contact map for each cluster
- the list of relevant contacts/residues which contribute more to the transition between clusters
- the correlation between snapshots of different moments of the MD simulation

Then every MD conformation have been divided into several groups by using a measure of similarity/dissimilarity. Snapshots that are placed in the same group are, according to some criterion, similar to each other and dissimilar from the conformations of other groups. The metric used was custom made by the team, whereas the clustering algorithm used to try and find an optimal clusterization was the **Affinity Propagation** algorithm. We compared the approach with the hierarchical clustering generated by calculating the RMSD.

Chapter 2

Materials and Methods

2.1 Manhattan distance

Clustering is a common technique used for the classification of similar objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait – often proximity according to some defined distance measure [12]. Because of the variety of feature types and scales, the distance measure (or measures) must be chosen carefully. We'll focus on a well-known distance measure used for patterns whose features are all continuous. One of the most popular metrics for continuous features is the Manhattan distance:

$$d_M(x_i, x_j) = \sum_{i=1}^{n} |x_i - x_j|$$

The Manhattan distance has an intuitive appeal as it is commonly used to evaluate the proximity of objects in two or three-dimensional space.

2.2 Clusters dendrogram

A dendrogram is a branching diagram that represents the relationships of similarity among a group of entities. Each branch is called a clade. The terminal end of each clade is called a leaf. Clades can have just one leaf (these are called simplicifolious) or they can have more than one. Two-leaved clades are bifolious, three-leaved are trifolious, and so on. There is no limit to the number of leaves in a clade. The arrangement of the clades tells us which leaves are most similar to each other. The height of the branch points indicates how similar or different they are from each other: the greater the height, the greater the difference. We used dendrograms to construct a hierarchy of clusters of snapshots based on the distance matrix. The Figure 2.1 shows a dendrogram with indications of its components.

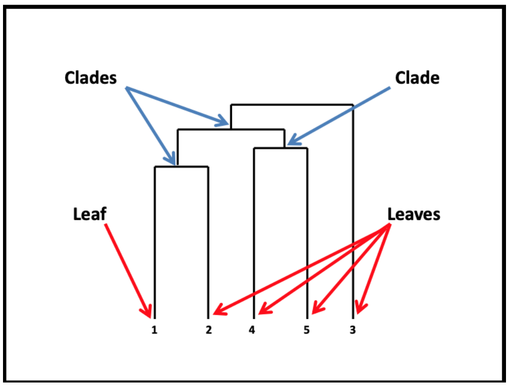

Figure 2.1: Dendrogram

2.3 Affinity propagation clustering

Affinity propagation (AP) is a centroid based clustering algorithm similar to k Means or K medoids, which does not require to configure the number of clusters before running the algorithm. Affinity propagation finds "**centroids**" or “**exemplars**” i.e. centers selected from actual data points that are the most representative for the cluster they belong to. Affinity propagation takes as input the similarities between the data points and identifies centroids based on certain criteria. Messages are exchanged between the data points until a high-quality set of centroids are obtained.
In this project we used the affinity propagation algorithm offered by the Scikit-learn library [10]. The main parameters of the algorithm are the damping factor and the preferences. Damping factor is the extent to which the current probability for a certain element of being an exemplar is maintained relative to the following updates. Preferences are values that can be set for each element, the ones which have a larger value are more likely to be chosen as exemplars.

Chapter 3

Implementation

The program that we created for this project, consists of the following steps:

1. Construction of the contact matrices for each snapshot of the trajectory of a protein;
2. Construction of the distance matrix representing the distance between each snapshot;
3. Development of clusters dendrogram;
4. Development of the optimal clusterization.

3.1 Construction of the contact matrices for each snapshot of the trajectory of a protein

In the project we first built the matrices for each snapshot that represent the contacts in each one of them. In order to build these matrices, we used the files "edges", generated by the software RING, that contain the contact maps from protein structures. Table 3.1 shows a typical extract from an edge file of one of the protein structures analyzed.

NodeId1	Interaction	NodeId2	Distance	Angle	Energy	Atom1	Atom2
A:152:_:GLN	VDW:SC_SC	A:155:_:TYR	3.574	-999.9	6.000	NE2	CE1
A:152:_:GLN	HBOND:MC_MC	A:184:_:ASN	3.055	24.025	17.000	N	O
A:152:_:GLN	VDW:MC_SC	A:237:_:VAL	3.871	-999.9	6.000	C	CB
A:154:_:PRO	VDW:SC_SC	A:187:_:PRO	4.011	-999.9	6.000	CB	CG
A:154:_:PRO	VDW:SC_SC	A:235:_:ASN	3.470	-999.9	6.000	CD	OD1
A:154:_:PRO	VDW:SC_MC	A:238:_:GLY	3.474	-999.9	6.000	CD	C
A:154:_:PRO	VDW:SC_SC	A:240:_:ILE	3.885	-999.9	6.000	CB	CD
A:155:_:TYR	HBOND:MC_MC	A:182:_:ALA	2.899	13.279	17.000	N	O

Table 3.1: An extract from an edge file of a protein structure

Before building the contact matrices, we fixed two thresholds:

- **energy threshold**: We decided to not consider Van der Waals interactions because they are the most weakest interactions and have the lowest energy (see Figure 3.2), therefore we consider non-covalent bonds with an energy value greater than the threshold, equal to 7.
- **contact threshold**: In order to keep all the remaining types of non-covalent interactions, we take the residues with a distance above the threshold, that is equal to 6. This parameter is due to the two different distance thresholds chosen by the software RING to represents strict and permissive parameters. We took a distance threshold with a value between the strict threshold of π - cation and the strict threshold of $\pi - \pi$ stacking.

We considered a null energy for the Inter-Atomic Contact(IAC) because we didn't find any information about the energy of this type of bond.

Interaction Type	**Strict** (Å)	**Relaxed** (Å)	**Energy**
Hydrogen Bond	3.5	5.5	17.0 (d>2.2 Å)
Ionic	4.0	5.0	20.0
Disulphide	2.5	3.0	167.0
Wan-der-Waals	0.5	0.8	6.0
π - cation	5.0	7.0	9.6
$\pi - \pi$ stacking	6.5	7.0	7.0

Table 3.2: Strict and Relaxed correspond to the optimized thresholds available in RING.

The contact matrices must be compared with each other, so we made sure that they had the following properties:

- The matrices of the different snapshots of a protein have the same number of rows (and columns) equal to the maximum number of the different residues involved in the contacts during the dynamics simulation.
- The indices of the contact matrices are the same both for the rows and for the columns. The indices consist of all the different residues and the corresponding bonds that are formed in the various snapshots of the Molecular Dynamic Simulation.

If the total number of residues is n, we need n^2 residues in the contact matrix. Because of these properties, the matrix is symmetric (the value between residue i and residue j is the same of that between j and i). We have generated weighed matrices for every snapshot: each contact between two residue, $C(i, j)$, is represented in the matrix by the multiplication between the **energy**, different for each type of non-covalent bond provided by RING (see Table 3.2), and a certain **weight**.
Weights are calculated for each snapshot according to the following points:

1. We calculate the **number of different non-covalent bonds**:

$numH$	number of Hydrogen bonds (HBOND)
$numV$	number of Van Der Wals bonds(VDW)
$numSS$	number of disulfide bridges bonds (SBOND)
$numI$	number of salt bridges bonds(IONIC)
$numPip$	number of $\pi - \pi$ stacking bonds(PIPISTACK)
$numPic$	number of π-cation bonds (PIPICATION)
$numIac$	number of Inter-Atomic Contact (IAC)

2. We calculate the **total number of non-covalent bonds**:

   $tot = numH + numV + numSS + numI + numPip + numPic + numIac$

3. We calculate calculate the **ratios**:

   $numH/tot$
   $numV/tot$
   $numSS/tot$
   $numI/tot$
   $numPip/tot$
   $numPic/tot$
   $numIac/tot$

4. We subtract the ratios from one:
   $weightHydrogenBonds = 1 - numH/tot$
   $weightVanDerWaalsBonds = 1 - numV/tot$
   $weightSboundBonds = 1 - numSS/tot$
   $weightIonicBonds = 1 - numI/tot$
   $weightPipistackBonds = 1 - numPip/tot$
   $weightPipicationBonds = 1 - numPic/tot$
   $weightInterAtomicContactBonds = 1 - numIac/tot$

In this way, a greater weight is given to the less frequent types of bonds and a lower weight to the more frequent ones.

3.2 Construction of the distance matrix representing the distance between each snapshot

Once the contact matrices of each snapshot were obtained, we calculated the matrix of distances between each snapshots of the trajectory. The distance between two snapshots is calculated according to the following points:

1. The contact matrices of the two snapshots are transformed into vectors;
2. The City Block (Manhattan) formula is applied, obtaining the distance between the two vectors;
3. The square root is made on the value obtained to lower it.

The distance matrix obtained represents the dissimilarity between the snapshots of the trajectory.

3.3 Development of clusters dendrogram

Using the distance matrices of each protein, we constructed the dendrogram that represent the hierarchical clustering between snapshots. We used the function scipy.cluster.hierarchy offered by the open source SciPy library on the distance matrix. [9]
To evaluate the clustering obtained, we compared it with the clustering generated by measuring the distance between structures. We calculated RMSD values for each configuration in a given trajectory with respect to every other configuration of the same trajectory.

$$RMSD(t_1, t_2) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} ||x_i(t_2) - x_i(t_1)||^2}$$

where $x_i(t)$ is the position of atom i at time t and N is the total number of atoms in the residue [11].

In order to build these matrices, we used the files "pdb", that contain the 3D coordinates of the MD snapshots from protein structures.

ATOM	1	N	PRO	A	150	19.130	33.580	73.100
ATOM	2	HN1	PRO	A	150	18.350	33.740	73.710
ATOM	3	HN2	PRO	A	150	18.760	33.600	72.160
ATOM	4	CD	PRO	A	150	19.770	32.210	73.300
ATOM	5	HD1	PRO	A	150	19.580	31.530	72.440
ATOM	6	HD2	PRO	A	150	19.250	31.830	74.210
ATOM	7	CA	PRO	A	150	20.160	34.610	73.420
ATOM	8	HA	PRO	A	150	20.050	34.800	74.480

Table 3.3: An extract from a PDB file of a protein structure showing how the atomic coordinates and other informations on each atom are deposited. The atoms are of a single proline residue in the protein. It can be seen that the x-, y-, z-coordinates of each atom are given to three decimal places.

3.4 Development of the optimal clusterization

The optimal clustering of the snapshots of a Molecular Dynamic simulation was obtained using the Affinity Propagation algorithm [10]. Affinity Propagation takes as input the similarity between elements so we had to convert our matrix of distances in a matrix of similarities. The output of this algorithm is a set of centroids. Each centroid corresponds to a snapshot of the input set and it is the element of a cluster that best represents all the other elements of that cluster, so we chose as the representative for each cluster the cluster centroid. We have identified the optimal clusters with an integer ranging from zero to the maximum number of optimal clusters minus one. Based on the centroid of each clusters, we calculated the distance between it and all its elements, as shown in the Table 3.4.

Cluster	Element	Distance from centroid
Cluster 0	3	23.104007133635253
	6	0.0
	8	23.65196680998397
	71	21.648067695557895
	75	23.489514289431337
	77	24.2845590000157
Cluster 1	12	23.36672599007286
	23	21.716196809835186
	24	21.604116831698317
	25	0.0
	49	22.039769878805004
	51	22.12938563850937
	95	23.1007081806295

Table 3.4: Part of the clustering results of the antibody protein

When the distance is equal to zero, it means that the element we are considering is the centroid.
Subsequently,having as representative for each cluster the centroid of the cluster,we considered the representative cluster contact map the one that coincides with the corresponding Ring-generated map of the snapshot identified as the centroid.
After an analysis of the contact maps that represent clusters, we construct a list of relevant contacts which contribute to the transition between the clusters. We considered as relevant contacts all of the contacts that the algorithm took in consideration during is process. The Table 3.5 shows a subset of the relevant contacts that contribute to the transition between cluster 0 and cluster 1.

Cluster 0 - Cluster 1
’A:157:_:THR’, ’HBOND’ ’A:180:_:PRO’, ’HBOND’
’A:163:_:GLU’, ’HBOND’ ’A:242:_:TYR’, ’HBOND’
’A:163:_:GLU’, ’IONIC’ ’C:49:_:ARG’, ’IONIC’
’A:164:_:LYS’, ’PICATION’ ’B:102:_:TYR’, ’PICATION’
’A:165:_:LYS’, ’HBOND’ ’B:101:_:ALA’, ’HBOND’

Table 3.5: Relevant contacts to the transition between cluster 0 and cluster1

Chapter 4

Results and conclusions

Understanding and interpreting a set of results is not an easy thing, especially if they concern an area of knowledge in which the programmers don't have much experience.

To make sense of our outcomes, we decided to compare the dendrogram obtained using our metric with the one obtained using RMSD, as suggested in the project specifications. Having a comparison would also have helped us tune our metric and the parameters of our program.

However, the metric indicated in the paper didn't have a python implementation so we used the tmscoring [13] module to compute the root mean square deviation.

As mentioned in Chapter 3, we have not considered Van der Waal interactions, and used an energy threshold of 7.000 and a contact threshold of 6.000. Below are the results for vhl in figure 4.2 and frataxin in figure 4.1.

The dendrograms appear to be similar but their similarity stops there. If we compare them in more detail, as in the table 4.1, we can observe that the majority of the snapshots are not in the same position nor a similar one.

This can be reasonably explained by the fact that RMSD is actually the mean of a distance, while our metric is more focused on the similarity of the structures, so they are not properly comparable.

This also meant that we would not have been able to make a comparison between the optimal clusters result, so we decided to use standard parameters values for our clustering algorithm and observe how it would behave.

We took in consideration and compared different types algorithms for clustering and in the end we chose Affinity Propagation. We thought that it would have been interesting to see if the algorithm could find an optimal clustering in structures and values quite similar to one another, and if its approach could have been able to infer some similarities between snapshots that would otherwise have gone undetected.

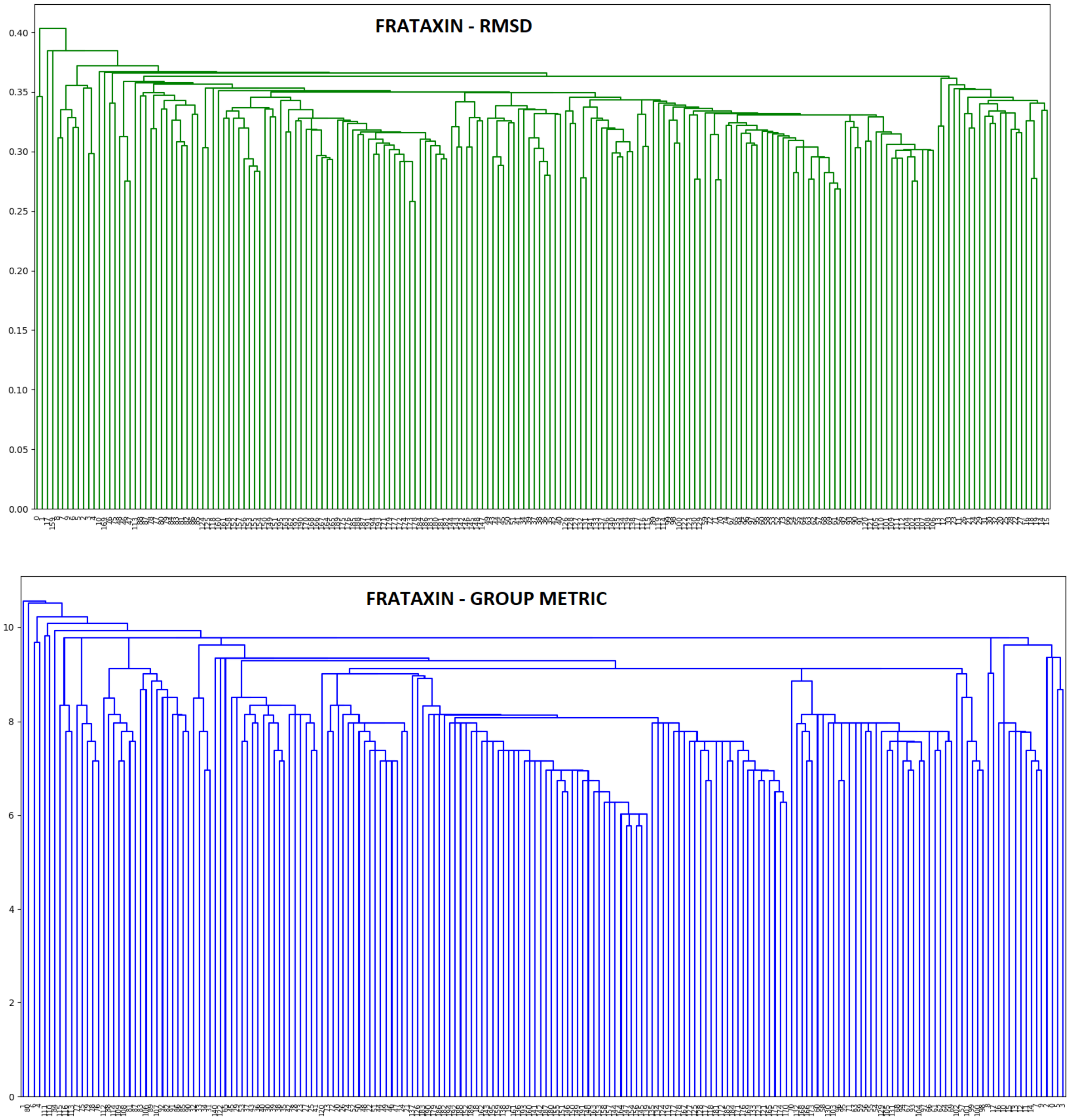

Figure 4.1: Frataxin comparison

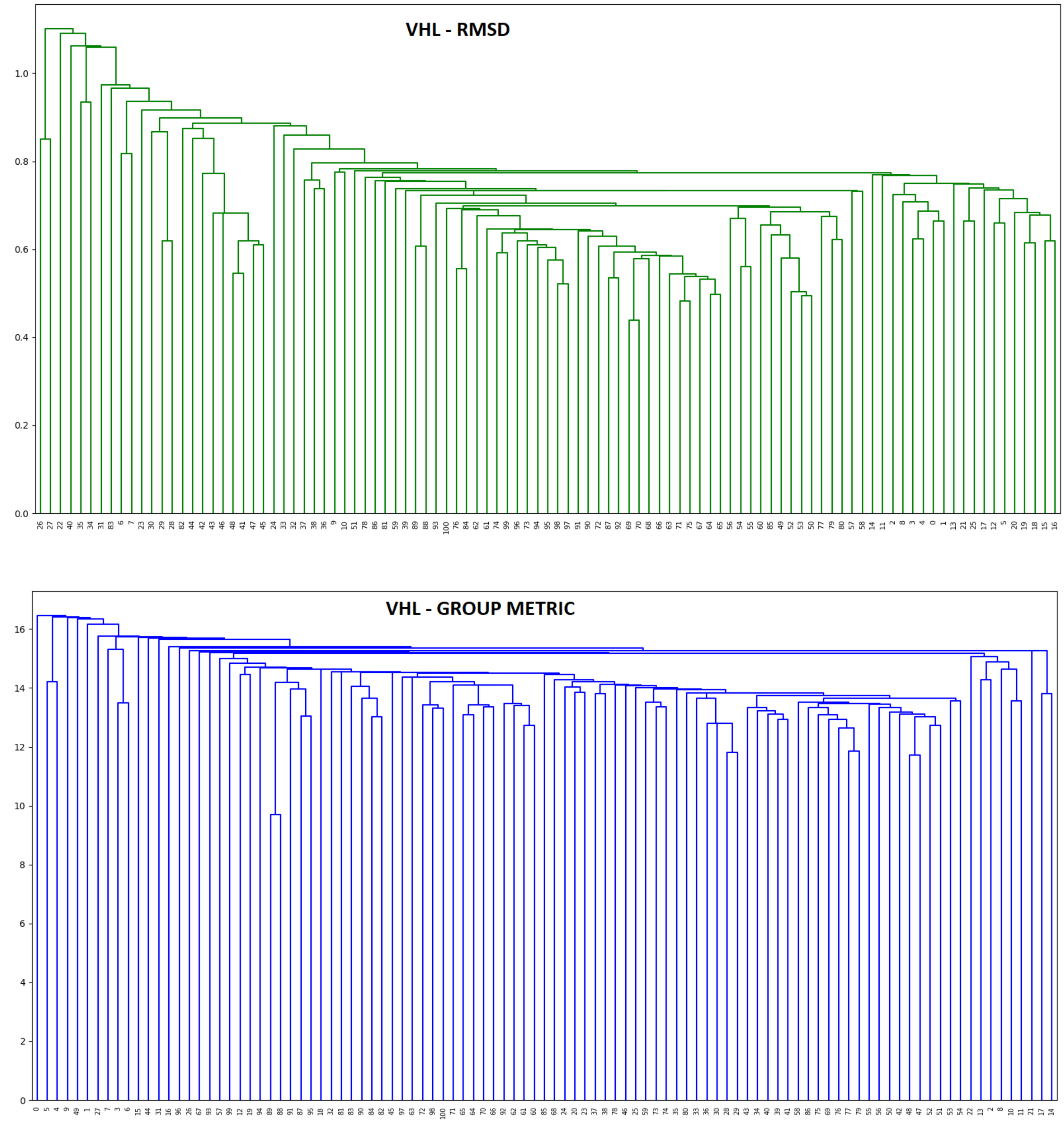

Figure 4.2: Vhl comparison

Frataxin-RMSD	Frataxin-group metric	vhl-RMSD	vhl-group metric
0	1	26	0
1	80	27	5
17	6	22	4
159	4	40	49
8	111	35	1
7	110	34	27
9	84	31	7
6	115	83	3
5	116	6	6
2	113	7	15

Table 4.1: First 10 labels placed on the bottom left corners of both sets of dendrograms

The standard parameters for Affinity Propagation are a damping factor of 0.5 and a preference for all values equal to the median between all of the similarities.

What we obtained as the optimal clustering was neither inconclusive nor exceptional. The pairs of elements that stood out the most in the dendrogram where correctly put in the same cluster, as shown in the following table 4.2 that takes in consideration vhl dendrogram and clustering results.

Close elements	Cluster
5,4	0
7,3,6	1
88,89	13
84,82	12
28,29	5

Table 4.2: Close elements in vhl dendrogram and the cluster they're in

The snapshots that didn't stood out and where closer in value, however, where randomly distributed between the clusters.

Overall, we have not obtained the desired results, we think that Affinity Propagation doesn't have the ability of finding the optimal clustering on it's own, but it has potential. A possible improvement could lie in the use of an optimization algorithm that evaluates the quality of the clusters and updates the preferences of the elements accordingly.

In any case, we will leave this considerations to people with a better expertise in this field and limit ourselves in reporting our findings.

Bibliography

[1] Doncheva,N.T., Klein,K., Domingues,F.S. and Albrecht,M. (2011) Analyzing and visualizing residue networks of protein structures. Trends Biochem. Sci., 36, 179–182.

[2] P. Csermely Creative elements: network-based predictions of active centres in proteins and cellular and social networks Trends Biochem. Sci., 33 (2008), pp. 569-576

[3] Cockroft,S.L. and Hunter,C.A. (2007) Chemical double-mutant cycles: dissecting non-covalent interactions. Chem. Soc. Rev., 36, 172–188.

[4] Dill,K.A., Ozkan,S.B., Shell,M.S. and Weikl,T.R. (2008) The protein folding problem. Annu. Rev. Biophys., 37, 289–316.

[5] Piovesan, D., Minervini, G., Tosatto, S.C.E. The RING 2.0 web server for high quality residue interaction networks. 2016. Nucleic Acids Research. doi:10.1093/nar/gkw315

[6] Doncheva,N.T., Assenov,Y., Domingues,F.S. and Albrecht,M. (2012) Topological analysis and interactive visualization of biological networks and protein structures. Nat. Protoc., 7, 670–685.

[7] Martin,A.J.M., Vidotto,M., Boscariol,F., Di Domenico,T., Walsh,I. and Tosatto,S.C.E. (2011) RING: networking interacting residues, evolutionary information and energetics in protein structures. Bioinformatics, 27, 2003–2005.

[8] http://protein.bio.unipd.it/ring

[9] `https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html`

[10] `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AffinityPropagation.html`

[11] W. Schreiner, R. Karch. (2012) Relaxation Estimation of RMSD in Molecular Dynamics Immunosimulations

[12] J. Abonyi, B. Feil. (2007) Cluster Analysis for Data Mining and System Identification.

[13] `https://pypi.org/project/tmscoring/`