Hackathon Report

Group 7:

Alon Fridman, Daniel Brown, Edan Patt, Linoy Tsaban and Yitzchak Vaknin

3D Data Processing in Structural Biology - 76562

July 20, 2021

Introduction

Nanobodies are a class of antigen-binding protein derived from camelids that achieve comparable binding affinities and specificities to classical antibodies, despite comprising of only a single 15 kDa variable domain [1]. Our goal in this project was to find whether or not there's a connection between the sequence similarity of nanobodies and their corresponding structures. We received a very large amount of data (1,000,000 samples) and used clustering algorithms for both their sequence similarity and structural similarity and analyzed the results.

Methods

Nanobody database

Throughout our research and analysis, we used the one million spike protein RBD binding nanobodies database provided to us in fasta format.

Sequence clustering with CD-Hit

For the first part of our work, clustering the nanobodies based on the protein sequences, we used the CD-Hit program. We ran the algorithm multiple times while trying different values for the 'c' parameter (which represents the sequence identity threshold): 0.95, 0.9 (90% match, the default value), 0.8 and 0.7. For the rest of the parameters we kept the default values.

Data preprocessing with TrainedNanoNet

We used the TrainNanoNet model for the first stage of obtaining a structural representation of each nanobody sequence. We fed the network an input of nanobody sequences and received an output of xyz coordinates in the format of a number_of_sequences x 140 x 3 matrix, where 140 is the maximum length of a single nanobody. We then converted the matrices into PDB files using the matrix_to_pdb function available in HackathonCatoPDB.py.

Calculating correspondence size for each nanobody

For the second part of our work, clustering the nanobodies based on structural features, we used the TrainedNanoNet output to generate a correspondence list for each pair of nanobodies by running a modified version (available in "cpp_code/correspondence_calc.cpp") of our alignment code from exercise 2. We removed the code that applied the transformation and calculated the distance between each pair of atoms (for each pair of nanobodies) and added the pairs with distance shorter than a given threshold (which we set once to 1 Å and once to 0.8 Å) to the list. We then wrote the size of the correspondence list for each pair to a csv file, creating a symmetrical matrix such that its ij entry contains the size of the match between nanobody i and nanobody j. Since the creation of the correspondence lists took too long, we opted for different more feasible approaches:

- From the clusters we got using the CD-Hit algorithm, we selected at random, 1,000 pdbs from the biggest cluster.
- From the clusters we got using the CD-Hit algorithm, we selected at random 1,000 pdbs from the top 200 biggest clusters, five representatives from each cluster.
- Out of all the one million samples we sampled 1,000 pdbs at random.

In each of these approaches we continued by creating a corresponding matrix for the selected representatives. Hyper parameters used are c=0.9 for CD-Hit and distance threshold of 1 Å for the matches calculation.

Verifying quality of sampling and CD-Hit clustering

In the second approach explained above, where we sampled five representatives from each of the 200 biggest clusters, we wanted to check whether there is indeed a lower RMSD between a quintuple from the same cluster and a higher RMSD between nanobodies from different quintuples.
We took the following five nanobodies that were assigned to the same cluster by CD-Hit:

1. 7168_4th_CoV2_BL_4943668-1
2. 7168_4th_CoV2_BL_13963198-1
3. 7168_4th_CoV2_BL_10539679-1
4. 7168_4th_CoV2_BL_5537038-1
5. 7168_4th_CoV2_BL_10539898-1

We aligned them onto each other with ChimeraX. The RMSDs we got by aligning all of them to the fourth one were between 0.3 Å and 0.7 Å. When we took nanobody 7168_4th_CoV2_BL_283813-1 which was assigned to a different cluster, the RMSD received was 2.4 Å (Fig. 1).

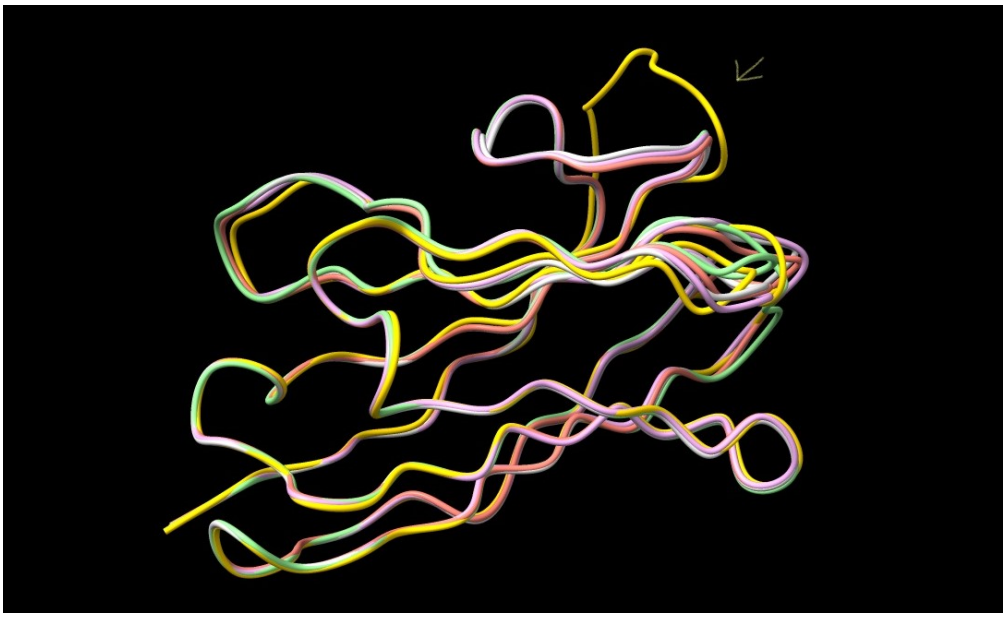


Figure 1: **Alignment of all six nanobodies in ChimeraX**. In yellow is the 7168_4th_CoV2_BL_283813-1 nanobody.

Another nanobody that was assigned to a different cluster than the one the five nanobodies above were assigned to (by CD-Hit) is 7168_4th_CoV2_BL_484036-2. The RMSD score we received for its alignment with the fourth nanobody is 1.2 Å (Fig. 2).

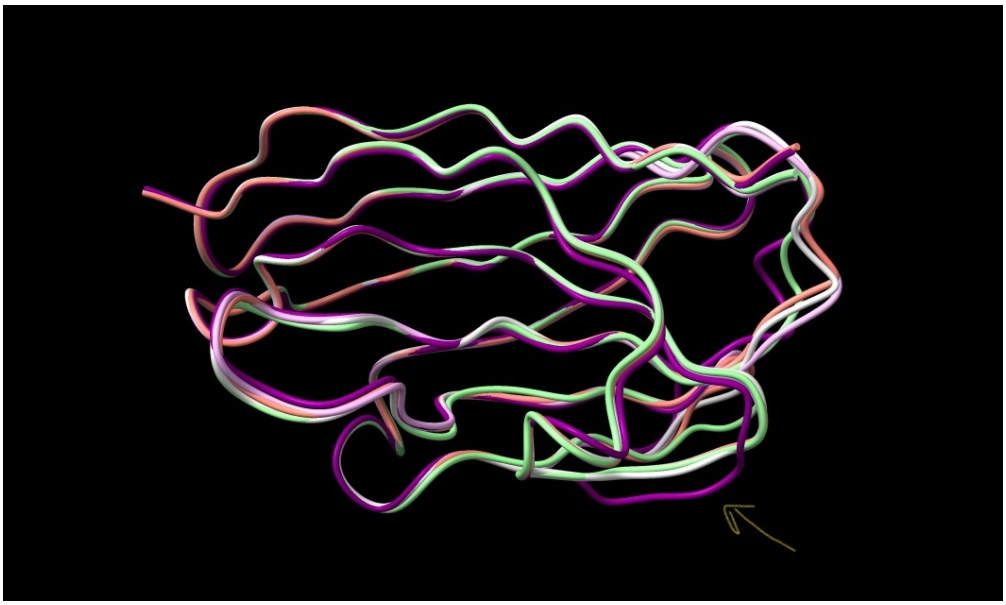


Figure 2: **Alignment of all six nanobodies in ChimeraX**. In purple is the 7168_4th_CoV2_BL_484036-2 nanobody.

These two examples, along with other examples that we didn't display here, contributed to our confidence in two things. First, that CD-Hit's clustering by sequences indeed tells us something about the nanobodies' structures. The second thing is that our five representatives from each cluster indeed represent a specific structural pattern that is unique for that cluster.

Structural clustering methods

Each set of data points went through the same pipeline:

1. Standardization/Normalization: This step is optional. In this stage data points (rows of the correspondence matrix) were either standardized or normalized using one of the methods: StandardScaler and MinMaxScaler.

2. Dimension Reduction: This step is also optional. Data points going through this have their feature dimension reduced by one of the two methods: PCA and t-SNE. We used this step mostly for visualization purposes (which is why we only allowed to reduce dimensions to either two or three).

3. Clustering Method: This is the only mandatory step in our pipeline. We allowed for five different clustering methods such as Kmeans, Spectral clustering, MiniBatchKmeans, Hierarchical clustering and DBScan

Clustering methods and parameters used

Choosing a clustering method

In this section we struggled to find the correct way to cluster our data. We noticed that trying to cluster millions of data points was not feasible (with respect to both run-time and memory considerations).
So we used our 1,000 random representatives (approach two described above - we selected at random five representatives from the top 200 biggest clusters we got from CD-Hit). After many trials, we decided the most suitable clustering method for the task at hand is MiniBatchKmeans. This algorithm runs Kmeans iteratively. It starts by randomly initiating centroids and sampling a minibatch of points. Then the centroids are updated on a per-sample basis. In each iteration another minibatch is randomly chosen and centroids are updated accordingly after the addition of each sample from the minibatch. This in theory could work for very large data sets. If we were to find a way of handling more data this method could scale up easily which is what guided us in choosing this method. While the more conventional methods such as Kmeans, DBScan and even Hierarchical clustering seemed like good choices for this task, MiniBatchKmeans has better computational complexity which is significant when dealing with large amounts of data while still achieving similar results to the classic Kmeans [2].

Clustering with MiniBatchKmeans

After creating the correspondence matrix for our 1,000 representatives, we ran the pipeline multiple times - with t-SNE dimensionality reduction and without and with either standard-

ization or normalization. Using MiniBatchKmeans we clustered the output points of the pipeline. We used the elbow method to find the appropriate k and followed that by using our labels to compare between the sequence based clusters and the correspondence based clusters.

Results

Sequence clustering with CD-Hit

After analyzing the results of the CD-Hit we ran with parameter c = 0.9, we noticed some interesting facts about the clusters -

- Nanobodies were divided into 226,797 different clusters
- 187,573 of them contain only one or two nanobodies
- The ten biggest clusters are in sizes:
  - 22,786
  - 8,916
  - 8,899
  - 7,061
  - 6,511
  - 6,175
  - 6,083
  - 5,188
  - 4,781
  - 4,484

There are significant differences between the sizes of the clusters, with most of the nanobodies assigned to their own cluster by themselves rather than bigger clusters. This is one of the reasons we chose to take five representatives only from the 200 biggest clusters (five from each) and 1,000 nanobodies only from the first cluster.

Structural clustering

As previously stated above, to optimize our structural clustering with MiniBatchKmeans, we used the elbow method to find a potentially optimal k. To do so, we used our 1,000 representatives and ran the clustering algorithm for k ranging from 3 to 300. We looked at the SSE scores of those clusters (sum of squared differences between each observation and its group's mean).
The graphs in Fig. 3 show the SSE scores when using different combinations of our pipeline steps for different values of k.

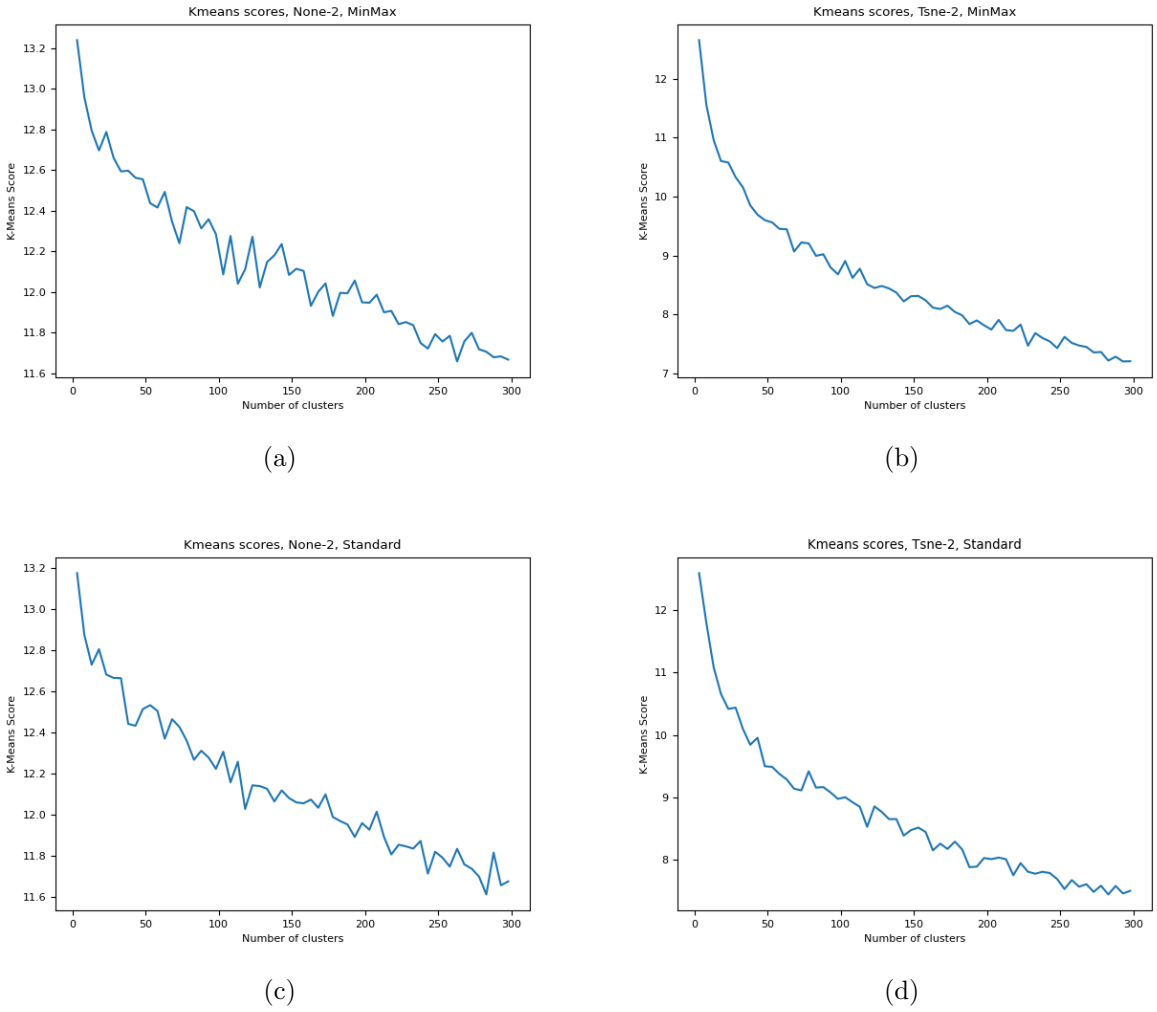


Figure 3: **Elbow method experiments.** (a) MiniBatchKmeans with MinMax normalization and without dimension reduction. (b) MiniBatchKmeans with MinMax normalization and with t-SNE. (c) MiniBatchKmeans with standard scaling and without dimension reduction. (d) MiniBatchKmeans with standard scaling and with t-SNE. For all these four sub figures, y axis values are in log scale.

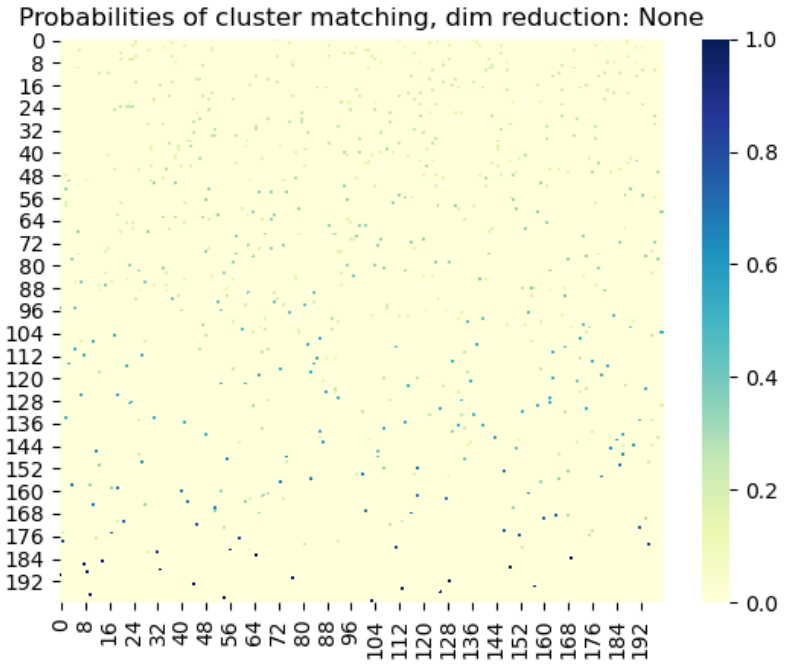
As we can see the SSE graphs are "choppy" although the score should decrease monotonously as k increases [3]. The reason we get small upticks is because MiniBatchKmeans works iteratively which causes some irregularity.
As seen in Fig. 3, we noticed that using t-SNE before clustering did decrease the SSE score, and caused for a more stable score in total (which shows in the graphs, they are more "smooth"). As for the k parameter, by looking at these graphs we could not spot an obvious plateau.

Cluster comparison

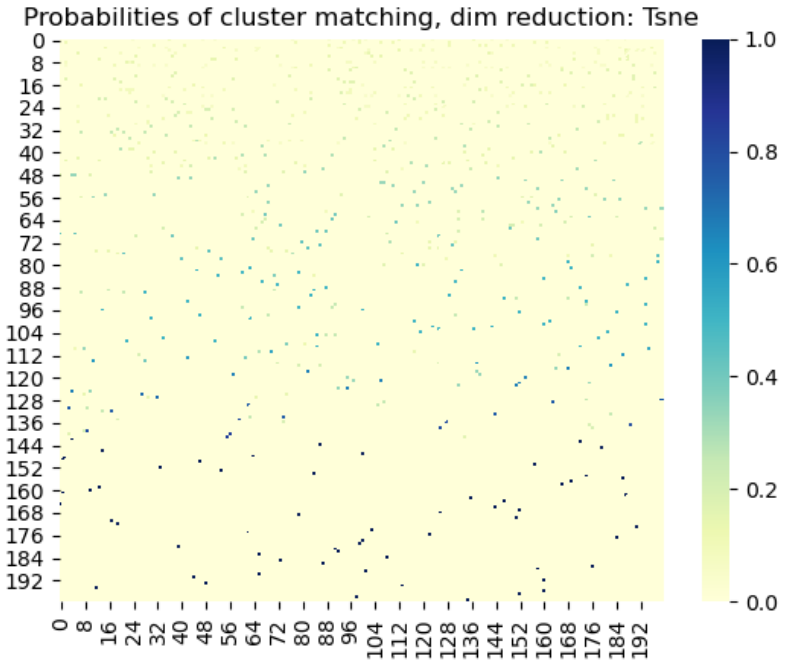
We wanted to check whether there is any correlation between the CD-Hit sequence based clusters and our structural clusters. We chose our structural clusters to be those achieved by using MiniBatchKmeans with k=200. While we could not find an "elbow", we thought that using the same amount of clusters as we have from the sequence based clustering is reasonable. For the structural clustering we used the 1,000 representatives whose correspondence matrix we fed to the pipeline with MinMax normalization (we chose this normalization due to technical issues preventing us from comparing the sequence based clusters to the structural clusters received when using the StandardScaler), once with t-SNE dimension reduction and once without.
We hypothesize that if there is indeed a significant connection between sequence similarity and structural similarity, then to some extent we would expect for nanobodies that were assigned to the same cluster by CD-Hit to also be assigned to the same cluster by our structural clustering algorithm.
Fig. 4 shows a heatmap of sorts: each row represents the distribution of nanobodies from a given structural cluster on all CD-Hit clusters. Meaning, a given (x,y) point on the graph shows the ratio of nanobodies from the structural cluster y that are were assigned to the same CD-Hit cluster x. The color of the point represents the ratio, such that if half of the nanobodies that were assigned to cluster number 120 in the structural clusters were assigned together to cluster 56 in the CD-Hit clusters, we would see a turquoise (0.5 in the color scale) colored point on the graph in coordinates (56,120).
Rows with one very dark point mean that the cluster translated well from the structural cluster to the sequential cluster, while rows with very light spots mean that the cluster was distributed more or less normally throughout different sequential clusters.
By taking the average over the maximal point of each distribution we found there to be a possibly significant correlation between the two different types of clusters. When clustering without dimension reduction in the pipeline, we received the average over the maximal point of each distribution to be 0.47 with a standard deviation of 0.23. When clustering after a t-SNE dimension reduction we received and average of 0.57 with a standard deviation of 0.31.
It is also probably important to note that when clustering the nanobodies using our structural clustering algorithm into 200 clusters, we found that on average there were 4.8 nanobodies in every cluster with a standard deviation of only 0.3.

Discussion

We learned a lot about working as a team, assigning tasks, working in tandem while also helping each other out when things didn't go as planned. If we were to do one thing differently it would have probably been during the preprocessing of the data. Since we created 1,000,000 different PDB files, our correspondence algorithm, which opens and close each file took too long to run.
We now know that we should have probably written larger PDB files, containing more than one nanobody each even though this could have caused a memory issue.



(a)



(b)

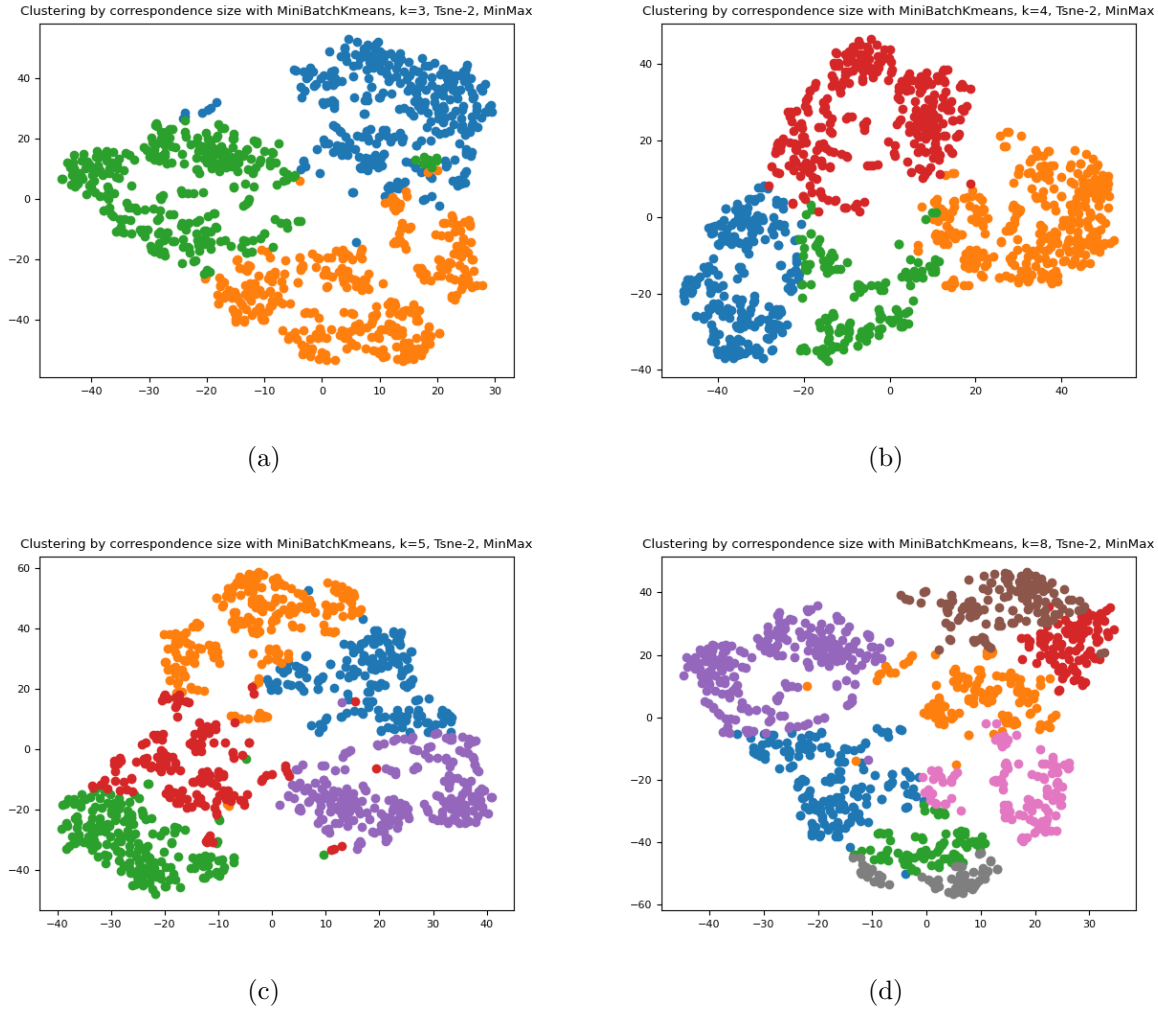
Figure 4: **Probability distribution heatmap.** (a) Distribution of nanobodies from structural clusters in CD-Hit sequence clusters (structural clustering without dimension reduction). (b) Distribution of nanobodies from structural clusters in CD-Hit sequence clusters (structural clustering with t-SNE dimension reduction).

During the process of dimension reduction and structural clustering we experimented in small scales and therefore for significant results there is need for larger scale experiments. For visualization purposes we tried both t-SNE and PCA, but in hindsight we should have tried UMAP as well. To conclude, the project raised a lot of very interesting questions, and many more ideas. Our program was ready to try out a variety of different techniques so we could compare different hyper parameters, clustering algorithms and dimension reduction strategies. Obviously everyone struggled with the lack of time but we'd be very interested in trying out and learning more about clustering and analyzing the results.

Future ideas

- Scaling to larger dataset ans trying different parameters
- Trying different structural representations of proteins
- Applying our program to different data - for example antibodies, and comparing results
- Applying the structural clustering pipeline on all members of a certain CD-Hit cluster
- Comparing clusters using different methods
- Using feature selection methods
- Choosing different representatives from the CD-Hit clustering

Supplementary Figures



(a) (b)

(c) (d)

Figure 5: **Clustering visualisation with t-SNE of the correspondence sizes. Clustering was done using MiniBatchKmeans after MinMax normalization with different values of k.** (a) MiniBatchKmeans with k=3. (b) MiniBatchKmeans with k=4. (c) MiniBatchKmeans with k=5. (d) MiniBatchKmeans with k=8.

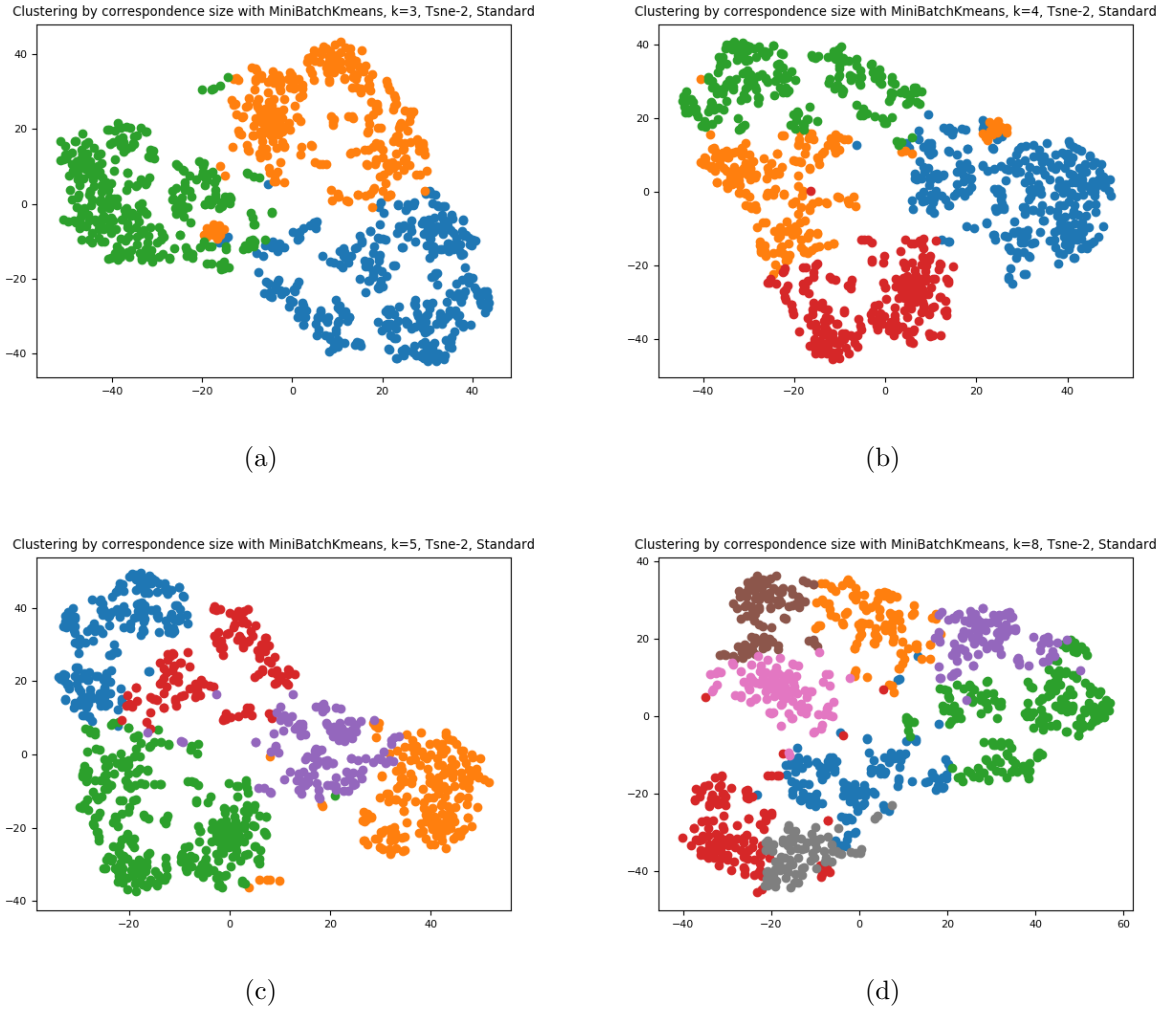
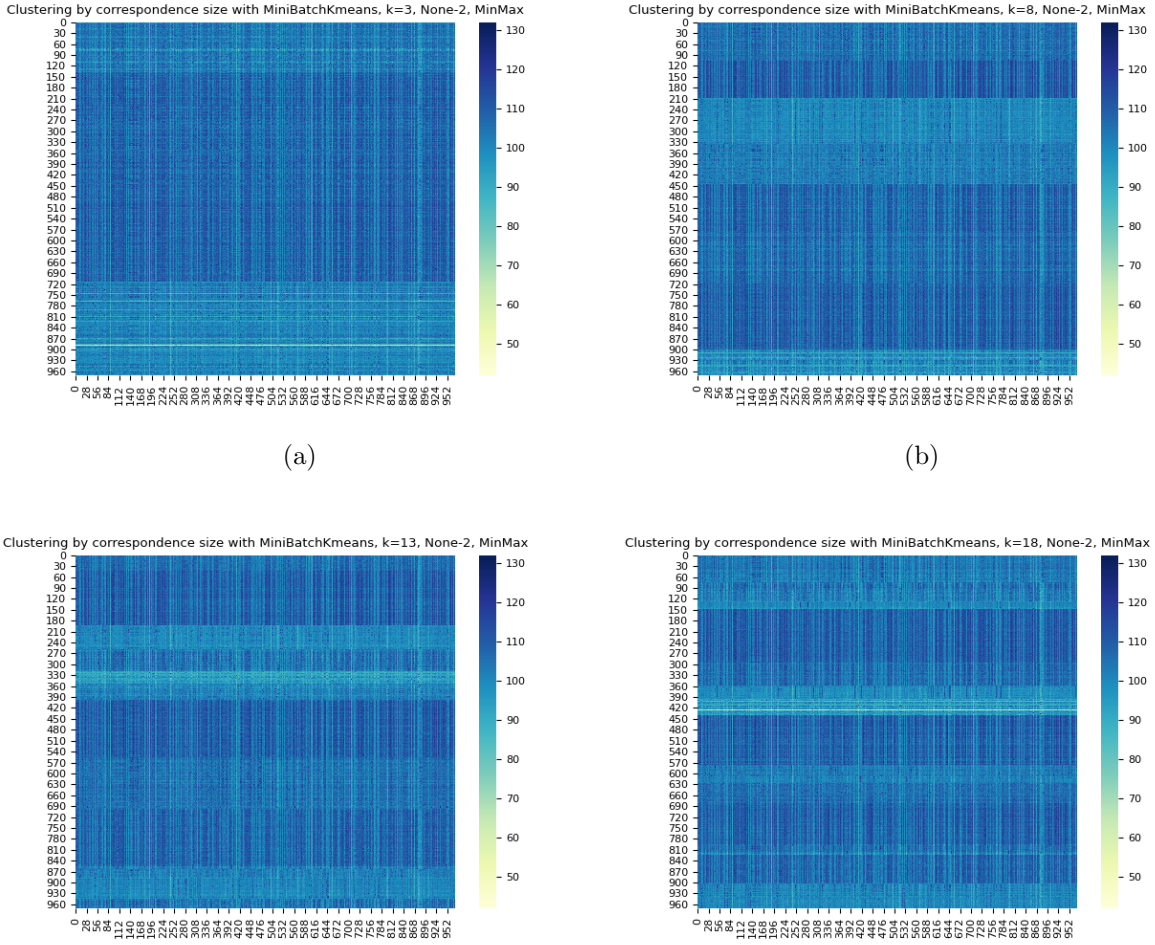


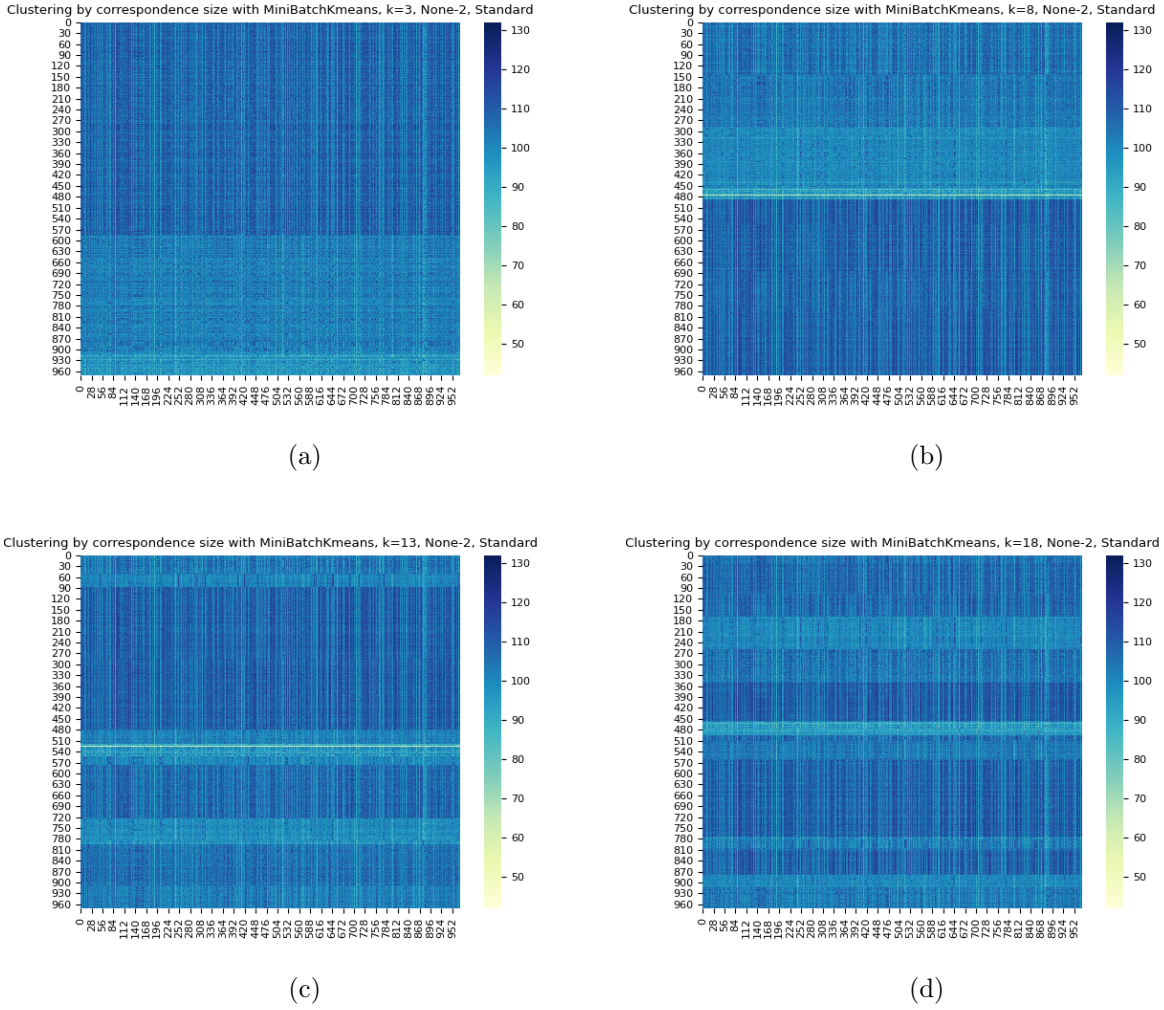
Figure 6: **Clustering visualisation with t-SNE of the correspondence sizes. Clustering was done using MiniBatchKmeans after standardization with different values of k.** (a) MiniBatchKmeans with k=3. (b) MiniBatchKmeans with k=4. (c) MiniBatchKmeans with k=5. (d) MiniBatchKmeans with k=8.



(a) (b)

(c) (d)

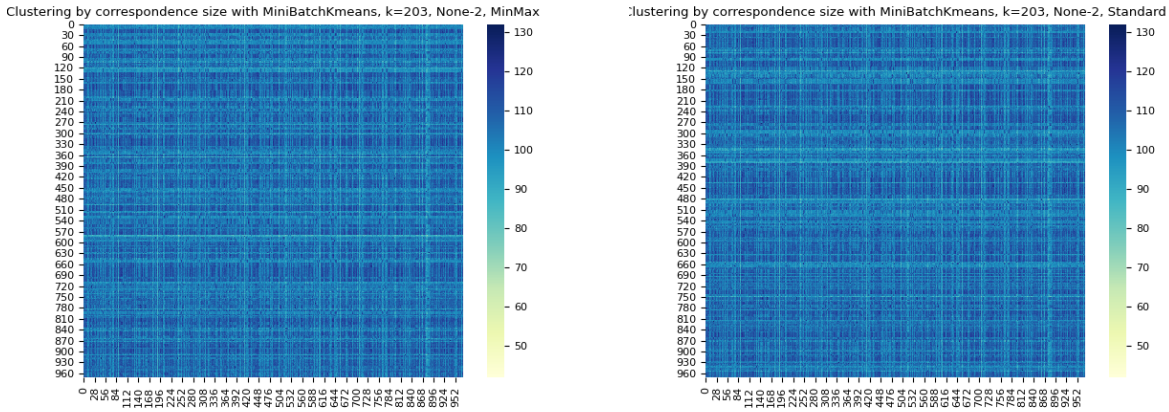
Figure 7: **Heatmaps of correspondence sizes clustering without dimension reduction. Clustering was done using MiniBatchKmeans after MinMax normalization with different values of k.** (a) MiniBatchKmeans with k=3. (b) MiniBatchKmeans with k=8. (c) MiniBatchKmeans with k=13. (d) MiniBatchKmeans with k=18.



(a) (b)

(c) (d)

Figure 8: **Heatmaps of correspondence sizes clustering without dimension reduction. Clustering was done using MiniBatchKmeans after standardization with different values of k.** (a) MiniBatchKmeans with k=3. (b) MiniBatchKmeans with k=8. (c) MiniBatchKmeans with k=13. (d) MiniBatchKmeans with k=18.



(a) (b)

Figure 9: **Heatmaps of correspondence sizes clustering without dimension reduction.** (a) MiniBatchKmeans clustering with $k \approx 200$ after MinMax normalization. (b) MiniBatchKmeans clustering with $k \approx 200$ after standardization.

References

[1] Laura Mitchell and Lucy Colwell. "Comparative Analysis of Nanobody Sequence and Structure Data". In: *Proteins: Structure, Function, and Bioinformatics* 86 (Mar. 2018).

[2] J. Alonso. "K-means vs Mini Batch K-means: a comparison". In: (2013).

[3] K. D. Joshi and Prakash S. Nalwade. "Modified K-Means for Better Initial Cluster Centres". In: (2013).