Clustering algorithms

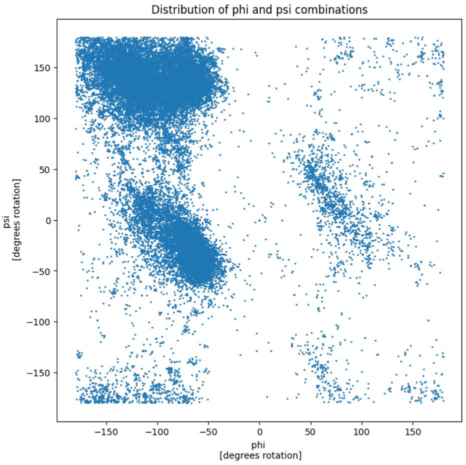
Johanna Norell

March 1, 2024

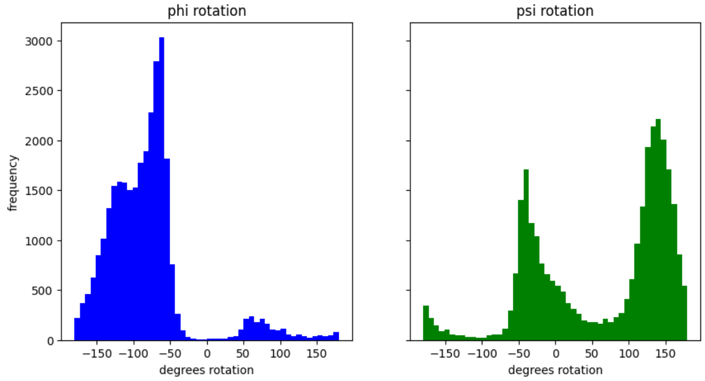
Visualising the data

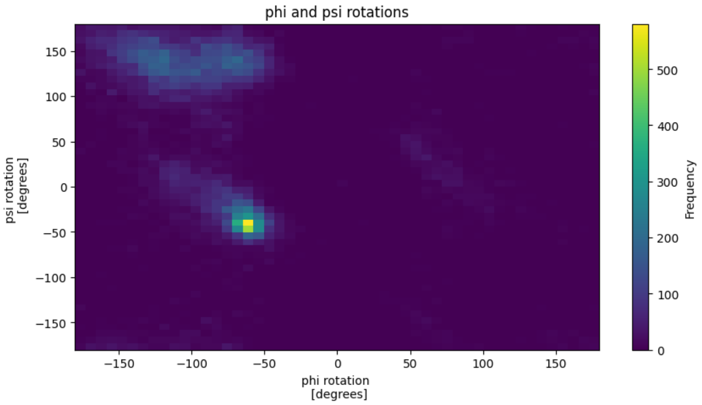
1. Show the distribution of phi and psi combinations using:

a) scatter plot



b) a 2D histogram

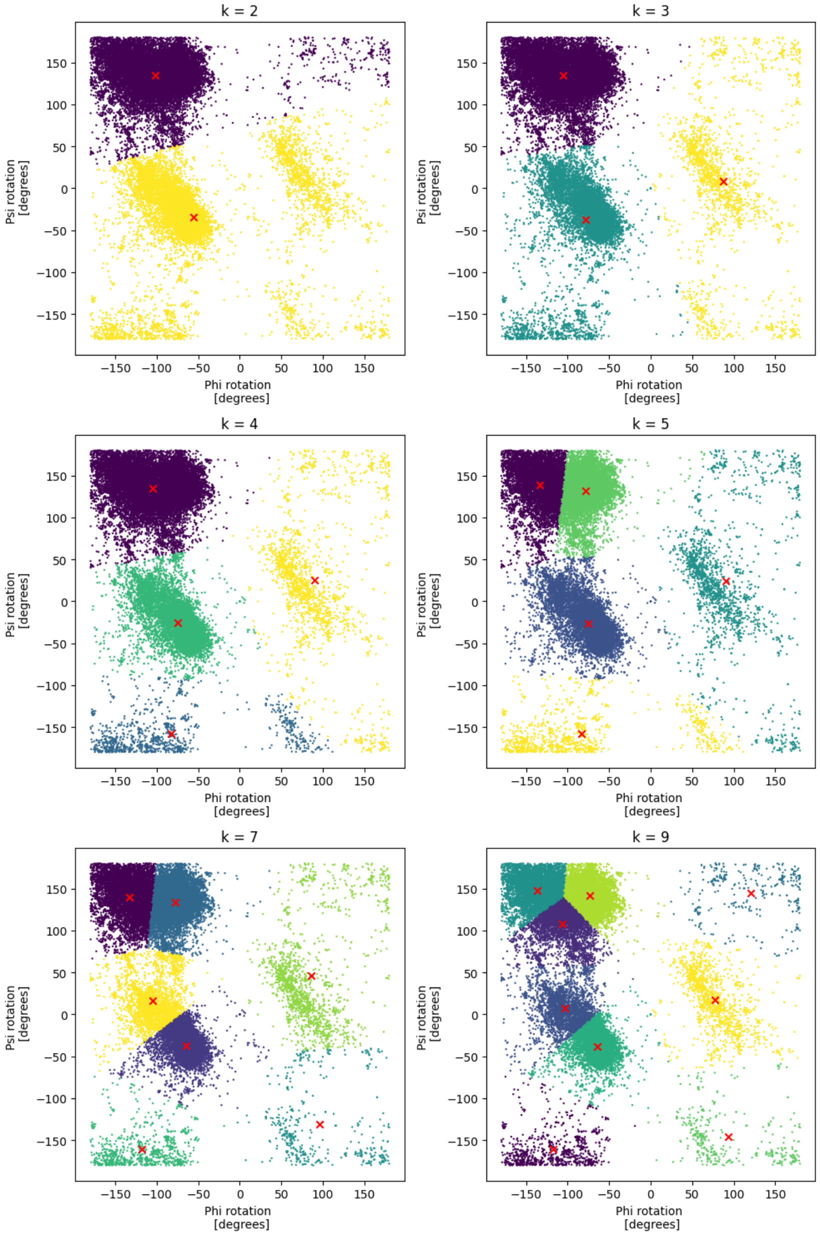



phi and psi rotations
psi rotation
[degrees]
150
100
50
0
−50
−100
−150
−150
−100
−50
0
50
100
150
phi rotation
[degrees]
Frequency
500
400
300
200
100
0

1 K-means clustering

2. Use the k-means clustering method to cluster the phi and psi angle combinations in the data file.

(a) Experiment with different values of k. Suggest an appropriate value of k for this task and motivate this choice



k = 2

k = 3

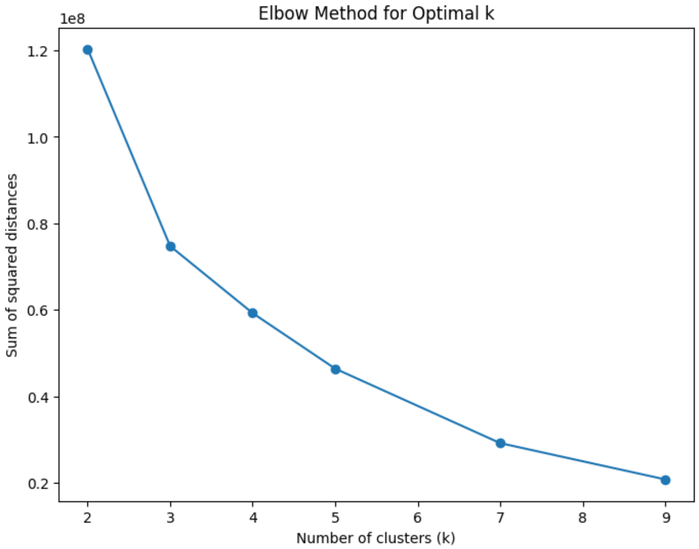
k = 4

k = 5

k = 7

k = 9

According to the plots above, we would say that k=3 or k=4 seems most reasonable, to conclude the optimal k, we can look at the elbbow plot below.



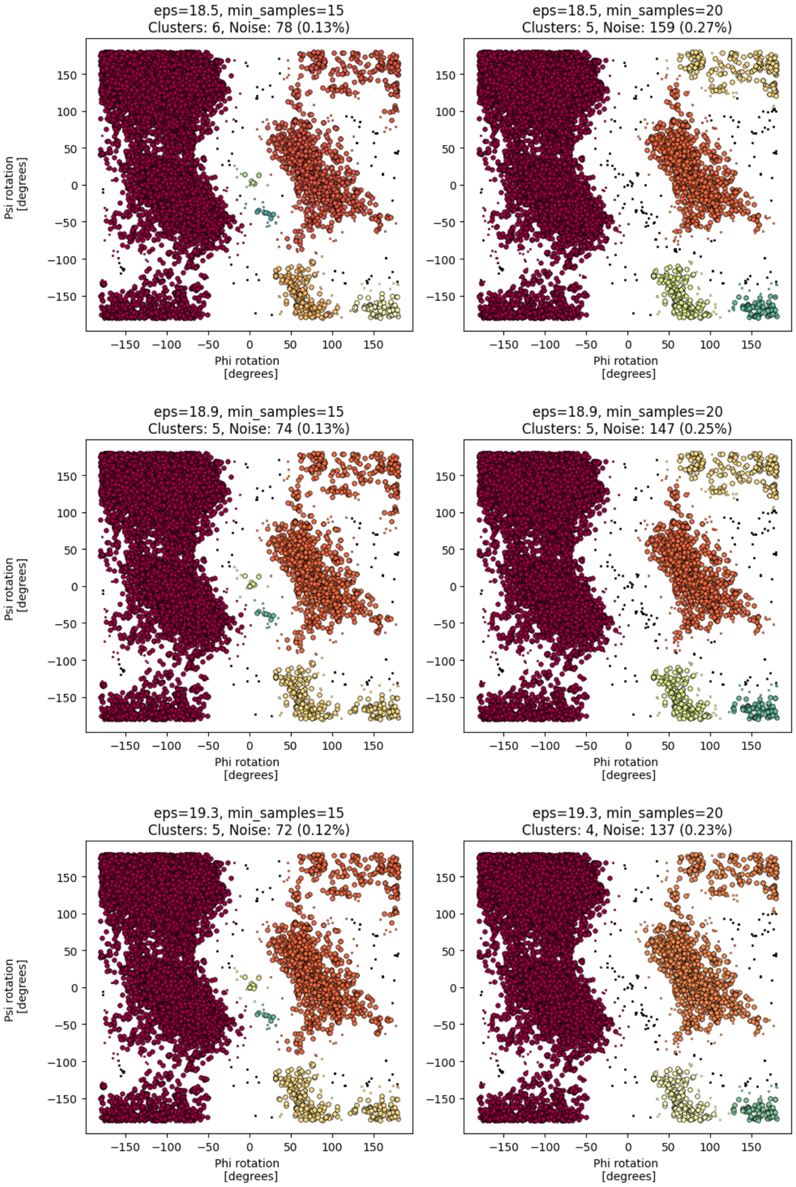
In the plot we are looking for the greatest difference in the derivative from ”incoming” and ”outgoing”. It is obvious in this plot that this is found on k=3, which tells us that k=3 is the most optimal k to choose.

(b) Do the clusters found in part (a) seem reasonable?

Like already touched upon, the clusters in k=4 is fairly reasonable. The biggest criticism to these clusters is that points which wishfully would have been considered noise is clustered into the yellow and blue one which becomes scattered. In conclusion, the clusters achieved are fairly good but far from perfect.

3. Use the DBSCAN method to cluster the phi and psi angle combinations in the data file.

(a) Motivate the choice of: i. the minimum number of samples in the neighborhood for a point to be considered as a core point, and ii. the maximum distance between two samples belonging to the same neighborhood (“eps” or “epsilon”). Compare the clusters found by DBSCAN with those found using k-means.



Since increasing the minimum number of samples in general results in more clusters and increasing epsilon generally implies fewer clusters, the combination of the two values determines the number of clusters. Looking at our data, we figured that we would like to have some points classified as noise, thus we did not want to set epsilon too high, since there seemed to be a fair amount of scattered data points which preferably we did not want to belong to any clusters.

To begin with, we search for an epsilon range by keeping the minimum sample size constant at the arbitrary chosen value 30. An upper limit for epsilon was concluded to be around 30, since that generated only two clusters. The lower limit was found at around 15 since that generated a very large amount of noise points.

Next step was to find an interval for the minimum sample size, and we therefore set epsilon to be 20. A value of 10 for the minimum sample size generated only two clusters, whilst a value of 15 generated three clusters, and we therefore concluded the minimum sample size to have an upper limit of 30 and lower of 15.

Iterative, we continued the search for an somewhat optimal combination of epsilon and the minimum sample size and ended up concluding a epsilon of 18-19 and minimum sample size of 15-20 for the algorithm to return a number of clusters around 4-6 which we resonated was reasonable just by looking at the scatter plot of the data. In the plot above, one can see three different epsilon values in each row and two different minimum sample size values in each column. We found that the values: epsilon = 18.9 and minimum sample size = 20 would be the best values of the chosen variables for this data set (plot showed in mid row to the right). This since the clusters are relatively tight, but still not too many.

A significant different between the clusters found using the DBSCAN algorithm in comparison to k-means is that the big cluster to the right could not be divided into two different ones. When trying different values of epsilon and the minimum sample size, it tends to add very small additional clusters, like the one in the top left corner, rather than splitting the big one.

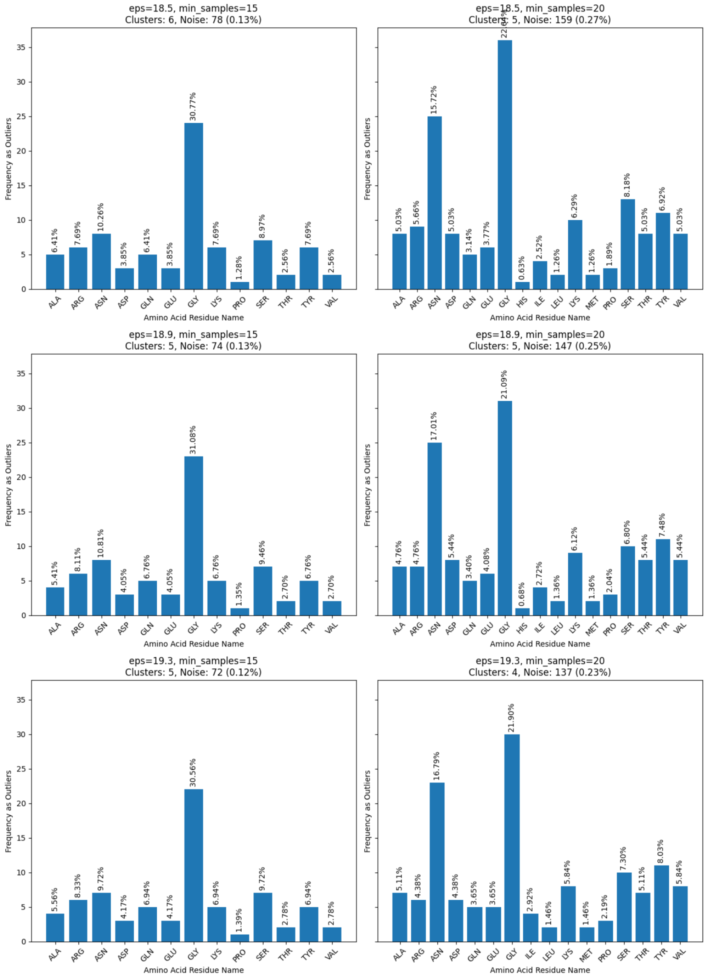
Another big difference between the two methods used is how they handle outliers. When using DBSCAN outliers will not get clustered but when using k-mean every point will get clustered. This can be seen with the DBSCAN having a lot of black points at x = 0 for the clustering made with min sample of 20. When comparing to the k-means each point at x = 0 gets classified to a cluster. The way DBSCAN handles outliers might be more suitable for our data since it is quite clear from observing the plots that the points at x = 0 should not be classified the way they are done in the k-means.

(b) Highlight the clusters found using DBSCAN and any outliers in a scatter plot.

This can be seen in the plots above.

2 Outliers

(c) How many outliers are found? Plot a bar chart to show how often each of the amino acid residue types are outliers.



eps=18.5, min_samples=15
Clusters: 6, Noise: 78 (0.13%)

eps=18.5, min_samples=20
Clusters: 5, Noise: 159 (0.27%)

eps=18.9, min_samples=15
Clusters: 5, Noise: 74 (0.13%)

eps=18.9, min_samples=20
Clusters: 5, Noise: 147 (0.25%)

eps=19.3, min_samples=15
Clusters: 5, Noise: 72 (0.12%)

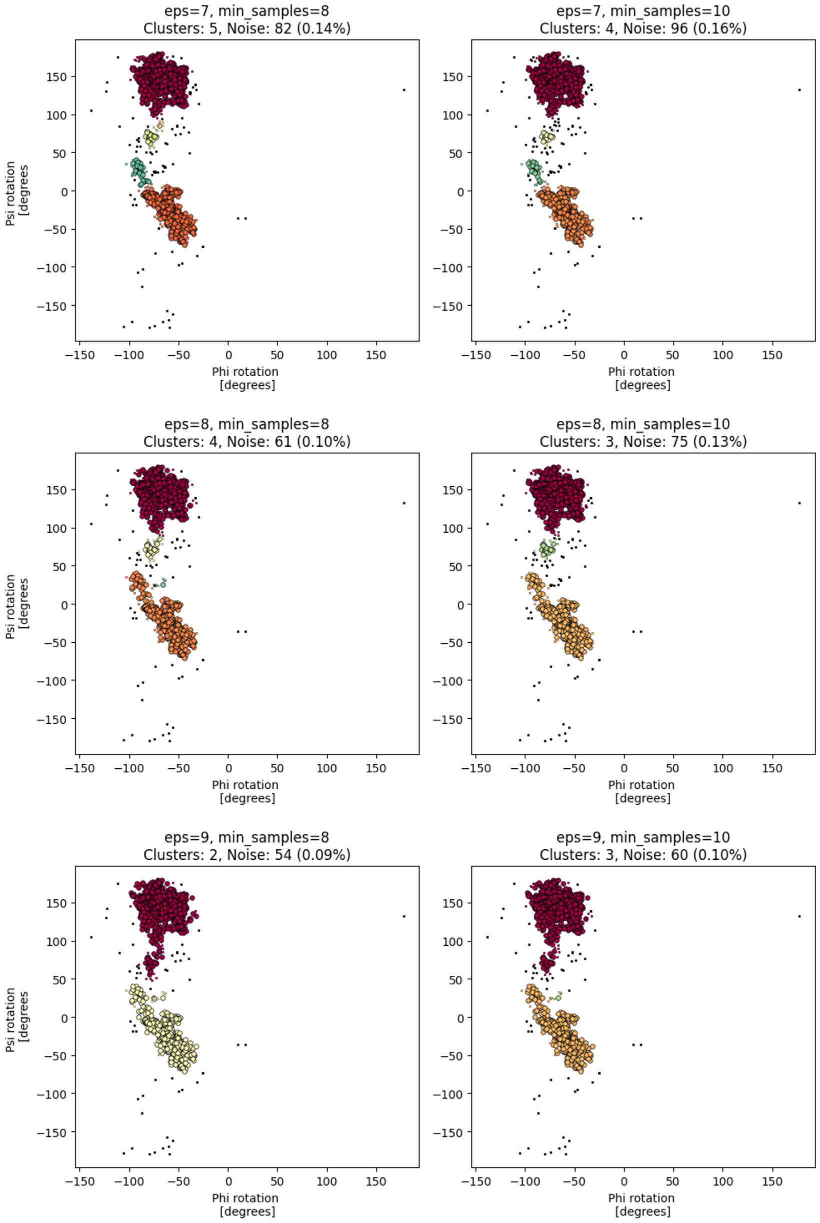
eps=19.3, min_samples=20
Clusters: 4, Noise: 137 (0.23%)

One can see that the amino acids with residue of type ASN or GLY are the ones with highest representation of outliers in all cases.

3 Sub-sampling data

4. The data file can be stratified by amino acid residue type. Use DBSCAN to cluster the data that have residue type PRO. Investigate how the clusters found for amino acid residues of type PRO differ from the general clusters (i.e., the clusters that you get from DBSCAN with mixed residue types in question 3). Note: the parameters might have to be adjusted from those used in question 3.

We can see in the plots that the clusters are found on the negative x-axis. No clusters are on the positive side of the x-axis which differs from the previous clusters. That is the biggest variation. This is due to the PRO amino acid apparently not having any positive Phi-rotation.


eps=7, min_samples=8
Clusters: 5, Noise: 82 (0.14%)
eps=7, min_samples=10
Clusters: 4, Noise: 96 (0.16%)
eps=8, min_samples=8
Clusters: 4, Noise: 61 (0.10%)
eps=8, min_samples=10
Clusters: 3, Noise: 75 (0.13%)
eps=9, min_samples=8
Clusters: 2, Noise: 54 (0.09%)
eps=9, min_samples=10
Clusters: 3, Noise: 60 (0.10%)
Psi rotation [degrees
Phi rotation [degrees]