Problem 2: Larger unlabeled subset
Part 1: Visualization
Question 1.1:
Provide at least one visualization which clearly shows the existence of the three main brain cell types described by the scientist, and explain how it shows this. Your visualization should support the idea that cells from a different group (for example, excitatory vs inhibitory) can differ greatly.

Answer:
The dimensionality of the $log_2(X+1)$-transformed data can be reduced by using PCA. In this case we need a high amount of PC's to explain a satisfying variance (see 1). For example, we need approximately 500 PC's to explain 60% of the variance. Therefore, there is a high chance that data points can differ quite a lot within each cluster.

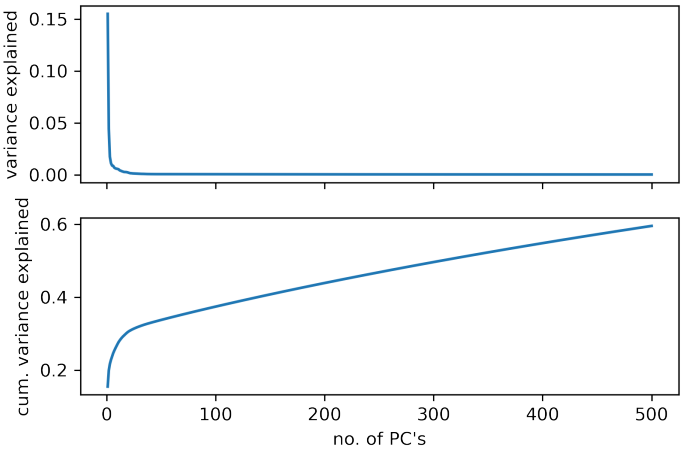

Figure 1: Variance explained (top) and cumulated variance explained (bottom)

For simplicity's sake, we only use 50 PC's to run the K-Means clustering function. Unfortunately, there is no set of labels corresponding to the data set. Therefore a comparison between the real cell types and the following shown clusters cannot be done. Nevertheless, the plot of the data can be split up into 3 clusters quite comfortably, which might be conform to three different cell types. This can either be done by plotting the data points over the first two PC's or be using T-SNE (see figure 2).

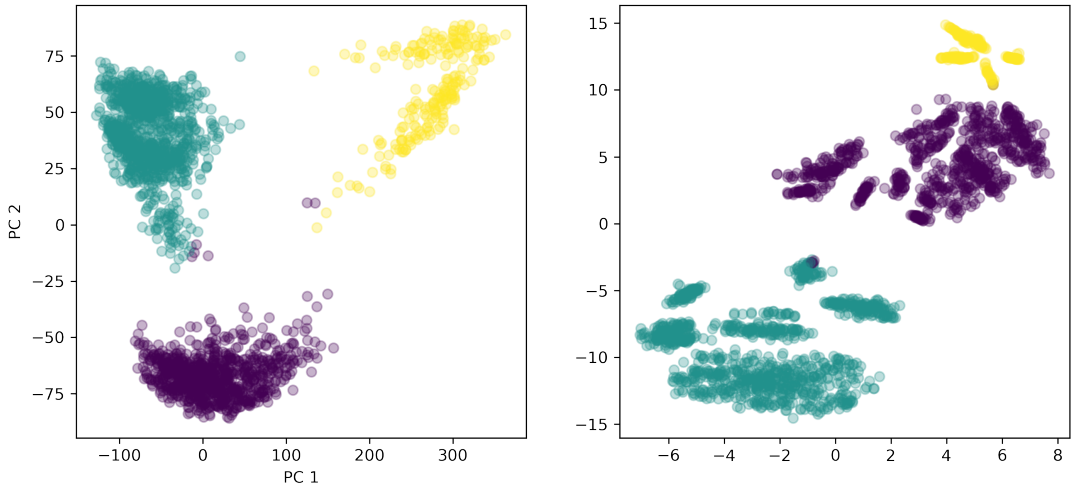

Figure 2: 3 clusters (over first two PC's, left), 3 clusters (T-SNE (perplexity=500), right)

Question 1.2:
Provide at least one visualization which supports the claim that within each of the three types, there are numerous possible sub-types for a cell. In your visualization, highlight which of the three main types these sub-types belong to. Again, explain how your visualization supports the claim.

Answer:
The "elbow" plot indicates, that the data could be clustered in 3 clusters (see figure 3). But because of a smooth transition towards a larger number of clusters, there could be as well multiple sub-clusters, as the scientist suggests.

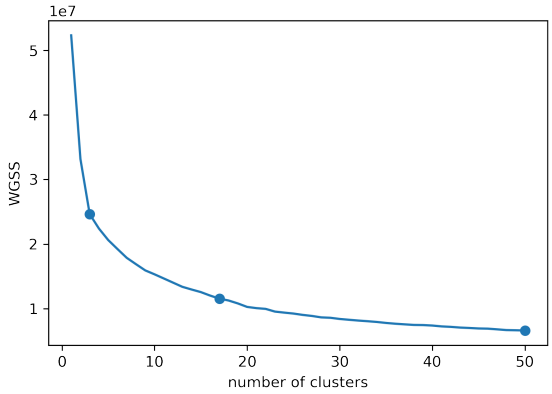

Figure 3: "Elbow" plot (WGSS (K-Means inertia) over number of clusters)

By reducing the perplexity of the T-SNE function to 100 the sub-clusters are moved apart, which makes the distinction easier (see figure 4). For demonstration purposes, 3 different number of clusters indicate that there are multiple sub-clusters. These might correspond to different cell types within the 3 main types. But there is a compromise between differentiation and merging of clusters. Big groups tend to melt together, whereas smaller groups are better distinguished with an increasing number of clusters. To find the "sweet spot", it is recommended to compare this model with labels of the real data set.

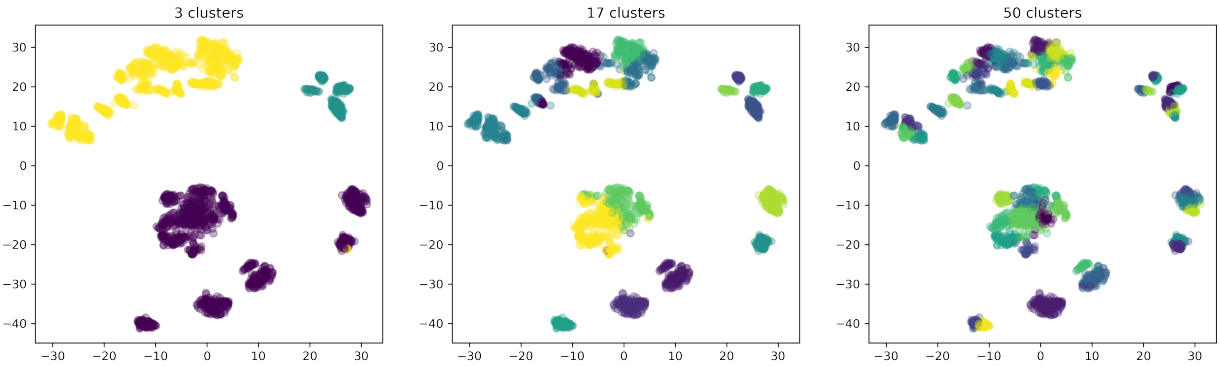

Figure 4: Different number of clusters for T-SNE (perplexity = 100)

To get further information, about how well the data was clustered, we could use a silhouette plot. The better the model represents the data, the higher the silhouette coefficient gets. This step is skipped at this point and done in the following part.

Part 2: Unsupervised Feature Selection
Question 2.1:
Using your clustering method(s) of choice, find a suitable clustering for the cells. Briefly explain how you chose the number of clusters by appropriate visualizations and/or numerical findings.

Answer:
As the "elbow" plot indicates, there are different cluster sizes which might fit to this case. Another tool to test the number of clusters for the data is the silhouette plot or the average silhouette score as a summarization of multiple silhouette plots (see figure 5). In general, the average silhouette score is quite low. For 3 clusters we achieve an average silhouette score of 0.3 which corresponds to the "elbow" of the "elbow" plot. In the range between 17 to 36 clusters we achieve a comparably high average silhouette score. But due to this wide range, we should not pick one number, especially with these low scores in general.

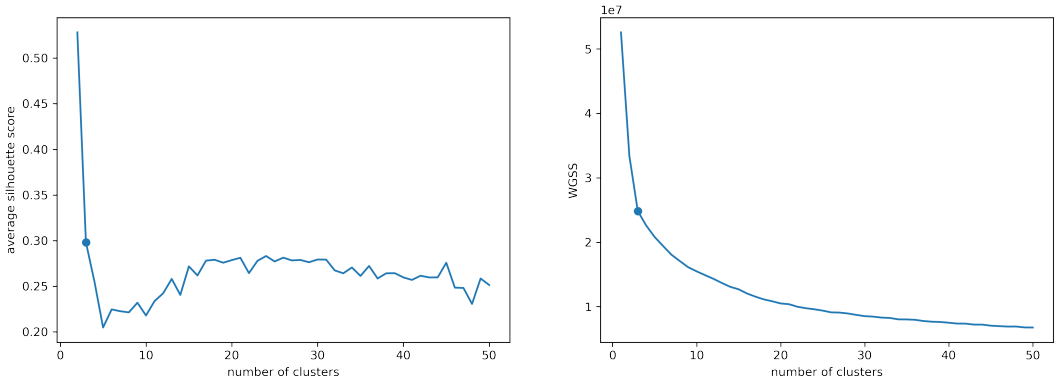

Figure 5: Average silhouette score (left) and "elbow" plot (right) over number of clusters

Therefore, the next considerations will focus on 3 clusters. This is supported by the information we have so far (e.g. by scientist's statement).

As seen in problem 1.1, we can easily plot 3 clusters in different plots, which makes the differentiation more robust (see figure 2)

Question 2.2:
We will now treat your cluster assignments as labels for supervised learning. Fit a logistic regression model to the original data (not principal components), with your clustering as the target labels. Since the data is high-dimensional, make sure to regularize your model using your choice of l_1, l_2, or elastic net, and separate the data into training and validation or use cross-validation to select your model. Report your choice of regularization parameter and validation performance.

Answer:
With using the cluster assignments as labels for the data set we receive the data:

X: $log_2(X_{original} + 1)$-transformed data from gene data set
y: cluster assignments from K-Means (3 clusters, given 50 PC's from PCA)

Further Boundaries:

- The train-test-split was done with an ratio of 0.33 ($\frac{n_{train}}{n_{test}} \approx \frac{2}{1}$).
- The function `sklearn.linear_model.LogisticRegression` was used.
- The regularization had minor effect on the score and was set to default settings (`penalty='l2'`).
- The maximum number of iterations had minor effect on the score and was set to default settings (`max_iter=100`).
- The `liblinear`-solver was used (`solver='liblinear'`).

Result: The test data could be classified with a score of 0.996 to the corresponding cluster.

Question 2.3:
Select the features with the top 100 corresponding coefficient values (since this is a multi-class model, you can rank the coefficients using the maximum absolute value over classes, or the sum of absolute values). Take the evaluation training data in p2_evaluation and use a subset of the genes consisting of the features you selected. Train a logistic regression classifier on this training data, and evaluate its performance on the evaluation test data. Report your score. (Don't forget to take the log transform $log_2(X+1)$ before training an testing.)
Compare the obtained score with two baselines: random features (take a random selection of 100 genes), and high-variance features (take the 100 genes with highest variance). Finally, compare the variances of the features you selected with the highest variance features by plotting a histogram of the variances of features selected by both methods.

Answer:
We draw the 100 most important features (highest sum of absolute values within each feature) from the problem above. These features are used to choose the features from the new dataset. The score on the test data set is 0.153[1].

By randomly choosing 100 features a score of 0.303 was achieved.

By choosing the 100 features of highest variance we can achieve a score of 0.924.

The first test score is very low due to a different setup of this problem compared with the problem above. One model has 3 clusters, whereas the new dataset has 36 different clusters. This has a great impact on the calculated labels and leads to the lowest score in this problem.

Choosing 100 features randomly leads to a higher score. But the highest score was achieved by choosing 100 features of highest variance. This makes sense, because the logic is comparable to the dimensionality reduction with PCA. Here we favor high PC's which represent high variances as well. The histograms show how far the variances of the second and the third approach are apart (see figure 6). Thus the labels can be calculated more reliable in the case of higher variances.

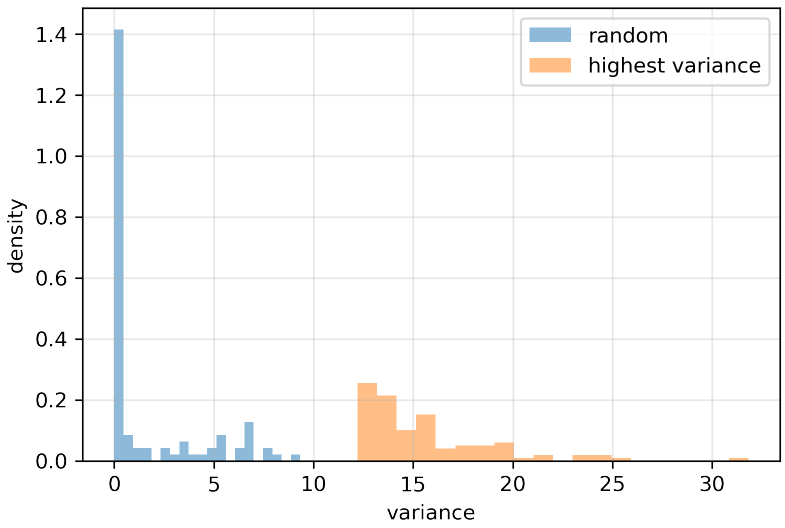

Figure 6: Histograms of 100 differently chosen features (integral sums to 1)

[1]The function parameters for `sklearn.linear_model.LogisticRegression` were set in the same manner as in question 2.2.

Problem 3: Influence of Hyper-parameters
Question 1:
When we created the T-SNE plot in Problem 1, we ran T-SNE on the top 50 PC's of the data. But we could have easily chosen a different number of PC's to represent the data. Run T-SNE using 10, 50, 100, 250, and 500 PC's, and plot the resulting visualization for each. What do you observe as you increase the number of PC's used?

Answer:
With the number of PC's we can influence the explained variance of the data after the reduction of the original data's dimensionality. The more PC's we add to the model, the more accurately we can reconstruct the original data.[2] But this does not necessarily mean that, we can get more insight from it.

Therefore we can influence the shape of the T-SNE plot by changing the amount of the PC's (see figure 7). The lower the number of considered PC's, the wider the spread between individual points in one cluster. Besides the density of the data points within one cluster, the distance between individual clusters is affected. With an increasing number of PC's, the clusters start to merge into one another.

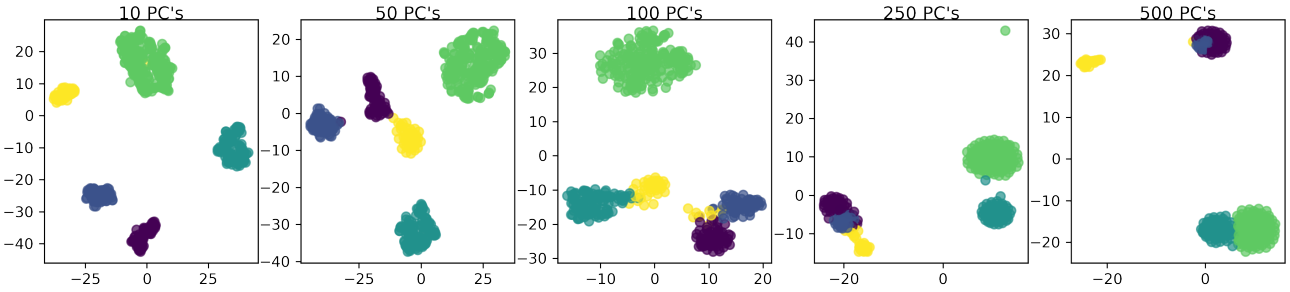

Figure 7: T-SNE plots for different number of PC's

Without knowing the clusters beforehand, we might expect 3 different clusters for 100, 250 or 500 PC's. Therefore, in this example the best visual distinction between the different clusters can be shown by using 10 or at maximum 50 PC's.

[2] In this case the $log_2(X+1)$-transformed data set from problem 1 was used.

Question 2:
Pick three hyper-parameters below and analyze how changing the hyper-parameters affect the conclusions that can be drawn from the data. Please choose at least one hyper-parameter from each of the two categories (visualization and clustering/feature selection). At minimum, evaluate the hyper-parameters individually, but you may also evaluate how joint changes in the hyper-parameters affect the results. You may use any of the datasets we have given you in this project. For visualization hyper-parameters, you may find it productive to augment your analysis with experiments on synthetic data, though we request that you use real data in at least one demonstration. [...]
For visualization hyper-parameters, provide substantial visualizations and explanation on how the parameter affects the image.
For clustering/feature selection, provide visualizations and/or numerical results which demonstrate how different choices affect the downstream visualizations and feature selection quality.
Provide adequate explanations in words for each of these visualizations and numerical results.

Answer:
Category A (visualization)
In the following task I will use the data from problem 1. Like before, I will move on with 50 PC's for the visualization with help of the T-SNE plot. The hyper-parameters of interest are the perplexity and the early exaggeration rate of the T-SNE function.

The perplexity parameter influences how the distance between individual points are set (see figure 8). With a high perplexity clusters start to merge into one another. In contrast, a too low perplexity results in clusters of low density. The documentation states that larger data sets require usually a larger parameter. The recommended range is between 5 and 50.

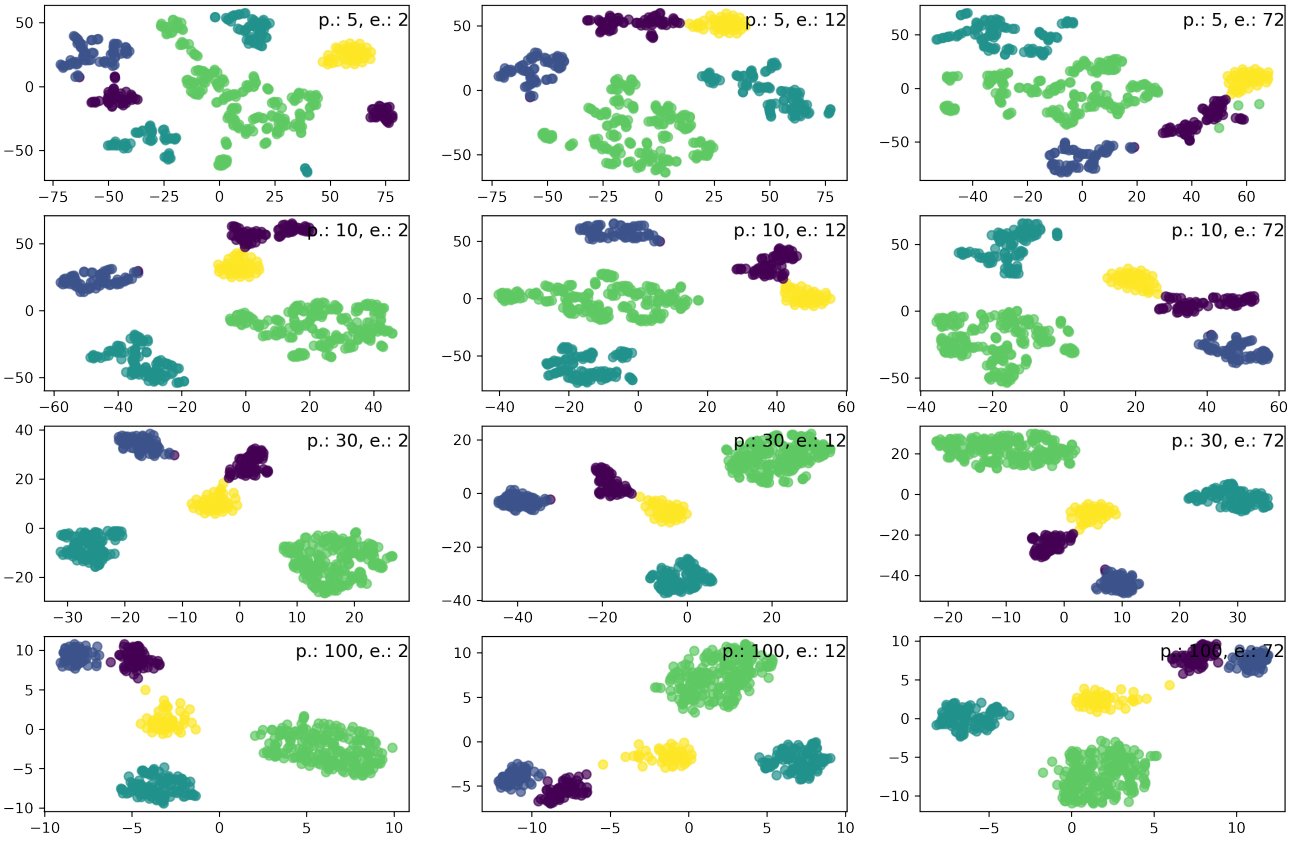

Figure 8: T-SNE plots for different hyper-parameters (p: perplexity, e: early exaggeration, 50 PC's)

According to the sklearn documentation the early exaggeration "controls how tight natural clusters in the original space are in the embedded space and how much space will be between them". By comparing the plots row-wise we can see small quality differences for constant perplexity parameters (see figure 8). The largest difference can be seen for a low perplexity of 5. Here the separation between the clusters for this perplexity was achieved. The default setting for this parameter is 12, which shows the best distinction between the different clusters.

In this comparison the best visualization was achieved with a perplexity of 30. The early exaggeration is in this case of minor weight due to comparable results.

Category B (clustering/feature selection)
For the next task I will use the same data set as before, but now as an unsupervised data set. Therefore I will analyze which number of clusters leads to the best results. The used tool is the K-Means algorithm.

The first approach is analyzing quantitative tools to estimate a reasonable number of clusters. Two different approaches are plotting the average silhouette score and the WGSS over the number of clusters (see figure 9). The silhouette plot gives three almost equally promising number of clusters with average silhouette scores around 0.36 to 0.37 for {3,4,5}. Here, we want to achieve high values to assign the data points to the clusters as good as possible. The second quantitative visualization is the "elbow" plot. Here, the "elbow" is a favorable candidate for the optimal number of clusters. Namely, the cluster numbers of {4,5,6} are good candidates.

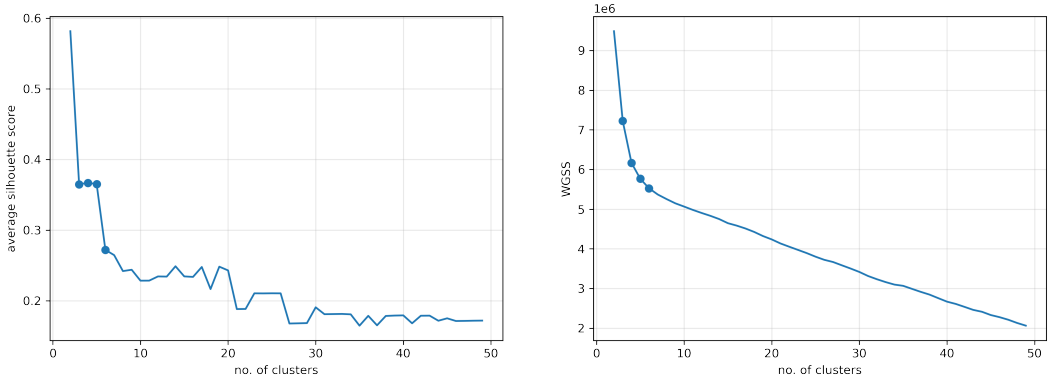

Figure 9: Silhouette and "elbow" plot (highlighted points $n = \{3, 4, 5, 6\}$, 50 PC's)

The visualization of the clustered data points over the first two PC's leads to two different possible conclusions (see figure 10). Without knowledge of the real number of clusters we could assume, that there are three, four or five different clusters. Each theory could be supported by the plots in figure 9 and the visualization in figure 10.

The most arguments lead to the conclusion that there are either three or five clusters. In case of three clusters, the neighbored points are assigned to one cluster with a fair space between these groups. In case of five clusters two clusters are sub-divided, which might be true as well. But in all visualizations of figure 10 there is space for subjective interpretations and therefore different decisions.

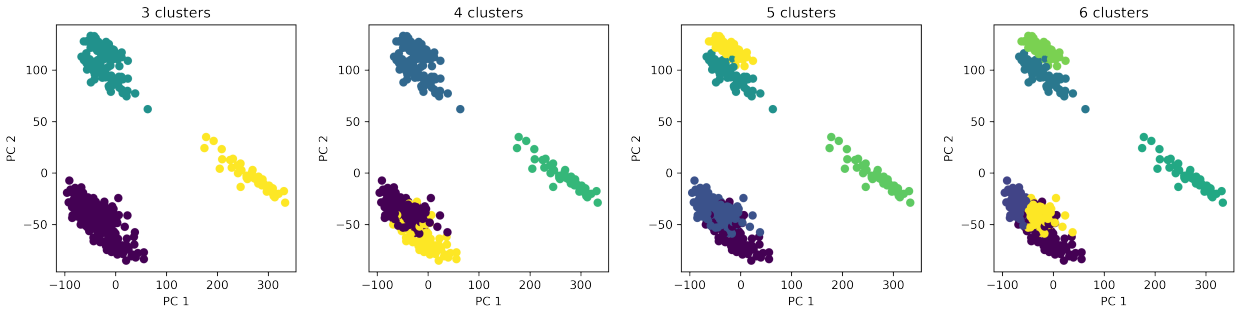

Figure 10: Visualization of clusters over the first two PC's (model used 50 PC's)