# Analysis of Various Clustering Algorithms

sduri2, kasivis2, jtrichm2, bommine2

## Introduction

When dealing with unstructured data sets, we often seek to find patterns of similarities within the dataset. While these similarities may not always be significant or meaningful, such patterns may sometimes reveal common groupings which are in fact significant; for example, we might notice that certain gene sequences correspond to a high chance of being diagnosed with certain medical conditions.

There are various methods by which we can try to find these patterns in unstructured data. Clustering is one common approach, where we seek simply to group together similar instances of data, for various definitions of “similar”. As we vary what we mean by “similar” – for example, we could define cluster similarity as the Euclidean or squared Euclidean distance between clusters’ center points – we are left with various algorithms and methods to cluster or group data points with each other. There are many algorithms for clustering including partitional methods such as k-means, minibatch k-means and Gaussian mixture models; and hierarchical methods such as agglomerative clustering, and so on. Each algorithm performs better or worse on certain types of data sets, and with certain types of data (for example, low vs. higher dimensional data).

As we become more capable of collecting large data sets of gene sequences and other biological data, the task of finding common groupings becomes harder and harder. One of the goals of research is identifying various new means of defining “similarity” as well as identifying the types of data sets a particular algorithm performs well (or poorly on), and vice-versa.

Our project seeks to compare various algorithms for DNA clustering on micro-arrays. We will compare the performances of the algorithms listed below on the same datasets and seek to determine the implications of these algorithms' performances.

The algorithms we compared were the following:

- K-means
- Ward Agglomerative Clustering
- Average Agglomerative Clustering
- Complete Agglomerative Clustering
- Spectral Clustering
- HDBScan

# Methods

We used two data sets for our evaluations: "The human whole miRNOme project version 1 (blood)" dataset, obtained here, and the "Mice Protein Expression Data Set" obtained from here. The former is a dataset of the blood expression profiles of 863 miRNA for 454 individuals; it originates from a study which sought to analyze this data, comprised of individuals diagnosed with different human diseases to test for disease-specific alterations. During preprocessing, we removed data columns which displayed low variance in order to increase the relevance of the data in the dataset for our clustering algorithms. The latter is a dataset of 77 proteins measured in the cerebral cortex of 8 classes of Down syndrome mice. Overall, we used these datasets as common datasets with which to compare clustering outputs between various algorithms. For each of the algorithms listed above we ran the clustering methods on the data sets and compared the outputs, both visually and quantitatively. Quantitatively, we measured the Silhouette and Calinski-Harabasz scores (Silhouette Scores being the mean Silhouette coefficient over all of the samples) as well as the runtime performance of these algorithms as measured by runtime. We used the first two as a measure of the effectiveness of each algorithm/method, and the latter as a measure of the cost of each algorithm.

# Results

## Mice Protein Data

| Algorithm/Method | Silhouette Score | Calinski-Harabasz | Runtime (sec.) |
|---|---|---|---|
| K-means | 0.1990 | 248.6167 | 0.1378 s |
| Ward Agglomerative Clustering | 0.1321 | 211.7696 | 0.0594 s |
| Average Agglomerative Clustering | 0.1638 | 102.9129 | 0.0556 s |
| Complete Agglomerative Clustering | 0.1624 | 118.5988 | 0.0553 s |
| Spectral Clustering | 0.1306 | 123.6942 | 0.1816 s |

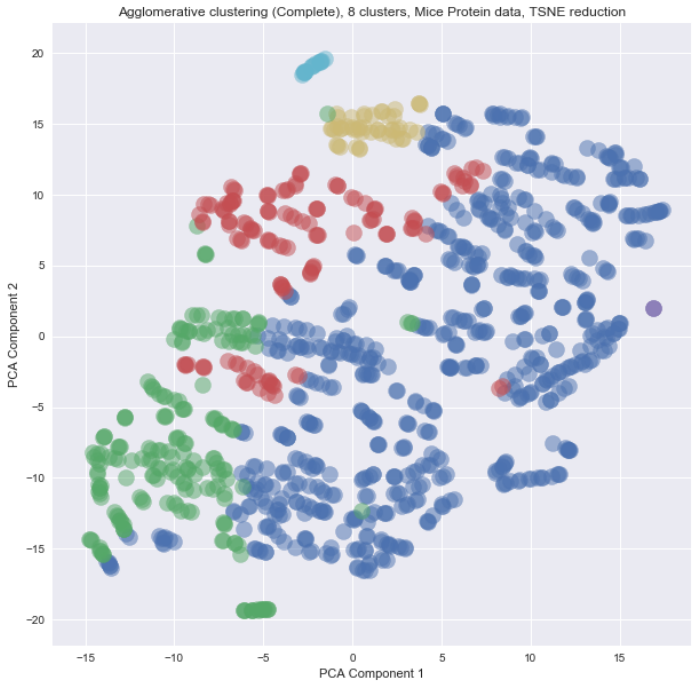
The Mice Protein data is a dataset of 77 proteins measured in the cerebral cortex of 8 classes of Down syndrome mice. The eight classes of mice come about based on varying features such as genotype, behavior and treatment applied. We see the results of applying the clustering algorithms to this dataset above. For each algorithm, the Silhouette Score and Calinski-Harabasz Scores and the average runtime in seconds over 100 repetitions of each algorithm were calculated. Silhouette Scores allow us to determine cluster quality, and measure how similar a data point is to its own cluster versus other clusters (comparing cohesion versus separation); values near 0 indicate the clusters overlapped each other, with values closer to 1 representing better scores. We see here that none of the algorithms had particularly high Silhouette scores, which suggests that the dataset did not lend itself well to being clustered. We worked with 8 clusters for this dataset as we know that there were 8 different classes of mice in this dataset. The fact that the Silhouette Scores were low for all of these algorithms suggests that this clustering configuration is not optimal, and that in the dataset the actual number of clusters represented by the data may in fact be lower or higher – the

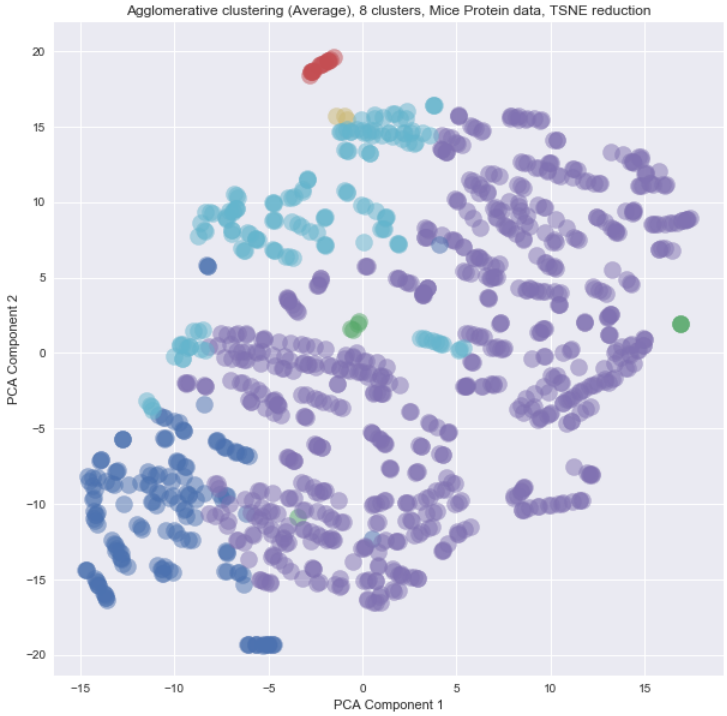
eight classes of mice may not actually be separable in the data they produce into eight separate classes.

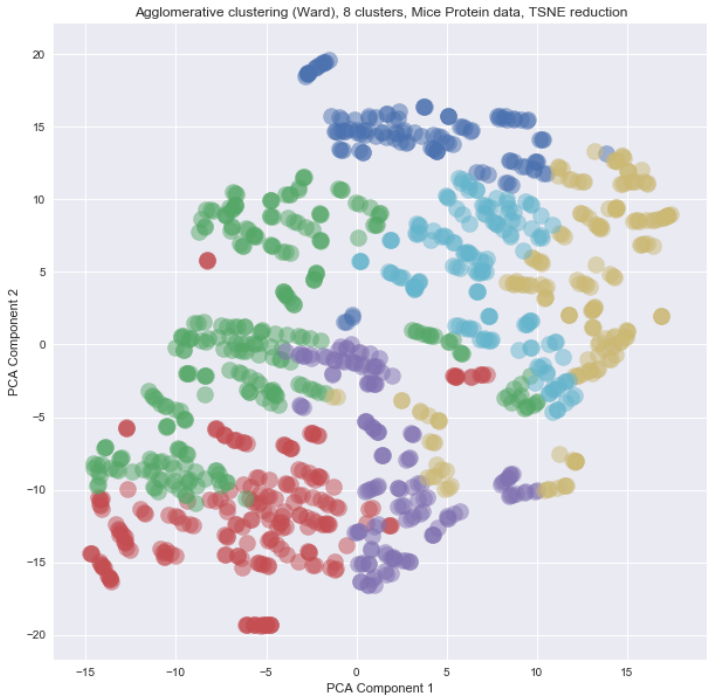
We see that K-means performs the best on this dataset, with the highest Silhouette Score of the lot. Calinski-Harabasz scores represent the ratio between within-cluster and between-cluster dispersions, with higher values being better. Interestingly, we see that Ward Agglomerative Clustering and K-Means have similar Calinski-Harabasz scores, better than the other algorithms, but the former runs about 2.5 times faster than the latter. Between the agglomerative algorithms, we see that Ward clustering performs better than the others in terms of the Calinski-Harabasz scores, with similar runtimes for all of these agglomerative algorithms. Overall, however, all of the algorithms performed relatively poorly on the Mice Protein dataset, with low Silhouette scores.

We see visual representations of these algorithms' clusters in the plots below. Note how the Average Agglomerative Clustering, has poor clustering overall.

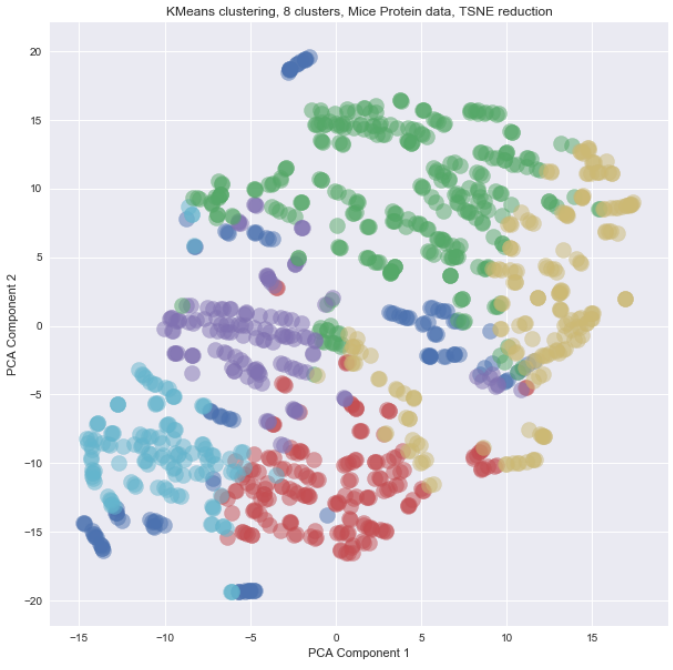
# Agglomerative Clustering Methods



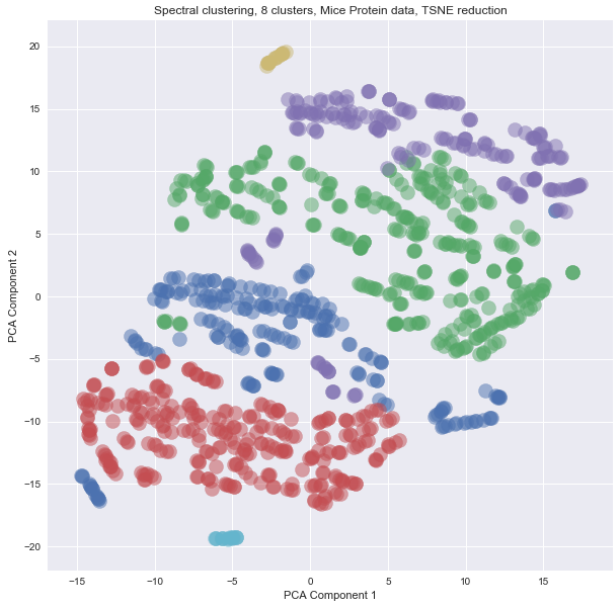




Agglomerative clustering (Ward), 8 clusters, Mice Protein data, TSNE reduction
PCA Component 2
PCA Component 1
20
15
10
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
15

## K-Means


KMeans clustering, 8 clusters, Mice Protein data, TSNE reduction
PCA Component 2
PCA Component 1
20
15
10
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
15

Spectral Clustering



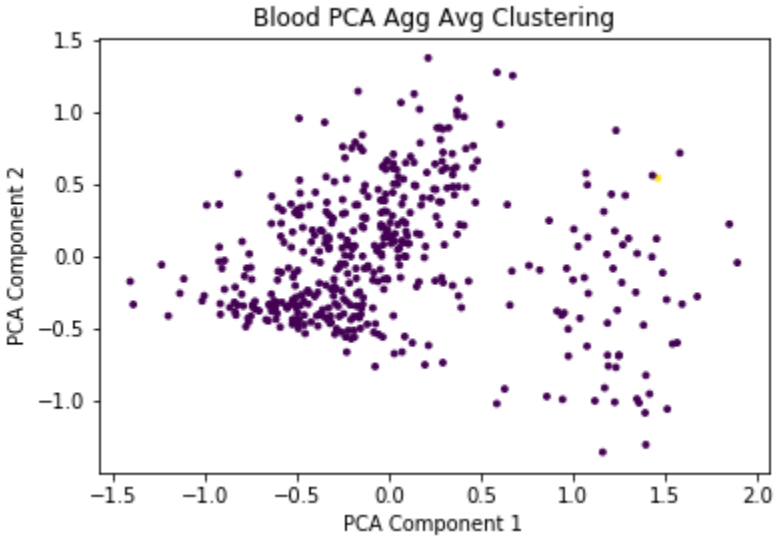

# Blood Dataset

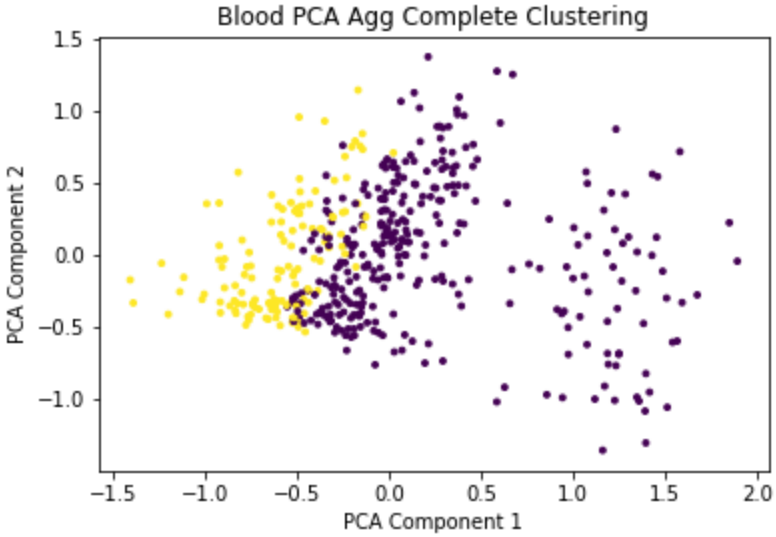
| Algorithm/Method | Silhouette Score | Calinski-Harabasz | Runtime (sec.) |
|---|---|---|---|
| K-means | 0.2605 | 174.7596 | 0.0541 s |
| Ward Agglomerative Clustering | 0.2130 | 140.2095 | 0.0090 s |
| Average Agglomerative Clustering | 0.4554 | 5.5234 | 0.0060 s |
| Complete Agglomerative Clustering | 0.1931 | 119.7707 | 0.0050 s |
| Spectral Clustering | 0.5517 | 383.5449 | 0.0511 s |
| HDBSCAN | 0.1369 | 21.5085 | 0.0160 s |

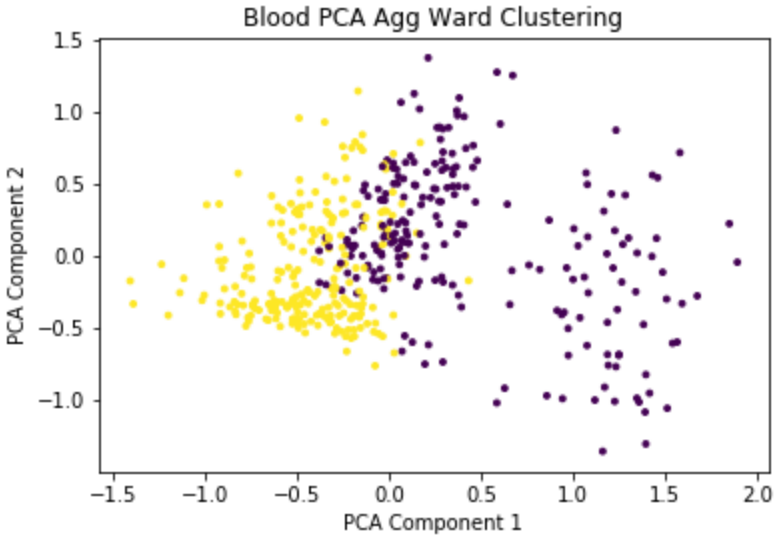
With the “The human whole miRNOme project version 1 (blood)” dataset (referred to simply as the Blood dataset henceforth), we see a different pattern emerge. Here the

Spectral Clustering algorithm performs the best, with higher Silhouette and Calinski-Harabasz scores than any other algorithm (significantly higher than all other algorithms except the Average Agglomerative Clustering, for the Silhouette Score). The Silhouette Score of 0.5517 suggests that the clusters generated were of relatively higher quality than we saw with the Mice Protein data, and thus the cluster configuration (the number of clusters we attempted to split the dataset into) was relatively appropriate. With the exception of Average Agglomerative Clustering, none of the other algorithms performed particularly strongly, with HDBSCAN performing among the worst between all of the algorithms we compared on this dataset. The fact that Average Agglomerative Clustering performed relatively well on the Silhouette Score but poorly on the Calinski-Harabasz score suggests that, although the clusters were relatively cohesive and separated, the clusters' data was not distributed in an optimal way for Calinski-Harabasz (which prefers spherical clusters which are compact in their middles).

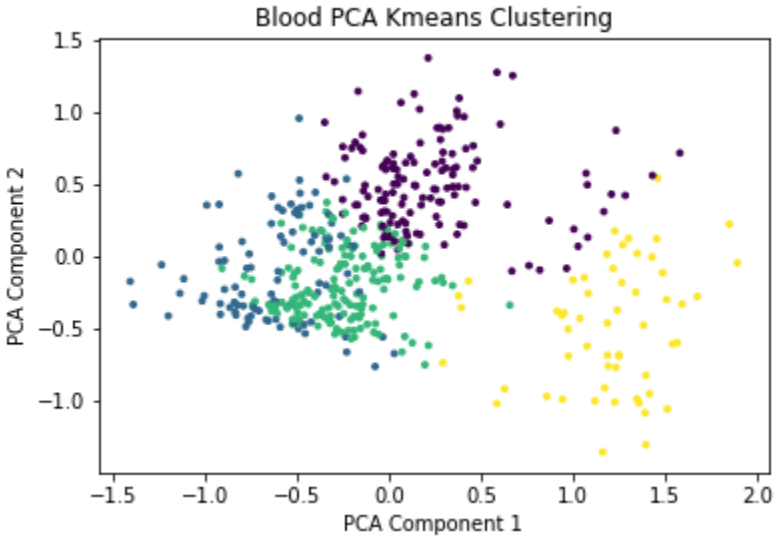
When we compare runtimes against the performance we achieve with the various algorithms, we see a few general points emerge. Although the three agglomerative algorithms run approximately five to ten times faster than Spectral Clustering, for example, the output they produce on this dataset is not worth the tradeoff. The additional time it takes for Spectral Clustering to run (as small as it may be on *this* particular dataset) is worth the performance gains as measured by the Silhouette and Calinski-Harabasz scores. Additionally, we see that Average Agglomerative Clustering is the best of the three agglomerative algorithms based on the Silhouette Scores, with relatively equivalent runtime performance between the three algorithms.
We see visual representations of these algorithms' clusters in the plots below.


Blood PCA Agg Avg Clustering
PCA Component 2
PCA Component 1
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

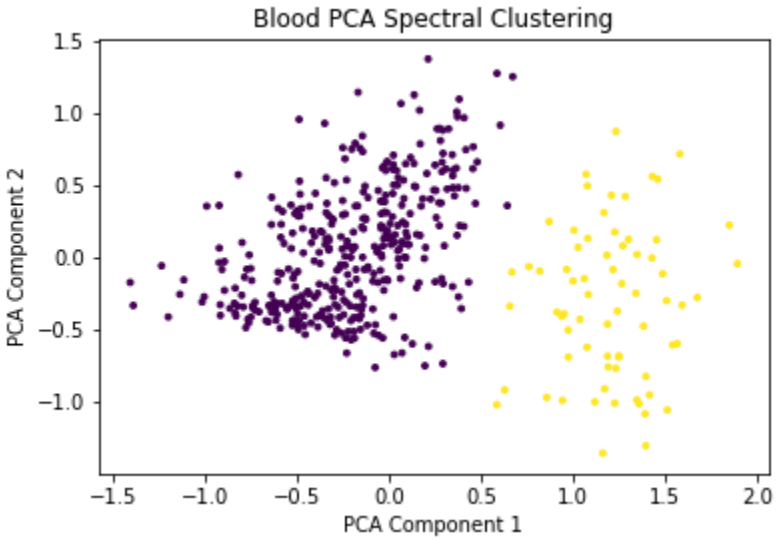

Blood PCA Agg Complete Clustering
PCA Component 2
PCA Component 1
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Blood PCA Agg Ward Clustering
PCA Component 2
PCA Component 1
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

## K-Means




## Spectral Clustering

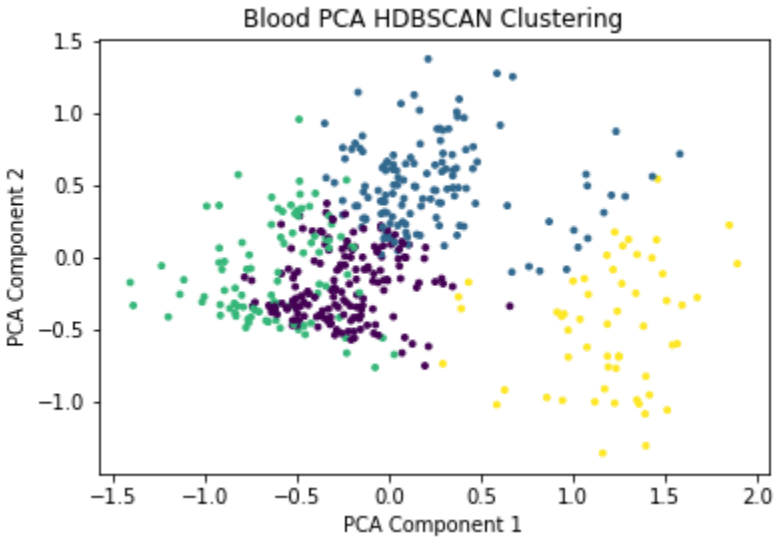



| Cluster 0 | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 |
|---|---|---|---|---|
| **acure myocard infarction** | 0.750000 | NaN | 0.250000 | NaN |
| **control** | 0.185714 | 0.171429 | 0.471429 | 0.171429 |

| | | | | |
|---|---|---|---|---|
| **copd** | 0.375000 | 0.041667 | 0.458333 | 0.125000 |
| **lung cancer** | 0.125000 | NaN | 0.406250 | 0.437500 |
| **melanoma** | 0.428571 | 0.057143 | 0.142857 | 0.371429 |
| **multiple sclerosis** | 0.434783 | NaN | 0.173913 | 0.391304 |
| **other pancreatic tumors and diseases** | 0.375000 | NaN | 0.270833 | 0.354167 |
| **ovarian cancer** | 0.133333 | NaN | 0.600000 | 0.266667 |
| **pancreatic cancer ductal** | 0.377778 | NaN | 0.311111 | 0.311111 |
| **pankreatitis** | 0.421053 | NaN | 0.289474 | 0.289474 |
| **periodontitis** | 0.055556 | 0.111111 | 0.833333 | NaN |
| **prostate cancer** | 0.217391 | 0.130435 | 0.260870 | 0.391304 |
| **sarcoidosis** | 0.044444 | 0.777778 | 0.155556 | 0.022222 |
| **tumor of stomach** | 0.384615 | NaN | 0.615385 | NaN |
| **wilms tumor** | NaN | NaN | 0.600000 | 0.400000 |

We see in the above table how the various diseases expressed in the dataset are distributed among the four clusters used in the Spectral Clustering. Note, for example, how 75% of those acure myocard infarction data points appear in Cluster 1; 78% of sarcoidosis points in Cluster 2 and 83% of periodontitis appear in Cluster 3.

## HDBSCAN




# Conclusion

We set out in our project to compare various clustering algorithms. We wanted to see how these algorithms performed on different datasets, with different intrinsic properties and data distributions. We used the “The human whole miRNOme project version 1 (blood)” and the “Mice Protein Expression Data Set” datasets to motivate our comparisons. When we compared algorithms’ performances on these datasets we used the Silhouette Score (the mean Silhouette coefficient for all of the samples) – which allow us to determine cluster quality, and measure how similar a data point is to its own cluster versus other clusters. We also used the Calinski-Harabasz score, which serves as a heuristic device allowing us to compare clustering algorithms. For the Calinski-Harabasz score we seek values as high as we can achieve. Finally, in comparing performances we used runtime as a basis of measuring the cost of achieving the varying cluster quality levels that the algorithms produced.

We found that, unsurprisingly, the most important factor of an algorithm’s performance is the dataset on which it is run. Algorithms which performed well for the Mice Protein dataset performed poorly on the Blood dataset, and vice-versa. We see that when we

use clustering algorithms there aren't necessarily any "go-to" algorithms. When determining what algorithm to use for a particular clustering task on a dataset it is clearly important to simply run all possible or plausible algorithms on the dataset and manually compare the results of their performances – there is no substitute to experimentation.