ISYE 6740 – Spring 2022
Final Report

Team Member Names: Michael Guo (903544000)

Project Title: Using Clustering to Identify Causes of Quality Metric Movement at a Biotechnology Company

Problem Statement

The Quality Group at a biotechnology company has noticed the Quality Metric 1 and Quality Metric 2 metrics for one of their products have undergone periods of increasing and decreasing over the past few years. They have also noticed a few sudden changes in the quality metric values in the same period. Traditional investigation methods (individually inspecting the characteristics of batches) have failed to reveal the cause of the periodicity due to the sheer number of variables (machines and reagent lots) used to run each batch. Therefore, I was asked to develop a repeatable method for investigating the cause of the rising and falling nature of these two metrics.

Establishing a data driven method for investigating patterns within the two quality metrics will allow the Quality Group to identify potential problems within the process as well as identify which components of the process can be changed for a more efficient/consistent process. Additionally, if this project successfully helps identify the root causes of the periodicity, it should be relatively simple to apply the method to other metrics across the enterprise. The factors examined in this project can be linked to several other metrics for the product.

This report focuses on determining the optimal clustering algorithm to use in the method, examining characteristics of the two datasets, and determining the design of the investigation method.

Data Source

Two datasets were obtained from a Snowflake database at the company, one for Quality Metric 1 and another for Quality Metric 2. The datasets each contain one quality metric as well as features which represent the various machines and reagent lots used in the process.

The first dataset records Quality Metric 1 for 3010 datapoints in chronological order, associated with 2789 unique batches. We have more datapoints than unique batches because each batch can be associated with multiple configurations of factors. The data have 15 feature columns each of which contain multiple factors. Across all the feature vectors there are a total of 97 different factors.

The second dataset records Quality Metric 2 for 14624 data points in chronological order, associated with 3727 unique batches. There are far more data points/unique batch, because this data set covers multiple configurations of the process (hence more factors) and because this

dataset contains many more steps of the process. The data have 34 feature columns, and a total of 278 factors across the columns. Measurements of Quality Metric 2 occur after Quality Metric 1 in the process so the 97 categorical features in the first data set are also be repeated in this dataset.

Methodology

Since the goal is to develop a technique to investigate patterns in the datasets and there are no objective class labels for when the quality metrics are changing, clustering will be used. The intention is to provide the general characteristics of each cluster to the Quality Group to help them identify the machines and reagent lots related to the rising/falling quality metrics. Several clustering algorithms are investigated to determine which method is best. Due to the categorical nature of the data features, two groups of clustering algorithms (Categorical and Numerical) are be investigated (**Figure 1**).

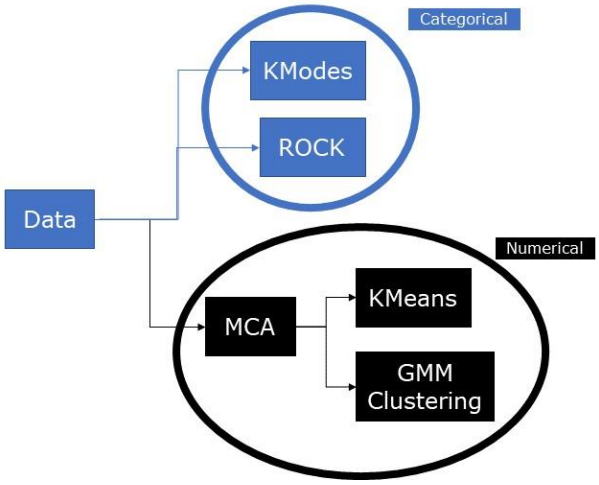


Figure 1: Clustering Models

In the categorical group, the KModes and ROCK clustering algorithms are used because they are better suited for clustering data with categorical features. The KModes algorithm was specifically chosen because it's an easy algorithm to explain to a non-technical audience like the Quality Group. The implementation of KModes in the **klaR** R package is used in this project [1].

The ROCK (RObust Clustering using linKs) algorithm was chosen as a candidate clustering method, because it integrates global information by assigning neighbors to each datapoint based on similarity and then clustering based on the number of common neighbors between points [2]. It should be noted that ROCK is a hierarchical clustering algorithm (as opposed to the centroid based KModes) which means that each data point starts off as its own cluster and as we advance through the algorithm, we merge these clusters together based on some similarity metric. The implementation of ROCK in the **cba** R package was used in this project [3].

For the numerical group, the data first have their dimensionality reduced to 2 dimensions via MCA (Multiple Correspondence Analysis), which allows us to perform dimensionality reduction on categorical features similarly to PCA. This also maps the features into numerical space allowing us to use numerical clustering methods. The initial plan in the proposal was to reduce the data to 1, 2, and 3 dimensions, but I only report dimensionality reduction to 2 dimensions in this report for the sake of brevity. The 2D data was specifically chosen because are easy to visualize and understand. The implementation of MCA in the **MASS** R package was used in this project [4].

The mapped data will then be clustered using KMeans and GMM Clustering Algorithms which operate on numerical data. The KMeans Algorithm (like KModes) was chosen because it is intuitively easy to explain for non-technical audiences. The implementation of KMeans in **stats**

package is used in this project [5]. The Hartigan-Wong algorithm was used as opposed to the traditional Lloyd's algorithm. The Hartigan-Wong algorithm randomly assigns points to centroids, minimizes the sum of squares for clusters to their data points when assigning data points, and calculates centroids as the mean of their assigned points on initialization and updates centroids when a point is reassigned.

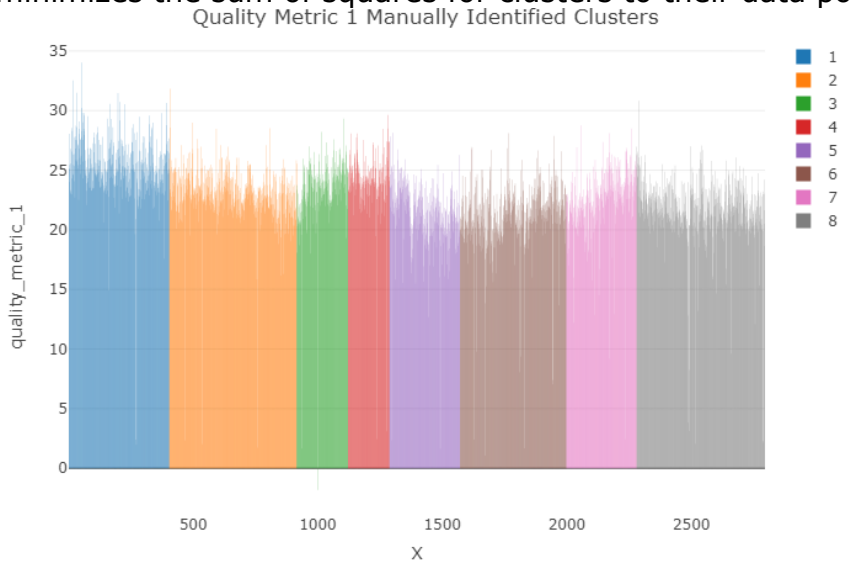


Figure 2: Manually identified clusters example

The GMM Clustering Algorithm was chosen because overlap between the borders of clusters is expected. We expect the borders between clusters to overlap because the Quality Group noticed periods of decreases and increases in quality metrics. This implies a gradual change in multiple features during the change period rather than simultaneous sudden changes in multiple features. The implementation of GMM Clustering in the **mclust** R package is used in this project [6].

The various clustering algorithms will be compared against data where the clusters have been manually identified by the Quality Group supervisor to determine their misclustering rate (**Figure 2**). The majority label within a manually identified cluster will be considered the "true" label of the cluster and any other labels in the manually identified data associated with that cluster will be considered misclustered.

$$miscluster\ rate_k = \frac{\#\ misclustered\ data\ points_k}{total\ \#\ data\ points_k}$$

Clustering algorithms with lower misclustering rates across all clusters will be considered more "accurate" and more appropriate for use than those with higher misclustering rates. It should be noted that the Quality Group supervisor's labels can be highly subjective. This means that the miscluster rate essentially measures how closely the clusters created by the algorithms match up with their expected clusters. This method may not select the most truly accurate clustering algorithm, but should be acceptable for our purposes since it selects the clustering algorithm that most closely matches the patterns the user (represented by the supervisor) is interested in.

Due to the random nature of the clustering algorithms investigated, when determining the miscluster rate and when determining the 'optimal' number of clusters for a given method, 5 iterations of the algorithm are be compared to generate average misclustering rates. This was not possible with ROCK Algorithm due to its computational complexity and hardware limitations. We are investigating the optimal number of clusters that minimize the misclustering rate to

determine if we should use the same number of clusters as identified by the user/Quality Group supervisor or we should vary the number of clusters investigated in the final method.

Recommendations for the optimal clustering algorithms are be presented in this paper. However, the final determination of what clustering algorithm(s) to ultimately use will be left to the Quality Group, which may prefer algorithms that perform better when identifying a specific type of cluster e.g. one with a descending trend. The final decision will likely be made based on a combination of algorithm miscluster rate and qualitative decisions.

Evaluation and Final Results

KModes Clustering

The miscluster rates for the KModes Clustering experiments on the Quality Metric 1 and Quality Metric 2 datasets can be seen in **Table 1** and **Table 2**, respectively.

Num Clusters	iteration-1	iteration-2	iteration-3	iteration-4	iteration-5	Average
10	0.461022	0.459424	0.579553	0.487859	0.448562	0.487284
11	0.459425	0.461981	0.492013	0.459425	0.480511	0.470671
12	0.477955	0.494569	0.463898	0.478275	0.482109	0.479361
13	0.450799	0.459105	0.463259	0.460064	0.487859	0.464217
14	0.490735	0.483067	0.440256	0.469329	0.455272	0.467732
15	0.455272	0.451757	0.432907	0.481150	0.448243	0.453866
16	0.442812	0.440575	0.441214	0.479872	0.446006	0.450096
17	0.476997	0.429712	0.454313	0.420128	0.462620	0.448754
18	0.446965	0.433866	0.425879	0.451757	0.436741	0.439042
19	0.451438	0.423642	0.438658	0.431310	0.401278	0.429265

Table 1: Quality Metric 1 KModes Miscluster Rate (best performing hyperparameter highlighted in green)

Num Clusters	iteration-1	iteration-2	iteration-3	iteration-4	iteration-5	average
12	0.384026	0.314825	0.235777	0.277216	0.341288	0.310626
13	0.369530	0.277352	0.267437	0.235435	0.342929	0.298537
14	0.262035	0.267847	0.282002	0.300123	0.242957	0.270993
15	0.262719	0.254923	0.291644	0.240358	0.26395	0.262719
16	0.250547	0.266890	0.274207	0.237828	0.242615	0.254417
17	0.235161	0.280156	0.258821	0.270514	0.290755	0.267082
18	0.244188	0.219776	0.227298	0.269489	0.283096	0.248769
19	0.250821	0.232905	0.292875	0.214374	0.207946	0.239784
20	0.151668	0.275643	0.197825	0.194338	0.245555	0.213006
21	0.253487	0.208219	0.203980	0.205689	0.205211	0.215317

Table 2: Quality Metric 2 KModes Miscluster Rate (best performing hyperparameter highlighted in green)

The algorithm performed poorly with the Quality Metric 1 dataset (best miscluster rate: 0.4293, 19 clusters), fairly well with the Quality Metric 2 dataset (best miscluster rate: 0.2130, 20 clusters). This may be caused by the Quality Metric dataset having less variables and less

informative variables compared to the second dataset. I believe the KModes clustering algorthm may perform better on Quality Metric 1 if some of the less variable features were removed, because their effect may be crowding out the effect of more variable features.

ROCK Clustering

The misclustering rates and number of clusters produced for the ROCK Clustering experiments on the Quality Metric 1 dataset and the number of clusters obtained are in **Table 3**. The theta is a hyperparameter which determines the threshold for how similar datapoints need to be to be considered neighbors.

theta	misclustering_rate	Num Clusters
0.1	0.576358	5
0.2	0.526518	9
0.3	0.531949	8
0.4	0.572843	5
0.5	0.623003	3
0.6	0.748882	2
0.7	0.756230	15
0.8	0.515655	251
0.9	0.942173	61

Table 3: Quality Metric 1 ROCK Performance (best performing hyperparameter highlighted in green)

ROCK Clustering performed very poorly on the Quality Metric 1 dataset (best miscluster rate: 0.5319, theta=0.3). Many of the misclusters counted are from clusters that were dropped for having negligibly small clusters. I believe this is a result of the homogeneous nature of the data. Many machines and lots of reagents are repeatedly used for the large stretches of time in the dataset. This level of similarity leads to many datapoints being considered neighbors resulting in a few big clusters and comparatively tiny satelite clusters. In other words, the Quality Metric 1 datapoints are too similar to use ROCK effectively.

The Quality Metric 2 dataset results were not reported, because attempts the ROCK algorithm failed to execute on a laptop representative of the computers used by the Quality Group. As mentioned in the initial paper by Guha et al [2], ROCK can be computationally very expensive with a worst case time complexity of

$$O\left(n^2 + nm_{max}m_{average} + n^2 log n\right), where\ n = num\ data\ points, and\ m = num\ neighbors$$

Given the poor misclustering rate in **Table 3**, and its unusable computation cost, ROCK can conclusively be eliminated as a possible clustering algorithm for investigating Quality Metric 1 and Quality Metric 2 data at the biotechnology company.

KMeans Clustering

The miscluster rates for the KMeans Clustering experiments on the Quality Metric 1 and Quality Metric 2 datasets can be seen in **Table 4** and **Table 5**.

Num Clusters	iteration-1	iteration-2	iteration-3	iteration-4	iteration-5	average
10	0.356869	0.334505	0.354952	0.356869	0.352077	0.351054
11	0.355272	0.350799	0.412141	0.353035	0.358147	0.365879
12	0.353994	0.338658	0.344089	0.368371	0.353994	0.351821
13	0.348882	0.350799	0.32492	0.333546	0.345048	0.340639
14	0.324281	0.330671	0.329393	0.350799	0.321406	0.33131
15	0.331629	0.350479	0.346645	0.333546	0.342812	0.341022
16	0.349201	0.343770	0.329393	0.337061	0.322364	0.336358
17	0.329712	0.319169	0.334824	0.331949	0.309904	0.325112
18	0.310224	0.339936	0.343770	0.322045	0.318850	0.326965
19	0.308946	0.319169	0.338339	0.324281	0.331629	0.324473

Table 4: Quality Metric 1 KMeans Miscluster Rate (best performing hyper parameter is highlighted in green)

Num Clusters	iteration-1	iteration-2	iteration-3	iteration-4	iteration-5	average
12	0.236119	0.288635	0.284190	0.255470	0.258753	0.264633
13	0.283575	0.259231	0.313799	0.283917	0.229075	0.273920
14	0.284122	0.220049	0.238375	0.288430	0.286447	0.263485
15	0.239948	0.241179	0.308260	0.233110	0.260325	0.256565
16	0.244051	0.237760	0.220938	0.200834	0.270925	0.234902
17	0.205142	0.264292	0.245008	0.228597	0.221417	0.232891
18	0.218135	0.212185	0.210681	0.220596	0.225109	0.217341
19	0.229964	0.232836	0.195501	0.259642	0.218545	0.227298
20	0.253146	0.212801	0.204458	0.262992	0.216972	0.230074
21	0.220801	0.213280	0.198578	0.192970	0.233315	0.211789

Table 5: Quality Metric 2 KMeans Miscluster Rate (best performing parameter is highlighted in green)

The algorithm performed relatively well with the Quality Metric 1 dataset (best miscluster rate: 0.3245, num clusters: 19), and had good performance with the Quality Metric 2 dataset (best miscluster rate: 0.2118, num clusters: 21). These miscluster rates are good results considering the large degree of cluster overlap observed in the dimensionally reduced data (**Appendix 1** and **Appendix 2**)

GMM Clustering

The miscluster rates for the GMM Clustering experiments on the Quality Metric 1 and Quality Metric 2 datasets can be seen in **Table 6** and **Table 7**.

Num Clusters	iteration-1	iteration-2	iteration-3	iteration-4	iteration-5	average
10	0.370288	0.368690	0.363898	0.39393	0.376997	0.374760
11	0.393930	0.394569	0.388179	0.38115	0.396805	0.390927
12	0.382748	0.377316	0.419808	0.376038	0.382748	0.387732
13	0.387859	0.385623	0.380192	0.386262	0.385942	0.385176

14	0.348243	0.360703	0.382109	0.386901	0.388818	0.373355
15	0.318850	0.375080	0.357508	0.398403	0.409904	0.371949
16	0.362300	0.327157	0.315016	0.316294	0.344728	0.333099
17	0.323323	0.315974	0.362620	0.372204	0.330032	0.340831
18	0.314377	0.327796	0.336102	0.307987	0.342173	0.325687
19	0.362939	0.373802	0.335144	0.306070	0.376997	0.350990

Table 6: Quality Metric 1 GMM Miscluster Rate (best performing hyperparameter is highlighted in green)

Num Clusters	iteration-1	iteration-2	iteration-3	iteration-4	iteration-5	average
12	0.315304	0.334108	0.290481	0.282002	0.279404	0.300260
13	0.282071	0.272839	0.304568	0.315098	0.315372	0.297990
14	0.281387	0.281387	0.316329	0.269283	0.316261	0.292929
15	0.310517	0.267642	0.280976	0.281113	0.280976	0.284245
16	0.269215	0.288225	0.290755	0.265112	0.300465	0.282754
17	0.272292	0.278720	0.258069	0.288567	0.278993	0.275328
18	0.255402	0.255539	0.282002	0.259437	0.271472	0.264770
19	0.301491	0.284874	0.251983	0.258206	0.283917	0.276094
20	0.244393	0.245487	0.253693	0.254240	0.252462	0.250055
21	0.240700	0.237623	0.244666	0.239880	0.257590	0.244092

Table 7: Quality Metric 2 GMM Miscluster Rate (best performing hyperparameter is highlighted in green)

The GMM algorithm had almost the same performance on Quality Metric 1 (best miscluster rate: 0.3257, num clusters: 18) and performed slightly worse on Quality Metric 2 (best miscluster rate: 0.2441, num clusters: 21) compared to the KMeans Algorithm. These results show us that GMM clustering does not help ameliorate the issue of overlapping "true" clusters. This makes sense since the dimension reduced data (**Appendix 1** and **Appendix 2**) do not appear to be normally distributed for the most part.

KMeans vs GMM Clustering

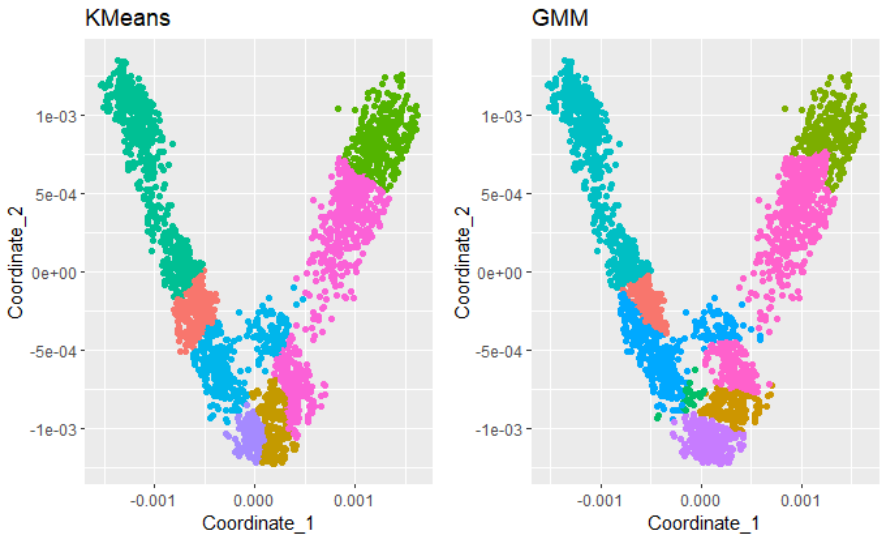


Figure 3: Quality Metric 1 Kmeans vs. GMM "True" Clusters

Since the misclustering rates of the KMeans and GMM Clustering methods are very similar, we present a visual comparison of the "true" clusters of each in **Figure 3** and **Figure 4**. The colors reflect the majority supervisor assigned label in the clusters created by the algorithms.

It can be observed that the general patterns of clusters of the data are observed in both the KMM and GMM plots. However, they differ at the borders, the

KMeans clusters have solid linear borders while the GMM plots tend to allow the clusters to "bleed" into each other more. Given the relative closeness of the two methods, I leave it to the Quality Group to choose the method they prefer.

It should also be noted that both methods produced less "true" clusters than the Quality Supervisor marked. For Quality Metric 1, KMeans generated 7 and GMM generated 8 compared to the manually selected 10. For Quality Metric 2, KMeans generated 9 and GMM generated 8 compared to the manually selected 12.

Number of Clusters to Use

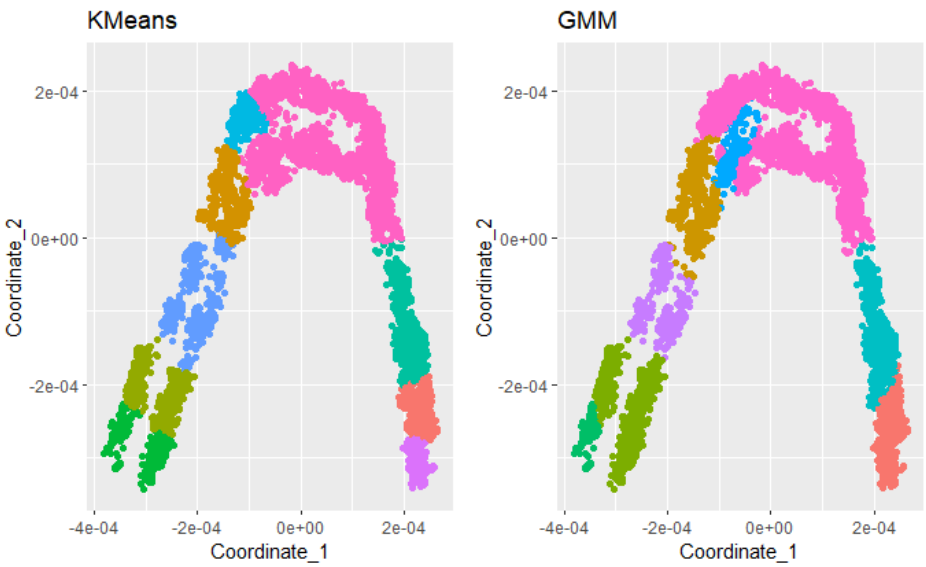


Figure 4: Quality Metric 2 KMeans vs. GMM "True" Clusters

Every clustering algorithm that uses number of clusters as a hyperparameter generally had better miscluster rates as the number of clusters increases. This combined with the fact that the clustering methods produced less "true" clusters than manually indicated show that the user may underestimate or overestimate the number and shape of clusters. Therefore, it may be prudent to cluster using multiple numbers of clusters and allow the user to choose the one that looks most appropriate.

Conclusion and Next Steps

Based on my investigations into the performance of the KModes, ROCK, KMeans, and GMM Clustering algorithms on the Quality Metric 1 and Quality Metric 2 datasets at a biotechnology company, KMeans and GMM Clustering following dimensionality reduction via MCA appear to be the most suitable algorithms for investigating patterns in their data. I believe a big contributor is MCA essentially highlighting the most variable factors while deprioritizing less variable ones. Additionally, the user should be presented with multiple possible clusters rather than one in the final method. Final determination of the exact algorithm to use will be decided after sharing the results of this project with the Quality Group.

Right now, my recommendation for a method to investigate quality metric data like Quality Metric 1 and Quality Metric 2 is as follows:

1. Obtain the Quality Metric Dataset
2. Perform Dimensionality Reduction via MCA to reduce to 2 or 3 dimensions.
3. Perform KMeans/GMM Clustering (exact method to be determined) on the reduced data for a range of k's, where k is the number of clusters.

4. Visualize each of the clustering results and show the user the plots.
5. Have the user pick the most appropriate plot.
6. Based on the k selected, generate summaries of the most variable pre-MCA factors for each cluster for future investigation.

The next steps are to:

1. Communicate the results of this report to the Quality Group to get their feedback on the proposed method.
2. Create a generalized prototype notebook/webapp which allows users to perform the recommended method without a lot of coding in R.
3. Work with the Quality Group to use the generalized notebook/webapp to investigate Quality Metric 1 and Quality Metric 2 as test cases.

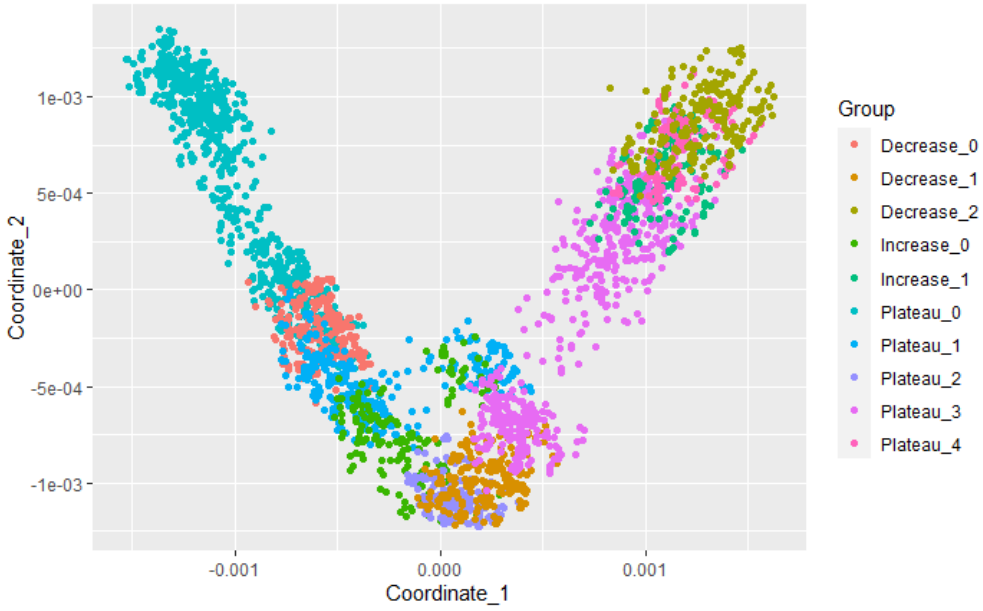
A mockup of a User Interface for selecting the appropriate k (steps 4 and 5) created using R Shiny can be seen in **Appendix C**. The exact format of the cluster summaries is still being determined.

References

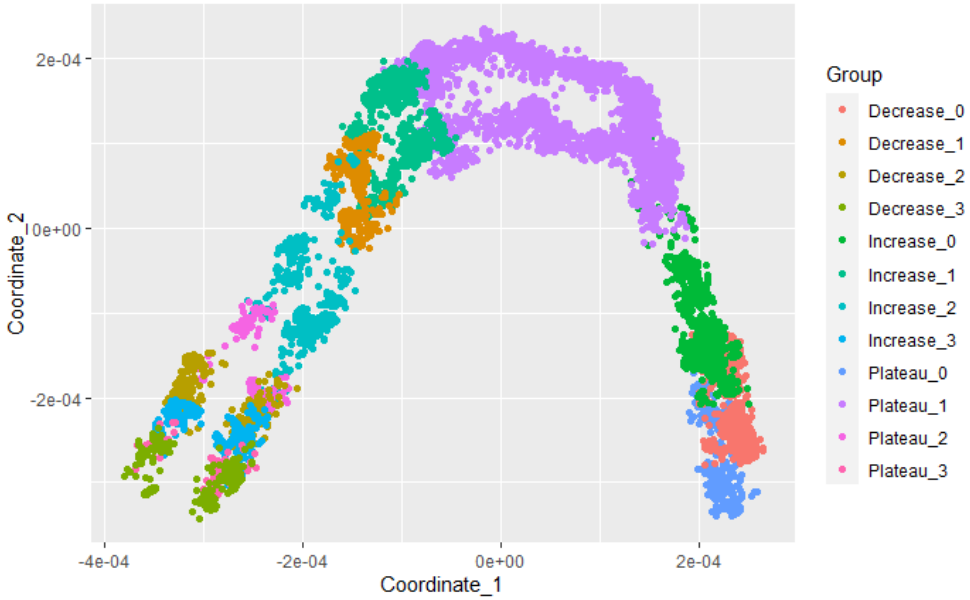
[1] https://cran.r-project.org/web/packages/klaR/index.html
[2] http://theory.stanford.edu/~sudipto/mypapers/categorical.pdf
[3] https://cran.r-project.org/web/packages/cba/cba.pdf
[4] https://cran.r-project.org/web/packages/MASS/index.html
[5] https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/kmeans
[6] https://cran.r-project.org/web/packages/mclust/index.html

Appendices

Appendix 1: Manually selected Quality Metric 1 clusters on 2D MCA Data



Appendix 2: Manually selected Quality Metric 2 clusters on 2D MCA Data



Appendix 3: User Interface Example