Part 1: Clustering of Gene Data

Introduction

The objective of my analyses was to train an unsupervised learning model for clustering. For the analyses presented here, data of the gene expression levels of 77 proteins/protein modifications that produced detectable signals in the nuclear fraction of the cortex was used.

There are eight classes of mice in the data classified according to genotype, behavior, and treatment. For genotype, mice are either control or trisomic. For behaviour, mice are either stimulated to learn or not stimulated to learn. For treatment, mice are either injected with saline or injected with memantine. 15 measurements were taken per protein for 72 mice in the experiment for a total of 1090 measurements per protein. The experiment was conducted to assess the effect of the drug memantine in recovering the ability to learn in trisomic mice.

Methods

Setup and Data Preparation

To setup, I loaded the cluster package to be used. I read the dataset from a csv file into R using the read.csv function specifying that there are headers in the data file, and to return the character "?" for missing values in the file. Since there is a large amount of missing values in the data, I used the na.omit function to create a dataset d with only complete information – without observations with missing values. I created a subset d.data with only the gene expression variables (V2-78) and a subset d.labels with only the class variable.

Using the apply function, I calculated the mean and variance of the gene expression variables for each observation. The scale of the variables looked very similar, so I decided not to scale the data in my subsequent analyses.

Partitioning by Medoids Clustering

First, I ran the Partitioning by Medoids (PAM) algorithm for different number of clusters to determine the optimal number of clusters that maximizes the silhouette width. To compare the silhouette width of models with 2 to 10 clusters, I created a function that ran the pam function for k = i+1, where i = 1 to 9, and used the silinfo$avg.width argument to extract the average silhouette widths. Then, I plotted the silhouette width for different numbers of clusters and joined the points using the plot and lines functions.

Since larger silhouette widths indicate better clustering, the optimal number of clusters was determined as the one with the highest silhouette width. As the optimal number of clusters

for PAM was determined to be 3, I used the pam function and the k = 3 argument to apply PAM to partition the gene data into 3 clusters around medoids.

Then, I printed the cluster size, cluster average silhouette widths, and the average silhouette width from the pam function. I used the silhouette function to compute silhouette information for the PAM clustering, and the plot function to generate a silhouette plot. Lastly, using the table function, I created tables of clusters by genotype, treatment, and behaviour to present the distribution of characteristics across clusters.

K-Means Clustering

I also performed K-means clustering to compare with PAM. First, I ran the K-means algorithm for different number of clusters to compare the silhouette width of models with 2 to 10 clusters. I used the silhouette function to compute silhouette information for the K-means clustering. Then, I plotted the silhouette width for different numbers of clusters and joined the points using the plot and lines functions.

Since larger silhouette widths indicate better clustering, the optimal number of clusters was determined as the one with the highest silhouette width. As the optimal number of clusters for K-means was determined to be 2, I used the kmeans function and the centers = 2 argument to perform K-means clustering to partition the gene data into 2 clusters around centroids.

I used the silhouette function to compute silhouette information for the PAM clustering, and the plot function to generate a silhouette plot. Then, I printed the cluster size, cluster average silhouette widths, and the average silhouette width using the summary function and the object created using the silhouette function. Using the table function, I created tables of clusters by genotype, treatment, and behaviour to present the distribution of characteristics across clusters.

Principal Component Analysis

To perform Principal Component Analysis (PCA) on the gene data, I used the prcomp function. Then, I printed the loadings of the 1st and 2nd principal components to investigate whether there are particular proteins that exhibit distinct expression patterns. Using the biplot function and scale = 0 argument, I generated a biplot to plot together the points and features based on the first 2 principal components.

Using the output from the prcomp function, I calculated and printed the proportion of variance explained by each principal component. I plotted the proportion of variance explained and the cumulative proportion of variance explained by each principal component using the plot function. Then, I printed the cumulative proportion of variance explained using the cumsum function. Lastly, using the plot and legend functions, I generated a plot of the 1st and 2nd principal components with colour-coded PAM clusters and a legend of the cluster ID.

Results

Partitioning by Medoids

The PAM algorithm was run for different number of clusters to determine the optimal number of clusters that maximizes the silhouette width. The plot of the silhouette width of PAM models with 2 to 10 clusters is presented in Figure 1 (Appendix B). The optimal number of clusters with the highest silhouette width was determined to be 3.

PAM was performed to partition the gene data into 3 clusters around medoids. The cluster sizes are 169, 220, and 163, respectively, and the cluster average silhouette widths are 0.19, 0.32, 0.28, respectively. The average silhouette width is 0.27. The silhouette plot is presented in Figure 2 (Appendix B). The distribution of clusters by genotype, treatment, and behaviour are shown in table 1 (Appendix A).

K-Means Clustering

The K-means algorithm was run for different number of clusters to determine the optimal number of clusters that maximizes the silhouette width. The plot of the silhouette width of K-means models with 2 to 10 clusters is presented in Figure 3 (Appendix B). The optimal number of clusters with the highest silhouette width was determined to be 2.

K-means clustering was performed to partition the gene data into 2 clusters around centroids. The cluster sizes are 265 and 287, respectively, and the cluster average silhouette widths are 0.21 and 0.34, respectively. The average silhouette width is 0.28. The silhouette plot is presented in Figure 4 (Appendix B). The distribution of clusters by genotype, treatment, and behaviour are shown in table 2 (Appendix A).

Principal Component Analysis

PCA was performed on the gene data. The biplot with the points and features based on the first 2 principal components is shown in Figure 5 (Appendix B). The plots of the proportion of variance explained and the cumulative proportion of variance explained by each principal component are shown in Figures 6 and 7 (Appendix B). Lastly, the plot of the 1st and 2nd principal components with colour-coded PAM clusters is shown in Figure 8 (Appendix B).

Discussion and Conclusions

Clustering of Gene Data

Since the average silhouette width of 0.28 for K-means clustering is the only slightly higher than the average silhouette width of 0.27 for PAM clustering, only the results of PAM will be discussed further since it is a more robust version of K-means.

After running the PAM algorithm for different number of clusters, the optimal number of clusters with the highest silhouette width was determined to be 3. The cluster average silhouette widths are 0.19, 0.32, 0.28, respectively. The average silhouette width for PAM clustering of the gene expression data with 3 clusters is 0.27. The low average silhouette width indicates that the clustering performance is not good and that clusters may not be distinct. The silhouette plot shows that some points have negative silhouette widths indicating that the points have been clustered incorrectly. This may help to explain the low average silhouette width.

Each cluster is more representative of a certain genotype, treatment, and behaviour. Cluster 1 is associated with the control genotype, saline treatment, and context-shock (stimulated to learn) behaviour. Cluster 2 is associated with the trisomy genotype, saline treatment, and context-shock (stimulated to learn) behaviour. Lastly, cluster 3 is associated with the trisomy genotype, memantine treatment, and shock-context (not stimulated to learn) behaviour.

The loadings of the 1st principal component for NR2A_N, ERK_N, pCAMKII_N, and Bcatenin_N are greater than 0.2 – otherwise, all other loadings are close to 0. Thus, the 1st principal component is largely influenced by these proteins/protein modifications. Similarly, the loadings of the 2nd principal component for NR2A_N, ERK_N, and pCAMKII_N are greater than 0.2 – otherwise, all other loadings are close to 0. As such, the 2nd principal component is largely influenced by these proteins/protein modifications.

The first 2 principal components explain about 73% of the data variation. The 1st principal component explains about 47% of the variation and the 2nd principal component explains about 26% of the variation. The plot of the 1st and 2nd principal components with colour-coded PAM clusters shows 3 distinct clusters with slight overlap. This shows that the NR2A_N, ERK_N, pCAMKII_N, and Bcatenin_N proteins/protein modifications which are largely explained in the first 2 principal components are almost sufficient to separate the clusters.

Lastly, the first 6 principal components explain about 92% of variation in the data. As such, this set of principal components explains a large amount of variation in the data and is a sufficient set of new features for representing the data.

Assumption of Independence

Although multiple measurements in the data came from the same mouse, we considered the measurements to be independent. Measurements from the same mouse are correlated and are more likely to be clustered. Our results may be biased since we ignored this correlation. As such, any distinct clusters observed may be a result of multiple measurements from the same mouse being considered independent.

To address the potential correlation, we could average the gene expression level measurements for each mouse so that each measurement analyzed for each protein is independent. Otherwise, we could perform multiple factor analysis (MFA) which is an extension of PCA for clustered quantitative data.

Part 2: Applying Unsupervised Learning Methods

Research Project Summary

I recently completed a project examining whether individuals with a diagnosis of influenza or pneumonia have an increased risk of an AMI diagnosis. We decided to investigate this research question because previous research has shown that acute respiratory infections may trigger AMI. To complete our analyses, we used data from the National Hospital Discharge Survey (NDHS), a national survey that collects inpatient discharge data from selected short-stay hospitals in the United States.

Since NHDS discharge records include up to 7 diagnoses, the exposure was defined as all-listed influenza or pneumonia diagnosis. As AMI is reordered to the first-listed diagnosis if present in the NHDS discharge record in any position, the outcome was defined as first-listed AMI diagnosis. The outcome and exposure were coded as binary variables with the categories yes and no. We fit a Poisson model with AMI diagnosis as the main exposure, influenza or pneumonia diagnosis as the outcome, and various demographic characteristics as covariates to answer the research question. We found that individuals who have a diagnosis of influenza or pneumonia are more likely to have a diagnosis of AMI in comparison to individuals without a diagnosis of influenza or pneumonia.

Applying Unsupervised Learning Methods

To build on the project described above, unsupervised learning methods such as K-means, hierarchical clustering, and PAM can be used to examine clustering of discharge data among individuals who have a diagnosis of AMI. When using NDHS data, an AMI diagnosis is always reordered to the first-listed diagnosis, so clustering based on diagnoses 2 to 7 can be examined. This may lead to the discovery of patterns/groups of secondary diagnoses among individuals who have a diagnosis of AMI. Since targets are not provided with unsupervised learning methods, this may allow us to identify conditions that co-occur with AMI which have not been studied before. In this way, unsupervised learning methods can be used to generate preliminary evidence and hypotheses for future studies.

As we used data from the NHDS for our previous project, we were limited by the data collected in the survey and could only adjust for demographic characteristics such as sex, age, race, and marital status. However, if the data to be analyzed contains information on many covariates, overfitting may occur if models are trained using all the variables. PCA may be useful in this situation to reduce the data to be analyzed. PCA can be used to identify a small subset of principal components that explain the largest amount of variation in the data. This set of principal components can then be used as predictors in supervised learning methods – for example, to fit a Poisson model as we did for our project.

Ivy Fong

Appendix A: Tables

Table 1. Cluster ID vs. genotype, treatment, and behaviour for PAM clustering

		Genotype		Treatment		Behaviour	
		Control	**Ts65Dn**	**Memantine**	**Saline**	**C/S**	**S/C**
Cluster ID	**1**	95	74	49	120	126	43
	2	101	119	80	140	118	102
	3	59	104	126	37	41	122

Table 2. Cluster ID vs. genotype, treatment, and behaviour for K-means clustering

		Genotype		Treatment		Behaviour	
		Control	**Ts65Dn**	**Memantine**	**Saline**	**C/S**	**S/C**
Cluster ID	**1**	121	144	153	112	118	147
	2	134	153	102	185	167	120

Appendix B: Figures

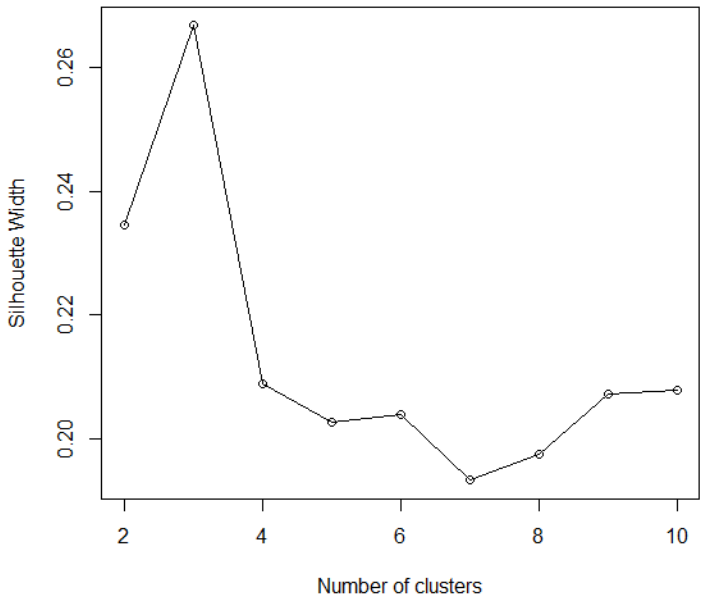

Figure 1. Silhouette widths vs. number of clusters for PAM clustering

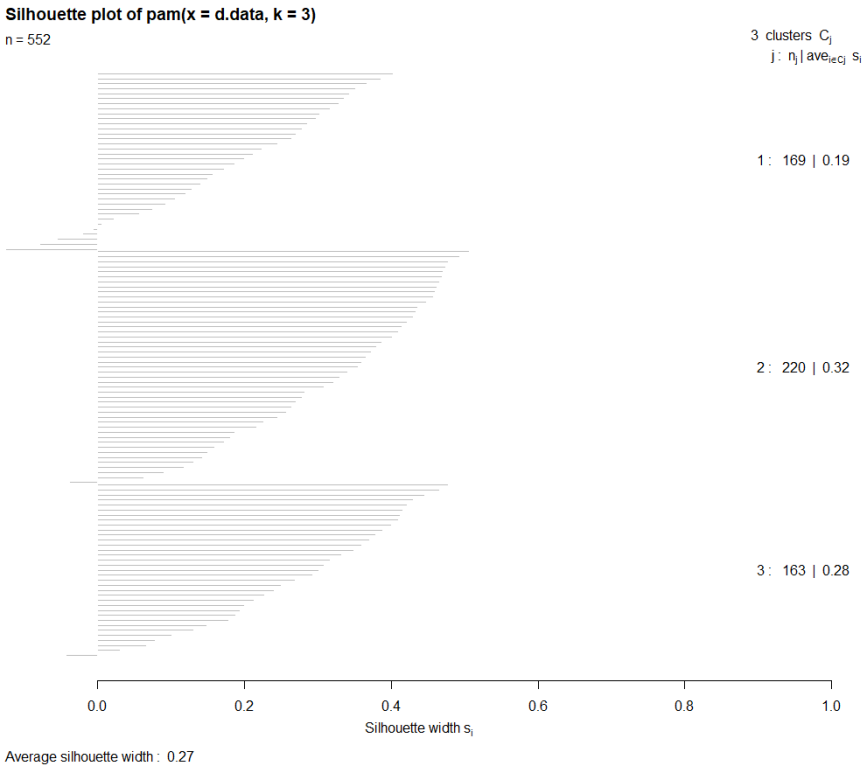

Figure 2. Silhouette plot for PAM clustering

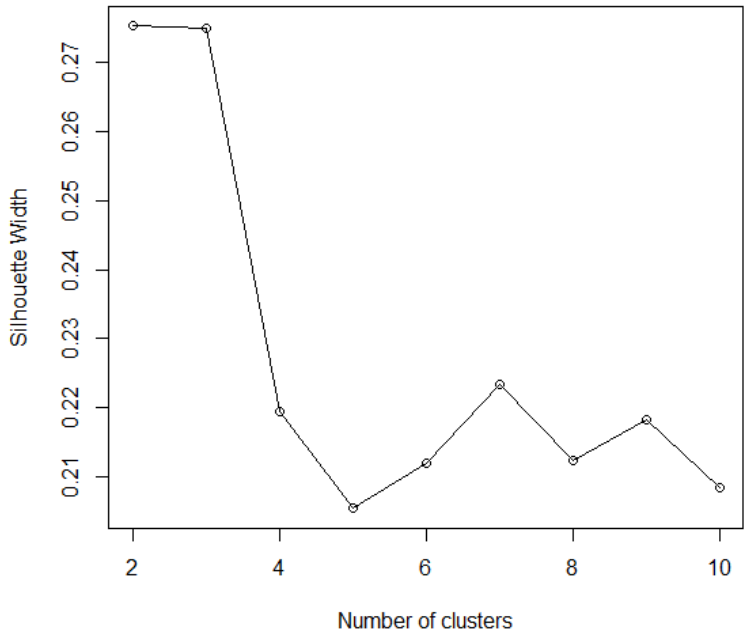

Figure 3. Silhouette widths vs. number of clusters for PAM clustering

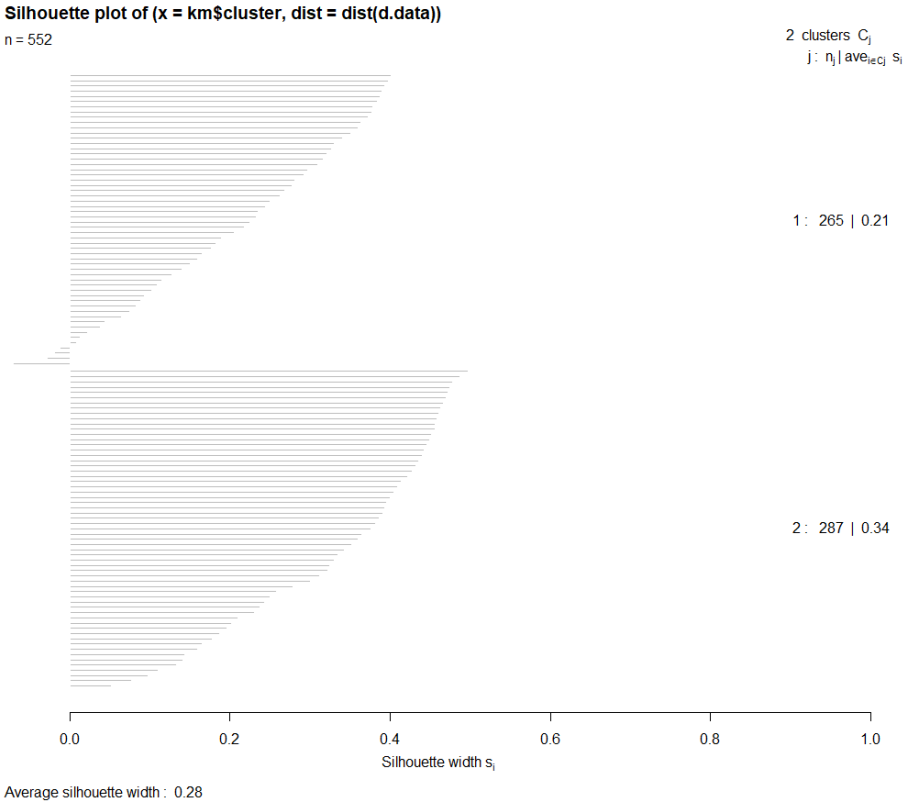

Figure 4. Silhouette plot for K-means clustering

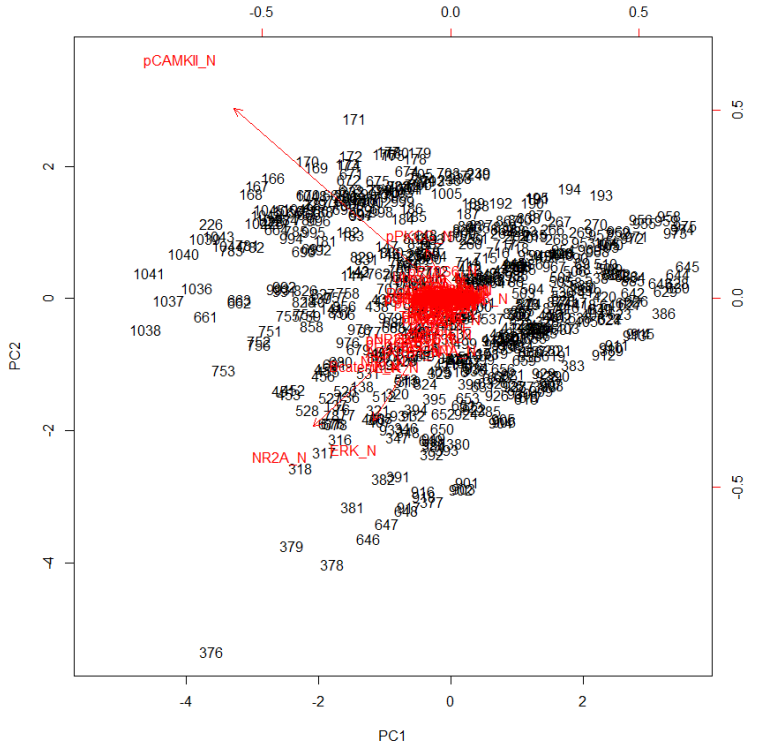

Figure 5. Biplot of the points and features based on the 1st and 2nd principal components

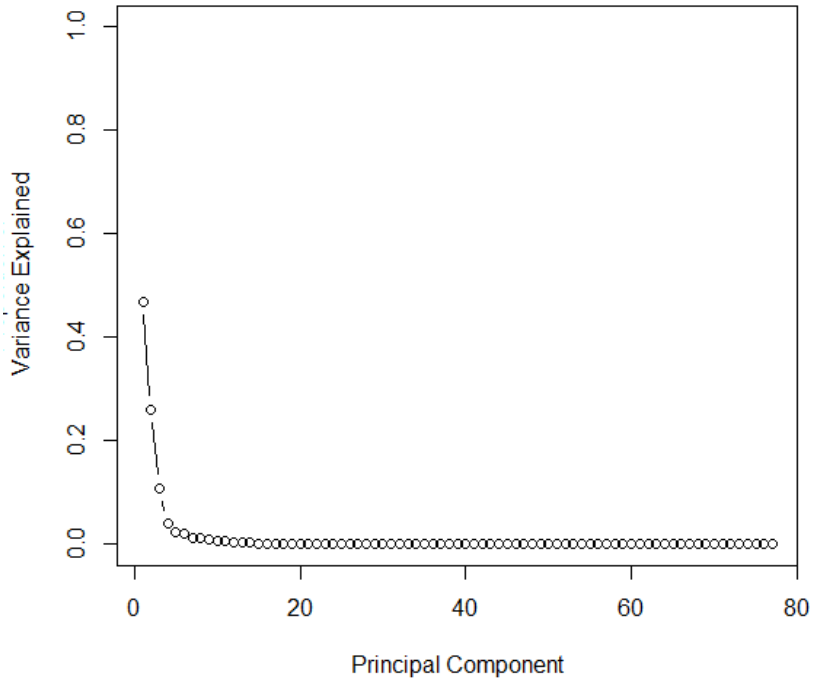

Figure 6. Proportion of variance explained by each principal component

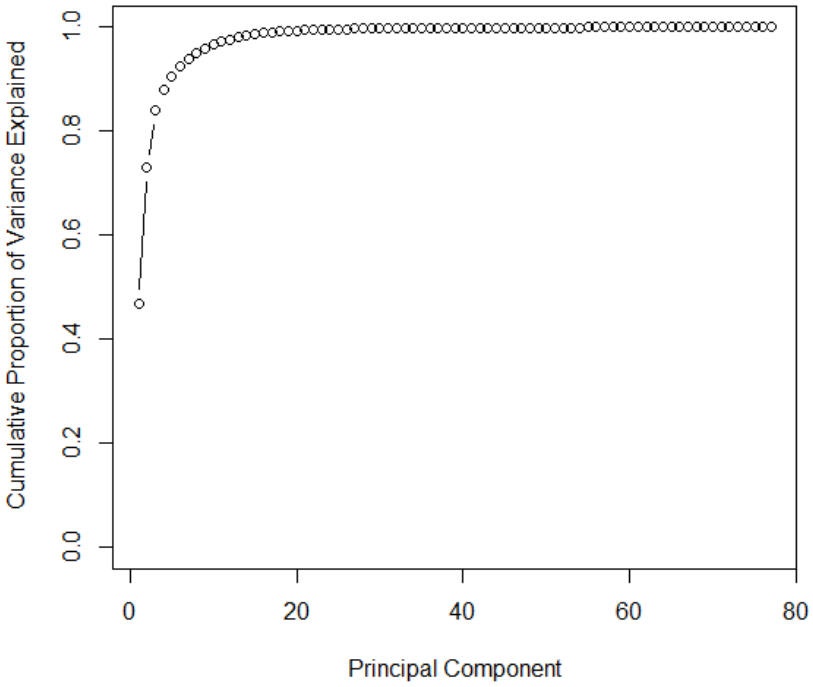

Figure 7. Cumulative proportion of variance explained by each principal component

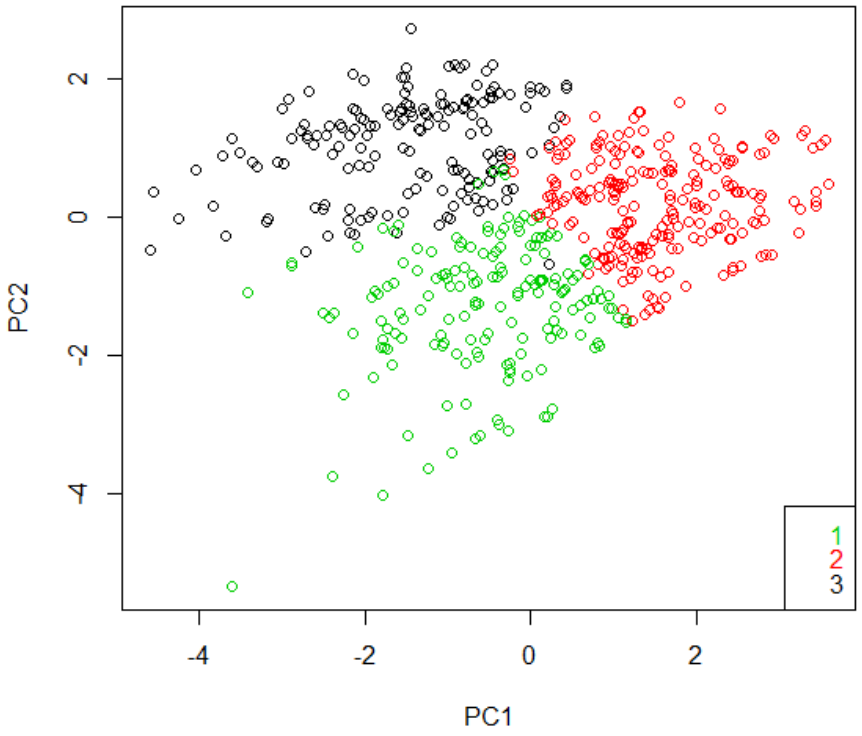

Figure 8. Plot of the 1st and 2nd principal components with colour-coded PAM clusters

Ivy Fong

Appendix C: R Code

Setup and Data Preparation

```
#load packages to be used
library(cluster)

#set seed
set.seed(123)

#load data and create dataset d without missing values
d <- read.csv("Data_Cortex_Nuclear.csv", header=T, na.strings="?") #read csv
data into R, specify variable names in header, return ? for missing values
d <- na.omit(d) #only keep observations with complete information
summary(d) #print summary of d dataset

#subset data
d.data <- d[,2:78] #create subset d.data gene expression variables
d.labels <- d[,82] #create subset d.labels with class variable
summary(d.data) #print summary of d.data dataset
summary(d.labels) #print summary of d.labels dataset

#calculate mean and variance of the gene expression variables for each observ
ation
apply(d.data, 2, mean) #print mean of gene variables
apply(d.data, 2, var) #print variance of gene variables
#scale of features looks similar - will not scale data
```

PAM Clustering

```
#run PAM algorithm for different number of clusters and compare the associate
d silhouette widths
sil_width <- c() #create sil_width vector

for(i in 1:9){
  pam_fit <- pam(d.data, k=i+1) #run PAM algorithm for 2-10 clusters
  sil_width[i] <- pam_fit$silinfo$avg.width #extract average silhouette width
from silinfo$avg.width to build vector
}

plot(2:10, sil_width, #plot sihouette width (higher is better)
     xlab = "Number of clusters",
     ylab = "Silhouette Width")
lines(2:10, sil_width) #highest silhouette width with 3 clusters

#apply PAM to partition gene data into 3 clusters around medoids - a more rob
ust version of K-means
pam <- pam(d.data, 3) #apply PAM to gene data with k = 3 clusters
pam$clusinfo #print PAM cluster information including cluster size = 169 220
163
```

```
pam$silinfo$clus.avg.widths #print cluster average silhouette widths = 0.1851
879 0.3212409 0.2780063
pam$silinfo$avg.width #print average silhouette width = 0.2668203

#create tables to present the distribution of outcomes across clusters
table(pam$cluster,d.labels) #create table of clusters by class
table(pam$cluster,d$Genotype) #create table of clusters by genotype
table(pam$cluster,d$Treatment) #create table of clusters by treatment
table(pam$cluster,d$Behavior) #create table of clusters by behaviour

#create silhouette plot
si <- silhouette(pam) #use silhouette function to compute silhouette informat
ion for PAM clustering
plot(si) #generate silhouette plot
```

K-Means Clustering

```
#run K-means algorithm for different number of clusters and compare the assoc
iated silhouette widths
km2 <- kmeans(d.data, centers=2, nstart=20)
si.km2 <- silhouette(km2$cluster, dist(d.data))
summary(si.km2) #k = 2, silhouette width = 0.2752

km3 <- kmeans(d.data, centers=3, nstart=20)
si.km3 <- silhouette(km3$cluster, dist(d.data))
summary(si.km3) #k = 3, silhouette width = 0.27490

km4 <- kmeans(d.data, centers=4, nstart=20)
si.km4 <- silhouette(km4$cluster, dist(d.data))
summary(si.km4) #k = 4, silhouette width = 0.21944

km5 <- kmeans(d.data, centers=5, nstart=20)
si.km5 <- silhouette(km5$cluster, dist(d.data))
summary(si.km5) #k = 5, silhouette width = 0.20543

km6 <- kmeans(d.data, centers=6, nstart=20)
si.km6 <- silhouette(km6$cluster, dist(d.data))
summary(si.km6) #k = 6, silhouette width = 0.21206

km7 <- kmeans(d.data, centers=7, nstart=20)
si.km7 <- silhouette(km7$cluster, dist(d.data))
summary(si.km7) #k = 7, silhouette width = 0.22344

km8 <- kmeans(d.data, centers=8, nstart=20)
si.km8 <- silhouette(km8$cluster, dist(d.data))
summary(si.km8) #k = 8, silhouette width = 0.2123

km9 <- kmeans(d.data, centers=9, nstart=20)
si.km9 <- silhouette(km9$cluster, dist(d.data))
```

```r
summary(si.km9) #k = 9, silhouette width = 0.21825

km10 <- kmeans(d.data, centers=10, nstart=20)
si.km10 <- silhouette(km10$cluster, dist(d.data))
summary(si.km10) #k = 10, silhouette width = 0.2085

#create sil_width2 vector with silhouette widths for K-means clustering with
2 to 10 clusters
sil_width2 <- c(0.2752, 0.27490, 0.21944, 0.20543, 0.21206, 0.22344, 0.2123,
0.21825, 0.2085)

#plot sihouette width (higher is better)
plot(2:10, sil_width2,
     xlab = "Number of clusters",
     ylab = "Silhouette Width")
lines(2:10, sil_width2)  #highest silhouette width with 2 clusters

#apply K-means to partition gene data into 2 clusters around centroids
km <- kmeans(d.data, centers=2, nstart=20) #perform K-means clustering 20 tim
es, create 2 clusters, pick the best solution
km$iter #print number of iterations needed to find solution = 1

#create tables to present the distribution of outcomes across clusters
table(km$cluster,d.labels) #create table of clusters by class
table(km$cluster,d$Genotype) #create table of clusters by genotype
table(km$cluster,d$Treatment) #create table of clusters by treatment
table(km$cluster,d$Behavior) #create table of clusters by behaviour
#no distinct clusters by genotype, behaviour, treatment, or class

#create silhouette plot
si2 <- silhouette(km$cluster, dist(d.data)) #compute silhouette information,
dist matrix is needed - dist outputs the distance between 1st and 2nd point a
nd so on
summary(si2) #print silhouette information summary
#cluster size = 265 287
#cluster average silhouette widths = 0.2094640 0.3359831
#average silhouette width = 0.2752
plot(si2) #generate silhouette plot
```

PCA

```r
#perform PCA
pr.out <- prcomp(d.data, scale=F) #perform PCA on gene data, don't scale the
data
pr.out$rotation[,1] #print pc1 loadings - except NR2A_N, ERK_N, pCAMKII_N, Bc
atenin_N, loadings > 0.2, all other pc1 loadings are close to 0
pr.out$rotation[,2] #print pc2 loadings - except NR2A_N, ERK_N, pCAMKII_N loa
dings > 0.2, all other pc2 loadings are close to 0
```

```
#create biplot
biplot(pr.out, scale=0) #generate biplot, plotting together the points and th
e features based on the first 2 pc's

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], lengt
h
## = arrow.len): zero-length arrow is of indeterminate angle and so skipped

#calculate pve = proportion of variance explained by each component
pr.var <- pr.out$sdev^2
pve <- pr.var/sum(pr.var)
pve #print proportion of variance explained by each component

#plot proportion of variance explained
plot(pve, xlab="Principal Component", ylab="Proportion of
     Variance Explained ", ylim=c(0,1) ,type="b")

#plot cumulative proportion of variance explained
plot(cumsum(pve), xlab="Principal Component", ylab="
     Cumulative Proportion of Variance Explained ", ylim=c(0,1) ,
     type="b")

#cumsum function gives cumulative sums of previous units - proportion of vari
ance explained
cumsum(pve) #print cumulative proportion of variance explained - first 6 pc's
explain about 92% of the data variation

#colour-code PAM clusters on plot of pc1 vs. pc2
plot(pr.out$x[,1:2], col=4-as.numeric(pam$cluster)) #generate plot of pc2 vs.
pc2 with colour-coded PAM clusters
legend("bottomright", legend=levels(as.factor(pam$cluster)), text.col=4-(1:3)
, y.intersp=0.8) #add legend with cluster ID
```