I0P16B: Applied Multivariate Statistical Analysis

Kristen Michelle Nader
Student Number: r0771801
Presented to: Professor Eddie Schrevens

January 24,2021

1 Introduction

This report aims to explore the chosen dataset using multivariate analysis techniques. I will begin by explaining the dataset to be used , then a description of my intention for this report. Finally, the plots and their interpretations will displayed. The last pages will contain the R code used to generate the plots and the analysis.

1.1 Description of the Data set, pre-processing

The data set used in the report is a micro array data set(GSE63061) from the Gene Expression Omnibus database that has been previously normalized and pre-preproccessed for quality control as a standard protocol in bioinformatics pipelines. The data consists of 2384 probes and 382 rows/observations. Using the series matrix provided by the GEO database(GSE63061_series_matrix), I mapped the probes to their respective genes for easy interpretation. As a word of caution, these variables, once read into R, should be removed from the environment after mapping probes to the genes. They are large files and will slow computation significantly. In addition, these gene names had to be cleaned to ensure the R read them properly. I will provided a small example of the variables included in the data set below before pre-processing. Each observation is marked as AD/MCI or CTL as individuals with Alzheimer's Disease, Mild Cognitive Impairment or Control Individuals.

Table 1: Variables before pre-processing of the raw normalized data. The dataset consists of 1 variable for class(AD/MCI/CTL) and 2384 gene probes

Variable	Description
class	AD:Alzheimer's Disease MCI: Mild Cognitive impairment CTL: Control
ILMN_1343291	probe expression values
ILMN_1343295	probe expression values

In addition, the first 50 Variables(including the class) were taken for this analysis. Therefore, the data set consisted of 382 observations and 50 variables. I decided to condense the data set further by considering AD and MCI as the Experimental group(EXP) and the CTL class as the control group.

Table 2: Variables after pre-processing. The class variable is maintained but under different coding (EXP/CTL) and 2384 gene probes under the name of their respective gene name for interpretability

Variable	Description
class	EXP:AD or MCI CTL: Control
hla_drb5	gene associated to probe ILMN_1343291 expression values
hla_a29_1	gene associated to probe ILMN_1343295 expression values

1.2 Description of the Problem

The reference paper[1] that published this set of micro-array data is entitled: "A novel multi-tissue RNA diagnostic of healthy ageing relates to cognitive health status". My aim is to explore multivariate techniques that is able to discriminate the experimental and control groups. Using a small set of probes, my goal is to see how well various multivariate technique perform in terms of accuracy and mis-classification. Having a reliably high accuracy could provide a pipeline for diagnostic using a smaller number of probes to aid in future healthcare. In addition, the objective would be to understand the biological system and the genes involved.

2 Choice of methodology and motivation

The exploration begins with basic plotting of variables. However, because of the large number of genes, plotting by pairs did not appear useful or interpretable. Principle component analysis did not offer a significant reduction in dimensionality . I started with a basic logistic regression to predict the class(Experimental or Control). The backward selection procedure was used to aid in choosing variables from the data set. Logistic regression was chosen due to the fact that my objective is to determine if a generalized linear model is able to discriminate between the two classes (known grouping structure) based on a function of the variables chosen using backward selection. I explored using tree based modeling for a more interpretable methodology using the full 49 variables and using the variables selected using the logistic backward regression in order to compare trees and accuracy. While the grouping structure is known, I decided to explore hierarchical clustering in order to determine if I could detect and distinguish a grouping structure. The reference provided explore using K Nearest Neighbors as a potential classification technique to discriminate groups. I also tried using this model. The data was split between training and testing data to determine how well each method does in classification of test observation to the experimental and the control class.

3 Exploration and Interpretation

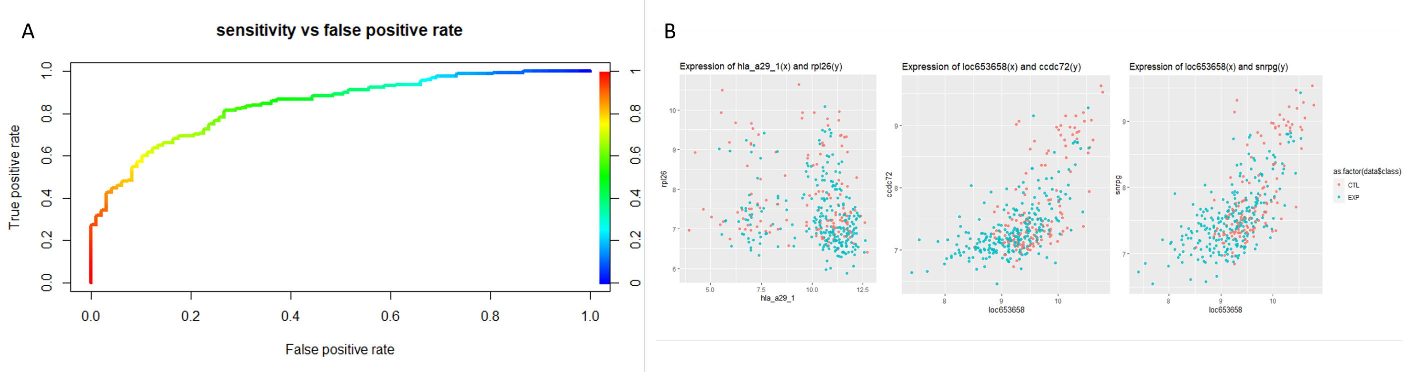

Figure 1: Results of the ROC curve(AUC=0.84) and general plots of gene expression plotted against each other

3.1 Logistic Discriminant Regression

The first methodology implemented was that of the Logistic regression, to take advantage of the grouping structure provided by the microarray data. An important aim of mine was to determine a small set of genes that could discriminate grouping structure. Therefore, this analysis started by using the backward selection to automattically select variables based on the Akaike Information Criteria. The resultant model is displayed below. From this method, 15 genes were chosen. Then for example, the odds of being in the experimental group(having Alzheimer's or mild cognitive impairment) is exp(b1)=exp(0.13)=1.13 times larger for each unit increase in the expression of HLA_A29_1 when all other variables are held constant.

3.2 Tree Based Modeling

All tree models created are not shown due to the fact that they are large and crowded. Therefore, the results of the pruned trees are shown in figure 2. I chose to look at the effect of pruning at different levels of terminal nodes. Sub figures A and C of figure2 are pruned at 5 terminal nodes, resulting in simpler and more interpretable trees. Sub figures B and D of figure2 are pruned at 10 terminal nodes and while they are larger, they still offer some level of interpretation. I chose to use trees due to their interpretability and ability to mimic human made decisions to discriminate groups. An important feature that these trees show is that in both cases when the backward selection variables and the full 49 variables are used, the gene LOC653658 is chosen as the best initial split. Trees provide

Table 3: Results of the Logistic regression for each variable

Explanatory Variables	Coefficient Estimate
(intercept)	44.38
hla_a29_1	0.13
rpl26	1.32
loc388588	0.37
folr3	-0.24
ifi44l	-0.46
loc648622	-2.25
cox7c	4.22
loc441377	1.73
loc653658	-3.1
ccdc72	-4.19
hbe1	-1.74
rpl13	1.84
loc644934	2.22
rps26	-3.07
hint1	2.69
klrb1	0.89

a measure of importance for each split and therefore it is critical that we investigate this gene. In addition, the mis-classification for each tree was measured and displayed in the table below.

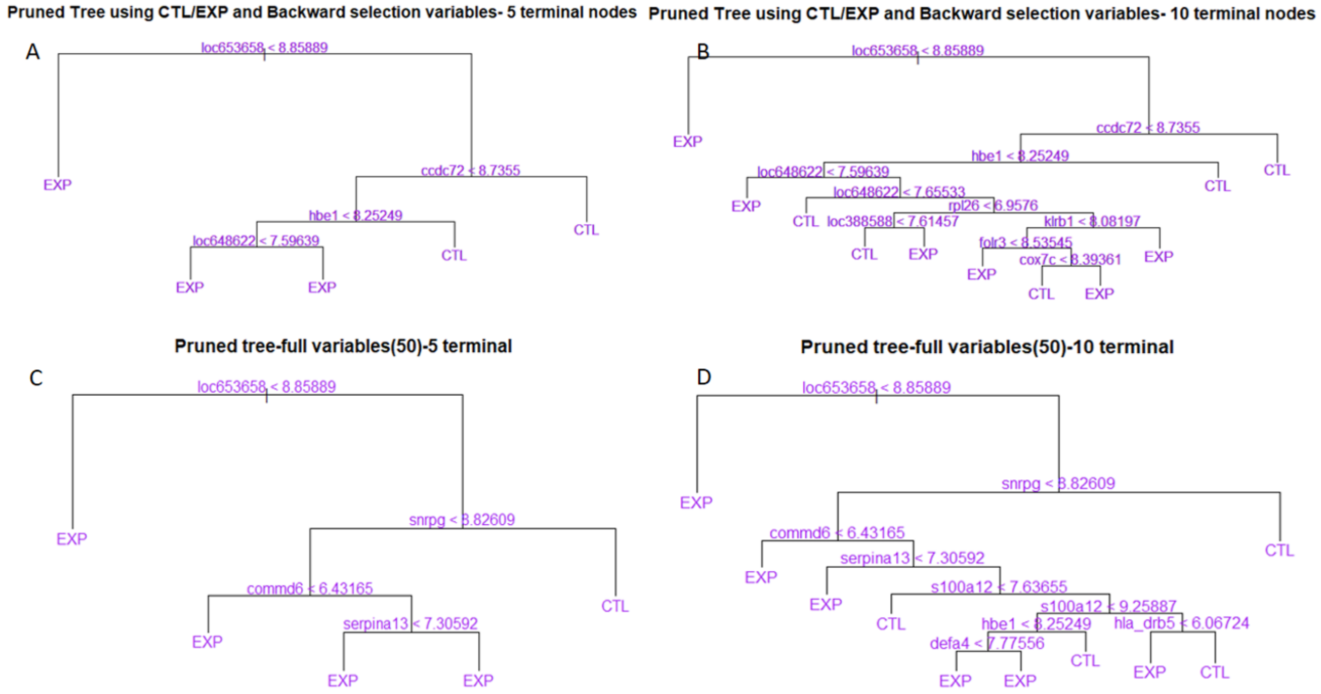

Figure 2: Results of Pruned tree-based discriminant analysis.A:5 terminal nodes Backward selection from logistic regression B: 10 terminal nodes from the backward selection. C: 5 terminal nodes using full 49 variable data set D: 10 terminal nodes using full 49 variable data set

After some research, LOC653658 appears to be a gene that is down-regulated in samples characterized by Alzheimer's disease. We can see this in the tree based models 2. Observations with expression levels less than 8.85 for LOC653658 are classified as EXP and is considered as one of the most important decision rules. The miss-classification rate for each model is displayed in table 4.

From 4, it can be seen that trees 1 and 4 have an extremely low mis-classification rate which may seem like a point in favor of using the trees. However , when cross validation trees are created, there is a large difference between the

Table 4: Mis-classification rates derived from all trees of different number of terminal nodes and different variables

Model	Mis-classification rate
tree 1: 15 variables	0.1024
tree 2: 15 variables pruned 5 terminal nodes	0.3071
tree 3: 15 variables pruned 10 terminal nodes	0.2008
tree 4: 49 variables	0.07874
tree 5: 49 variables pruned 5 terminal nodes	0.3189
tree 6: 49 variables pruned 10 terminal nodes	0.2244

mis-classification rates. This shows the importance of cross validation and underestimation of the error depicted by models trained on the training set. In addition, the table shows that the mis-classification rate decreases as as function of the number of terminal nodes. In this way, trees with more terminal nodes will have lower mis-classification rate at the expense of very large and deep trees. These trees will continue splitting until the terminal node consists of 1 observation each which is results in a tree that is un-interpretable In addition, there is not much of a difference between mis-classification rates of trees [2 and 5] and [3 and 6]. Then at the risk of a larger mis-classification rate, one could opt for tree model 2 in hopes of better interpretation. However, if there is a cost for mis-classification and hence a cost for false positive or false negatives, accuracy should be prioritized and model 3 could be investigated. Once again, this is in the context of this particular subset of data.

3.3 Hierarchical Clustering

While we know there is a grouping structure, I wanted to investigate if cluster analysis could detect this structure. Therefore, as part of the exploratory analysis, a profile plot (figure **??**) was done where each line represents an independent observation. Each observation has been colored in order to visualize the experimental and control group. However, this plot did not provide much information and based on the resultant pairs plot using cluster analysis, I decided that hierarchical clustering model was not the best model to use for such a large data set.

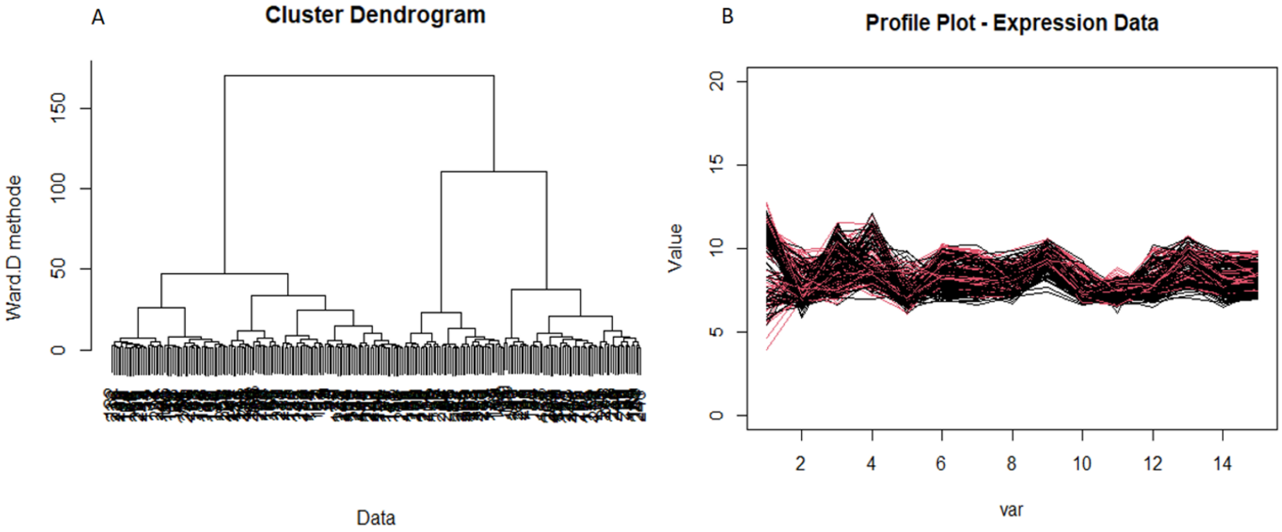

Figure 3: Results of cluster-based discriminant analysis. Not very informative. Dendrogram(A) shows some group structure

3.4 K-Nearest Neighbors

The K-Nearest Neighbors Classification method will classify observations according the the K nearest observations. Therefore, observations close to the new data point will "vote" towards the classification of the new observation - majority wins. The data has already been normalized. After reading that the paper actually uses K-NN to investigate grouping structure, I decided to try this method. On the training set, the optimal 'K' is chosen based on training accuracy. This will be an overestimation of the true accuracy. Figure 4 displays the results for each value of k between

1 and 5. Using 1-NN is an obvious overestimation and over fitting. Interestingly enough, 3-NN had the highest training accuracy. Therefore revealing that there could in fact be 3 classes in the data set. This is in fact correct. Earlier,I considered observations with Alzheimer's disease and mild cognitive impairment into 1 group and control observations into another. Using the training and testing sets, 3-NN had a training accuracy of 0.83 and a testing accuracy of 0.64. The advantages of K-NN is that it is easy to understand and flexible. However, it is also subject to over fitting.

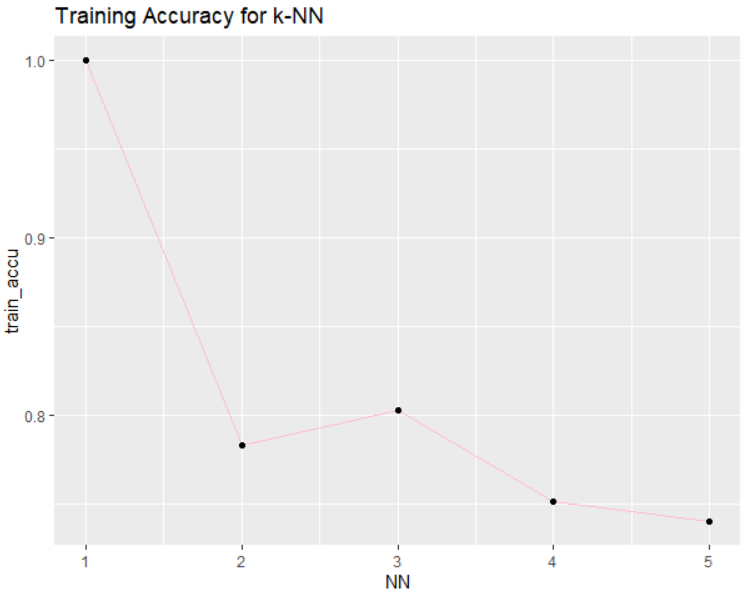

Figure 4: Results of K-NN

4 Conclusion

It is important to remember the pre-processing steps done to the data set. For simplicity, I considered the classes AD and MCI as the experimental class. However, the Nearest Neighbor approach actually displayed a higher accuracy for 3 classes rather than 2. In addition, only 49 genes were taken from the 2384 set of genes provided by the micro-array experiment. An improvement would be to identify marker genes for Alzheimer's disease and use these as the probes for the micro-array data. A smaller set of validated genes could be used to construct multivariate analysis models. A note of caution, from this research, I do not expect to get the same significant genes that were found in the respective paper due to the fact that I am only using 49 genes from a set over 2000genes. Another instance to consider is the reason for developing the model. If it is for diagnostic and a binary yes/no is required, one could look into more complex models that value accuracy over interpretation. However, for this report, I was interested in exploring the data and understanding/ interpreting the link between genes and groups. In this case, interpretable models such as tree-based models are preferred because they mimic human decisions.

5 Appendix

```

install.packages("janitor")
install.packages("class")
install.packages("corrplot")
install.packages("tree")

library("janitor")
library("class")
library("ggplot2")
library("gridExtra")
library("corrplot")
library("GEOquery")
library("grr")
library("leaps")
library("tree")
library("ROCR")
library("biomaRt")
library("cluster")


## logistic regression
data=read.table("C:\\Users\\user\\Desktop\\amsa_data\\GSE63061_reduced2.txt",sep=" ",header=TRUE)

### change gene names
series61 <- getGEO(filename = "C:\\Users\\user\\Desktop\\amsa_data\\GSE63061_series_matrix.txt")
GSE63061_expression <- as.data.frame(t(assayData(series61)$exprs))

probes <- colnames(GSE63061_expression)


probeData61 <- as.data.frame(matrix(nrow = length(probes)))
probeData61$probes <- probes
head(probeData61$probes)
probeData61$EntrezID <- featureData(series61)[["Entrez_Gene_ID"]][matches(probeData61$probes, featureData(series61)[[1]], all.y = F)$y]
probeData61$GeneName <- featureData(series61)[["Symbol"]][matches(probeData61$probes,featureData(series61)[[1]], all.y = F)$y]


k=colnames(data)[-1]
l=probeData61$probes
for (val in 1:length(k)) {
  for (j in 1:length(probeData61$probes))
    if (k[val]==probeData61$probes[j])
      # print(k[val])
      k[val]=probeData61$GeneName[j]
}

colnames(data)=c("class",k)
data <- clean_names(data)

## clean up workspace series 61 is a large expression set and will freeze the computer( froze my comp)
rm(series61,GSE63061_expression,probes,probeData61,l,val,j)

data=data[,c(1:50)]


## transform to binary and use logistic regression


data$class=ifelse(data$class=="CTL", "CTL","EXP")
```

Listing 1: Read data, load packages, reformat and clean data

```
training=data[c(1:254),c(1:50)]
testing=data[c(255:382),c(1:50)]
cutoff=0.5
```

```r
set.seed(1)





glm1=glm(training$class~.,family=binomial,data=training)
summary(glm1)

slm.back<- step(glm(as.factor(training$class) ~., family=binomial,training), direction = "backward")
summary(slm.back)


glm1_data_pred <- predict(slm.back, testing, type = "response")
predicted.glm1 <- as.numeric(glm1_data_pred > cutoff)
table(testing$class, predicted.glm1)
missclassification_logistic=(22+23)/(15+22+23+68)

predict<-fitted(slm.back)
pred<-prediction(predict,training$class)
perf<-performance(pred,measure="tpr",x.measure="fpr")
plot(perf,main="sensitivity vs false positive rate",colorize=TRUE,
     colorkey.relwidth=0.5,lwd=4.5)
perf_auc=performance(pred,measure="auc")
perf_auc@y.values
```

Listing 2: Logistic regression

```r
 data.tree1 <- tree(as.factor(training$class)~hla_a29_1+rpl26+loc388588+folr3+ifi44l
                   +loc648622+cox7c+loc441377
                   +loc653658+ccdc72+hbe1+rpl23+loc644934
                   +rps26+hint1+klrb1
                     ,method="recursive.partition",split="deviance",data=training)

tree.control(nobs=254)
attributes(data.tree1)
summary(data.tree1)
data.tree1
plot(data.tree1)
text(data.tree1,splits=T,all=T,pretty = 0)              # puts names on tree plot
title("Tree selection of variables-Backward selection variables ")

## check misclass of this tree
tree1.pred=predict(data.tree1,testing,type="class")
table(tree1.pred,testing$class)
missclass_tree1=(20+29)/(17+20+29+62)

## misclassification of the pruned tree

data.tree.cv1 <- cv.tree(data.tree1,FUN=prune.tree)
plot(data.tree.cv1$size,data.tree.cv1$k,type="l")
plot(data.tree.cv1$size,data.tree.cv1$dev,type="l",xlab="Number of end nodes",ylab="Deviance")
T.ind1<-prune.tree(data.tree1,best=5)
summary(T.ind1)

plot(T.ind1)
text(T.ind1, col="purple")
title("Pruned Tree using CTL/EXP and Backward selection variables- 5 terminal nodes")

## missclass
T.ind1.pred=predict(T.ind1,testing,type="class")
table(T.ind1.pred,testing$class)
missclass_tree_T.ind1=(31)/(7+1+30+90)



T.ind10<-prune.tree(data.tree1,best=10)
summary(T.ind10)

plot(T.ind10)
```

```
text(T.ind10,col="purple")
title("Pruned Tree using CTL/EXP and Backward selection variables- 10 terminal nodes")


T.ind10.pred=predict(T.ind10,testing,type="class")
table(T.ind10.pred,testing$class)
missclass_tree_T.ind10=(7+26)/(11+7+26+84)


## try using the full data variables with the training/testing
data.tree2 <- tree(as.factor(training$class) ~ ., data=training, method="recusive.partition",
                   split="deviance")
tree.control(nobs=254)

attributes(data.tree2)
summary(data.tree2)

data.tree2
plot(data.tree2)
text(data.tree2,splits=T,all=T)              # puts names on tree plot
title("tree selection of variables-full number of variables(50)")

## misclassif

tree2.pred=predict(data.tree2,testing,type="class")
table(tree2.pred,testing$class)
missclass_tree_2=(28+17)/(20+17+28+63)




data.tree.cv2 <- cv.tree(data.tree2,FUN=prune.tree)
plot(data.tree.cv2$size,data.tree.cv2$k,type="l")
plot(data.tree.cv2$size,data.tree.cv2$dev,type="l",xlab="Number of end nodes",ylab="Deviance")

T.ind2<-prune.tree(data.tree2,best=5)

summary(T.ind2)

par(mfrow=c(1,1))
plot(T.ind2)
text(T.ind2,pretty = 1,col="purple")
title("Pruned tree-full variables(50)-5 terminal")

T.ind2.pred=predict(T.ind2,testing,type="class")
table(T.ind2.pred,testing$class)
missclass_tree_2=(30+0)/(7+0+30+91)

T.ind3<-prune.tree(data.tree2,best=10)

summary(T.ind3)

par(mfrow=c(1,1))
plot(T.ind3)
text(T.ind3,pretty = 1,col="purple")
title("Pruned tree-full variables(50)-10 terminal")

T.ind3.pred=predict(T.ind3,testing,type="class")
table(T.ind3.pred,testing$class)
missclass_tree_3=(4+26)/(11+4+26+87)
```

Listing 3: Tree based models

```
data_subset=training[,c(3,10,13,16,18,20,22,25,27,30,31,37,40,44,45,50)]

data.clusD=hclust(dist(data_subset),method="ward.D")
plot(data.clusD,xlab="Data",ylab="Ward.D methode",sub="")



plot(c(1,15),c(-0,20),type="n",xlab="var",ylab="Value",main="Profile Plot - Expression Data")
# Use a loop to generate profiles for each observation.
for (k in (1:254))
{
  if(training$class[k]=="EXP")
    points(1:15,data_subset[k,],type="l",col=1)
  else
    points(1:15,data_subset[k,],type="l",col=2)
}

data_subset_cluster <- cutree(data.clusD,k=2)
table(data_subset_cluster)
table(training$class)


pairs(data_subset,col=data_subset_cluster,main="Predicted grouping structure in the data")
pairs(data_subset,col=as.factor(data$Class),main="Observed grouping structure in the data")



p1 <- ggplot(data, aes(x=hla_a29_1, y=rpl26, col=as.factor(data$class))) +
  geom_point(alpha=0.8)

p2 <- ggplot(data, aes(x=loc653658, y=snrpg, col=as.factor(data$class))) +
  geom_point(alpha=0.8)

p3 <- ggplot(data, aes(x=loc653658, y=ccdc72, col=as.factor(data$class))) +
  geom_point(alpha=0.8)

grid.arrange(p1 ,ncol=1)
print(p1 + ggtitle("Expression of hla_a29_1(x) and rpl26(y)"))

grid.arrange(p2 ,ncol=1)
print(p2 + ggtitle("Expression of loc653658(x) and snrpg(y)"))

grid.arrange(p3 ,ncol=1)
print(p3 + ggtitle("Expression of loc653658(x) and ccdc72(y)"))
```

Listing 4: "H-Clustering"

```
train_temp=training[,-1]
test_temp=testing[,-1]

train_acc <- c()


for (i in 1:5){
  set.seed(1)
  train_pred <- knn(train_temp, train_temp, training$class, k=i)
  train_acc <- c(train_acc, mean(train_pred == training$class))
}

tep_df<- data.frame(NN=c(1:5), train_accu=train_acc)


p4<- ggplot(data=tep_df, aes(x=NN, y=train_accu, group=1)) +
  geom_line(color="pink")+
  geom_point()


grid.arrange(p4 ,ncol=1)
print(p4 + ggtitle("Training Accuracy for k-NN"))


train_pred <- knn(train_temp, train_temp, training$class, k=3)
accuracy <- mean(train_pred == training$class)


testing_pred=knn(train_temp, test_temp, training$class, k=3)
```

```
testing_accuracy=mean(testing_pred==testing$class)
```

Listing 5: K-Nearest Neighbors

```
ensembl_human <- useMart("ensembl", dataset = "hsapiens_gene_ensembl")
attrib<-listAttributes(ensembl_human)



structureData_human <- getBM(mart = ensembl_human,
                             attributes = c("definition_1006"),
                             filters = "external_gene_name",
                             values = c("LOC653658"))
head(structureData_human)


```

Listing 6: Search Gene Ontology

References

[1] Sood, S., Gallagher, I. J., Lunnon, K., Rullman, E., Keohane, A., Crossland, H., Phillips, B. E., Cederholm, T., Jensen, T., van Loon, L. J., Lannfelt, L., Kraus, W. E., Atherton, P. J., Howard, R., Gustafsson, T., Hodges, A., and Timmons, J. A. A novel multi-tissue RNA diagnostic of healthy ageing relates to cognitive health status. *Genome Biology 16*, 1 (2015), 1–17.