Project two - Supervised and Unsupervised learning methods

Jishan Ahmed

December 9, 2017

Goverview of the data set

We can implement supervised and unsupervised machine learning techniques in analysis of genomic data. In particular, principal components analysis(PCA), hierarchical clustering and random forest are very useful machine learning tools to analyze genomic data. Illustration of these techniques has been presented in this project using the mRNA microarray data which is commonly known as Wang data. Wang data consists of 52580 gene expression measurements on 22 samples. We have used count table and phenotype table in our study. From Phenotype table, we have extracted different cell type information to implement our random forest methods. We do not make use of the phenotype information in performing PCA and clustering, as these are unsupervised techniques. But after performing PCA and clustering, we will check to see the extent to which these phenotype agree with the results of these unsupervised techniques. In this project, we have attempted to use the results of PCA in order to implement supervised learning algorithm Random forest.

First we read Count table and phenotype table to apply PCA, hierarchical clustering and random forest algorithms.

Data loading

```
# Read data files from local disk
gene <- read.csv("C:/Users/User 1/Desktop/gene.csv")
pheno <- read.csv("C:/Users/User 1/Desktop/pheno.csv")
dim(gene)
```

```
## [1] 52580    23
```

```
length(pheno)
```

```
## [1] 6
```

Data preprocessing

We begin by examining the Phenotype.

```
wang.pheno=pheno$cell.type
gene_Gene <- gene[,-1]
```

Dimension reduction method

Principal Components Analysis (PCA)

We first perform PCA on the data without scaling the variables (genes), because it is better not to scale the genes. It is also required to transpose our matrix.

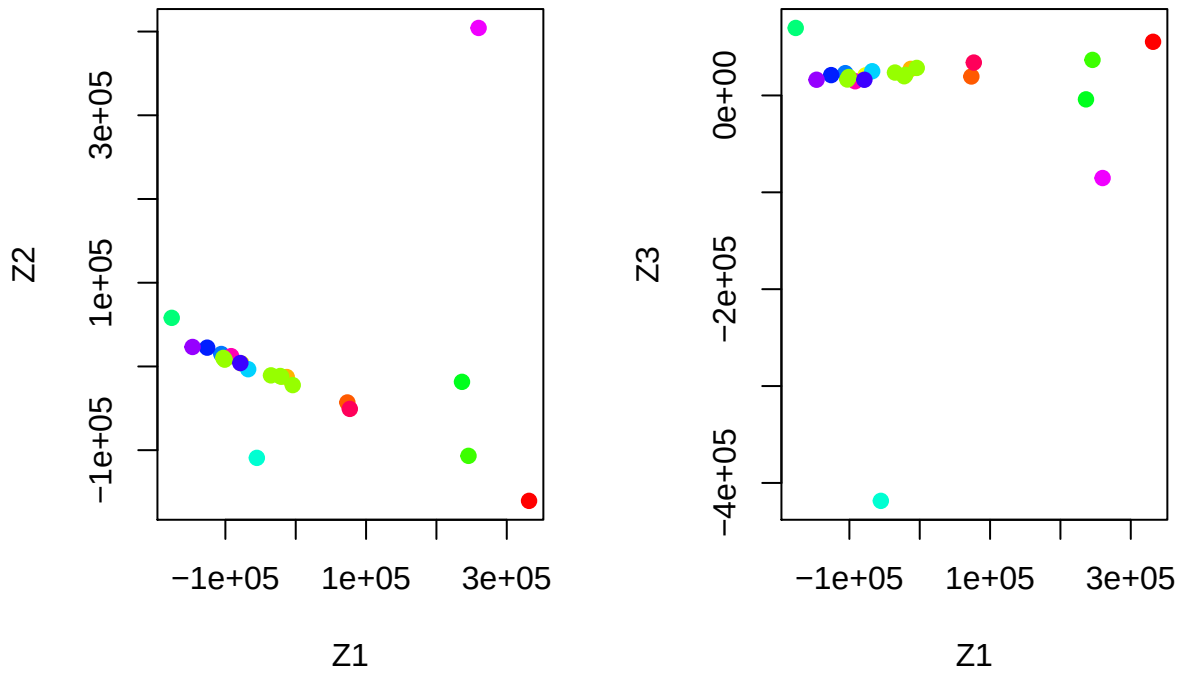
```
wang.data=t(gene_Gene)
```

```
pr.out2 =prcomp (wang.data, scale=FALSE)
```

We now plot the first few principal component score vectors, in order to visualize the data. The observations corresponding to a given cell type will be plotted in the same color, so that we can see to what extent the observations within a cell type are similar to each other. We first create a simple function that assigns a distinct color to each element of a numeric vector.

```
Cols=function(vec){
  cols=rainbow (length(unique(vec )))
  return (cols[as.numeric(as.factor (vec))])
}

par(mfrow =c(1,2))
plot(pr.out2$x[,1:2], col =Cols(wang.pheno), pch =19, xlab ="Z1",ylab="Z2")
plot(pr.out2$x[,c(1,3)],col =Cols(wang.pheno), pch =19, xlab ="Z1",ylab="Z3")
```



We see that it would not have been possible to visualize the data without using a dimension reduction method such as PCA, since based on the full data set there are huge possible scatterplots, none of which would have been particularly informative. We can obtain a summary of the proportion of variance explained (PVE) of the first few principal components using the

$$summary()$$

method for a

$$prcomp$$

object.

```
summary (pr.out2)
```

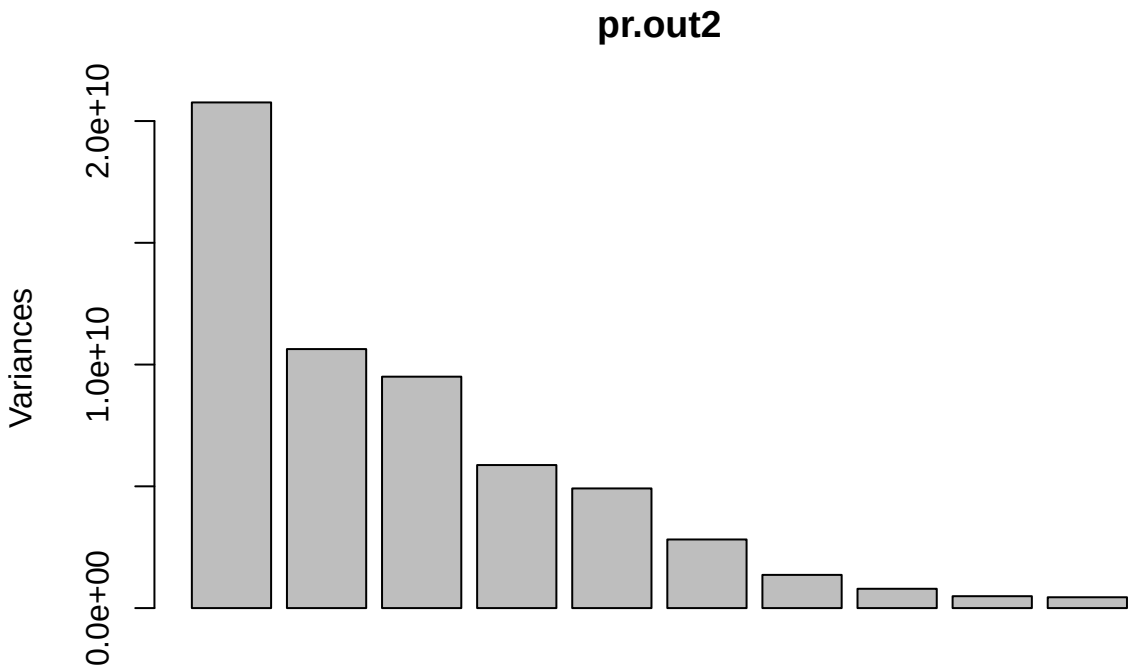
```
## Importance of components:
##                             PC1       PC2       PC3       PC4       PC5
## Standard deviation     1.441e+05 1.031e+05 9.748e+04 7.664e+04 7.009e+04
## Proportion of Variance 3.548e-01 1.817e-01 1.624e-01 1.004e-01 8.395e-02
## Cumulative Proportion  3.548e-01 5.365e-01 6.989e-01 7.993e-01 8.832e-01
##                             PC6       PC7       PC8       PC9      PC10
## Standard deviation     5.308e+04 3.693e+04 2.816e+04 2.210e+04 2.108e+04
## Proportion of Variance 4.813e-02 2.330e-02 1.355e-02 8.350e-03 7.600e-03
## Cumulative Proportion  9.313e-01 9.546e-01 9.682e-01 9.765e-01 9.841e-01
##                            PC11      PC12      PC13      PC14      PC15
## Standard deviation     1.749e+04 1.199e+04 1.176e+04 1.067e+04 9.577e+03
## Proportion of Variance 5.230e-03 2.460e-03 2.360e-03 1.950e-03 1.570e-03
## Cumulative Proportion  9.893e-01 9.918e-01 9.942e-01 9.961e-01 9.977e-01
##                            PC16      PC17      PC18      PC19      PC20
## Standard deviation     8.916e+03 5.180e+03 3.535e+03 3.031e+03 2.051e+03
## Proportion of Variance 1.360e-03 4.600e-04 2.100e-04 1.600e-04 7.000e-05
## Cumulative Proportion  9.990e-01 9.995e-01 9.997e-01 9.999e-01 1.000e+00
##                            PC21      PC22
## Standard deviation     1.72e+03 3.421e-10
## Proportion of Variance 5.00e-05 0.000e+00
## Cumulative Proportion  1.00e+00 1.000e+00
```

Using the

$$plot()$$

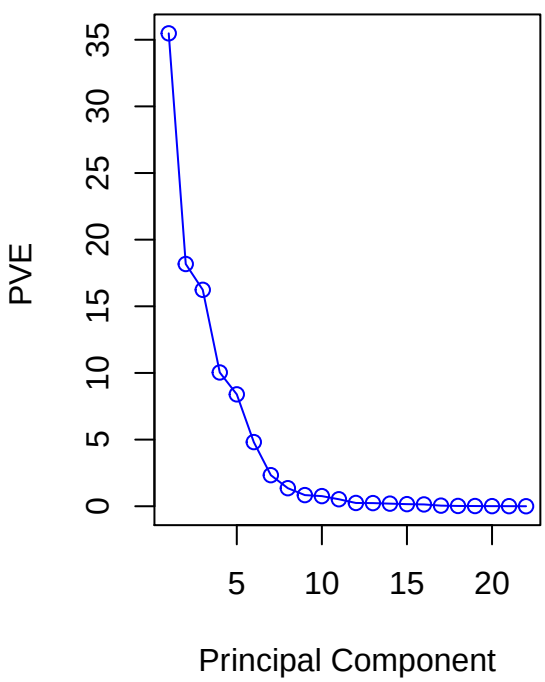
function, we can also plot the variance explained by the first few principal components.

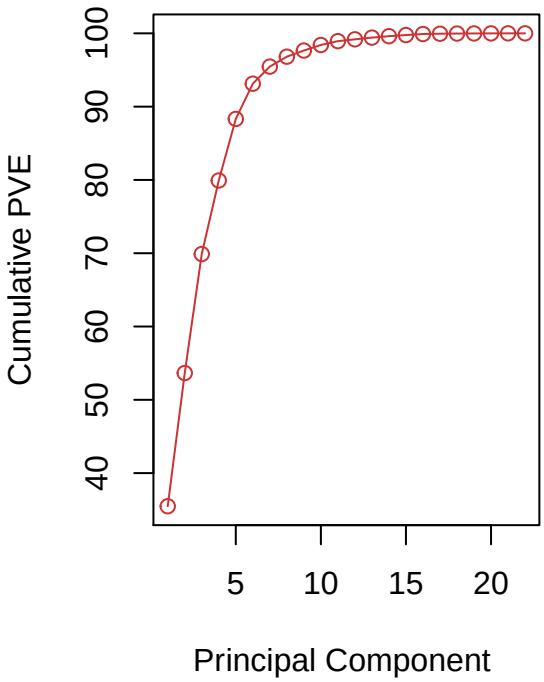
```
plot(pr.out2)
```



We observe that the height of each bar in the bar plot is given by squaring the corresponding element of pr.out$sdev. However, it is more informative to plot the PVE of each principal component which is known as a scree plot.

```
pve = 100* pr.out2$sdev ^2/ sum(pr.out2$sdev ^2)
par(mfrow =c(1,2))
plot(pve , type ="o", ylab="PVE ", xlab="Principal Component", col ="blue")
plot(cumsum (pve ), type="o", ylab ="Cumulative PVE", xlab="Principal Component ", col ="brown3 ")
```





The elements of pve can also be computed directly as

```
summary(pr.out2)$importance[2,]
```

```
##     PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9
## 0.35479 0.18173 0.16237 0.10037 0.08395 0.04813 0.02330 0.01355 0.00835
##    PC10    PC11    PC12    PC13    PC14    PC15    PC16    PC17    PC18
## 0.00760 0.00523 0.00246 0.00236 0.00195 0.00157 0.00136 0.00046 0.00021
##    PC19    PC20    PC21    PC22
## 0.00016 0.00007 0.00005 0.00000
```

```
summary(pr.out2)$importance[3,]
```

```
##     PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9
## 0.35479 0.53652 0.69889 0.79926 0.88320 0.93134 0.95464 0.96819 0.97653
##    PC10    PC11    PC12    PC13    PC14    PC15    PC16    PC17    PC18
## 0.98413 0.98935 0.99181 0.99418 0.99612 0.99769 0.99905 0.99951 0.99972
##    PC19    PC20    PC21    PC22
## 0.99988 0.99995 1.00000 1.00000
```

We see that together, the first 10 principal components explain around 98 %. However, looking at the scree plot, we see that while each of the first 10 principal components explain a substantial amount of variance, there is a marked decrease in the variance explained by further principal components. That is, there is an elbow in the plot after approximately the tenth principal component. This suggests that there may be little benefit to examining more than tenth or so principal components (though even examining ten principal components may be difficult).Together, all principal components explain 100% of the variance.

Unsupervised learning method

Hierarchical clustering

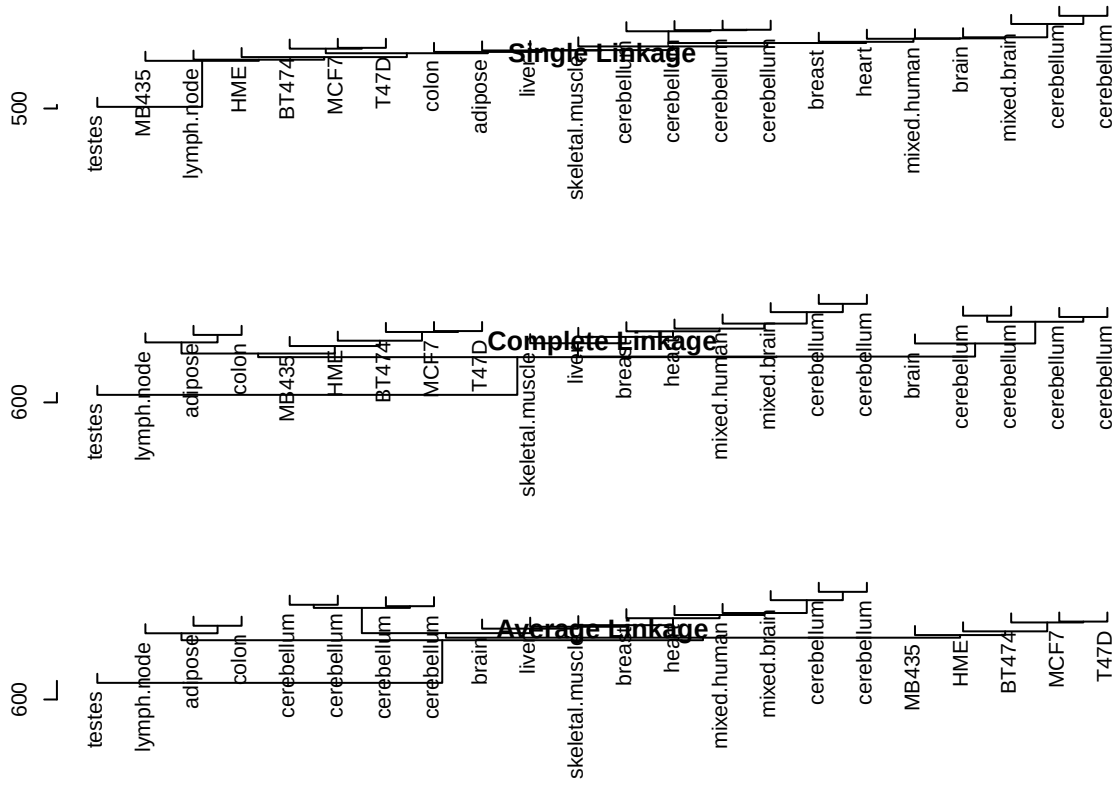
We have implemented hierarchical clustering algorithm to hierarchically cluster the cell type in the Wang data, with the goal of finding out whether or not the observations cluster into distinct types of cell type. To begin, we standardize the variables to have mean zero and standard deviation one in order to to keep each gene on the same scale.

```
# Standardize data to have mean 0 and stdev 1
sd.data=scale(wang.data)
```

We now perform hierarchical clustering of the observations using complete, single, and average linkage. Euclidean distance has been used as a distance metric to measure dissimilarity.

```
# Create distance matrix
data.dist=dist(sd.data)
```

```
hc.single =hclust(data.dist, method ="single")
hc.complete =hclust(data.dist, method ="complete")
hc.average =hclust(data.dist, method ="average")
par(mfrow =c(3,1))
plot(hc.single, labels =wang.pheno, main="Single Linkage", xlab="",sub ="", ylab="" )
plot(hc.complete ,labels =wang.pheno, main ="Complete Linkage", xlab="", sub ="", ylab="" )
plot(hc.average , labels =wang.pheno, main ="Average Linkage", xlab="", sub ="", ylab="" )
```



We see that the choice of linkage certainly does affect the results obtained. Complete and average linkage tend to yield evenly sized clusters whereas single linkage tends to yield extended clusters to which single

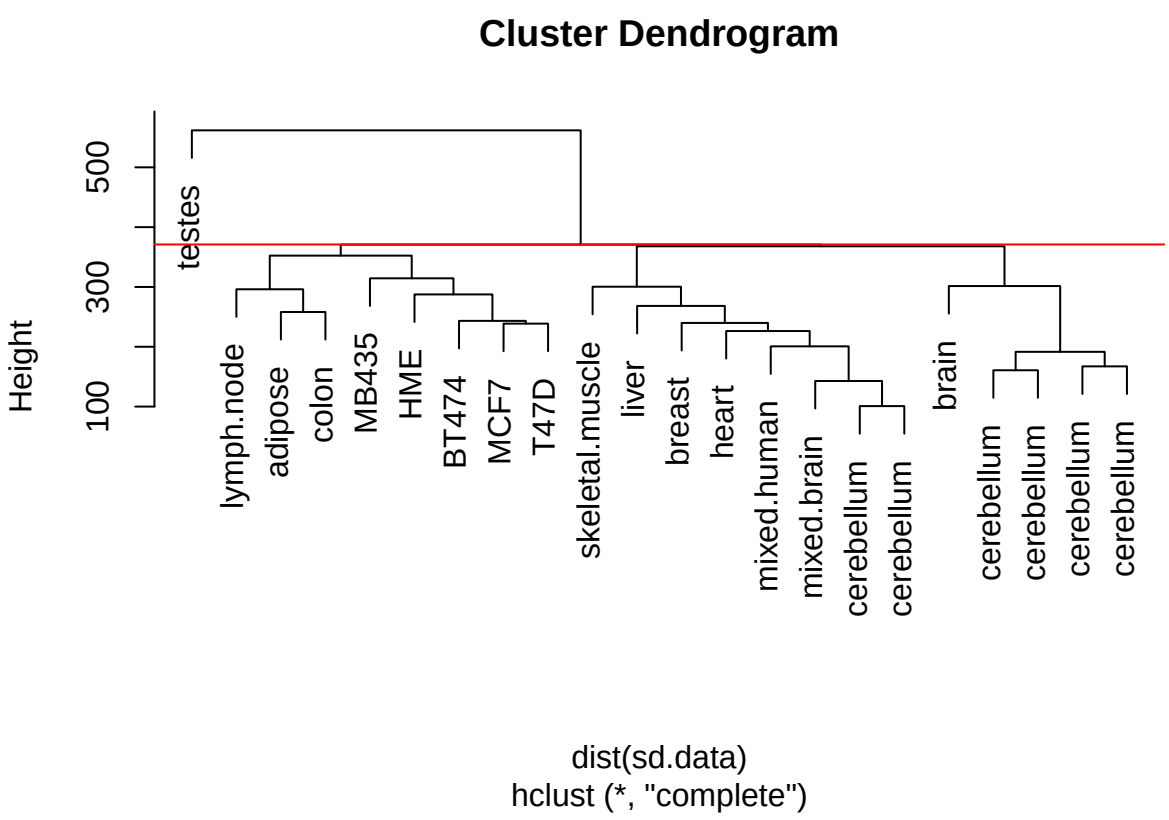
leaves are fused one by one. We will use complete linkage hierarchical clustering to analyze our findings. We can cut the dendrogram at the height that will yield a particular number of clusters. We have chosen three in this project.

```
hc.out =hclust(dist(sd.data))
hc.clusters =cutree (hc.out ,3)
table(hc.clusters ,wang.pheno)
```

```
##             wang.pheno
## hc.clusters adipose brain breast BT474 cerebellum colon heart HME liver
##           1       1     0      0     1          0     1     0   1     0
##           2       0     1      1     0          6     0     1   0     1
##           3       0     0      0     0          0     0     0   0     0
##             wang.pheno
## hc.clusters lymph.node MB435 MCF7 mixed.brain mixed.human skeletal.muscle
##           1          1     1    1           0           0               0
##           2          0     0    0           1           1               1
##           3          0     0    0           0           0               0
##             wang.pheno
## hc.clusters T47D testes
##           1    1      0
##           2    0      0
##           3    0      1
```

There are some clear patterns. All the cerebellum fall in same cluster. We can plot the cut on the dendrogram that produces three clusters:

```
par(mfrow =c(1,1))
plot(hc.out , labels =wang.pheno)
abline (h=371, col ="red")
```



The

$$abline()$$

function draws a straight line on top of any existing plot in

$$R$$

. The argument

$$h = 371$$

plots a horizontal line at height

$$371$$

on the dendrogram; This is the height that results in three distinct clusters. It is easy to verify that the resulting clusters are the same as the ones we obtained using

$$cutree(hc.out, 3)$$

.

Printing the output of hclust gives a useful brief summary of the object:

```
hc.out
```

```
##
## Call:
## hclust(d = dist(sd.data))
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 22
```

Supervised learning method

Random forest

We have found ten dominant features in our PCA analysis. Here we have performed Random forest using reduced dimension of feature space. We have considered ten features from PCA to implement random forest.

```
dimension.data <- data.frame(wang.pheno, pr.out2$x)
pca.data <- dimension.data[,1:11]
dim(pca.data)
```

```
## [1] 22 11
```

```
str(pca.data)
```

```
## 'data.frame':    22 obs. of  11 variables:
##  $ wang.pheno: Factor w/ 17 levels "adipose","brain",..: 4 8 11 12 16 1 2 3 5 5 ...
##  $ PC1       : num  -77042 -176294 -105813 -125894 -91801 ...
##  $ PC2       : num  3500 58042 14856 22449 12314 ...
##  $ PC3       : num  20727 69648 22990 21003 14639 ...
##  $ PC4       : num  -29044 317066 -22203 -27195 -29136 ...
##  $ PC5       : num  -6242 -687 -4633 4273 6656 ...
##  $ PC6       : num  4954 -38699 28779 24661 4670 ...
##  $ PC7       : num  17985 -7670 17558 21717 21045 ...
##  $ PC8       : num  -7608 2133 -21940 -12475 -873 ...
##  $ PC9       : num  -15112 427 -5106 -10522 -15412 ...
##  $ PC10      : num  27574 -3077 28362 23676 34829 ...
```

We have used

$$randomForest()$$

$$R$$

package in this project. By default, it uses

$$\sqrt{p}$$

variables when building a random forest of classification trees. We choose here default case that is

$$m_{try} = \sqrt{p}$$

.

```
set.seed(81)
library(MASS)
library(rpart)
library(randomForest)
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

Modelling with Random forest

Cell type has been used here as a response variable. We can consider this classification as multi class classification problems. We have attempted to classify 17 different cell type, namely BT474, HME , MB435, MCF7, T47D, adipose, brain, breast, cerebellum, colon, heart, liver, lymph node„ mixed brain, mixed human, skeletal muscle, testes.

```
rf.model <- randomForest(wang.pheno ~ ., data=pca.data, importance=TRUE)
```

```
rf.model
```

```
##
## Call:
##  randomForest(formula = wang.pheno ~ ., data = pca.data, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 72.73%
## Confusion matrix:
##                  adipose brain breast BT474 cerebellum colon heart HME
## adipose                0     0      0     0          0     1     0   0
## brain                  0     0      0     0          1     0     0   0
## breast                 0     0      0     0          1     0     0   0
## BT474                  0     0      0     0          0     0     0   0
## cerebellum             0     0      0     0          6     0     0   0
## colon                  0     0      0     0          1     0     0   0
## heart                  0     0      0     0          1     0     0   0
## HME                    0     0      0     0          1     0     0   0
## liver                  0     0      0     0          0     0     1   0
## lymph.node             0     0      0     0          0     0     0   0
## MB435                  0     0      0     0          0     0     0   0
## MCF7                   0     0      0     0          0     0     0   0
## mixed.brain            0     0      0     0          1     0     0   0
## mixed.human            0     0      0     0          0     0     0   0
## skeletal.muscle        0     0      0     0          0     0     0   1
## T47D                   0     0      0     1          0     0     0   0
## testes                 0     0      0     0          0     0     0   0
##                  liver lymph.node MB435 MCF7 mixed.brain mixed.human
## adipose              0          0     0    0           0           0
## brain                0          0     0    0           0           0
## breast               0          0     0    0           0           0
## BT474                0          0     0    0           0           0
## cerebellum           0          0     0    0           0           0
## colon                0          0     0    0           0           0
## heart                0          0     0    0           0           0
## HME                  0          0     0    0           0           0
## liver                0          0     0    0           0           0
## lymph.node           0          0     1    0           0           0
## MB435                0          0     0    1           0           0
## MCF7                 0          0     1    0           0           0
## mixed.brain          0          0     0    0           0           0
## mixed.human          0          0     0    0           1           0
## skeletal.muscle      0          0     0    0           0           0
## T47D                 0          0     0    0           0           0
## testes               0          1     0    0           0           0
##                  skeletal.muscle T47D testes class.error
## adipose                        0    0      0           1
## brain                          0    0      0           1
## breast                         0    0      0           1
## BT474                          0    1      0           1
```

```
## cerebellum                    0    0      0           0
## colon                         0    0      0           1
## heart                         0    0      0           1
## HME                           0    0      0           1
## liver                         0    0      0           1
## lymph.node                    0    0      0           1
## MB435                         0    0      0           1
## MCF7                          0    0      0           1
## mixed.brain                   0    0      0           1
## mixed.human                   0    0      0           1
## skeletal.muscle               0    0      0           1
## T47D                          0    0      0           1
## testes                        0    0      0           1
```

```
#plot(rf.model)
```

We see that the number of variables has been tried at each split is 3. We have found OOB estimate of error rate is 72.73%. We have also used our entire data as a training set to see our model performance.

```
#colnames(pca.data)
#Prediction on training data
predicted.values <- predict(rf.model, pca.data[1:11])
d_pca <- table(predicted.values, pca.data$wang.pheno)
print(d_pca)
```

```
##
## predicted.values  adipose brain breast BT474 cerebellum colon heart HME
##   adipose               1     0      0     0          0     0     0   0
##   brain                 0     1      0     0          0     0     0   0
##   breast                0     0      1     0          0     0     0   0
##   BT474                 0     0      0     1          0     0     0   0
##   cerebellum            0     0      0     0          6     0     0   0
##   colon                 0     0      0     0          0     1     0   0
##   heart                 0     0      0     0          0     0     1   0
##   HME                   0     0      0     0          0     0     0   1
##   liver                 0     0      0     0          0     0     0   0
##   lymph.node            0     0      0     0          0     0     0   0
##   MB435                 0     0      0     0          0     0     0   0
##   MCF7                  0     0      0     0          0     0     0   0
##   mixed.brain           0     0      0     0          0     0     0   0
##   mixed.human           0     0      0     0          0     0     0   0
##   skeletal.muscle       0     0      0     0          0     0     0   0
##   T47D                  0     0      0     0          0     0     0   0
##   testes                0     0      0     0          0     0     0   0
##
## predicted.values  liver lymph.node MB435 MCF7 mixed.brain mixed.human
##   adipose             0          0     0    0           0           0
##   brain               0          0     0    0           0           0
##   breast              0          0     0    0           0           0
##   BT474               0          0     0    0           0           0
##   cerebellum          0          0     0    0           0           0
##   colon               0          0     0    0           0           0
##   heart               0          0     0    0           0           0
##   HME                 0          0     0    0           0           0
##   liver               1          0     0    0           0           0
```

```
##   lymph.node          0          1     0    0           0           0
##   MB435               0          0     1    0           0           0
##   MCF7                0          0     0    1           0           0
##   mixed.brain         0          0     0    0           1           0
##   mixed.human         0          0     0    0           0           1
##   skeletal.muscle     0          0     0    0           0           0
##   T47D                0          0     0    0           0           0
##   testes              0          0     0    0           0           0
##
## predicted.values  skeletal.muscle T47D testes
##   adipose                       0    0      0
##   brain                         0    0      0
##   breast                        0    0      0
##   BT474                         0    0      0
##   cerebellum                    0    0      0
##   colon                         0    0      0
##   heart                         0    0      0
##   HME                           0    0      0
##   liver                         0    0      0
##   lymph.node                    0    0      0
##   MB435                         0    0      0
##   MCF7                          0    0      0
##   mixed.brain                   0    0      0
##   mixed.human                   0    0      0
##   skeletal.muscle               1    0      0
##   T47D                          0    1      0
##   testes                        0    0      1
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
confusionMatrix(predicted.values,pca.data$wang.pheno)
```

```
## Confusion Matrix and Statistics
##
##                 Reference
## Prediction       adipose brain breast BT474 cerebellum colon heart HME
##   adipose              1     0      0     0          0     0     0   0
##   brain                0     1      0     0          0     0     0   0
##   breast               0     0      1     0          0     0     0   0
##   BT474                0     0      0     1          0     0     0   0
##   cerebellum           0     0      0     0          6     0     0   0
##   colon                0     0      0     0          0     1     0   0
##   heart                0     0      0     0          0     0     1   0
##   HME                  0     0      0     0          0     0     0   1
##   liver                0     0      0     0          0     0     0   0
##   lymph.node           0     0      0     0          0     0     0   0
```

```
##   MB435                0     0      0     0          0     0     0   0
##   MCF7                 0     0      0     0          0     0     0   0
##   mixed.brain          0     0      0     0          0     0     0   0
##   mixed.human          0     0      0     0          0     0     0   0
##   skeletal.muscle      0     0      0     0          0     0     0   0
##   T47D                 0     0      0     0          0     0     0   0
##   testes               0     0      0     0          0     0     0   0
##                  Reference
## Prediction        liver lymph.node MB435 MCF7 mixed.brain mixed.human
##   adipose             0          0     0    0           0           0
##   brain               0          0     0    0           0           0
##   breast              0          0     0    0           0           0
##   BT474               0          0     0    0           0           0
##   cerebellum          0          0     0    0           0           0
##   colon               0          0     0    0           0           0
##   heart               0          0     0    0           0           0
##   HME                 0          0     0    0           0           0
##   liver               1          0     0    0           0           0
##   lymph.node          0          1     0    0           0           0
##   MB435               0          0     1    0           0           0
##   MCF7                0          0     0    1           0           0
##   mixed.brain         0          0     0    0           1           0
##   mixed.human         0          0     0    0           0           1
##   skeletal.muscle     0          0     0    0           0           0
##   T47D                0          0     0    0           0           0
##   testes              0          0     0    0           0           0
##                  Reference
## Prediction        skeletal.muscle T47D testes
##   adipose                       0    0      0
##   brain                         0    0      0
##   breast                        0    0      0
##   BT474                         0    0      0
##   cerebellum                    0    0      0
##   colon                         0    0      0
##   heart                         0    0      0
##   HME                           0    0      0
##   liver                         0    0      0
##   lymph.node                    0    0      0
##   MB435                         0    0      0
##   MCF7                          0    0      0
##   mixed.brain                   0    0      0
##   mixed.human                   0    0      0
##   skeletal.muscle               1    0      0
##   T47D                          0    1      0
##   testes                        0    0      1
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.8456, 1)
##     No Information Rate : 0.2727
##     P-Value [Acc > NIR] : 3.855e-13
##
##                   Kappa : 1
```

```
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: adipose Class: brain Class: breast
## Sensitivity                 1.00000      1.00000       1.00000
## Specificity                 1.00000      1.00000       1.00000
## Pos Pred Value              1.00000      1.00000       1.00000
## Neg Pred Value              1.00000      1.00000       1.00000
## Prevalence                  0.04545      0.04545       0.04545
## Detection Rate              0.04545      0.04545       0.04545
## Detection Prevalence        0.04545      0.04545       0.04545
## Balanced Accuracy           1.00000      1.00000       1.00000
##                      Class: BT474 Class: cerebellum Class: colon
## Sensitivity               1.00000            1.0000      1.00000
## Specificity               1.00000            1.0000      1.00000
## Pos Pred Value            1.00000            1.0000      1.00000
## Neg Pred Value            1.00000            1.0000      1.00000
## Prevalence                0.04545            0.2727      0.04545
## Detection Rate            0.04545            0.2727      0.04545
## Detection Prevalence      0.04545            0.2727      0.04545
## Balanced Accuracy         1.00000            1.0000      1.00000
##                      Class: heart Class: HME Class: liver
## Sensitivity               1.00000    1.00000      1.00000
## Specificity               1.00000    1.00000      1.00000
## Pos Pred Value            1.00000    1.00000      1.00000
## Neg Pred Value            1.00000    1.00000      1.00000
## Prevalence                0.04545    0.04545      0.04545
## Detection Rate            0.04545    0.04545      0.04545
## Detection Prevalence      0.04545    0.04545      0.04545
## Balanced Accuracy         1.00000    1.00000      1.00000
##                      Class: lymph.node Class: MB435 Class: MCF7
## Sensitivity                    1.00000      1.00000     1.00000
## Specificity                    1.00000      1.00000     1.00000
## Pos Pred Value                 1.00000      1.00000     1.00000
## Neg Pred Value                 1.00000      1.00000     1.00000
## Prevalence                     0.04545      0.04545     0.04545
## Detection Rate                 0.04545      0.04545     0.04545
## Detection Prevalence           0.04545      0.04545     0.04545
## Balanced Accuracy              1.00000      1.00000     1.00000
##                      Class: mixed.brain Class: mixed.human
## Sensitivity                     1.00000            1.00000
## Specificity                     1.00000            1.00000
## Pos Pred Value                  1.00000            1.00000
## Neg Pred Value                  1.00000            1.00000
## Prevalence                      0.04545            0.04545
## Detection Rate                  0.04545            0.04545
## Detection Prevalence            0.04545            0.04545
## Balanced Accuracy               1.00000            1.00000
##                      Class: skeletal.muscle Class: T47D Class: testes
## Sensitivity                         1.00000     1.00000       1.00000
## Specificity                         1.00000     1.00000       1.00000
## Pos Pred Value                      1.00000     1.00000       1.00000
## Neg Pred Value                      1.00000     1.00000       1.00000
```

```
## Prevalence                    0.04545     0.04545     0.04545
## Detection Rate                0.04545     0.04545     0.04545
## Detection Prevalence          0.04545     0.04545     0.04545
## Balanced Accuracy             1.00000     1.00000     1.00000
```

Results

In this project, we have used Wang data which is a high dimensional genomic data. From our PCA analysis, we have found that the first ten principal components are capable of explaining approximately 98% of the variance of the data. It is reasonable to use this ten components for our further investigation. We have then performed hierarchical clustering of the observations using complete, single, and average linkage along with Euclidean distance. We see that complete linkage performs better. It is noticeable that cerebellums were clustered in cluster three. Three distinct clusters are clearly visible in our denrogram. To explore more information regarding this data, we have implemented supervised learning algorithm Random forest. Random forest is really a good choice to classify this data. It has only 22 samples. This is too small to implement other available ML tools. We know that Random forest utilizes bootstrap method. It has built in feature to perform cross validation. It gives us Out of Bag error which can be considered as test error rate. Hence we can skip cross validation. We have found estimate of OOB error rate is 72.73% which seems very high but this result is convincing. We have predicted pretty large number of classes i.e 17 classes from 22 samples using 10 predictors. To convince ourselves, we have made a prediction using our training data. We have observed that our model performed perfectly. Our classification accuracy, kappa accuracy, sensitivity, and specificity were 100%.

Conclusion

In conclusion, it is always challenging to unfold the mystery of genomic data. Most of the ML tools are incapable of handling data with large predictors. In this project, we have successfully implemented dimension reduction method to explore the genomic data. Clustering methods also have helped us to visualize some patterns in our data. We have also explored the beauty of dimension reduction in implementing Random forest classification. It is hard to classify these kinds of genomic data with large number of predictors without dimension reduction. In this project, we have experienced that data preparation and transition from supervised to unsupervised learning are really a fascinating jobs.