STAT646 - Homework 2

Daniel Osorio - dcosorioh@tamu.edu
Department of Veterinary Integrative Biosciences
Texas A&M University

1. For the yeast data, do the following:

```
> yeastFiles <- list.files("yeastData/", full.names = TRUE)
> yeastInfo <- lapply(yeastFiles, function(file){
+   read.csv(file, sep = "\t", comment.char = "!")
+ })
> spots <- table(unlist(lapply(yeastInfo, function(X){X[,1]})))
> spots <- names(spots)[spots == length(yeastInfo)]
> yeastInfo <- lapply(yeastInfo, function(X){
+     X[X[,1] %in% spots,c("Ch1.Intensity..Median.","Ch2.Intensity..Median.")]
+ })
> sNames <- gsub(".txt$","",basename(yeastFiles))
> sNames <- gsub("[[:alpha:]]","",sNames)
> chanel1 <- sapply(yeastInfo, function(X){X[,1]})
> chanel2 <- sapply(yeastInfo, function(X){X[,2]})
> colnames(chanel1) <- colnames(chanel2) <- sNames
> rownames(chanel1) <- rownames(chanel2) <- spots
> Y <- log2(chanel2) - log2(chanel1)
```

   (a) For each of average, complete, and single linkage, carry out hierarchical clustering on the samples using Euclidean distance.

```
> distanceMatrix <- dist(t(Y))
> avgHclust <- hclust(distanceMatrix, method = "average")
> sglHclust <- hclust(distanceMatrix, method = "single")
> cptHclust <- hclust(distanceMatrix, method = "complete")
```

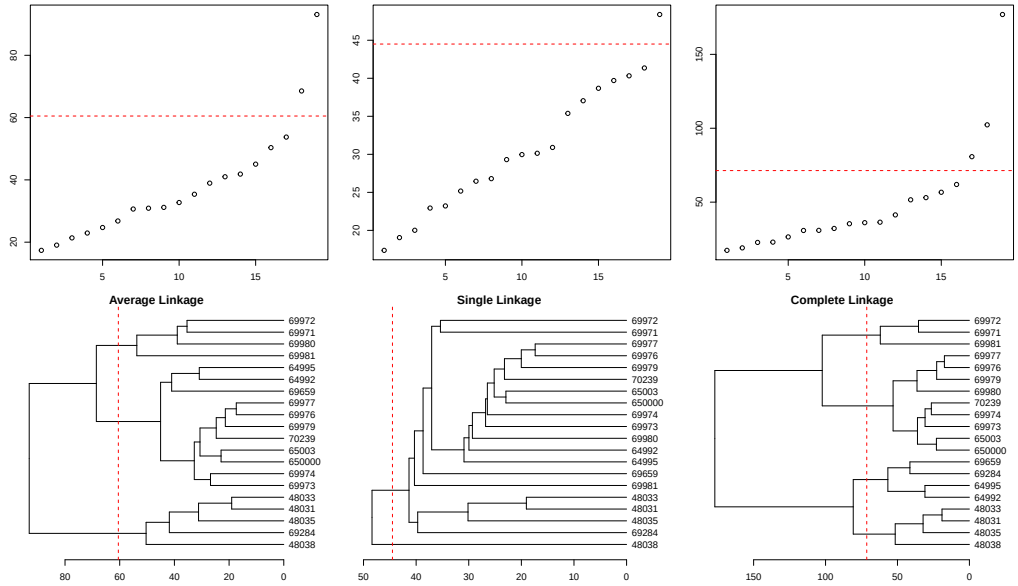
      i. Which two samples are merged first, and what is the distance between them?

```
> c(sNames[abs(avgHclust$merge[1,])], round(avgHclust$height[1],2))
[1] "69976" "69977" "17.37"
> c(sNames[abs(sglHclust$merge[1,])], round(sglHclust$height[1],2))
[1] "69976" "69977" "17.37"
> c(sNames[abs(cptHclust$merge[1,])], round(cptHclust$height[1],2))
[1] "69976" "69977" "17.37"
```

      ii. Report a height plot showing merge distances. How many clusters would you say there are? *Depending of the linkage method, between 2 and 4.*

```
> par(mfrow=c(2,3), mar=c(3,3,1,1))
> plot(avgHclust$height)
> abline(h = 60.5, col = "red", lty = 2)
```

```
> plot(sglHclust$height)
> abline(h = 44.5, col = "red", lty = 2)
> plot(cptHclust$height)
> abline(h = 71.35, col = "red", lty = 2)
> par(mar=c(3,2,1,4))
> plot(as.dendrogram(avgHclust), horiz = TRUE,
+      main = "Average Linkage")
> abline(v = 60.5, lty = 2, col= "red")
> plot(as.dendrogram(sglHclust), horiz = TRUE,
+      main = "Single Linkage")
> abline(v = 44.5, lty = 2, col= "red")
> plot(as.dendrogram(cptHclust), horiz = TRUE,
+      main = "Complete Linkage")
> abline(v = 71.35, lty = 2, col= "red")
```

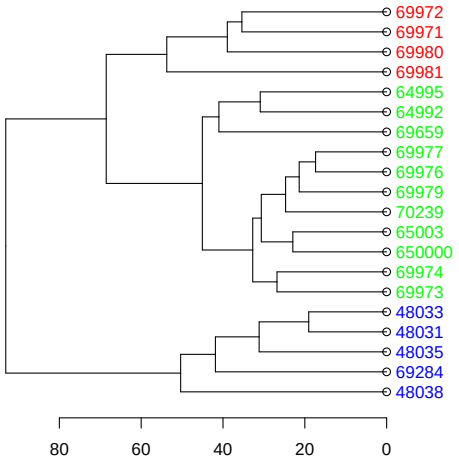


iii. Interpret the clusters that you found with respect to the "meta" data for this dataset *The clusters are mainly driven by the strain type, fuel (10) samples are always clustered together as well as the wine's (4) samples, depending on the linkage method, the association between strains is recovered differentially.*

(b) Carry out K-means clustering, with K = 3. How do the results compare to the hierarchical clustering results you obtained using average linkage? *Main clusters are recovered identically*

```
> avgHclust <- as.dendrogram(avgHclust)
> set.seed(11)
> K <- kmeans(distanceMatrix, centers = 3)
> labelCol <- function(x) {
+   if (is.leaf(x)) {
+     label <- attr(x, "label")
+     K1 <- names(K$cluster)[K$cluster == 1]
+     K2 <- names(K$cluster)[K$cluster == 2]
+     K3 <- names(K$cluster)[K$cluster == 3]
+     attr(x, "nodePar") <-
+       list(lab.col=if(label %in% K1){
+       "red"} else {if (label %in% K2){"blue"} else{"green"}})}
+   return(x)
```

```
+ }
> avgHclust <- dendrapply(avgHclust, labelCol)
> nodePar <- list(pch = c(NA,NA))
> par(mar=c(3,15,1,15))
> plot(avgHclust, horiz = TRUE, nodePar = nodePar)
```



2. The cardiothoracic data, in file 'GDS4308.soft', are described here:

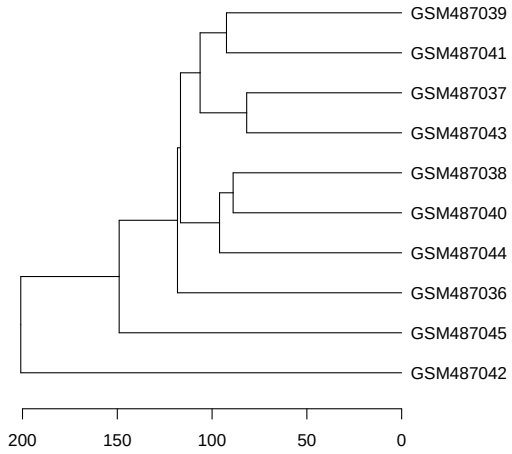
   https://www.ncbi.nlm.nih.gov/sites/GDSbrowser?acc=GDS4308

   Note that the first 97 lines of the data file consist of meta data, so the gene intensities start on line 98. Note also that the last line of the data file contains a table description and should not be included with the gene intensities. In what follows, work with the log2-transformed intensities. Inspect the meta data in "GDS4308.soft" (you can just open it in a text editor) to figure out what the column names correspond to. Note that these are paired data.

```
> GSE19533 <- read.csv(file = "GDS4308_full.soft",
+                      sep = "\t", skip = 122,
+                      comment.char = "!")
> GSE19533 <- log2(GSE19533[,3:12])
```

   (a) Carry out hierarchical clustering of the individuals, using Euclidean distance and complete linkage. Interpret the results.

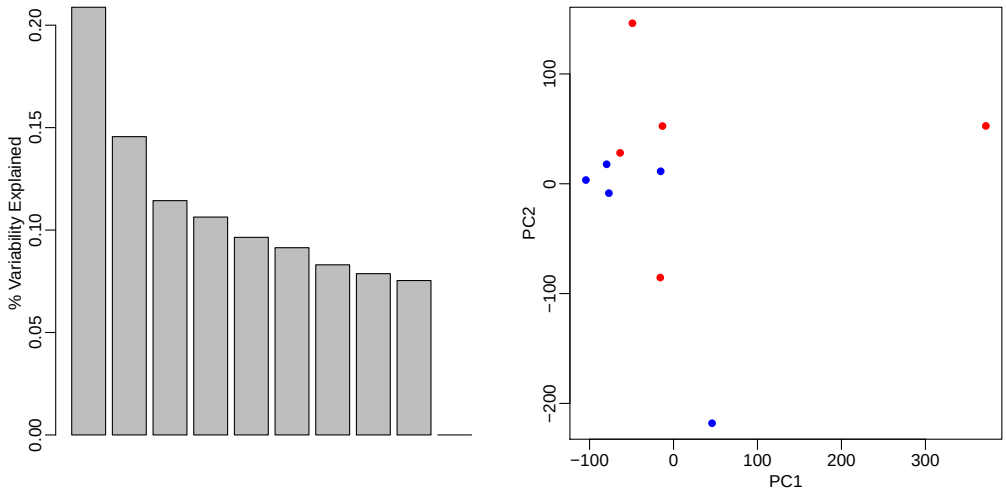
```
> dGSE19533 <- dist(t(GSE19533))
> hc <- hclust(dGSE19533, method = "complete")
> par(mar=c(3,15,1,15))
> plot(as.dendrogram(hc), horiz = TRUE)
```



Clusters are driven by the state of the surgery, pre- and post- surgery samples are mainly clustered together.

(b) Carry out a principal component analysis of the intensities, treating individuals (columns) as variables and features (rows) as replicates. Interpret the results.

```
> cGSE19533 <- t(scale(t(GSE19533)))
> PC <- prcomp(t(cGSE19533))
> colors <- ifelse(
+   test = rownames(PC$x) %in% paste0("GSM4870",seq(37,45,2)),
+   yes = "blue",
+   no = "red")
> par(mfrow=c(1,2), mar=c(3,3,1,1), mgp=c(1.5,0.5,0))
> barplot(PC$sdev/sum(PC$sdev), ylab = "% Variability Explained")
> plot(PC$x[,1:2], col = colors, pch = 16)
```

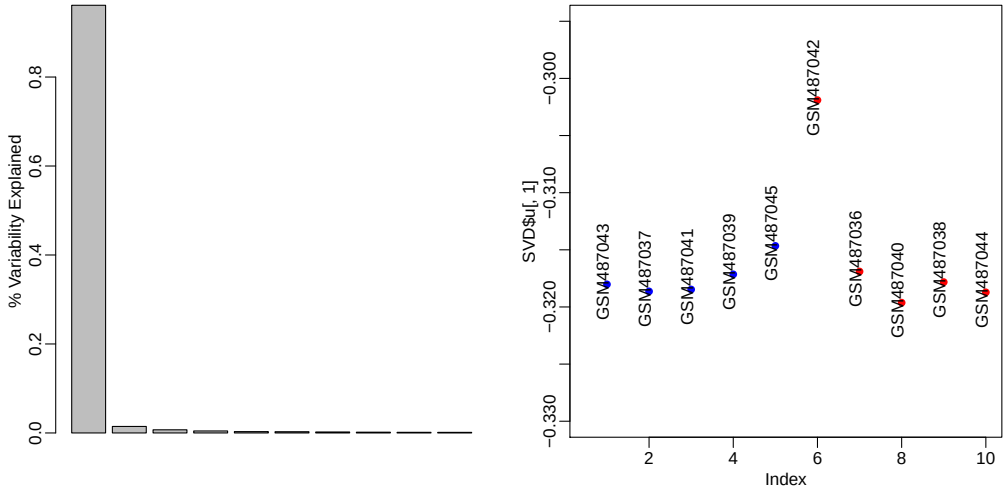


The first two principal components explain the 21% and the 15% of the variability respectively.

(c) Carry out singular value decomposition of the column-centered intensities. Interpret the results.

```
> SVD <- svd(t(scale(GSE19533)))
> par(mfrow=c(1,2), mar=c(3,3,1,1), mgp=c(1.5,0.5,0))
> barplot((SVD$d^2)/sum(SVD$d^2), ylab = "% Variability Explained")
> plot(SVD$u[,1], col= colors, pch =16,
+      ylim = c(-0.33,-0.295), xlim = c(0.5,10))
```

```
> text(x = 1:10-0.2, y =SVD$u[,1] + 0.002,
+      colnames(cGSE19533),pos = 1, srt = 90)
```



This pattern explains the 96% of the variability. Sample GSM487042 is the most different one.

(d) Compute one-sample t-statistics for each gene to search for features for which the mean paired difference is not 0. Use the bootstrap to obtain p-values. Use p.adjust to translate the p-values to FDR estimates (specify argument method = 'fdr'). How many features are significant at an estimated FDR of 0.05? I will give tips on R code for this problem in class and Q&A.

```
> pre <- paste0("GSM4870", seq(37,45,2))
> pos <- paste0("GSM4870", seq(36,44,2))
> diff <- GSE19533[,pos] - GSE19533[,pre]
> # Statistic under the original values
> tStat <- apply(diff,1,function(X){t.test(X)$statistic})
> # Centering the values to make simulation under the null hypothesis
> diff <- t(scale(t(diff)))
> n <- nrow(diff)
> # Bootstrap
> boot <- sapply(1:100, function(b){
+   sColumn <- sample(1:5,replace = TRUE)
+   if(length(unique(sColumn)) > 1){
+     apply(diff[,sColumn],1,function(X){t.test(X)$statistic})
+   } else {
+     rep(NA, n)
+   }
+ })

> # p-Value
> pValues <- rowMeans(abs(boot) > abs(tStat), na.rm = TRUE)

> # How many features are significant at an estimated FDR of 0.05?
> pValues <- p.adjust(pValues, method = "fdr")
> sum(pValues < 0.05)

[1] 307
```