**Lab 3**. Introduction to gene expression data: normalization and exploratory cluster analysis. There are two main technologies for generating high-throughput gene expression data: microarray and RNA-Seq. In this lab, we will work with an RNA-Seq based gene expression data set from a study of major depressive disorder. Include plots in your Word doc report and use a color to indicate your answers.

**A**. Load Data. Download the RNA-Seq data from Harvey (sense.filtered.cpm.Rdata). Load the data into an RStudio with an R script.

**1.** How many genes and samples are there? The rows are genes and the columns are samples.

157 samples, 8923 genes

# load gene expression data

**# set the current working directory first (setwd())**

load("sense.filtered.cpm.Rdata")

dim(sense.filtered.cpm)

colnames(sense.filtered.cpm)

Download the demographic (sex, age,…) and clinical diagnostic (depression) data from Harvey (Demographic\_symptom.csv).

# phenotype (mdd/hc) is in a separate file

# match phenotypes to expression data subject ids

subject.attrs <- read.csv("Demographic\_symptom.csv", stringsAsFactors = FALSE)

dim(subject.attrs) # 160 subjects x 40 attributes

colnames(subject.attrs) # interested in X (sample ids) and Diag (diagnosis)

subject.attrs$X

subject.attrs$Diag

There are more samples in the demographic data than in the gene expression data, so we need to match gene expression samples with their diagnosis.

**2.** How many cases and controls are there?

[in]sum(subject.attrs$Diag == "MDD")

[out] 80

[in]sum(subject.attrs$Diag == "HC ")

[out] 80

80 healthy controls, 80 diagnosed with depression disorder

library(dplyr) # install.packages("dplyr")

# create a phenotype vector

# grab X (subject ids) and Diag (Diagnosis) from subject.attrs that

# intersect %in% with the RNA-Seq data

phenos.df <- subject.attrs %>%

filter(X %in% colnames(sense.filtered.cpm)) %>%

dplyr::select(X, Diag)

colnames(phenos.df) # $Diag is mdd diagnosis

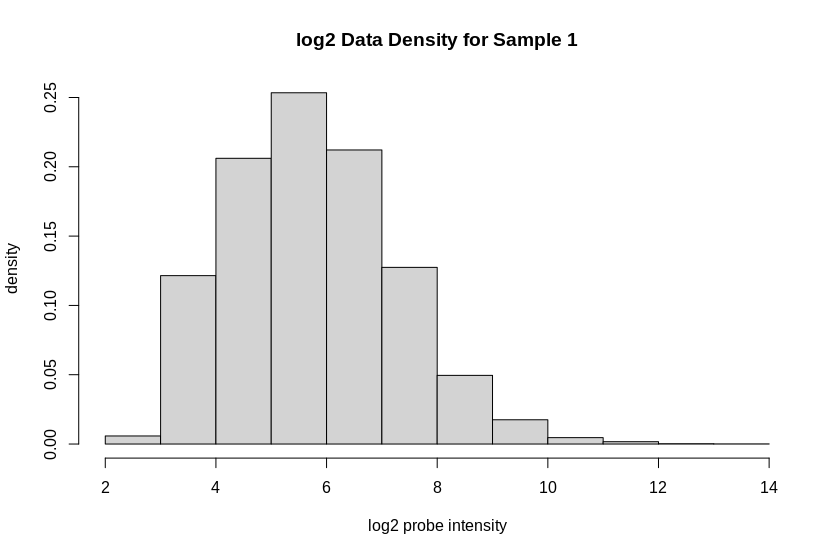
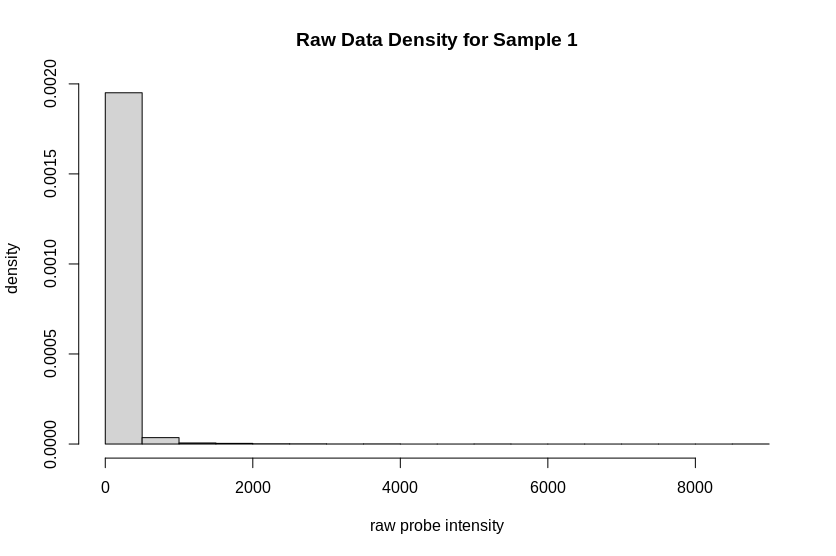
# grab Diag column and convert character to factor

mddPheno <- as.factor(phenos.df$Diag) # this is our phenotype/class vector

summary(mddPheno) # MDD -- major depressive disorder, HC -- healthy control

**B**. Normalization. The data has already been normalized by counts per million reads, but we would like to explore the data to determine whether additional normalization and transformations of the data are needed. The log2 transformation is meant to make the distribution of expression levels symmetric within a subject. Use the code below to visualize the distribution of probe intensities for all samples. The boxplots show expression of all genes for each sample, where samples are arranged across the x-axis. The histograms plot the density of expresson levels for one sample.

**3.** Show your plots and discuss the differences between the original data (raw cpm) and data that has been log2 transformed.



The immediate visible difference is that the raw data’s values are mostly small and all with about the same value except for a number of outliers with massively larger values. Without any kind of normalization this makes the data hard to read and not very useful. Normalizing the data makes it much more readable and compacts it onto a smaller interval. Now the distribution looks much more “normal,” so we can use more tools on it more easily.

### raw cpm boxplots and histogram of one sample

boxplot(sense.filtered.cpm,range=0,ylab="raw probe intensity", main="Raw", names=mddPheno)

hist(sense.filtered.cpm[,1], freq=F, ylab="density", xlab="raw probe intensity", main="Raw Data Density for Sample 1")

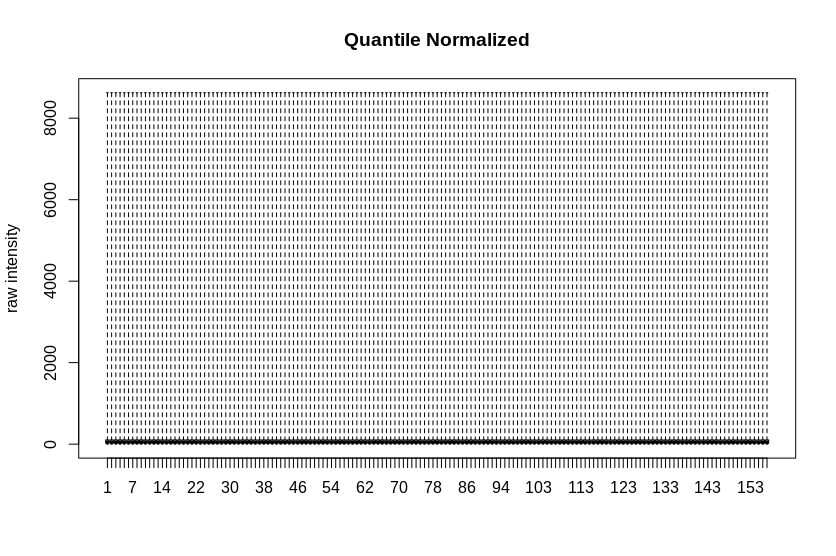
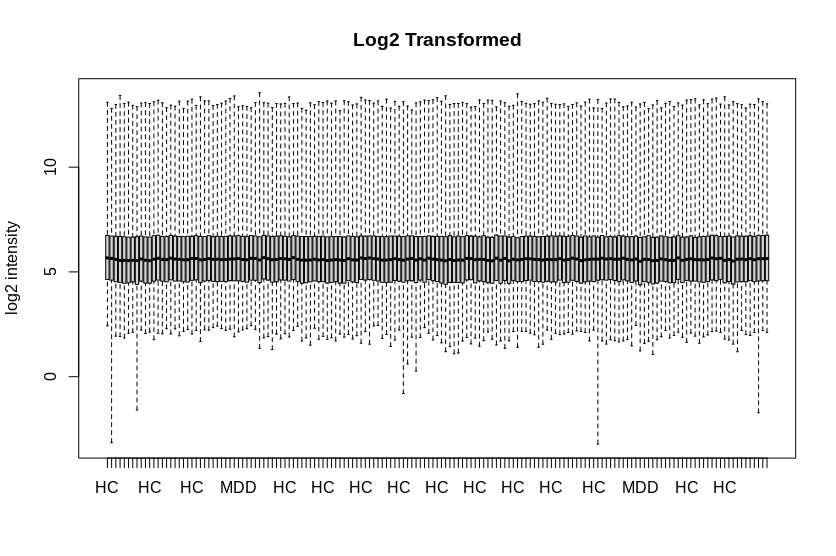
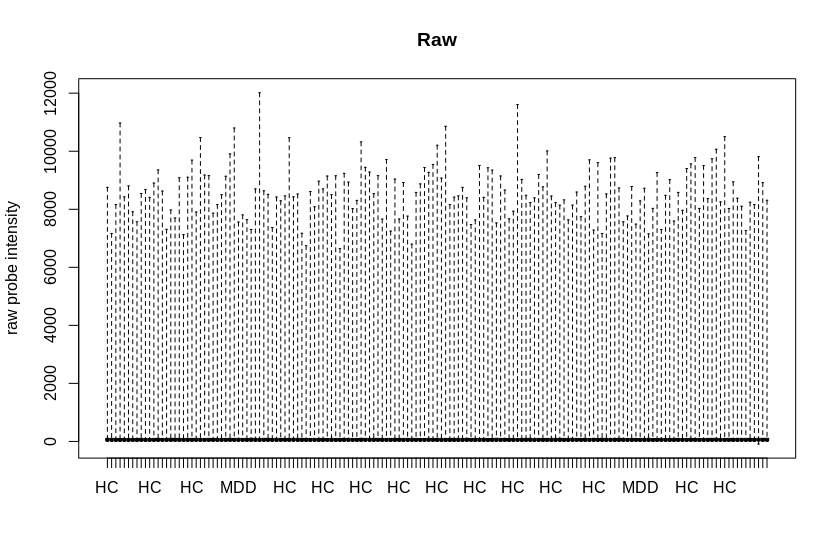
### log2 transformed boxplots and histogram

boxplot(log2(sense.filtered.cpm), range=0,ylab="log2 intensity", main="Log2 Transformed", names=mddPheno)

hist(log2(sense.filtered.cpm[,1]), freq=F, ylab="density", xlab="log2 probe intensity", main="log2 Data Density for Sample 1")

Run the code below to perform quantile normalization of the data.

**4.** How does quantile normalization affect each sample’s distribution of gene expression in comparison to the other samples in the boxplots (show the boxplots)?



The interquartile range of the quantile normalized box plot is really small compared to the log2 transform, but the max and min values are the same for each sample. Having a definite range can allow the use of certain tools, but also can muddy up the data. Any type of transform on the data can have benefits but it also can reduce the amount of information. The quantile transformation reduces the amount of information available by forcing everything to have the same values, but with reduction in information, we can make certain assumptions about the data which can prove helpful. The distribution is the largest compared to the other methods but it is also the most uniform across samples.

# install quantile normalize

#install.packages("BiocManager")

library(BiocManager)

BiocManager::install("preprocessCore")

library(preprocessCore)

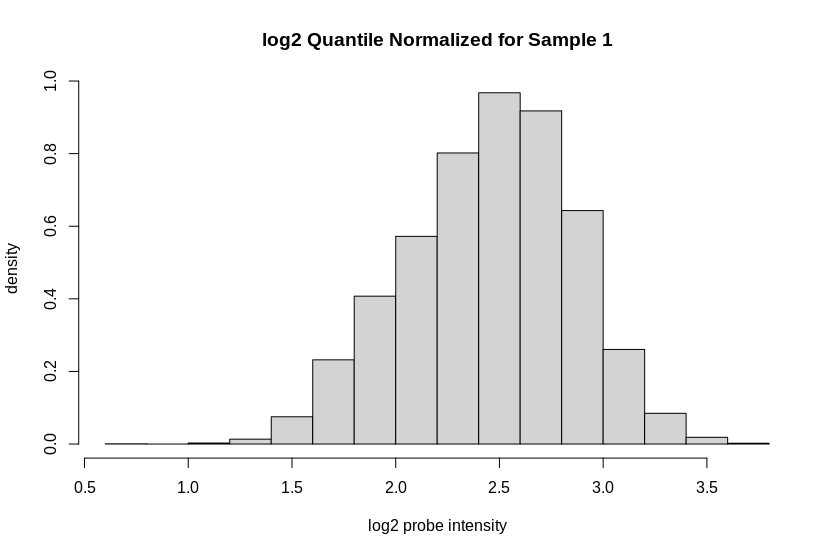
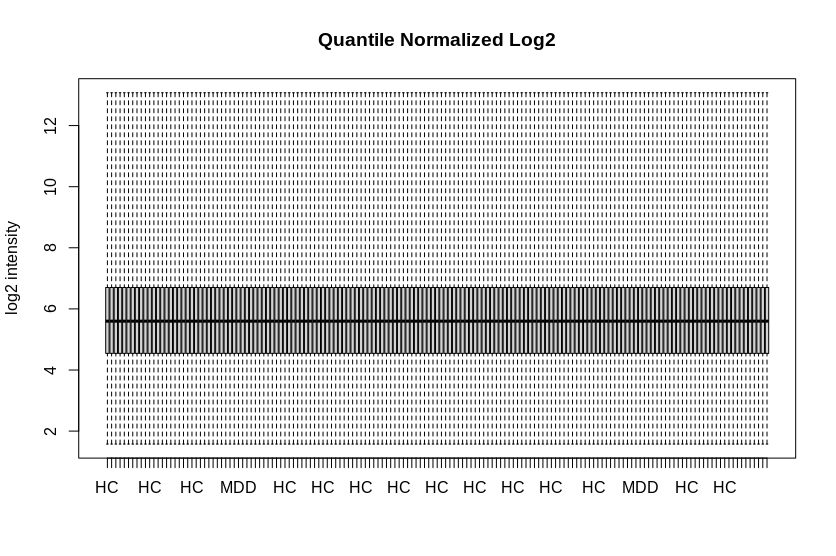
# apply quantile normalization

mddExprData\_quantile <- normalize.quantiles(sense.filtered.cpm)

boxplot(mddExprData\_quantile,range=0,ylab="raw intensity", main="Quantile Normalized")

Now let’s apply the log2 transformation to the normalized data to make the normalized data more symmetric.

**5**. Show the plots of the final processed data.



The log2 and quantized transformations gives us the benefit of both transformations individually. We get the larger interquartile range of the log2 transform - making the distribution look normal - but we also get the standardized values of the quantile transformation.

mddExprData\_quantileLog2 <- log2(mddExprData\_quantile)

# add phenotype names to matrix

colnames(mddExprData\_quantileLog2) <- mddPheno

boxplot(mddExprData\_quantileLog2,range=0,ylab="log2 intensity", main="Quantile Normalized Log2")

hist(log2(mddExprData\_quantileLog2[,1]), freq=F, ylab="density", xlab="log2 probe intensity", main="log2 Quantile Normalized for Sample 1")

**6.** Based on quantile normalization, why is the output from the following commands expected?

because the quantile normalization forces the same values to each sample so with the same values in each column we should have the same mean and distribution.

mean(mddExprData\_quantileLog2[,1])

colMeans(mddExprData\_quantileLog2)

**C**. Exploratory clustering of microarray data. Below we run multi-dimensional scaling (MDS) and hierarchical clustering to get an idea of how the samples are related based on the similarity of their gene expression. Later we will swap dimensions to see how the genes are related. In addition to clustering samples, the first two dimensions of MDS explain the most variation in the data. Ideally this variation in the expression data is due to biological differences between the samples.

**7.** Run the code below and show plots. What is the size of the mdd.mds and what does it represent?

2 columns of length 157 each. it is a dimensional reduction of the data set we read in earlier.

**8.** How are the MDD/HC samples distributed/mixed in the mds space? Are there any unusual samples?

In the graph there is one HC sample (HC.70) that has a relatively massive value (an order of magnitude larger than the rest) in the first dimension. HC also has a couple more points that are on the edge of being outliers in the second dimension, but do not seem to warrant removal without some other reason.

# transpose data matrix and convert to data frame

# ggplot wants data frame and subjects as rows

expr\_SxG <- data.frame(t(mddExprData\_quantileLog2)) # Subject x Gene

colnames(expr\_SxG) <- rownames(sense.filtered.cpm) # add gene names

## MDS of subjects

#d<-dist(expr\_SxG) # Euclidean metric

mddCorr<-cor(t(expr\_SxG)) # distance based on correlation

d <- 1-mddCorr

mdd.mds <- cmdscale(d, k = 2)

x <- mdd.mds[,1]

y <- mdd.mds[,2]

mdd.mds.df <- data.frame(mdd.mds)

colnames(mdd.mds.df) <- c("dim1","dim2")

#install.packages("ggplot2") # if not already installed

#BiocManager::install("ggplot2")("ggplot2")

library(ggplot2)

p <- ggplot() # initialize empty ggplot object

p <- p + geom\_point(data=mdd.mds.df, aes(x=dim1, y=dim2, color=mddPheno, shape=mddPheno), size=3)

p <- p + ggtitle("MDS") + xlab("Dim 1") + ylab("Dim 2")

print(p)

Below, use the distance matrix to create a dendrogram to visualize the relationships between samples. Show the plot.

**9.** How are the relationships between samples that you observed in the mds plot reflected in the tree? In other words, how are the MDD/HC samples distributed in the tree (the table command might also help you interpret the distribution) and are there any unusual samples in the tree?

The one outlier we spotted before is all alone :( and does not merge until the end. Other than that there appears to be 4 clusters, or 3 if the definition of a cluster is harsher, but that is much harder to observe in the plot.

## hierarchical cluster of subjects

mddTree = hclust(as.dist(d))

mddTree$labels <- mddPheno

plot(mddTree)

num.clust <- 5

mddCuts <- cutree(mddTree,k=num.clust)

table(names(mddCuts),mddCuts)

#install.packages("dendextend")

library(dendextend)

dend <- as.dendrogram(mddTree)

dend=color\_labels(dend, k=num.clust)

#dend <- dend %>% color\_branches(k = 4)

plot(dend) # selective coloring of branches AND labels

Use the code below to identify clusters using the dynamic cutree from WGCNA.

#library(BiocManager)

#BiocManager::install("WGCNA")

library(WGCNA)

# Plot the dendrogram and colors underneath

sizeGrWindow(8,6)

dynamicMods = cutreeDynamic(dendro = mddTree, distM = d,

deepSplit = 2, pamRespectsDendro = FALSE,

minClusterSize = 2)

mddColors = labels2colors(dynamicMods)

table(mddColors)

table(mddColors,names(mddCuts))

plotDendroAndColors(mddTree, mddColors, "Dynamic Clusters",

dendroLabels = NULL, # hang = -1,

addGuide = TRUE, #guideHang = 0.05,

main = "Clustering with WGCNA")

The second table above shows how often a dynamic cluster (color) co-occurs with a phenotype label (MDD or HC).

**10.** Show the plot and second table. Which clusters (colors) are skewed towards one phenotype or the other? Which clusters are most evenly mixed?

mddColors HC MDD

black 4 5

blue 14 19

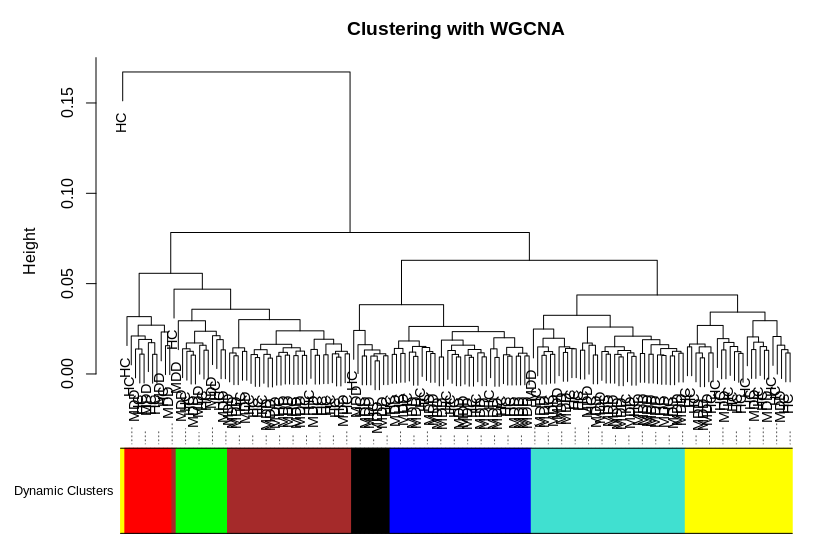
brown 17 13

green 6 6

red 7 4

turquoise 12 24

yellow 19 7



black, green are clearly even but are of small sizes so it is hard to definitively say if the trend will always continue that way.

blue, brown are not dominated by one phenotype but are slightly skewed toward MDD and HC respectively

red is even more strongly skewed but is such a small sample it is hard to say it is definitively skewed toward HC or luck just threw it that way.

yellow, turquoise are clearly skewed toward HC and MDD respectively.

because of the small sample size of some of the clusters, like red and black, it is hard to say whether the differences are significant or not. More analysis should be done before conclusions are drawn on all the data.

Note: We hope that the data shows some genes that are different between MDD/HC groups (differentially expressed), which we will investigate in the next lab. However, when clustering MDD/HC groups based on all of the genes, we expect the effect of a few differentially expressed genes to be washed out (hidden) among the many genes that are not related to differences between MDD/HC.