Lab4 732A51 Bioinformatics Group 9

Duc Duong, Martin Smelik, Raymond Sseguya

8 December 2018

## Question 1

In overall, the code provided will analysis of gene expression data from HUVEC1 and Ocular Vascular Endothelial2 Cells.  
The first step is to download the data with the code GSE20986 using getGEOsuppFiles function. Then untar, unzip it to data folder. A data frame called phenodata is created to hold the metadata of the data. It’s also written to a file with the same name.

library(GEOquery)  
#The data folder should be empty  
  
x = getGEOSuppFiles("GSE20986")  
x

## size  
## C:/Users/smelo/Documents/Skola/bioinformatics/Bioinformatics\_Labs/Lab 4/GSE20986/GSE20986\_RAW.tar 56360960  
## isdir  
## C:/Users/smelo/Documents/Skola/bioinformatics/Bioinformatics\_Labs/Lab 4/GSE20986/GSE20986\_RAW.tar FALSE  
## mode  
## C:/Users/smelo/Documents/Skola/bioinformatics/Bioinformatics\_Labs/Lab 4/GSE20986/GSE20986\_RAW.tar 666  
## mtime  
## C:/Users/smelo/Documents/Skola/bioinformatics/Bioinformatics\_Labs/Lab 4/GSE20986/GSE20986\_RAW.tar 2018-12-11 22:14:29  
## ctime  
## C:/Users/smelo/Documents/Skola/bioinformatics/Bioinformatics\_Labs/Lab 4/GSE20986/GSE20986\_RAW.tar 2018-12-11 22:13:27  
## atime  
## C:/Users/smelo/Documents/Skola/bioinformatics/Bioinformatics\_Labs/Lab 4/GSE20986/GSE20986\_RAW.tar 2018-12-11 22:13:27  
## exe  
## C:/Users/smelo/Documents/Skola/bioinformatics/Bioinformatics\_Labs/Lab 4/GSE20986/GSE20986\_RAW.tar no

untar("GSE20986/GSE20986\_RAW.tar", exdir = "data")  
cels = list.files("data/", pattern = "[gz]")  
sapply(paste("data", cels, sep = "/"), gunzip)

## data/GSM524662.CEL.gz data/GSM524663.CEL.gz data/GSM524664.CEL.gz   
## 13555726 13555055 13555639   
## data/GSM524665.CEL.gz data/GSM524666.CEL.gz data/GSM524667.CEL.gz   
## 13560122 13555663 13557614   
## data/GSM524668.CEL.gz data/GSM524669.CEL.gz data/GSM524670.CEL.gz   
## 13556090 13560054 13555971   
## data/GSM524671.CEL.gz data/GSM524672.CEL.gz data/GSM524673.CEL.gz   
## 13554926 13555042 13555290

phenodata = matrix(rep(list.files("data"), 2), ncol =2)  
class(phenodata)

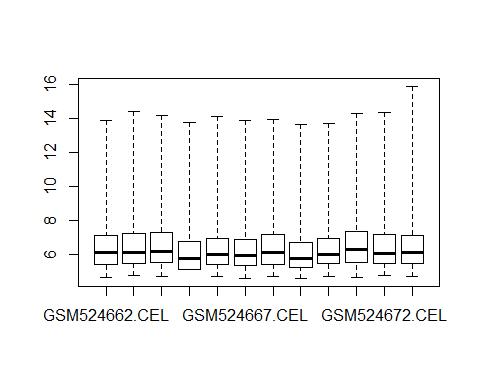
## [1] "matrix"

phenodata <- as.data.frame(phenodata)  
colnames(phenodata) <- c("Name", "FileName")  
phenodata$Targets <- c("iris",   
 "retina",   
 "retina",   
 "iris",   
 "retina",   
 "iris",   
 "choroid",   
 "choroid",   
 "choroid",   
 "huvec",   
 "huvec",   
 "huvec")  
#Write the list of downloaded content to a file  
write.table(phenodata, "data/phenodata.txt", quote = F, sep = "\t", row.names = F)

The, they use the read.affy function to read the data and stored it in an object called celfiles. The boxplot function will display the microarray distributions. The values in boxplots are the log base 2 intensities of both pm and mm probes.

library(simpleaffy)  
#Using read.affy function to read..   
celfiles <- read.affy(covdesc = "phenodata.txt", path = "data")  
boxplot(celfiles)

##

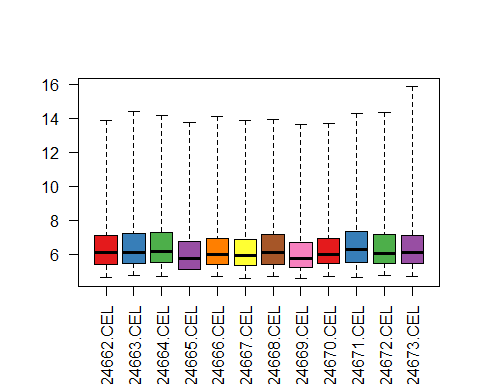


The second boxplot is still the same. But, it is couloured and the labels are made verticaled for easier reading.

library(RColorBrewer)  
cols = brewer.pal(8, "Set1")  
eset <- exprs(celfiles)  
samples <- celfiles$Targets  
colnames(eset)

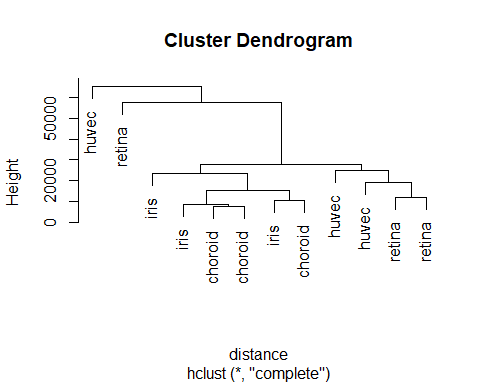
## [1] "GSM524662.CEL" "GSM524663.CEL" "GSM524664.CEL" "GSM524665.CEL"  
## [5] "GSM524666.CEL" "GSM524667.CEL" "GSM524668.CEL" "GSM524669.CEL"  
## [9] "GSM524670.CEL" "GSM524671.CEL" "GSM524672.CEL" "GSM524673.CEL"

colnames(eset) <- samples  
boxplot(celfiles, col = cols, las = 2) #las=2 make the axis labels horizontal



In the next step, they use dist function to calculate the distance of the data from 12 samples. Then, use hclust function to analysis hierarchical clusters and then plot it as a cluster dendrogram.

distance <- dist(t(eset), method = "maximum")  
clusters <- hclust(distance)  
plot(clusters)

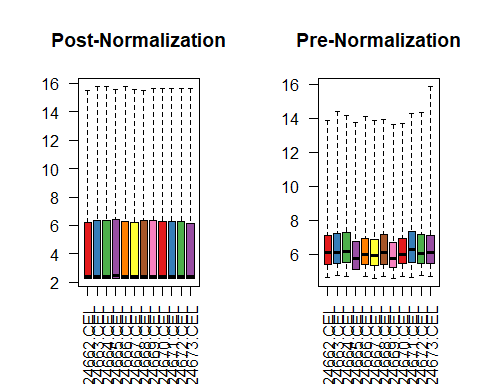


The below block will convert celfiles objects (AffyBatch type) into an ExpressionSet though gcrma function. This function will use the robust multi-array average (RMA) expression measure with help of probe sequence. When converting, the data is being normalized. Two boxplots show the data before and after normalized is drawn to compare.

require(simpleaffy)  
require(affyPLM)  
celfiles.gcrma = gcrma(celfiles)

## Adjusting for optical effect............Done.  
## Computing affinities.Done.  
## Adjusting for non-specific binding............Done.  
## Normalizing  
## Calculating Expression

par(mfrow=c(1,2))  
boxplot(celfiles.gcrma, col = cols, las = 2, main = "Post-Normalization")   
boxplot(celfiles, col = cols, las = 2, main = "Pre-Normalization")



And then, they draw the cluster dndrogram of the normilizated data.

dev.off()

## null device   
## 1

distance2 <- dist(t(exprs(celfiles.gcrma)), method = "maximum")  
clusters2 <- hclust(distance2)  
plot(clusters2)

In the next step, a matrix call design is created. It contains the name of the names of genes and which samples it belongs to. A contrast matrix is also created by the makeContrasts function. It includes three pairs of having versus the others.

library(limma)  
phenodata

## Name FileName Targets  
## 1 GSM524662.CEL GSM524662.CEL iris  
## 2 GSM524663.CEL GSM524663.CEL retina  
## 3 GSM524664.CEL GSM524664.CEL retina  
## 4 GSM524665.CEL GSM524665.CEL iris  
## 5 GSM524666.CEL GSM524666.CEL retina  
## 6 GSM524667.CEL GSM524667.CEL iris  
## 7 GSM524668.CEL GSM524668.CEL choroid  
## 8 GSM524669.CEL GSM524669.CEL choroid  
## 9 GSM524670.CEL GSM524670.CEL choroid  
## 10 GSM524671.CEL GSM524671.CEL huvec  
## 11 GSM524672.CEL GSM524672.CEL huvec  
## 12 GSM524673.CEL GSM524673.CEL huvec

samples <- as.factor(samples)  
design <- model.matrix(~0+samples)  
colnames(design)

## [1] "sampleschoroid" "sampleshuvec" "samplesiris" "samplesretina"

colnames(design) <- c("choroid", "huvec", "iris", "retina")  
design

## choroid huvec iris retina  
## 1 0 0 1 0  
## 2 0 0 0 1  
## 3 0 0 0 1  
## 4 0 0 1 0  
## 5 0 0 0 1  
## 6 0 0 1 0  
## 7 1 0 0 0  
## 8 1 0 0 0  
## 9 1 0 0 0  
## 10 0 1 0 0  
## 11 0 1 0 0  
## 12 0 1 0 0  
## attr(,"assign")  
## [1] 1 1 1 1  
## attr(,"contrasts")  
## attr(,"contrasts")$samples  
## [1] "contr.treatment"

contrast.matrix = makeContrasts(  
 huvec\_choroid = huvec - choroid,   
 huvec\_retina = huvec - retina,   
 huvec\_iris = huvec - iris,   
 levels = design)

In this step. They use the design matrix to fit the linear model celfiles.gcrma expressionSet created before by using the LMFit function. The result called fit is used in contrasts.fit function with the contrast matrix. They continue with extracting some t value, F value.. by the eBayes function.

fit = lmFit(celfiles.gcrma, design)  
huvec\_fit <- contrasts.fit(fit, contrast.matrix)  
huvec\_ebay <- eBayes(huvec\_fit)

In the next step, the topTable function with number = 100000 will extract the top-ranked genes from the result before. getSYMBOL function is called to map that 100000 genes with the hgu133plus2. The final result is printed below.

library(hgu133plus2.db)  
library(annotate)  
  
probenames.list <- rownames(topTable(huvec\_ebay, number = 100000))  
getsymbols <- getSYMBOL(probenames.list, "hgu133plus2")  
results <- topTable(huvec\_ebay, number = 100000, coef = "huvec\_choroid")  
results <- cbind(results, getsymbols)  
summary(results)

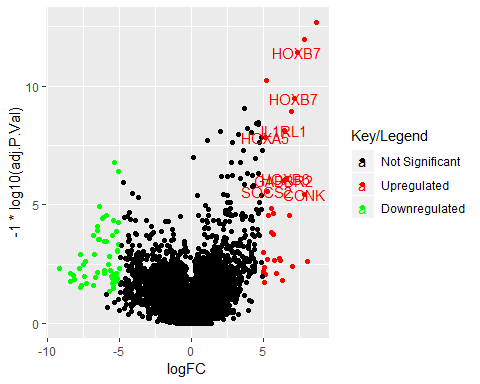
## logFC AveExpr t P.Value   
## Min. :-9.19111 Min. : 2.279 Min. :-39.77473 Min. :0.0000   
## 1st Qu.:-0.05967 1st Qu.: 2.281 1st Qu.: -0.70649 1st Qu.:0.1523   
## Median : 0.00000 Median : 2.480 Median : 0.00000 Median :0.5079   
## Mean :-0.02353 Mean : 4.375 Mean : 0.07441 Mean :0.5346   
## 3rd Qu.: 0.03986 3rd Qu.: 6.241 3rd Qu.: 0.67455 3rd Qu.:1.0000   
## Max. : 8.67086 Max. :15.541 Max. :296.84201 Max. :1.0000   
##   
## adj.P.Val B getsymbols   
## Min. :0.0000 Min. :-7.710 YME1L1 : 22   
## 1st Qu.:0.6036 1st Qu.:-7.710 HFE : 15   
## Median :1.0000 Median :-7.451 CFLAR : 14   
## Mean :0.7436 Mean :-6.582 NRP2 : 14   
## 3rd Qu.:1.0000 3rd Qu.:-6.498 ARHGEF12: 13   
## Max. :1.0000 Max. :21.290 (Other) :41857   
## NA's :12740

The results are grouped into three groups. Group 3 includes genes that adj.P.Val < 0.05 and logFC < -5. Group 2 contains gene that adj.P.Val < 0.05 and logFC > 5, and the rest is group 1. Number of gene in each groups is printed. Data in group 1 means Not Significant, group 2 means “Upregulated” and group 3 means “Downregulated”. A scatter plot is draw, in which x = logFC and y = -1\*log10(adj.P.Val)

results$threshold <- "1"  
a <- subset(results, adj.P.Val < 0.05 & logFC > 5)  
results[rownames(a), "threshold"] <- "2"  
b <- subset(results, adj.P.Val < 0.05 & logFC < -5)  
results[rownames(b), "threshold"] <- "3"  
table(results$threshold)

##   
## 1 2 3   
## 54587 33 55

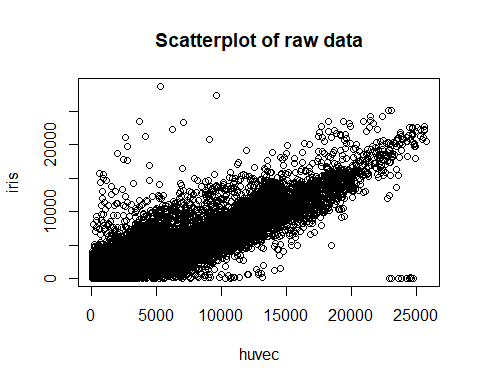
library(ggplot2)  
volcano <- ggplot(data = results,   
 aes(x = logFC, y = -1\*log10(adj.P.Val),   
 colour = threshold,   
 label = getsymbols))  
  
volcano <- volcano +   
 geom\_point() +   
 scale\_color\_manual(values = c("black", "red", "green"),   
 labels = c("Not Significant", "Upregulated", "Downregulated"),   
 name = "Key/Legend")  
  
volcano +   
 geom\_text(data = subset(results, logFC > 5 & -1\*log10(adj.P.Val) > 5), aes(x = logFC, y = -1\*log10(adj.P.Val), colour = threshold, label = getsymbols) )



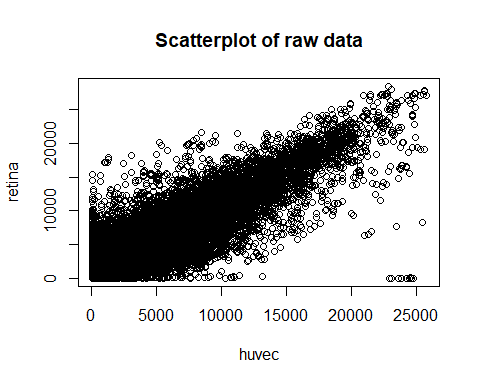
## Question2

The three constrast are: + huvec - choroid, + huvec - retina, + huvec - iris We will choose the first sample of each type to make analysis. Here is the plots of raw data.

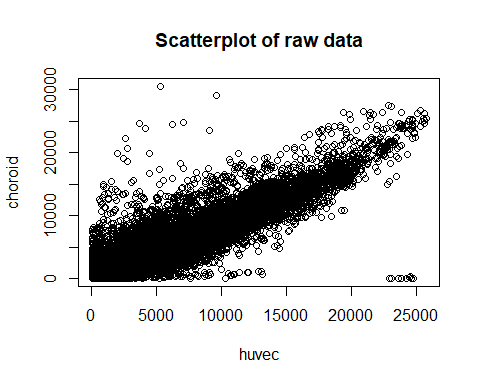
iris <- eset[,1]  
retina <- eset[,2]  
choroid <- eset[,7]  
huvec <- eset[,10]  
  
plot(x=huvec ,y=iris,xlab="huvec",ylab="iris", main="Scatterplot of raw data")



plot(x=huvec ,y=retina,xlab="huvec",ylab="retina", main="Scatterplot of raw data")

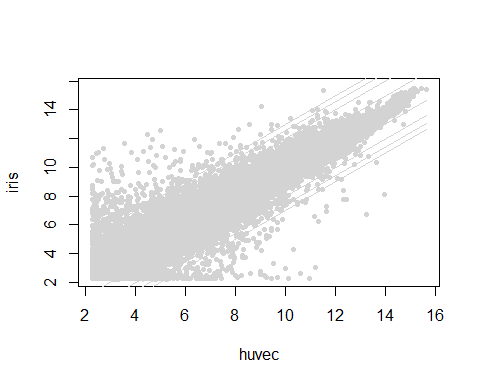


plot(x=huvec ,y=choroid,xlab="huvec",ylab="choroid", main="Scatterplot of raw data")

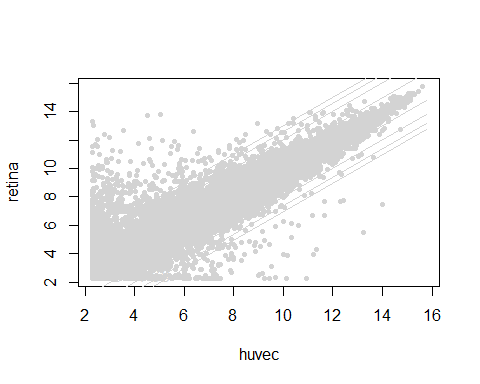


And here, for the normalized data:

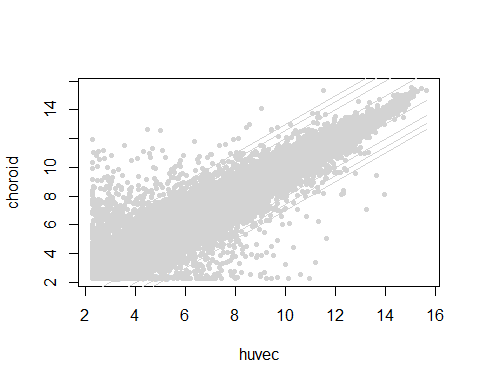
huvec\_iris <- pairwise.comparison(celfiles.gcrma,"Targets",c("huvec","iris"))  
huvec\_retina <-pairwise.comparison(celfiles.gcrma,"Targets",c("huvec","retina"))  
huvec\_choroid <-pairwise.comparison(celfiles.gcrma,"Targets",c("huvec","choroid"))  
plot(huvec\_iris)



plot(huvec\_retina)

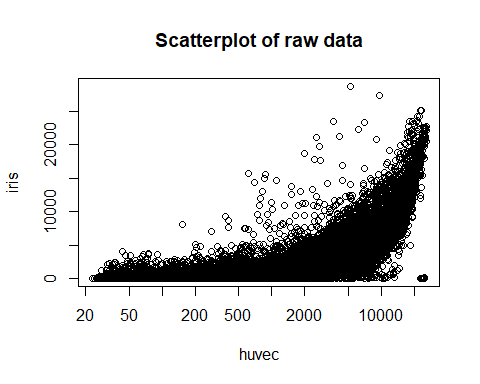


plot(huvec\_choroid)

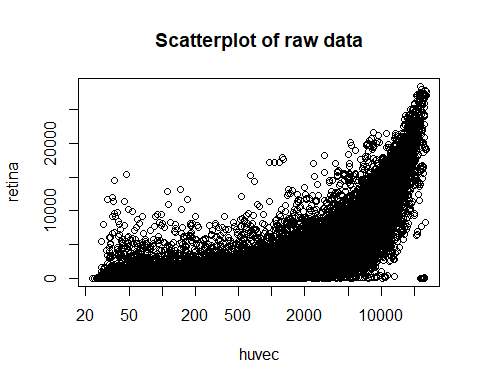


And here is log-scaled graph

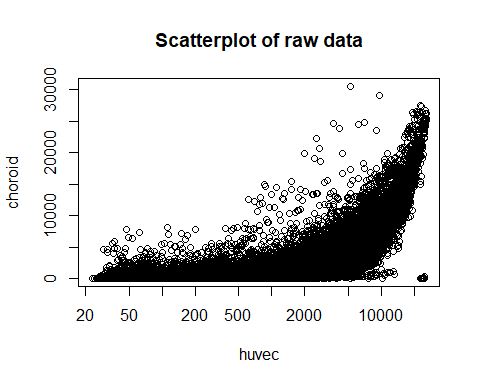
plot(x=huvec ,y=iris,xlab="huvec",ylab="iris", main="Scatterplot of raw data",log=c('x','y'))



plot(x=huvec ,y=retina,xlab="huvec",ylab="retina", main="Scatterplot of raw data",log=c('x','y'))

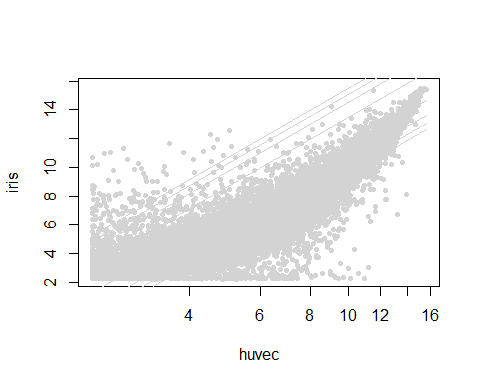


plot(x=huvec ,y=choroid,xlab="huvec",ylab="choroid", main="Scatterplot of raw data",log=c('x','y'))

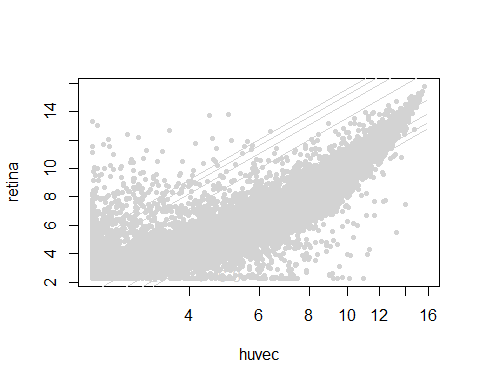


Here is log-scaled graph tirh normalized data

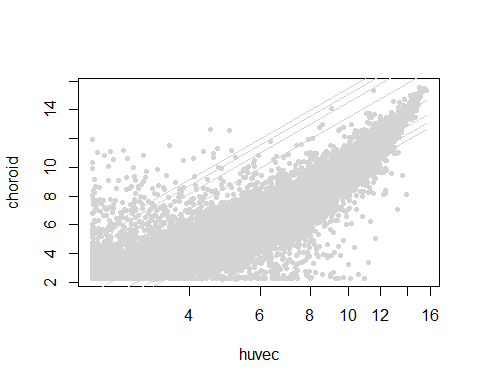
huvec\_iris <- pairwise.comparison(celfiles.gcrma,"Targets",c("huvec","iris"))  
huvec\_retina <-pairwise.comparison(celfiles.gcrma,"Targets",c("huvec","retina"))  
huvec\_choroid <-pairwise.comparison(celfiles.gcrma,"Targets",c("huvec","choroid"))  
plot(huvec\_iris,log=c('x','y'))



plot(huvec\_retina,log=c('x','y'))

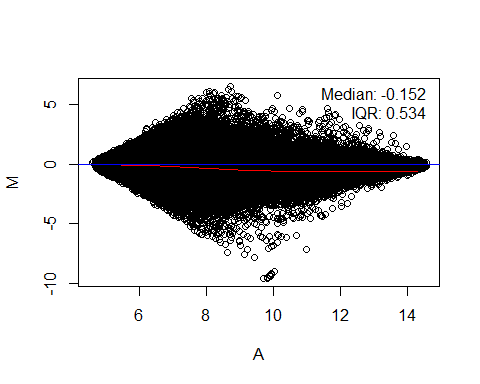


plot(huvec\_choroid,log=c('x','y'))

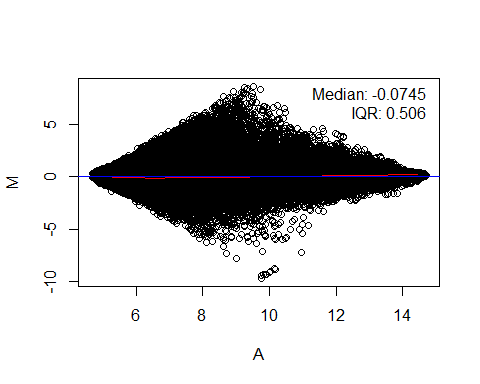


Here is MA plot

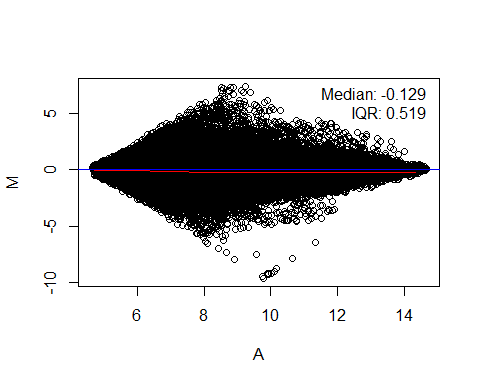
iris\_huvec <- eset[,c(1,10)]  
retina\_huvec <- eset[,c(2,10)]  
chronoid\_huvec <- eset[,c(7,10)]  
library(affy)  
#inspired by wiki  
  
ma.plot( rowMeans(log2(iris\_huvec)), log2(iris\_huvec[, 1])-log2(iris\_huvec[, 2]), cex=1 )



ma.plot( rowMeans(log2(retina\_huvec)), log2(retina\_huvec[, 1])-log2(retina\_huvec[, 2]), cex=1 )

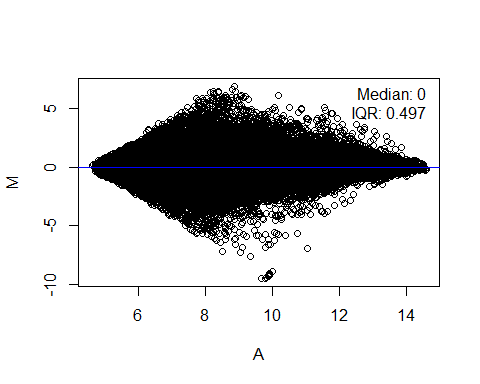


ma.plot( rowMeans(log2(chronoid\_huvec)), log2(chronoid\_huvec[, 1])-log2(chronoid\_huvec[, 2]), cex=1 )

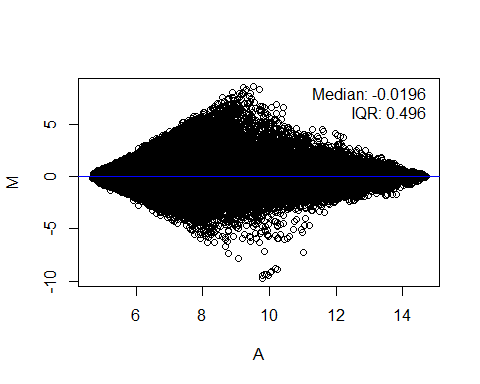


And MA plot with normalized data

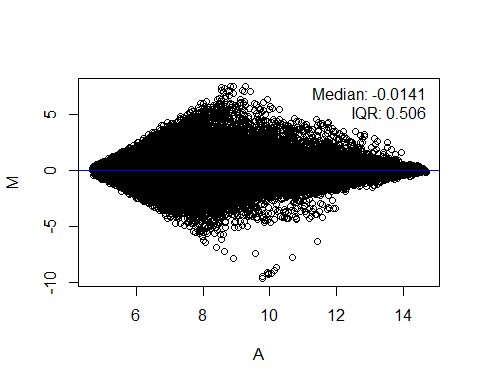
library(preprocessCore)  
  
#do a quantile normalization  
norm\_iris\_huvec <- normalize.quantiles(iris\_huvec)  
norm\_retina\_huvec <- normalize.quantiles(retina\_huvec)  
norm\_chronoid\_huvec <- normalize.quantiles(chronoid\_huvec)  
  
  
  
##normalized  
ma.plot( rowMeans(log2(norm\_iris\_huvec)), log2(norm\_iris\_huvec[, 1])-log2(norm\_iris\_huvec[, 2]), cex=1 )



ma.plot( rowMeans(log2(norm\_retina\_huvec)), log2(norm\_retina\_huvec[, 1])-log2(norm\_retina\_huvec[, 2]), cex=1 )

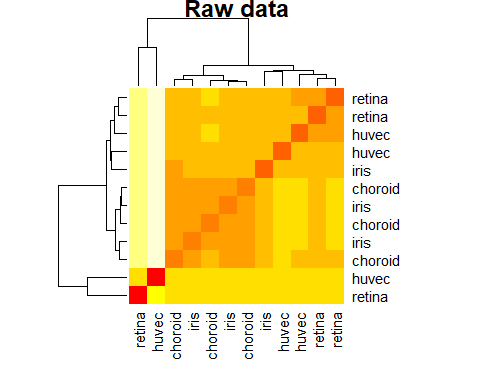


ma.plot( rowMeans(log2(norm\_chronoid\_huvec)), log2(norm\_chronoid\_huvec[, 1])-log2(norm\_chronoid\_huvec[, 2]), cex=1 )

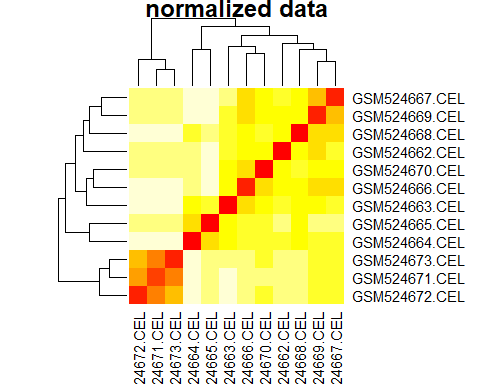


And here is the heat map:

par(mfrow=c(1,2))  
heatmap(as.matrix(distance), main = "Raw data")



heatmap(as.matrix(distance2), main = "normalized data")



## Question 3

Volcano plots:

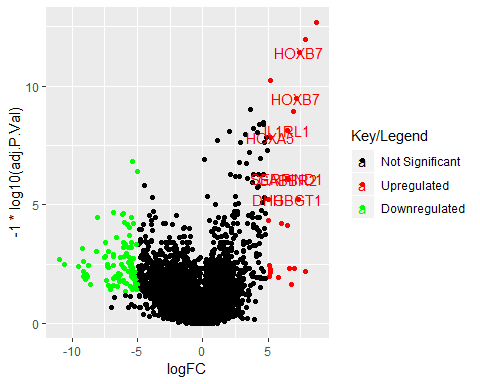
results <- topTable(huvec\_ebay, number = 100000, coef = "huvec\_retina")  
results <- cbind(results, getsymbols)  
summary(results)

## logFC AveExpr t P.Value   
## Min. :-10.97621 Min. : 2.279 Min. :-40.23456 Min. :0.0000   
## 1st Qu.: -0.05050 1st Qu.: 2.281 1st Qu.: -0.60778 1st Qu.:0.1564   
## Median : 0.00000 Median : 2.480 Median : 0.00000 Median :0.5238   
## Mean : -0.03205 Mean : 4.375 Mean : 0.08621 Mean :0.5414   
## 3rd Qu.: 0.04602 3rd Qu.: 6.241 3rd Qu.: 0.72396 3rd Qu.:1.0000   
## Max. : 8.67086 Max. :15.541 Max. :296.84201 Max. :1.0000   
##   
## adj.P.Val B getsymbols   
## Min. :0.0000 Min. :-7.710 YME1L1 : 22   
## 1st Qu.:0.6128 1st Qu.:-7.710 HFE : 15   
## Median :1.0000 Median :-7.469 CFLAR : 14   
## Mean :0.7525 Mean :-6.614 NRP2 : 14   
## 3rd Qu.:1.0000 3rd Qu.:-6.522 ARHGEF12: 13   
## Max. :1.0000 Max. :21.290 (Other) :41857   
## NA's :12740

results$threshold <- "1"  
a <- subset(results, adj.P.Val < 0.05 & logFC > 5)  
results[rownames(a), "threshold"] <- "2"  
b <- subset(results, adj.P.Val < 0.05 & logFC < -5)  
results[rownames(b), "threshold"] <- "3"  
table(results$threshold)

##   
## 1 2 3   
## 54557 24 94

library(ggplot2)  
volcano <- ggplot(data = results,   
 aes(x = logFC, y = -1\*log10(adj.P.Val),   
 colour = threshold,   
 label = getsymbols))  
  
volcano <- volcano +   
 geom\_point() +   
 scale\_color\_manual(values = c("black", "red", "green"),   
 labels = c("Not Significant", "Upregulated", "Downregulated"),   
 name = "Key/Legend")  
  
volcano +   
 geom\_text(data = subset(results, logFC > 5 & -1\*log10(adj.P.Val) > 5), aes(x = logFC, y = -1\*log10(adj.P.Val), colour = threshold, label = getsymbols) )



results <- topTable(huvec\_ebay, number = 100000, coef = "huvec\_iris")  
results <- cbind(results, getsymbols)  
summary(results)

## logFC AveExpr t P.Value   
## Min. :-8.26243 Min. : 2.279 Min. :-42.52934 Min. :0.0000   
## 1st Qu.:-0.08709 1st Qu.: 2.281 1st Qu.: -1.14547 1st Qu.:0.1252   
## Median : 0.00000 Median : 2.480 Median : 0.00000 Median :0.3678   
## Mean :-0.02251 Mean : 4.375 Mean : -0.00841 Mean :0.4888   
## 3rd Qu.: 0.03905 3rd Qu.: 6.241 3rd Qu.: 0.66164 3rd Qu.:1.0000   
## Max. : 8.67086 Max. :15.541 Max. :296.84201 Max. :1.0000   
##   
## adj.P.Val B getsymbols   
## Min. :0.0000 Min. :-7.710 YME1L1 : 22   
## 1st Qu.:0.5008 1st Qu.:-7.710 HFE : 15   
## Median :0.7355 Median :-7.230 CFLAR : 14   
## Mean :0.6798 Mean :-6.440 NRP2 : 14   
## 3rd Qu.:1.0000 3rd Qu.:-6.319 ARHGEF12: 13   
## Max. :1.0000 Max. :21.290 (Other) :41857   
## NA's :12740

results$threshold <- "1"  
a <- subset(results, adj.P.Val < 0.05 & logFC > 5)  
results[rownames(a), "threshold"] <- "2"  
b <- subset(results, adj.P.Val < 0.05 & logFC < -5)  
results[rownames(b), "threshold"] <- "3"  
table(results$threshold)

##   
## 1 2 3   
## 54601 25 49

library(ggplot2)  
volcano <- ggplot(data = results,   
 aes(x = logFC, y = -1\*log10(adj.P.Val),   
 colour = threshold,   
 label = getsymbols))  
  
volcano <- volcano +   
 geom\_point() +   
 scale\_color\_manual(values = c("black", "red", "green"),   
 labels = c("Not Significant", "Upregulated", "Downregulated"),   
 name = "Key/Legend")  
  
volcano +   
 geom\_text(data = subset(results, logFC > 5 & -1\*log10(adj.P.Val) > 5), aes(x = logFC, y = -1\*log10(adj.P.Val), colour = threshold, label = getsymbols) )

