ADCC Analysis

Elana J Fertig

July 13, 2015

## R packages

library('simpleaffy')

## Loading required package: BiocGenerics  
## Loading required package: parallel  
##   
## Attaching package: 'BiocGenerics'  
##   
## The following objects are masked from 'package:parallel':  
##   
## clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,  
## clusterExport, clusterMap, parApply, parCapply, parLapply,  
## parLapplyLB, parRapply, parSapply, parSapplyLB  
##   
## The following object is masked from 'package:stats':  
##   
## xtabs  
##   
## The following objects are masked from 'package:base':  
##   
## anyDuplicated, append, as.data.frame, as.vector, cbind,  
## colnames, do.call, duplicated, eval, evalq, Filter, Find, get,  
## intersect, is.unsorted, lapply, Map, mapply, match, mget,  
## order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,  
## rbind, Reduce, rep.int, rownames, sapply, setdiff, sort,  
## table, tapply, union, unique, unlist  
##   
## Loading required package: Biobase  
## Welcome to Bioconductor  
##   
## Vignettes contain introductory material; view with  
## 'browseVignettes()'. To cite Bioconductor, see  
## 'citation("Biobase")', and for packages 'citation("pkgname")'.  
##   
## Loading required package: affy  
## Loading required package: genefilter  
##   
## Attaching package: 'genefilter'  
##   
## The following object is masked from 'package:base':  
##   
## anyNA  
##   
## Loading required package: gcrma

library('lumi')

## Warning: replacing previous import by 'graphics::image' when loading  
## 'methylumi'

##   
## Attaching package: 'lumi'  
##   
## The following objects are masked from 'package:affy':  
##   
## MAplot, plotDensity

library('gplots')

## Warning: package 'gplots' was built under R version 3.1.3

##   
## Attaching package: 'gplots'  
##   
## The following object is masked from 'package:stats':  
##   
## lowess

## Data Preprocessing

Processing raw data, adapted from 2013-01-03 Illumina lumi Analysis.R.

Read in the raw data using the lumi Bioconductor package.

sample.probe.file <- "../Data/sample\_probe\_profile.txt"  
qc.probe.file <- "../Data/control\_probe\_profile.no\_hyb.txt"  
  
lumi.data <- lumiR.batch(sample.probe.file, annotationColumn=c("ENTREZ\_GENE\_ID", "SYMBOL", "CHROMOSOME", "DEFINITION"))

## Inputting the data ...  
##   
## Adding nuID to the data ...  
## Please provide Illumina ID Mapping library!  
## Perform Quality Control assessment of the LumiBatch object ...

lumi.data <- addControlData2lumi(qc.probe.file, lumi.data)

## Inputting the data ...

sampleNames(lumi.data) <- c(paste(30:35, "S", sep="-"), paste(30:35, "R", sep="-"))

Quantile normalization of the data.

lumi.data.analyzed <- lumiExpresso(  
 lumi.data,  
 varianceStabilize.param=list(method="log2"),   
 normalize.param=list(method="quantile")  
)

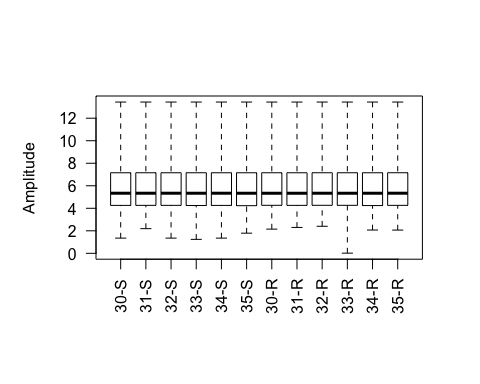
## Background Correction: bgAdjust   
## Variance Stabilizing Transform method: log2   
## Normalization method: quantile   
##   
##   
## Background correction ...  
## Perform bgAdjust background correction ...  
## done.  
##   
## Variance stabilizing ...  
## Perform forcePositive background correction ...  
## Perform log2 transformation ...  
## done.  
##   
## Normalizing ...  
## Perform quantile normalization ...  
## done.  
##   
## Quality control after preprocessing ...  
## Perform Quality Control assessment of the LumiBatch object ...  
## done.

Subset to annotated genes that were detected and annotated to a gene.

lumi.data.analyzed.gsea <- lumi.data.analyzed[ fData(lumi.data.analyzed)$SYMBOL != "", ]  
  
lumi.data.analyzed.gsea.selected <- lumi.data.analyzed.gsea[ detectionCall(lumi.data.analyzed.gsea) > 0, ]

Double checking normalization

boxplot(lumi.data.analyzed.gsea.selected,las=2)

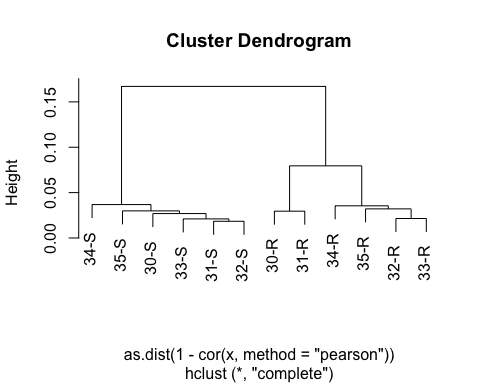


Retain only those probes which have an overall fold change between any condition of at least 1.

normLumiDat.logFC <- lumi.data.analyzed.gsea.selected[apply(exprs(lumi.data.analyzed.gsea.selected),1,function(x){max(x)-min(x)}) > 1,]

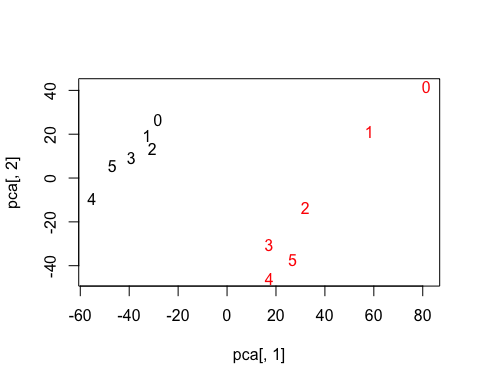
Dominant clustering appears to be by whether lines are sensitive or resistant. Subclusters appear to distinguish timing, although recognize it may be confounded with technicalities relating to batch.

plot(standard.pearson(exprs(normLumiDat.logFC)))



PCA analysis

pca <- prcomp(t(exprs(normLumiDat.logFC)),  
 center=TRUE, scale=TRUE)$x  
plot(pca[,1], pca[,2],col=ifelse(substr(row.names(pca),4,4)=='S','black','red'),  
 pch=substr(row.names(pca),2,2))

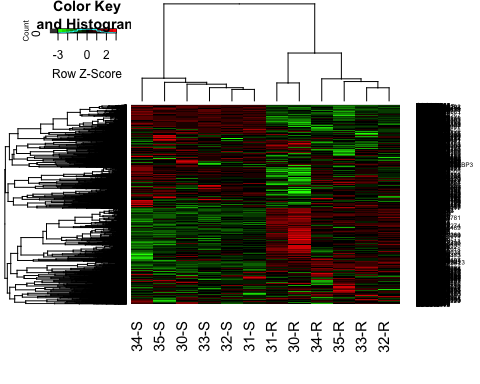


Get gene level data for analysis with CoGAPS.

normLumiDat.Gene <- apply(exprs(normLumiDat.logFC),2,function(x){  
 tapply(x,fData(normLumiDat.logFC)$SYMBOL,mean)  
})  
normLumiDat.Gene.SD <- apply(exprs(normLumiDat.logFC),2,function(x){  
 tapply(x,fData(normLumiDat.logFC)$SYMBOL,sd)  
})

Heatmap of mean gene expression data

heatmap.2(normLumiDat.Gene,scale='row',trace='none',  
 col=greenred,hclust=function(x) hclust(x,method="complete"),  
 distfun=function(x) as.dist((1-cor(t(x)))/2))



Export the data for CoGAPS analysis.

save(list=ls(pattern='Gene'),  
 file='NormGeneData.Rda')