suppressPackageStartupMessages(library(DESeq2))

suppressPackageStartupMessages(library(gplots))

suppressPackageStartupMessages(library(ggplot2))

suppressPackageStartupMessages(library(RColorBrewer))

suppressPackageStartupMessages(library(biomaRt))

suppressPackageStartupMessages(library(ggrepel))

suppressPackageStartupMessages(library(gridExtra))

suppressPackageStartupMessages(library(sva))

suppressPackageStartupMessages(source('~/scripts/make.html.report.R'))

hmcol <- colorRampPalette(brewer.pal(9,"GnBu"))(100)

data <- read.table("counts.tsv", row.names=1, header=T, sep="\t")

colnames(data) <- gsub("X", "", colnames(data))

colnames(data) <- gsub('[.]', "-", colnames(data))

data <- data[,order(colnames(data))]

col <- unique(grepl("[\_]",colnames(data)))

if(all(col)) {col <- as.data.frame(t(sapply((strsplit(colnames(data),"[\_]")),unlist)))} else {col <- as.data.frame(colnames(data))}

rownames(col) <- colnames(data); col

colnames(col) <- intgroup <- c('type','rep')

dds <- DESeqDataSetFromMatrix(countData=data, colData=col, design=~type)

dds <- DESeq(dds)

rld <- rlog(dds,blind=F); rlogMat <- assay(rld); distsRL <- dist(t(rlogMat));

mat <- as.matrix(distsRL); hc <- hclust(distsRL)

if(length(colnames(col)) > 1) {rownames(mat) <- colnames(mat) <- do.call(paste,c(col[,intgroup]))} else rownames(mat) <- colnames(mat) <- paste(col[,"type"])

### PCA ###

pca\_data <- plotPCA(rld,intgroup=intgroup, returnData=T)

perc <- round(100 \* attr(pca\_data, "percentVar"))

p <- ggplot(pca\_data, aes(PC1, PC2, color=paste(type))) + geom\_point(size=2) + geom\_text\_repel(aes(label=paste(rep)), size = 2.5, force = 2, max.iter = 1000) + xlab(paste0("PC1: ",perc[1],"% variance")) + ylab(paste0("PC2: ",perc[2],"% variance")); p$labels$colour <- ""

#plot(p)

pdf("PCA.pdf",title="PCA", height=10,width=10); p; dev.off()

### correlation matrix ###

pdf("correlation.pdf",title="correlation"); heatmap.2(mat, Rowv=as.dendrogram(hc), symm=T, trace="none", col = rev(hmcol), margin=c(13,13), cexRow=0.8, cexCol=0.8); dev.off()

### normalized counts ###

write.table(counts(dds, normalized=T), sep="\t", file="counts\_norm.tsv", quote=F, col.names=NA)

### get results ###

contrasts <- c('type','time')

conditions <- character()

for(x in contrasts){conditions <- c(conditions,as.vector(unique(col[,x])))}

res <- list()

for(z in contrasts){for(x in conditions){for(y in conditions){

if(x==y) next; res[[paste0(y,"\_vs\_",x)]] <- lfcShrink(dds,contrast=c(z,y,x))}}}

names(res)

res\_org <- res

res <- res[c(1,4,28)]

### report ###

htmlreport(res)

### make tsv ###

for(x in names(res)){

write.table(res[[x]], file=paste0(x,".tsv"), quote=F, sep="\t")}

### remove PCA batch effect ###

batch <- col$batch

mod <- model.matrix(~type, data=col)

rldfilt <- subset(assay(rld), rowMeans(assay(rld)) > 0)

rldbatch <- ComBat(dat=rldfilt, batch=batch, mod=mod,par.prior=TRUE, prior.plots=FALSE); rv <- rowVars(rldbatch)

select <- order(rv, decreasing = TRUE)[seq\_len(min(500,length(rv)))]

pca <- prcomp(t(rldbatch[select, ])); percentVar <- pca$sdev^2/sum(pca$sdev^2)

intgroup.df <- as.data.frame(colData(rld)[, intgroup, drop = FALSE])

group <- factor(apply(intgroup.df, 1, paste, collapse = " : "))

pca\_data <- data.frame(PC1 = pca$x[, 1], PC2 = pca$x[, 2], group = group,intgroup.df, names = colnames(rld))

attr(pca\_data, "percentVar") <- percentVar[1:2]

distsRL <- dist(t(rldbatch)); mat <- as.matrix(distsRL); hc <- hclust(distsRL)

if(length(colnames(col)) > 1) {rownames(mat) <- colnames(mat) <- do.call(paste,c(col[,intgroup]))} else rownames(mat) <- colnames(mat) <- paste(col[,"type"])

perc <- round(100 \* attr(pca\_data, "percentVar"))

p <- ggplot(pca\_data, aes(PC1, PC2, color=paste(type))) + geom\_point(size=2) + geom\_text\_repel(aes(label=paste(rep)), size = 2.5, force = 2, max.iter = 1000) + xlab(paste0("PC1: ",perc[1],"% variance")) + ylab(paste0("PC2: ",perc[2],"% variance")); p$labels$colour <- ""

#plot(p)

pdf("PCA.pdf",title="PCA", height=10,width=10); p; dev.off()