Report Sleep Deprivation

giugno 01, 2018

Contents

Description

This is the report of the analysis made for the paper *(TITLE) AND AUTHORS. INSERT ABSTRACT*

Importing data

Importing data and filtering out those genes with cpm lesser than 1. We use the *filtered.data* method in *NOISeq* package.

```
countMatrix <- ReadDataFrameFromTsv(file.name.path="../../data/refSEQ_countMatrix.txt")
# head(countMatrix)

designMatrix <- ReadDataFrameFromTsv(file.name.path="../../design/all_samples_short_names_noRS2HC7.tsv")
# head(designMatrix)

filteredCountsProp <- filterLowCounts(counts.dataframe=countMatrix,
                                      is.normalized=FALSE,
                                      design.dataframe=designMatrix,
                                      cond.col.name="gcondition",
                                      method.type="Proportion")
```

```
## Filtering out low count features...
## 14454 features are to be kept for differential expression analysis with filtering method 3
```

Control Genes

Negative control genes

Loading Negative Control Genes to normalize data

```
library(readxl)

sd.ctrls <- read_excel(path="../../data/controls/Additional File 4 full list of BMC genomics SD&RS2.xlsx
sd.ctrls <- sd.ctrls[order(sd.ctrls$adj.P.Val),]

sd.neg.ctrls <- sd.ctrls[sd.ctrls$adj.P.Val > 0.9, ]

sd.neg.ctrls <- sd.neg.ctrls$`MGI Symbol`
sd.neg.ctrls <- sd.neg.ctrls[-which(is.na(sd.neg.ctrls))]

int.neg.ctrls <- sd.neg.ctrls
int.neg.ctrls <- unique(int.neg.ctrls)
neg.map <- convertGenesViaMouseDb(gene.list=int.neg.ctrls, fromType="SYMBOL",
                                  "ENTREZID")
# sum(is.na(neg.map$ENTREZID))
neg.ctrls.entrez <- as.character(neg.map$ENTREZID)


ind.ctrls <- which(rownames(filteredCountsProp) %in% neg.ctrls.entrez)
counts.neg.ctrls <- filteredCountsProp[ind.ctrls,]
```

positive control genes

Loading Positive Control Genes to detect them during the differential expression step.

```
## sleep deprivation
sd.lit.pos.ctrls <- read_excel("../../data/controls/SD_RS_PosControls_final.xlsx",
                        sheet=1)
colnames(sd.lit.pos.ctrls) <- sd.lit.pos.ctrls[1,]
sd.lit.pos.ctrls <- sd.lit.pos.ctrls[-1,]


sd.est.pos.ctrls <- read_excel("../../data/controls/SD_RS_PosControls_final.xlsx",
                        sheet=3)

sd.pos.ctrls <- cbind(sd.est.pos.ctrls$`MGI Symbol`, "est")
sd.pos.ctrls <- rbind(sd.pos.ctrls, cbind(sd.lit.pos.ctrls$Gene, "lit"))

sd.pos.ctrls <- sd.pos.ctrls[-which(duplicated(sd.pos.ctrls[,1])),]
sd.pos.ctrls <- sd.pos.ctrls[-which(is.na(sd.pos.ctrls[,1])),]
```

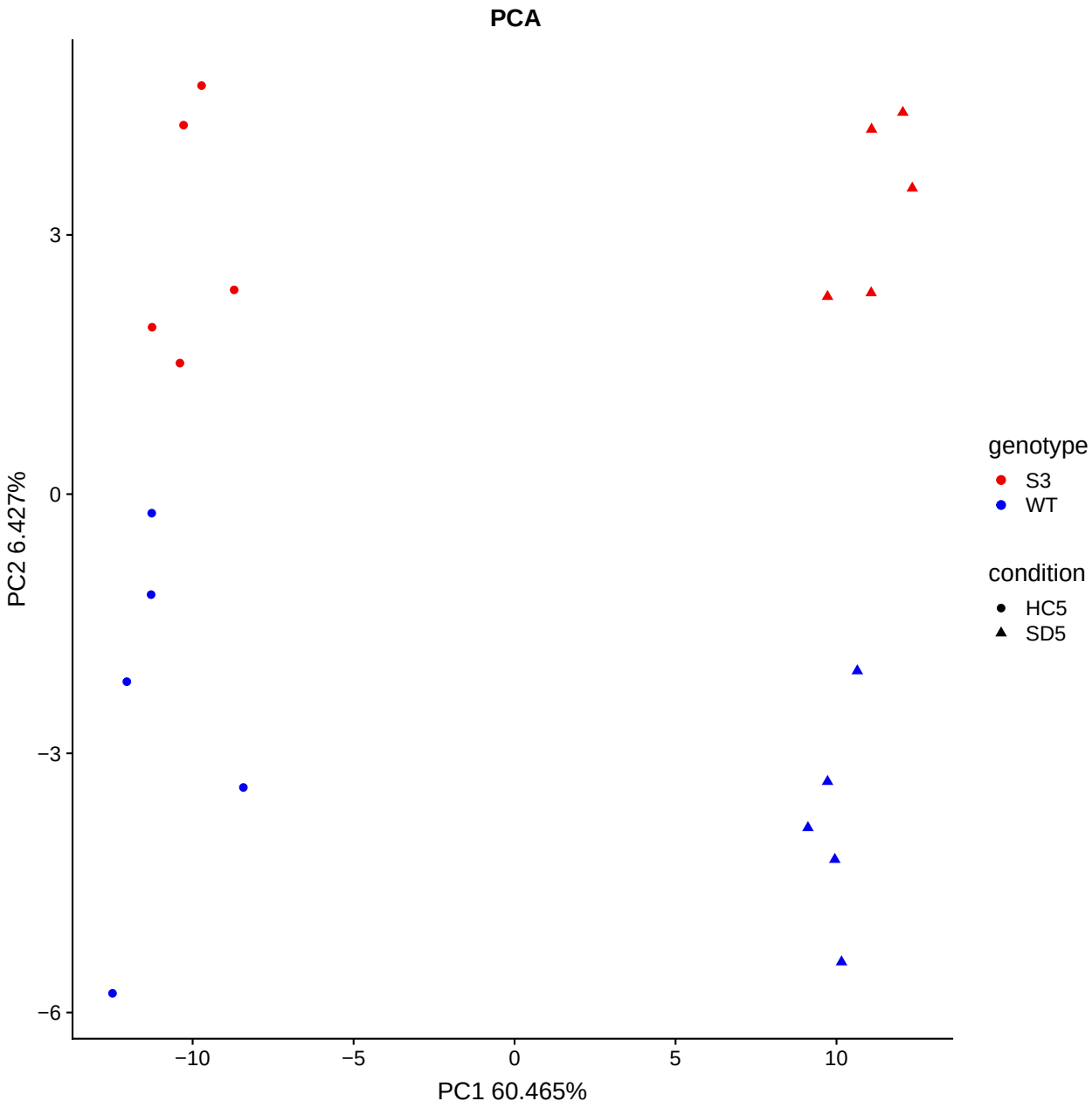
Normalizations

TMM + RUVs Normalization

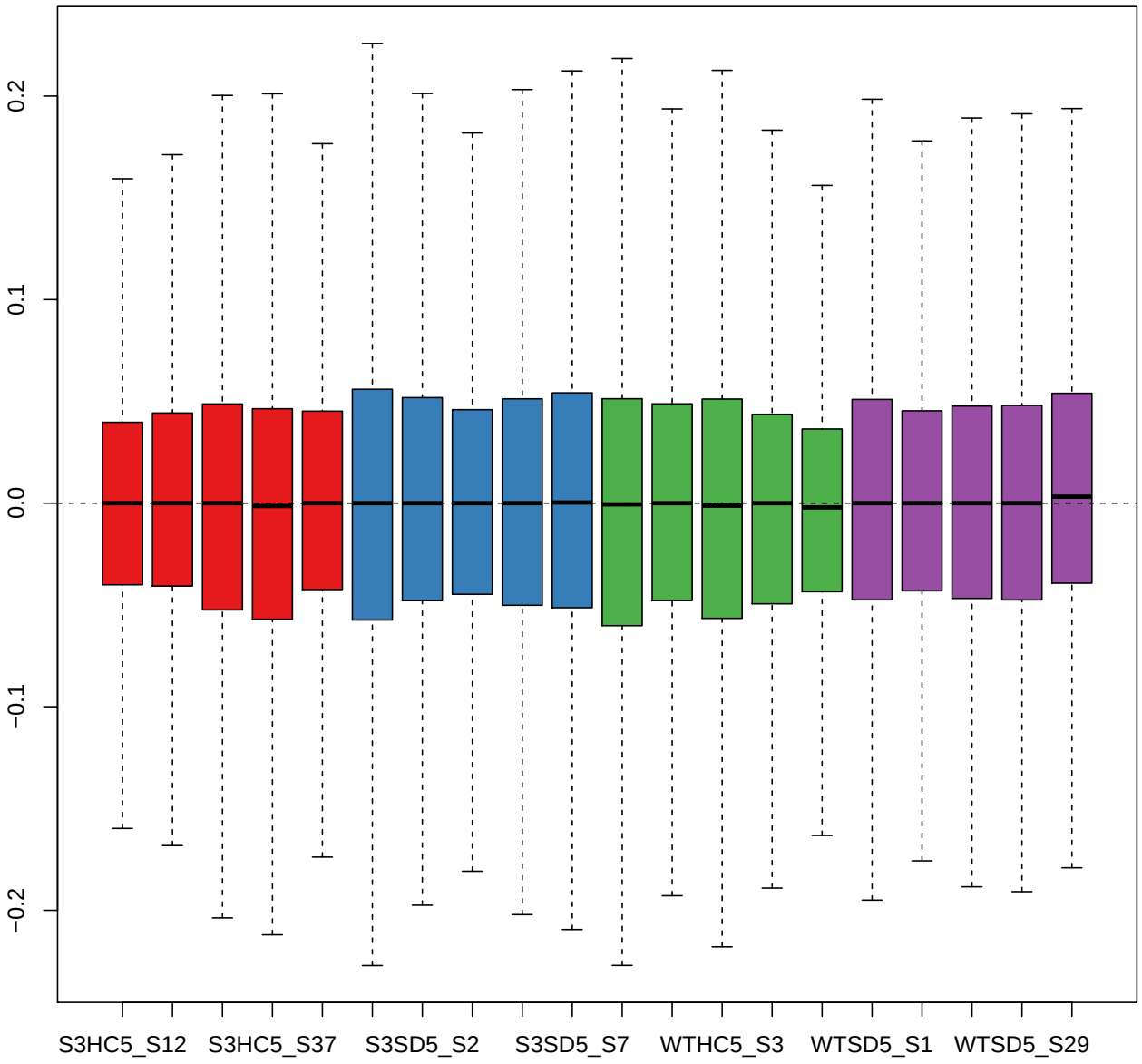
Normalizing data with *TMM*, as implemented in *edgeR* package, and applying a *RUVs* method of *RUVSeq* package on normalized data, in order to adjust the counts for the unwanted variation. We plot a PCA and an RLE plot on these data.

```
normPropCountsUqua <- NormalizeData(data.to.normalize=filteredCountsProp,
                                    norm.type="tmm",
                                    design.matrix=designMatrix)
neg.ctrl.list <- rownames(counts.neg.ctrls)
groups <- makeGroups(designMatrix$gcondition)
ruvedSExprData <- RUVs(as.matrix(round(normPropCountsUqua)), cIdx=neg.ctrl.list,
                       scIdx=groups, k=5)

normExprData <- ruvedSExprData$normalizedCounts

ggp <- PlotPCAPlotlyFunction(counts.data.frame=log1p(normExprData),
                    design.matrix=designMatrix, shapeColname="condition",
                    colorColname="genotype", xPCA="PC1", yPCA="PC2",
                    plotly.flag=FALSE, show.plot.flag=TRUE, save.plot=FALSE,
                    prefix.plot=NULL)
```

```
## [1] FALSE
```




```
# ggplotly(ggp)
dir.create("plots")
save_plot(filename="plots/PCA.pdf", plot=ggp)

pal <- RColorBrewer::brewer.pal(9, "Set1")
plotRLE(normExprData, outline=FALSE, col=pal[designMatrix$gcondition])
```



edgeR Differential Expression Analysis

Making differential expression analysis with *edgeR* package on four different contrasts.

Here is a brief legend:

- WTHC5: Wild Type Home Cage Control 5 days
- WTSD5: Wild Type Sleep Deprivation 5 days.
- S3HC5: Shank3 Home Cage Control 5 days.
- S3SD5: Shank3 Sleep Deprivation 5 days.

```
padj.thr <- 0.05
venn.padgj.thr <- 0.1
desMat <- cbind(designMatrix, ruvedSExprData$W)
colnames(desMat) <- c(colnames(designMatrix), colnames(ruvedSExprData$W))
```

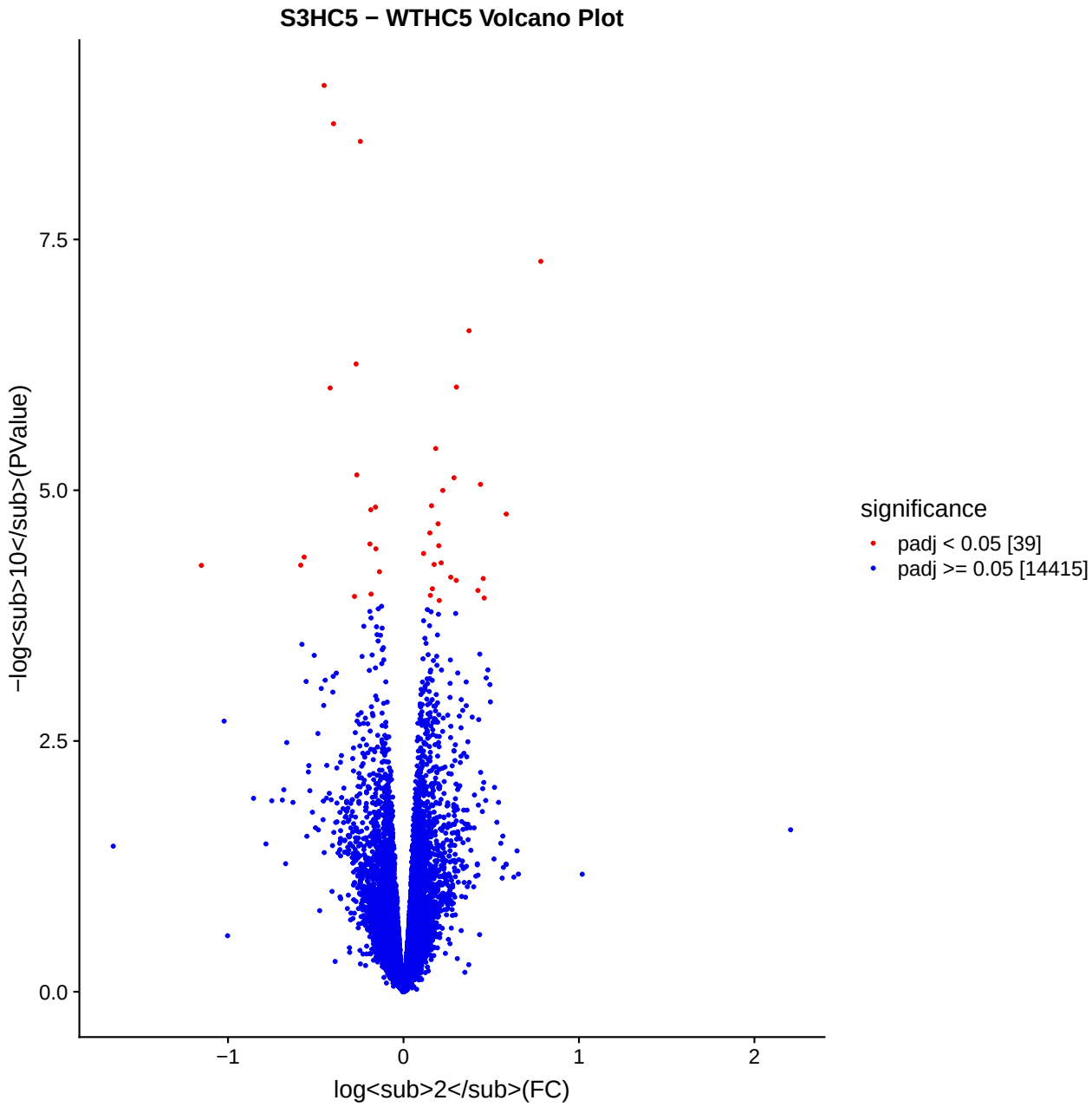
```
cc <- c("S3HC5 - WTHC5", "S3SD5 - WTSD5")

rescList1 <- applyEdgeR(counts=filteredCountsProp, design.matrix=desMat,
                        factors.column="gcondition",
                        weight.columns=c("W_1", "W_2", "W_3", "W_4", "W_5"),
                        contrasts=cc, useIntercept=FALSE, p.threshold=1,
                        is.normalized=FALSE, verbose=TRUE)
names <- names(rescList1)
rescList1 <- lapply(seq_along(rescList1), function(i)
{
    attachMeans(normalized.counts=normExprData, design.matrix=desMat,
                factor.column="gcondition", contrast.name=names(rescList1)[i],
                de.results=rescList1[[i]])
})
names(rescList1) <- names
```

Shank3 Home Cage control VS Wild Type Home Cage Controls

volcano plot

A volcano plot of differential expressed genes.

```
res.o.map1 <- convertGenesViaMouseDb(gene.list=rownames(rescList1[[1]]),
                             fromType="ENTREZID")

res.o <- attachGeneColumnToDf(mainDf=rescList1[[1]],
                                genesMap=res.o.map1,
                                rowNamesIdentifier="ENTREZID",
                                mapFromIdentifier="ENTREZID",
                                mapToIdentifier="SYMBOL")
WriteDataFrameAsTsv(data.frame.to.save=res.o,
                    file.name.path=paste0(names(rescList1)[1], "_edgeR"))

vp <- luciaVolcanoPlot(res.o, prefix=names(rescList1)[1],
                        positive.controls.df=NULL,
                        threshold=padj.thr)
# ggplotly(vp)
plot(vp)
```




```
de <- sum(res.o$FDR < padj.thr)
nde <- sum(res.o$FDR >= padj.thr)
detable <- cbind(de,nde)
rownames(detable) <- names(rescList1)[1]
ddetable <- detable

tot.ctrls <- dim(sd.pos.ctrls)[1]
idx.pc <- which(tolower(res.o$gene) %in% tolower(sd.pos.ctrls[,1]))
tot.pc.de <- sum(res.o$FDR[idx.pc] < padj.thr)
tot.pc.nde <- length(idx.pc) - tot.pc.de

wt <- res.o[which(res.o$FDR < padj.thr),]
wt.sign.genes.entrez <- rownames(res.o)[which(res.o$FDR < venn.padgj.thr)]
```

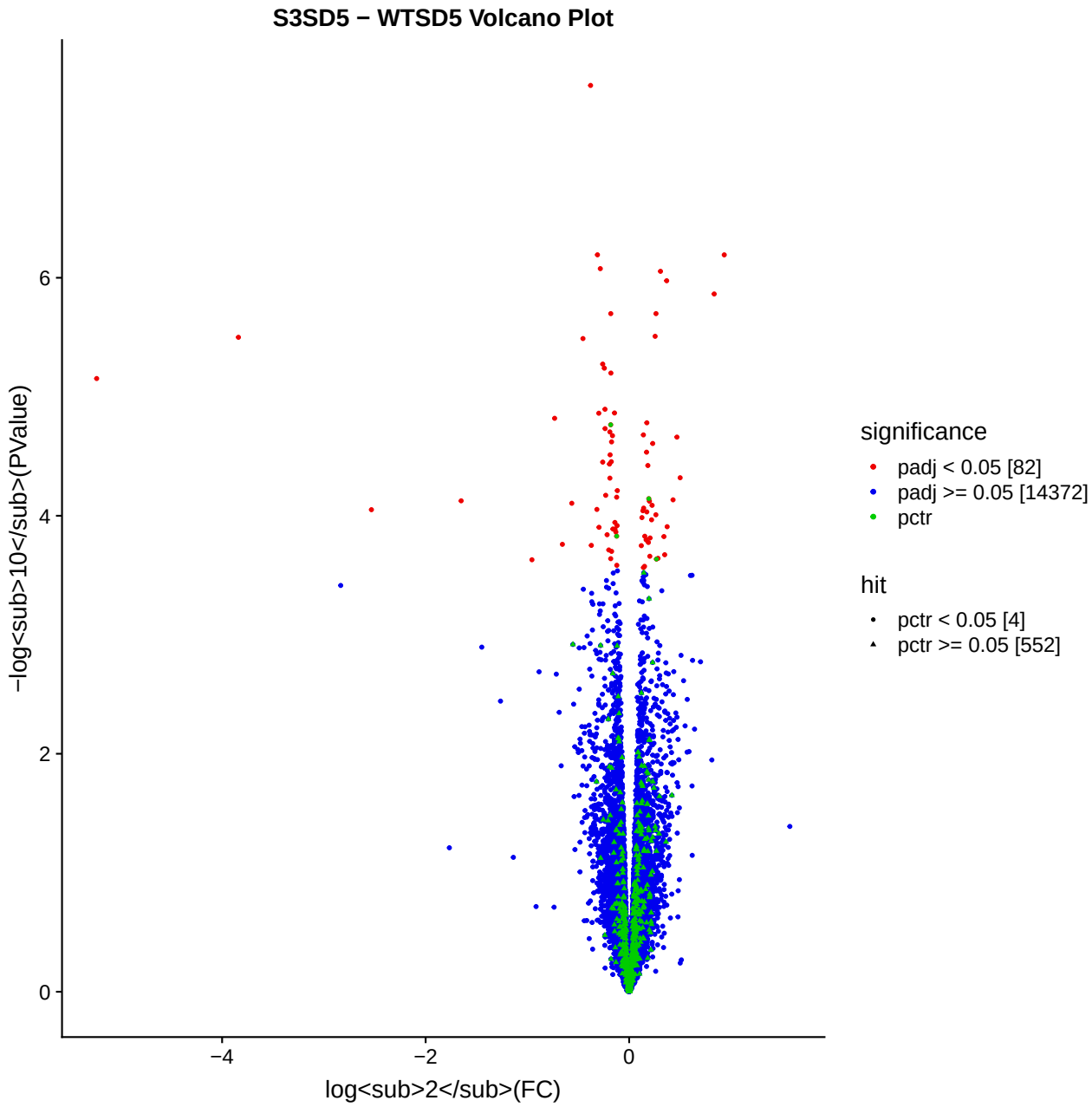
```
kowthc5 <- res.o[which(res.o$FDR < padj.thr),]
kowthc5.sign.genes.entrez <- rownames(res.o)[which(res.o$FDR < venn.padgj.thr)]
```

Shank3 Sleed Deprivation VS Wild Type Sleep Deprivation

volcano plot

A volcano plot of differential expressed genes.

```
rs2.o.map <- convertGenesViaMouseDb(gene.list=rownames(rescList1[[2]]),
                            fromType="ENTREZID")

res.rs2.o <- attachGeneColumnToDf(mainDf=rescList1[[2]],
                                  genesMap=rs2.o.map,
                                  rowNamesIdentifier="ENTREZID",
                                  mapFromIdentifier="ENTREZID",
                                  mapToIdentifier="SYMBOL")
WriteDataFrameAsTsv(data.frame.to.save=res.rs2.o,
                    file.name.path=paste0(names(rescList1)[2], "_edgeR"))

vp <- luciaVolcanoPlot(res.rs2.o, positive.controls.df=sd.pos.ctrls,
                       prefix=names(rescList1)[2],
                       threshold=padj.thr)
# ggplotly(vp)
plot(vp)
```




```
de <- sum(res.rs2.o$FDR < padj.thr)
nde <- sum(res.rs2.o$FDR >= padj.thr)
detable <- cbind(de,nde)
rownames(detable) <- names(rescList1)[2]
ddetable <- rbind(ddetable, detable)
pos.df <- cbind(tot.ctrls, tot.pc.de, tot.pc.nde)
colnames(pos.df) <- c("total_p.ctrl", "p.ctrl_de_mapped",
                      "p.ctrl_notde_mapped")
rownames(pos.df) <- names(rescList1)[2]

kowtsd5 <- res.rs2.o[which(res.rs2.o$FDR < padj.thr),]
kowtsd5.sign.genes.entrez <- rownames(res.rs2.o)[which(res.rs2.o$FDR < venn.padgj.thr)]
```

DE TABLE + Positive Controls table

We present a summarization of the results. The first table is a summarization on how many genes are Differentially Expressed. The second table explains on the first column how many positive controls we have, on the second column how many positive controls have been identified over the differentially expressed genes, and, finally, on the third column how many positive controls have beed identified on the NOT differentially expressed genes.

```
ddetable
```

```
##                de   nde
## S3HC5 - WTHC5 39 14415
## S3SD5 - WTSD5 82 14372
```

```
pos.df
```

```
##                total_p.ctrl p.ctrl_de_mapped p.ctrl_notde_mapped
## S3SD5 - WTSD5           579                3                 553
```

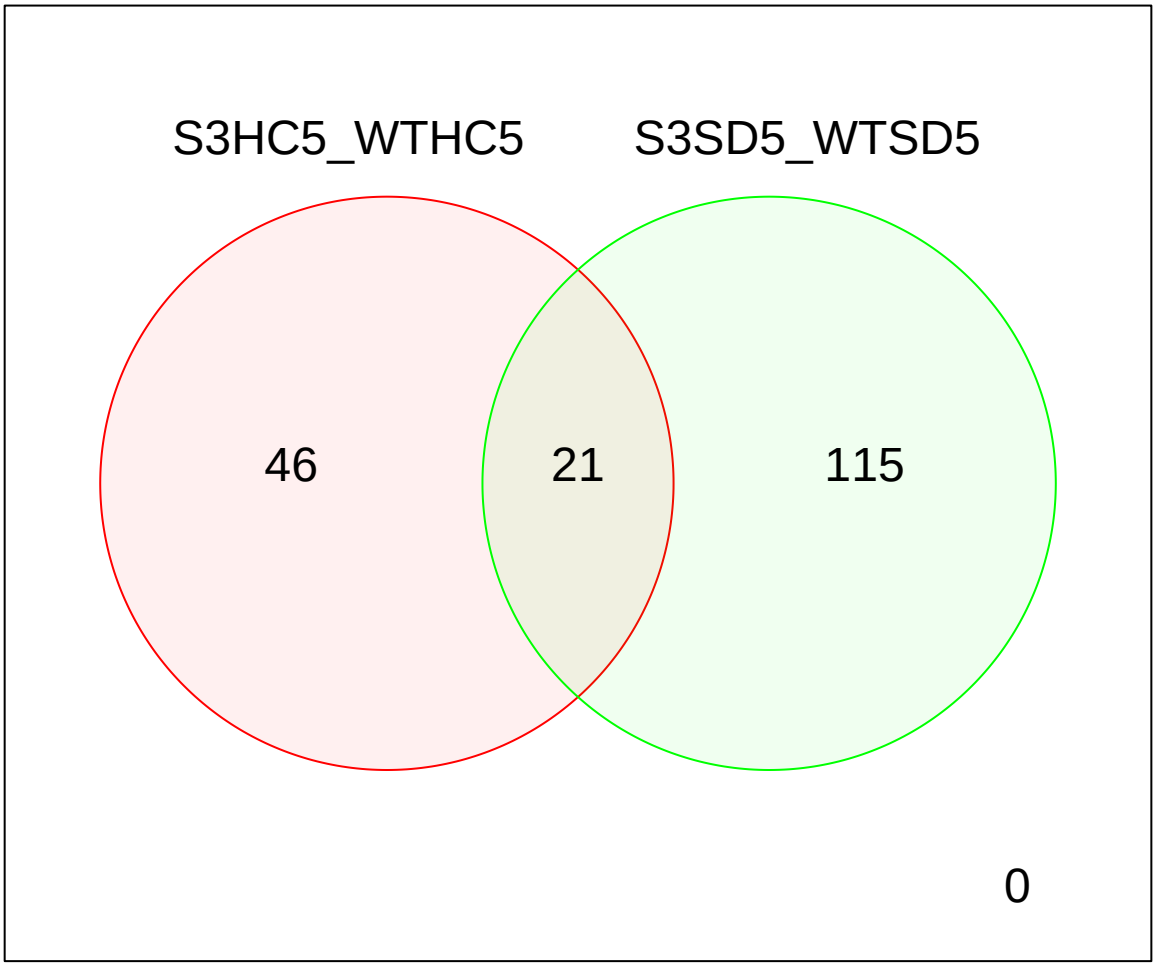
Venn Diagram

S3HC5-WTHC5 vs S3SD5-WTSD5

We take the results of the two contrasts. *Shank3 Sleed Deprivation VS Wild Type Sleep Deprivation* and *Shank3 Home Cage control VS Wild Type Home Cage Controls*. And plot the results in a Venn Diagram

```
source("../../R/venn2.R")

gene.map <- convertGenesViaMouseDb(gene.list=rownames(normExprData),
                                   fromType="ENTREZID", toType="SYMBOL")
venn <- Venn2de(x=kowthc5.sign.genes.entrez, y=kowtsd5.sign.genes.entrez,
        label1="S3HC5_WTHC5", label2="S3SD5_WTSD5",
        title="S3HC5_WTHC5 venn S3SD5_WTSD5", plot.dir="./",
        conversion.map=gene.map)
```

S3HC5_WTHC5 venn S3SD5_WTSD5



Heatmaps

Setting up the data structures for the heatmap.

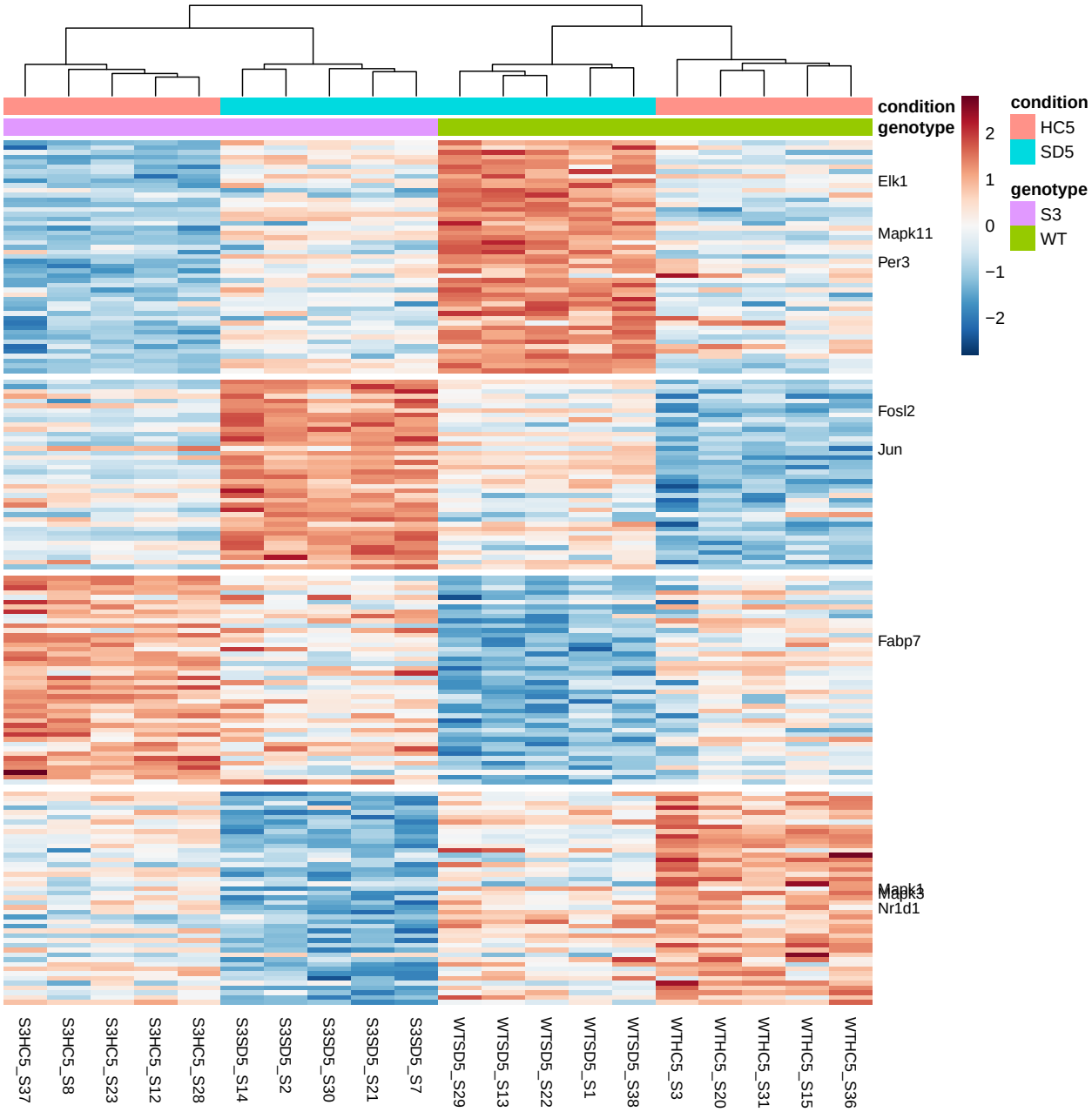
```
source("../../R/heatmapFunctions.R")
de.genes.entr <- union(rownames(venn$int), rownames(venn$XnoY))
de.genes.entr <- union(de.genes.entr, rownames(venn$YnoX))

gene.map <- convertGenesViaMouseDb(gene.list=de.genes.entr,
                            fromType="ENTREZID")
```

```
de.genes.symb <- attachGeneColumnToDf(as.data.frame(de.genes.entr,
                                                    row.names=de.genes.entr),
                                      genesMap=gene.map,
                                      rowNamesIdentifier="ENTREZID",
                                      mapFromIdentifier="ENTREZID",
                                      mapToIdentifier="SYMBOL")

# de.genes.symb[which(is.na(de.genes.symb$gene)),]
de.genes.symb$gene[which(de.genes.symb$de.genes.entr=="100039826")]  <- "Gm2444" ## not annotated in nc
de.genes.symb$gene[which(de.genes.symb$de.genes.entr=="210541")]  <- "Entrez:210541" ## not annotated i


de.genes.counts <- normExprData[match(de.genes.symb$de.genes.entr, rownames(normExprData)),]
rownames(de.genes.counts) <- de.genes.symb$gene

de.gene.means <- computeGeneMeansOverGroups(counts=de.genes.counts,
                            design=designMatrix, groupColumn="gcondition")

library(gplots)
library(clusterExperiment)
color.palette = clusterExperiment::seqPal3#c("black", "yellow")
pal <- colorRampPalette(color.palette)(n = 1000)


library(pheatmap)
filter2 <- rowMeans(de.gene.means)>0
filter <- apply(de.gene.means, 1, function(x) log(x[4]/x[3]) * log(x[2]/x[1]) < 0)
filter[is.na(filter)] <- FALSE
```

heatmap gene by gene organized as kmeans cluster

```
idx <- which(!(rownames(de.genes.counts) %in% c("Nr1d1", "Fabp7", "Per3",
                        "Jun", "Elk1", "Fosl2", "Mapk1", "Mapk3", "Mapk11")))
de.genes.counts1 <- de.genes.counts
rownames(de.genes.counts1)[idx] <- ""
ann.col <- desMat[, c(1:2)]
de.heatmap <- de.genes.counts[filter2,]
set.seed(0)
ph1 <- pheatmap(log(de.heatmap+1), cluster_cols=TRUE, scale="row",
            color=pal, border_color=NA, fontsize_row=10, kmeans_k=4, annotation_col=ann.col,
            silent=TRUE)
clusterized.genes <- as.data.frame(ph1$kmeans$cluster)

gene.map <- convertGenesViaMouseDb(gene.list=rownames(clusterized.genes), fromType="SYMBOL")
converted.clusterized.gens <- attachGeneColumnToDf(mainDf=clusterized.genes, genesMap=gene.map,
                    rowNamesIdentifier="SYMBOL", mapFromIdentifier="SYMBOL", mapToIdentifier="ENTREZID"

converted.clusterized.gens$gene[which(rownames(converted.clusterized.gens)=="Gm2444")]  <- "100039826" #
converted.clusterized.gens$gene[which(rownames(converted.clusterized.gens)=="Entrez:210541")]  <- "2105
converted.clusterized.gens <- converted.clusterized.gens[order(converted.clusterized.gens$`ph1$kmeans$cl

ord.de.genes.counts <- de.heatmap[match(rownames(converted.clusterized.gens), rownames(de.heatmap)),]
```

```
idx <- which(!(rownames(ord.de.genes.counts) %in% c("Nr1d1", "Fabp7", "Per3",
                        "Jun", "Elk1", "Fosl2", "Mapk1", "Mapk3", "Mapk11")))

rownames(ord.de.genes.counts)[idx] <- ""
gaps.row <- c()
for(i in c(1:4))
{
    li <- length(which(converted.clusterized.gens$`ph1$kmeans$cluster`==i))
    l <- ifelse(i!=1, gaps.row[i-1]+li, li)
    gaps.row <- c(gaps.row, l)
}


ph1 <- pheatmap(log(ord.de.genes.counts+1), cluster_cols=TRUE, scale="row",
            color=pal, border_color=NA, fontsize_row=9, fontsize_col=9, cluster_rows=FALSE,
            annotation_col=ann.col, gaps_row=gaps.row)
```




```
save_pheatmap_pdf(filename="plots/heatmap_gg_k4.pdf", plot=ph1, width=20, height=20)
```

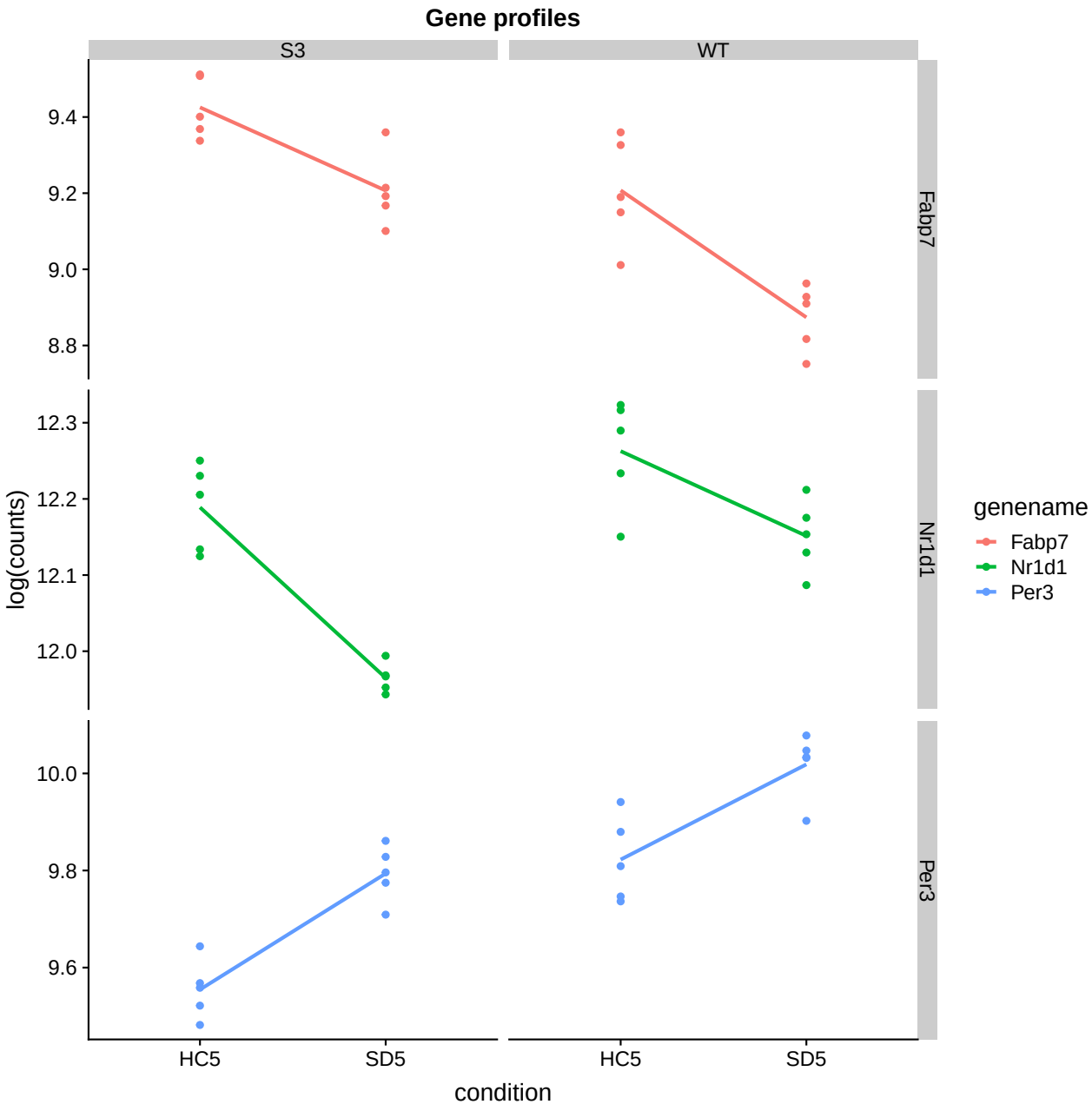
```
## pdf
##   2
```

Group gene profiles

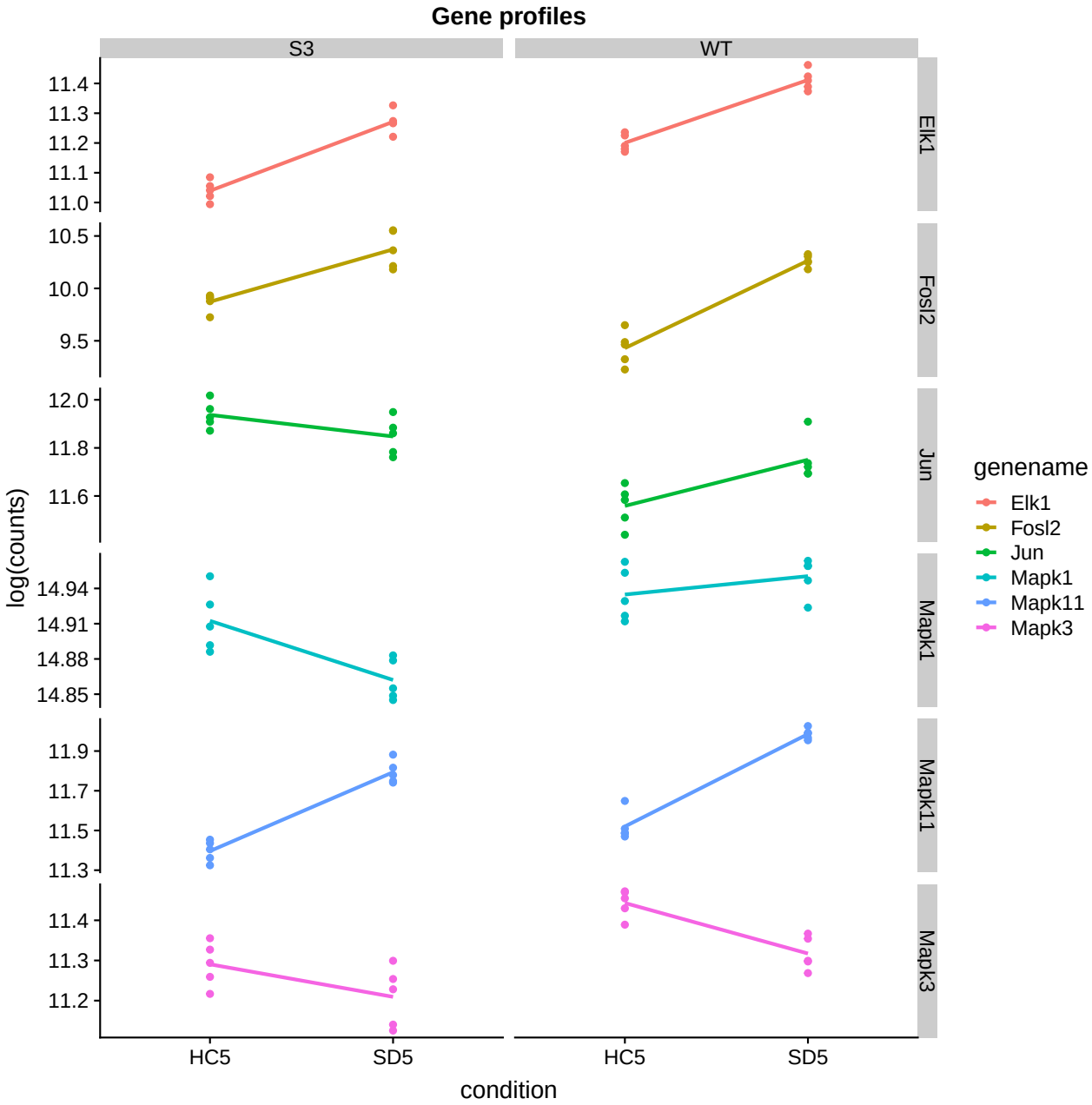
Group gene profiles by genotype

```
g <- geneGroupProfileRows(normalized.counts=normExprData, design.matrix=designMatrix,
            gene.names=c("Nr1d1", "Fabp7", "Per3"),
```

```
            res.o=de.genes.symb, show.plot=TRUE, plotly.flag=FALSE, log.flag=TRUE)
# ggplotly(g)
plot(g)
```




```
save_plot(filename=paste0("plots/", "Nr1d1_Fabp7_Per3", "_log_gene_profile_genotype.pdf"), plot=g,
        base_height=15, base_width=15)

g <- geneGroupProfileRows(normalized.counts=normExprData, design.matrix=designMatrix,
            gene.names=c("Jun", "Elk1", "Fosl2", "Mapk1", "Mapk3", "Mapk11"),
            res.o=de.genes.symb, show.plot=TRUE, plotly.flag=FALSE, log.flag=TRUE)
# ggplotly(g)
plot(g)
```




```
save_plot(filename=paste0("plots/", "Jun_Elk1_Fosl2_Mapk", "_log_gene_profile_genotype.pdf"), plot=g,
        base_height=15, base_width=15)
```