title: "DEG (Defferential gene expression analysis, RNA-seq)_July2019_Kopp lab"

author: "Soheila Zarei" date: '2019-07-28'

{r, echo=TRUE}

knitr::opts_chunk$set(error = TRUE)

```
knitr::opts_chunk$set(echo = TRUE, eval = FALSE)

Read_count_all_removedp53_July28_2018<-
read.csv("Data/Read_count_all_removedp53_Oct22_2018.csv", header =
TRUE, row.names = 1)
head(Read_count_all_removedp53_July28_2018)

##     Gene  Du2  Du3  Du7 Du10 Du11 Du12 Du17 Du18_19 Du20 Du22 Du23
Du24
## 1   Xkr4    7    6    6    1    0    4    0       6    8    1    4
5
## 2    Rp1    0    0    0    0    0    1    0       2    0    1    0
0
## 3  Sox17 2462 2402 3573 2393   75 3124 1037   16911  296 1124  811
760
## 4 Mrpl15 1797 1680 2070 1528 1219 1687  869    3050 2451 1511 1768
1783
## 5 Lypla1 2526 2475 2630 1943  887 1922  622    4767 2234 1862 1495
1728
## 6  Tcea1 3829 3609 2099 2900 1226 2465 1015    2957 2194 1432 1607
1779
##   Du25  AC3  AC5  AC7  AC8 AC10 AC12 AC13 AC14 AC16 AC17 AC19 AC23
## 1    9    5    0    2    2    6    2    0    0    2    3    2    1
## 2    0    0    0    1    1    0    0    0    0    1    0    0    0
## 3  662 3833 3094 6873 2318  589 6373 6072 5273 6859 4167 5033  292
## 4 1587 1515 1521 1001 1316 1387 1083 1455 1772  995 1182 1061 1351
## 5 2033 2583 2313  967 2224 2905 1857 1762 1852 1642 1525 1569 1059
## 6 1893 2836 2983 1090 2934 2837  914 1395 1425 1069 1105  902 1171

all_reads_July2019<- Read_count_all_removedp53_July28_2018
View(all_reads_July2019)

sort(all_reads_July2019$Gene)
row.names(all_reads_July2019) <- all_reads_July2019$Gene
head(all_reads_July2019)
cts_July_2019 <- all_reads_July2019[,-1]
head(cts_July_2019)
sampleinfo_Oct23<-
read.csv("Data/phenotype_removed_Com_all_samples_Oct23_2018.csv",
row.names = 1)
```

```
library(DESeq2)
sampleinfo_Oct23
colnames(cts_July_2019)
rownames(sampleinfo_Oct23)
colData<- sampleinfo_Oct23
all(rownames(colData) %in% colnames(cts_July_2019))
```

With the count matrix, cts, and the sample information, coldata, we can construct a DESeqDataSet:

```
dds <- DESeqDataSetFromMatrix ( countData = cts_July_2019,
                                colData,
                                design = ~ B + P + Sex + Condition )

dds
```

Pre-filtering

```
library(dplyr)
library("BiocParallel")
register(MulticoreParam(4))
colData ( dds) %>% head
assay ( dds) %>% head
rowRanges ( dds) %>% head
dds <- dds[ rowSums ( counts ( dds)) > 1, ]
```

Note on factor levels Setting the factor levels

```
dds$Condition<- factor(dds$Condition, levels = c("A", "D"))
dds$Condition<- relevel(dds$Condition, ref = "A")
```

Using parallelization To speed the process

```
library("BiocParallel")
register(MulticoreParam(4))
```

Data transformations and visualization Count data transformations

```
#vsd <- vst(dds, blind=FALSE)
rld <- rlog(dds, blind=FALSE)
head(assay(rld), 3)

# this gives log2(n + 1)
ntd <- normTransform(dds)
library("vsn")
meanSdPlot(assay(ntd))

meanSdPlot(assay(vsd))

meanSdPlot(assay(rld))
```

Data quality assessment by sample clustering and visualization

Heatmap of the sample-to-sample distances transformed data is used for sample clustering.

```
sampleDists <- dist(t(assay(rld)))
sampleDists
jpeg("cluster_July2019", width = 8, height = 6, units = 'in', res =
300)
plot ( hclust ( sampleDists ),
       labels = colnames ( sampleDists),
       main = " rld transformed read counts \ ndistance : Pearson
correlation ")
#obtain regularized log - transformed values

library("RColorBrewer")
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(vsd$condition, vsd$type, sep="-")
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
         clustering_distance_rows=sampleDists,
         clustering_distance_cols=sampleDists,
         col=colors)
```

Principal component plot of the samples batch effect detector

```
plotPCA(rld, intgroup= c("Condition", "Sex"))
plotPCA(rld, intgroup= c("Condition"))

library(ggplot2)
jpeg("PCA_July2019", width = 4, height = 4, units = 'in', res = 300)

pcaData <- plotPCA(rld, intgroup=c("Condition", "Sex"),
returnData=TRUE)
percentVar <- round(100 * attr(pcaData, "percentVar"))
ggplot(pcaData, aes(PC1, PC2, color=Condition, shape=Sex)) +
  geom_point(size=3) +
  xlab(paste0("PC1: ",percentVar[1],"% variance")) +
  ylab(paste0("PC2: ",percentVar[2],"% variance")) +
  coord_fixed()
dev.off()
```

Variations to the standard workflow Wald test individual steps The function DESeq runs the following functions in order:

```
dds <- estimateSizeFactors(dds)
dds <- estimateDispersions(dds)
dds <- nbinomWaldTest(dds)
```

Differential expression analysis Likelihood ratio test (LRT) for testing multiple terms at once

```
dds <- DESeq(dds, test="LRT",  reduced = ~ P + B + Sex, parallel=TRUE)
res <- results(dds)
res
```

Log fold change shrinkage for visualization and ranking

```
resultsNames(dds)
```

p-values and adjusted p-values We can order our results table by the smallest p value:

```
resOredered_July2019<- res[order(res$pvalue),]
```

We can summarize some basic tallies using the summary function.

```
summary(res)
```

How many adjusted p-values were less than 0.1?

```
sum(res$padj< 0.05, na.rm = TRUE)

resall <- results(dds, alpha = 0.05)
summary(resall)
write.csv(resall, "resSig_D_A_July2019_all.csv")
```

filter for significant genes, according to some chosen threshold for the false dicovery rate (FDR),

```
resOredered_July2019<- res[order(res$padj),]

resSig_D_A_05_July2019 = subset(resOredered_July2019, padj < 0.05)
resSig_D_A_05_July2019
write.csv(resSig_D_A_05_July2019, "resSig_D_A_05_July2019.csv")

#####heatmap for 0.05 DEG
DEgenes_D_AJUly2019_0.05 <- rownames(resOredered_July2019
[resOredered_July2019$padj<0.05 & !is.na(resOredered_July2019$padj),
])#[1:200]
DEgenes_D_AJUly2019_0.05
DEgenes_D_AJUly2019_0.05 <- assay(rld)[DEgenes_D_AJUly2019_0.05, ]
DEgenes_D_AJUly2019_0.05
DEgenes_D_AJUly2019_0.05 <- DEgenes_A_D_Oct23_0.05 -
rowMeans(DEgenes_D_AJUly2019_0.05)
DEgenes_D_AJUly2019_0.05
write.csv(DEgenes_D_AJUly2019_0.05, "DEgenes_D_AJUly2019_0.05.csv")


#pdf("heatmap_DEgenes_D_AJUly2019_0.05.pdf", width=5, height=25)
jpeg("heatmap_July2019_600_35", width = 7, height = 35, units = 'in',
res = 600)
df <- as.data.frame(colData(rld)[,c("Condition","Sex")])
```

```
pheatmap(DEgenes_D_AJUly2019_0.05, annotation_col=df,
         show_rownames=T,
         cluster_cols=T,
         cluster_rows=T,
         scale="row",
         clustering_distance_rows="euclidean",
         clustering_distance_cols="euclidean",
         clustering_method="complete",
         border_color=FALSE,
         cex=0.5)
dev.off()

plotMA(res, ylim= c(-2,2))
```

Plot counts

```
plotCounts(dds, gene=which.min(res$padj), intgroup="Condition")
```

More information on results columns

```
mcols(res)$description
```

For customized plotting, a data.frame for plotting with ggplot.

```
d <- plotCounts(dds, gene=which.min(res$padj), intgroup="Condition",
                returnData=TRUE)
library("ggplot2")
ggplot(d, aes(x=Condition, y=count)) +
  geom_point(position=position_jitter(w=0.1,h=0)) +
  scale_y_log10(breaks=c(25,100,400))

par(mar=c(8,5,2,2))
boxplot(log10(assays(dds)[["cooks"]]), range=0, las=2)
```