Deseq analysis of iN.d5 example

Yogita Sharma and Per Ludvik Brattås

March, 2019

Tutorial on DESeq2

Analysing iN-d5 RNA-sequencing data

Download public data from NCBI GEO

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE90068

1. Setup of workspace

```
# clean environment and workspace
rm(list=ls())
gc()

# path to project
setwd("~/Dropbox (MN)/RNA-seq workshop/R analysis/")
path <- getwd()
```

2. Read data

```
# path and filename of raw featureCounts file
fc.file <- "Data/GSE90068_hg38.mRNA.NCBI.fCounts.primary.tab.txt"
fc.file.path <- paste(path,fc.file,sep = "/")

# read data - count table from featureCounts
data <- read.csv(file = fc.file.path,header = T,skip = 1)
head(data)
```

3. Define data

```
# We only want C2 NEG CTL and C2 REST_sh.PBPA data
colnames(data)

library(tidyverse)
countdata <- data %>%
  select(contains("C2.NEG.CTL"),contains("C2.REST_sh.PBPA"))
head(countdata)
rownames(countdata) <- make.names(data$Geneid,unique = T)
head(countdata)

# take the column names of the rawCount table from featureCounts
cnames <- colnames(countdata)

# remove the substrings that are not needed for identifying the samples, noise from filenames
sampleNames <- gsub(pattern = ".trimmedAligned.out.sorted.bam",replacement = "",x = cnames)
sampleNames <- gsub(pattern = "hg38.C2.",replacement = "",x = sampleNames)
```

```
sampleNames

# make colnames
colnames(countdata) <- sampleNames
```

4. colData data.frame that describes the data

```
# make one vector describing if the sample has REST inhibition or not
condition <- rep(c("CTL","RESTsh_pBpA"),each=3)

# ColData
colData <- data.frame(condition=condition)
```

5. Run DESeq

```
# 4. make a DESeq object
library(DESeq2)

# create object
dds <- DESeqDataSetFromMatrix(countData = countdata,colData = colData,design = ~condition)

# run DESeq
dds <- DESeq(dds)
```

6. Peak at the data

```
## Get normalized count data
nCounts <- counts(dds,normalized=T)
head(nCounts)

## Calculate average coutns in each condition
aCounts <- sapply( levels(dds$condition), function(lvl) rowMeans( counts(dds,normalized=TRUE)[,dds$cond
head(aCounts)

aCounts["SOX2",]
aCounts["ASCL1",]
```

6.1 PCA

```
## Log2 Transform the data
vst <- varianceStabilizingTransformation(dds,blind = T)
avst <- data.frame(assay(vst))
head(avst)

# pca
pca <- prcomp(t(avst))
head(pca)
head(pca$x)
pcax <- data.frame(pca$x)
```

```
pcax$condition <- rep(c("CTL","REST"),each=3)

library(ggplot2)
ggplot(data = data.frame(pca$x),mapping = aes(x = PC1,y = PC2,color=condition)) +
  geom_point(size=4) +
  theme_classic() +
  ggtitle(label = "PCA")
```

6.2 Heatmaps

```
# heatmaps
library(pheatmap)
library(RColorBrewer)

## Visualize most variable genes
avst.sd <- apply(avst,1,sd)
head(avst)
avst <- avst[order(-avst.sd),]
head(avst)

color <- rev(colorRampPalette(brewer.pal(10,"RdBu"))(200))
pheatmap(avst[1:10,],color = color)
pheatmap(avst[1:50,],color = color)
pheatmap(avst[1:250,],color = color,show_rownames = F)
pheatmap(avst[1:1000,],color = color,show_rownames = F)
```

7. Differential expression

```
res <- results(dds)
res
res["TRIM28",]
res["NEUROD1",]
res["SOX2",]
# create new variable for plotting
res.df <- data.frame(res)
plotData <- res.df
# log2 transform baseMean levels
plotData$log2baseMean <- log2(plotData$baseMean+1)
plotData$log10padj <- -log10(plotData$padj)

p <- ggplot(data = plotData,mapping = aes(x = log2baseMean,y = log2FoldChange))
p
p + theme_classic()
p + geom_point()

# change style
p +
  geom_point() +
  theme_classic()

p +
  geom_point(size=.5) +
  theme_classic()
```

```
## scale size on significance
p +
  geom_point(mapping = aes(size=log10padj)) +
  theme_classic()

p +
  geom_point(mapping = aes(size=log10padj)) +
  theme_classic() +
  scale_size_continuous(range = c(0,3))

# color
p +
  geom_point(mapping = aes(size=log10padj,color=log10padj)) +
  theme_classic() +
  scale_size_continuous(range = c(0,3))

# alpha
p +
  geom_point(mapping = aes(size=log10padj,color=log10padj,alpha=log10padj)) +
  theme_classic() +
  scale_size_continuous(range = c(0,5))

# coloring
library(RColorBrewer)
p +
  geom_point(mapping = aes(size=log10padj,color=log10padj,alpha=log10padj)) +
  theme_classic() +
  scale_size_continuous(range = c(0,5)) +
  scale_colour_gradient2(low = "blue", mid = "red",
                         high = "darkred", midpoint = 35, space = "Lab",
                         na.value = "grey50", guide = "colourbar", aesthetics = "colour")

# title

p +
  geom_point(mapping = aes(size=log10padj,color=log10padj,alpha=log10padj)) +
  theme_classic() +
  scale_size_continuous(range = c(0,5)) +
  scale_colour_gradient2(low = "blue", mid = "red",
                         high = "darkred", midpoint = 35, space = "Lab",
                         na.value = "grey50", guide = "colourbar", aesthetics = "colour") +
  ggtitle(label = "RESTi + pB.pA vs Control")


## plot based on significant genes
head(plotData)
plotData$signUP <- ifelse(test = !is.na(plotData$padj) & plotData$padj<0.05 & plotData$log2FoldChange>0
plotData$signDOWN <- ifelse(test = !is.na(plotData$padj) & plotData$padj<0.05 & plotData$log2FoldChange<

p <- ggplot(data = plotData,mapping = aes(x = log2baseMean,y = log2FoldChange)) +
  geom_point(size=.5,color="grey") +
  theme_classic()
p
```

```
## add sign UP and DOWN
p + geom_point(data = plotData[plotData$signUP==1,],
               color= "red",size=2) +
  geom_point(data = plotData[plotData$signDOWN==1,],
             color= "blue",size=2)


## change cutoff
padj <- 0.0001
plotData$signUP <- ifelse(test = !is.na(plotData$padj) & plotData$padj<padj & plotData$log2FoldChange>0
plotData$signDOWN <- ifelse(test = !is.na(plotData$padj) & plotData$padj<padj & plotData$log2FoldChange<

p <- ggplot(data = plotData,mapping = aes(x = log2baseMean,y = log2FoldChange)) +
  geom_point(size=.5,color="grey") +
  theme_classic()
p

## add sign UP and DOWN
p +
  geom_point(data = plotData[plotData$signUP==1,],
               color= "darkred",size=2,alpha=.6) +
  geom_point(data = plotData[plotData$signDOWN==1,],
             color= "darkblue",size=2,alpha=.6) +
  ggtitle(label = "RESTi + pB.pA",subtitle = paste("Sign genes, p-adj<",padj,sep = ""))
```

8. Write data to file

```
### write sign genes to file
# write the deseq statistics

# all sign genes
write.table(x = res.df[!is.na(res.df$padj) & res.df$padj<padj,],file = paste("allSignGenes.",padj,".txt

# print normalized count data
sign.up.names <- rownames(plotData[plotData$signUP==1,])
sign.down.names <- rownames(plotData[plotData$signDOWN==1,])

head(aCounts)

aCounts.up <- aCounts[sign.up.names,]
head(aCounts.up)
aCounts.down <- aCounts[sign.down.names,]
head(aCounts.down)

write.table(x = aCounts.down,file = paste("Count.average_Sign.down.padj.",padj,".txt",sep = ""))
write.table(x = aCounts.up,file = paste("Count.average_Sign.up.padj.",padj,".txt",sep = ""))
```