Differential Expression Analysis

Leonard Sparring

5/21/2019

```
sampleCondition <- c('BH','BH','BH','Serum','Serum','Serum')

sampleFiles <- grep("csv",list.files("/home/leo/Documents/GenomeAnalysis/GenomeAnalysis/genome_analysis/

sampleTable <- data.frame(sampleName = c('BH_1','BH_2','BH_3','Serum_1','Serum_2','Serum_3'),
                          fileName = sampleFiles,
                          condition = sampleCondition)

ddsHTSeq <- DESeqDataSetFromHTSeqCount(sampleTable = sampleTable,
                                       directory = "/home/leo/Documents/GenomeAnalysis/GenomeAnalysis/g
                                       design= ~ condition)
ddsHTSeq
```

```
## class: DESeqDataSet
## dim: 3044 6
## metadata(1): version
## assays(1): counts
## rownames(3044): DNPH1 Int-Tn_1 ... zur zwf
## rowData names(0):
## colnames(6): BH_1 BH_2 ... Serum_2 Serum_3
## colData names(1): condition
```

```
# Filtering out the hypothetical proteins
mykeep <- !grepl("KAEIAEFF*", rownames(counts(ddsHTSeq)))
names(mykeep) <- rownames(counts(ddsHTSeq))
nrow(ddsHTSeq)
```

```
## [1] 3044
```

```
ddsHTSeq <- ddsHTSeq[mykeep,]
nrow(ddsHTSeq)
```

```
## [1] 1622
```

```
#counts(ddsHTSeq)

#Pre-filtering
keep <- rowSums(counts(ddsHTSeq)) >= 10
ddsHTSeq <- ddsHTSeq[keep,]
#counts(ddsHTSeq)
nrow(ddsHTSeq)
```

```
## [1] 1598
```

```
# Differential expression analysis

ddsHTSeq <- DESeq(ddsHTSeq)
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

```
res <- results(ddsHTSeq)
res
```

```
## log2 fold change (MLE): condition Serum vs BH
## Wald test p-value: condition Serum vs BH
## DataFrame with 1598 rows and 6 columns
##                  baseMean     log2FoldChange             lfcSE
##                 <numeric>          <numeric>         <numeric>
## DNPH1    4635.04059508242  -4.08038970764076 0.0666418494854734
## Int-Tn_1 2205.86144127244   1.24269140348199 0.0723663399621564
## Int-Tn_2 2622.65996329734  0.521634748107719 0.0631981487471043
## abfA_1   310.968697238851   0.61752847076642  0.129106086108519
## abfA_2   61.5152044199668  0.175509482479994  0.272113948536257
## ...                    ...                ...                ...
## znuC_3   1058.03583813624 -0.819452012849907  0.082181939954124
## zosA     4529.15761589868   -2.1570306723654 0.0645943574678357
## zupT     1494.62126179883 -0.110114967484217 0.0801282434086996
## zur      608.602947790472  -1.09130111278351  0.112450966515208
## zwf      6248.75661743442 -0.675380498183197 0.0543632765293364
##                      stat               pvalue                 padj
##                 <numeric>            <numeric>            <numeric>
## DNPH1      -61.22863844783                    0                    0
## Int-Tn_1  17.1722295770638   4.2863436550845e-66  1.44201624438422e-65
## Int-Tn_2  8.25395614347991  1.53235605390003e-16   2.7606594973306e-16
## abfA_1    4.78310890973305  1.72604542672381e-06  2.47818561716501e-06
## abfA_2   0.644985247629116     0.518936745871944     0.544491739923419
## ...                    ...                  ...                  ...
## znuC_3   -9.97119334621871  2.03765667530179e-23  4.16390711909496e-23
## zosA     -33.3934844609218 1.70457114810273e-244 1.53892920602721e-243
## zupT     -1.37423413767563     0.169369005848513     0.191273265968851
## zur      -9.70468415347877  2.87966798596027e-22  5.73779232115276e-22
## zwf      -12.4234693215877  1.94921589174396e-35  4.69811009804953e-35
```

```
# Ordered genes by log2foldchanges

resOrdered <- res[order(res$log2FoldChange),]

summary(res)
```

```
##
## out of 1598 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 683, 43%
## LFC < 0 (down)     : 690, 43%
## outliers [1]       : 0, 0%
## low counts [2]     : 0, 0%
## (mean count < 2)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
res05 <- results(ddsHTSeq, alpha=0.05)
summary(res05)
```

```
##
## out of 1598 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 661, 41%
## LFC < 0 (down)     : 675, 42%
## outliers [1]       : 0, 0%
## low counts [2]     : 0, 0%
## (mean count < 2)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
sum(res$padj < 0.1, na.rm=TRUE) # Way too many?
```

```
## [1] 1373
```

```
# Exporting most significant genes to csv

resSig <- subset(resOrdered, padj < 0.01)
resSig
```

```
## log2 fold change (MLE): condition Serum vs BH
## Wald test p-value: condition Serum vs BH
## DataFrame with 1282 rows and 6 columns
##                baseMean    log2FoldChange             lfcSE
##               <numeric>         <numeric>         <numeric>
## lacS   1243.13434446642 -6.00710949894311  0.301890779522987
## lacC_2 14064.2950255556   -5.901952429736  0.102088067639778
## acpA_1 22083.7400890089 -5.55569969793153 0.0905417865232242
## fabD   26935.9170695946 -5.52105495958673 0.0719063540285022
## fruA_2 12854.3809841882 -5.38089872325391 0.0606233643616279
## ...                ...               ...               ...
## purF   45012.3134250264  8.92249174501702   0.41214276147301
## purL   96387.3912602996  9.21155327406957   0.37895597657934
## purQ    31078.115940355  9.25048821328815  0.575296911463555
## purC   32734.5694609888  9.37024486639198  0.282026319308283
## purS     8026.641979105  9.83602431504477   0.20304826691937
##                    stat               pvalue                 padj
##               <numeric>            <numeric>            <numeric>
## lacS   -19.8982874151866  4.21071413661346e-88  1.71651050773171e-87
## lacC_2 -57.8123630526661                     0                    0
## acpA_1 -61.3606149300631                     0                    0
## fabD   -76.7811834458787                     0                    0
## fruA_2 -88.7594870379677                     0                    0
## ...                 ...                   ...                  ...
## purF    21.6490317896832 6.20705460535326e-104 2.82588981747992e-103
## purL    24.3077134109824 1.62470464874409e-130 8.92191762437475e-130
## purQ    16.0795026515177  3.55219219670719e-58  1.09795031534586e-57
## purC    33.2247177829859 4.73351147315531e-242 4.22578286821351e-241
## purS    48.4418038345072                     0                    0
```

```
write.csv(as.data.frame(resSig),
          file="condition_Serum_vs_BH.csv")
```

```
write.csv(as.data.frame(resSig[ order( -resSig$log2FoldChange, -resSig$baseMean ),]), file="UpRegulated
```

```
# Log fold change shrinkage for visualization and ranking

resultsNames(ddsHTSeq)
```

```
## [1] "Intercept"             "condition_Serum_vs_BH"
```

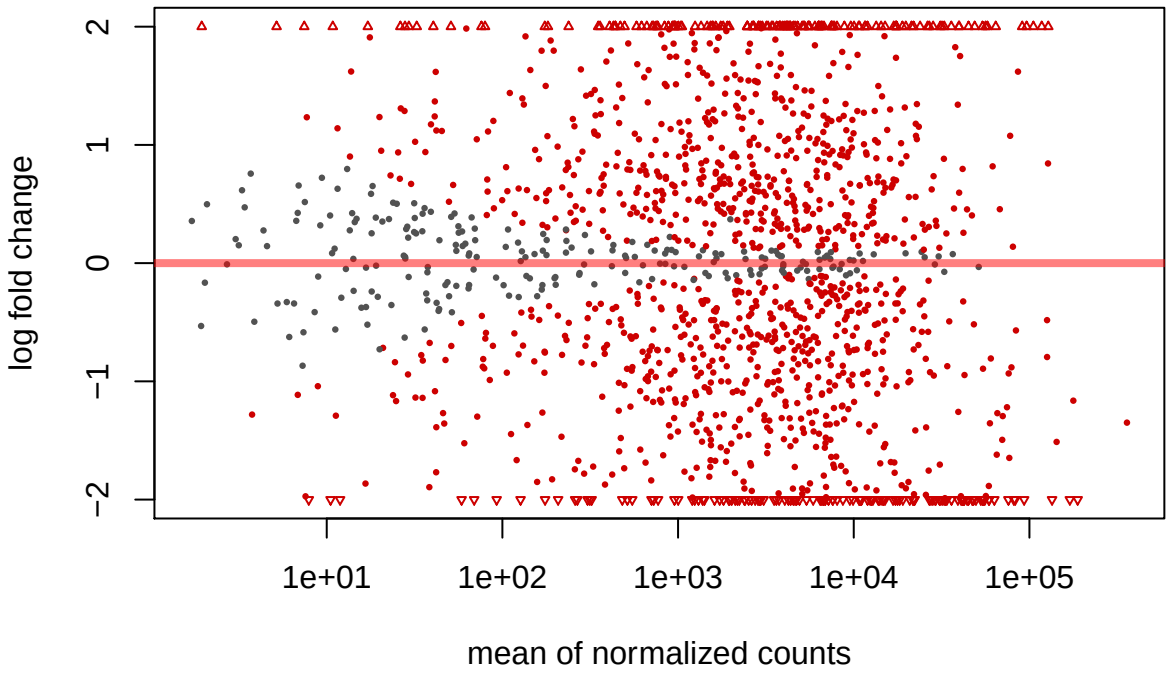
```
resLFC <- lfcShrink(ddsHTSeq, coef="condition_Serum_vs_BH", type="apeglm")
```

```
## using 'apeglm' for LFC shrinkage. If used in published research, please cite:
##     Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for
##     sequence count data: removing the noise and preserving large differences.
##     Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895
```

```
resLFC
```

```
## log2 fold change (MAP): condition Serum vs BH
## Wald test p-value: condition Serum vs BH
## DataFrame with 1598 rows and 5 columns
##                  baseMean     log2FoldChange              lfcSE
##                 <numeric>          <numeric>          <numeric>
## DNPH1    4635.04059508242  -4.07840903143185 0.0666389839811106
## Int-Tn_1 2205.86144127244   1.23920358572849 0.0722543643063449
## Int-Tn_2 2622.65996329734  0.519920138143993 0.0631085256048916
## abfA_1   310.968697238851  0.609369455242493  0.128737633520562
## abfA_2   61.5152044199668  0.164488378172411  0.263365120778065
## ...                   ...                ...                ...
## znuC_3   1058.03583813624 -0.815360875004469 0.0820873083697473
## zosA     4529.15761589868  -2.15429476749742  0.064588205717323
## zupT     1494.62126179883  -0.10934262660239 0.0798727096994216
## zur      608.602947790472  -1.08274574512776  0.112365846293235
## zwf      6248.75661743442  -0.67437362804007 0.0543263035903089
##                        pvalue                 padj
##                     <numeric>            <numeric>
## DNPH1                       0                    0
## Int-Tn_1   4.2863436550845e-66  1.44201624438422e-65
## Int-Tn_2  1.53235605390003e-16   2.7606594973306e-16
## abfA_1    1.72604542672381e-06  2.47818561716501e-06
## abfA_2       0.518936745871944     0.544491739923419
## ...                        ...                  ...
## znuC_3    2.03765667530179e-23  4.16390711909496e-23
## zosA     1.70457114810273e-244 1.53892920602721e-243
## zupT         0.169369005848513     0.191273265968851
## zur       2.87966798596027e-22  5.73779232115276e-22
## zwf       1.94921589174396e-35  4.69811009804953e-35
```

```
# Plot of shrunken log2 fold changes
plotMA(resLFC, ylim=c(-2,2))
```



Heatmap of downregulated differentially expressed genes. normalized counts ordered by most differentially log fold changes

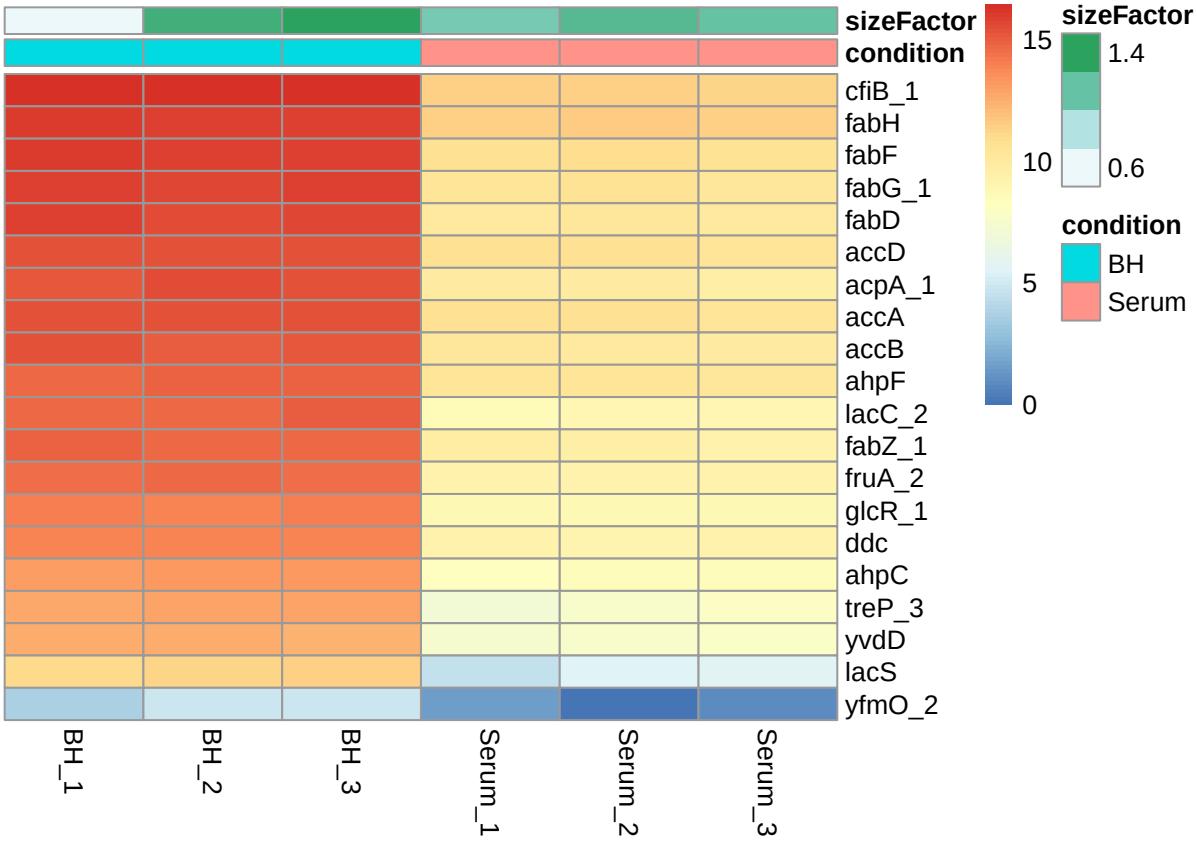
```
library("pheatmap")

ntd <- normTransform(ddsHTSeq)

select <- order(rowMeans(counts(ddsHTSeq,normalized=TRUE)),
                decreasing=TRUE)

rowMeansCounts <- assay(ntd)[select,]
topVarGenes <- subset(rowMeansCounts, rownames(rowMeansCounts) %in% row.names(resOrdered)[1:20])


df <- as.data.frame(colData(ddsHTSeq)[,c("condition","sizeFactor")])
pheatmap(topVarGenes, cluster_rows=FALSE, show_rownames=TRUE,
         cluster_cols=FALSE, annotation_col=df)
```



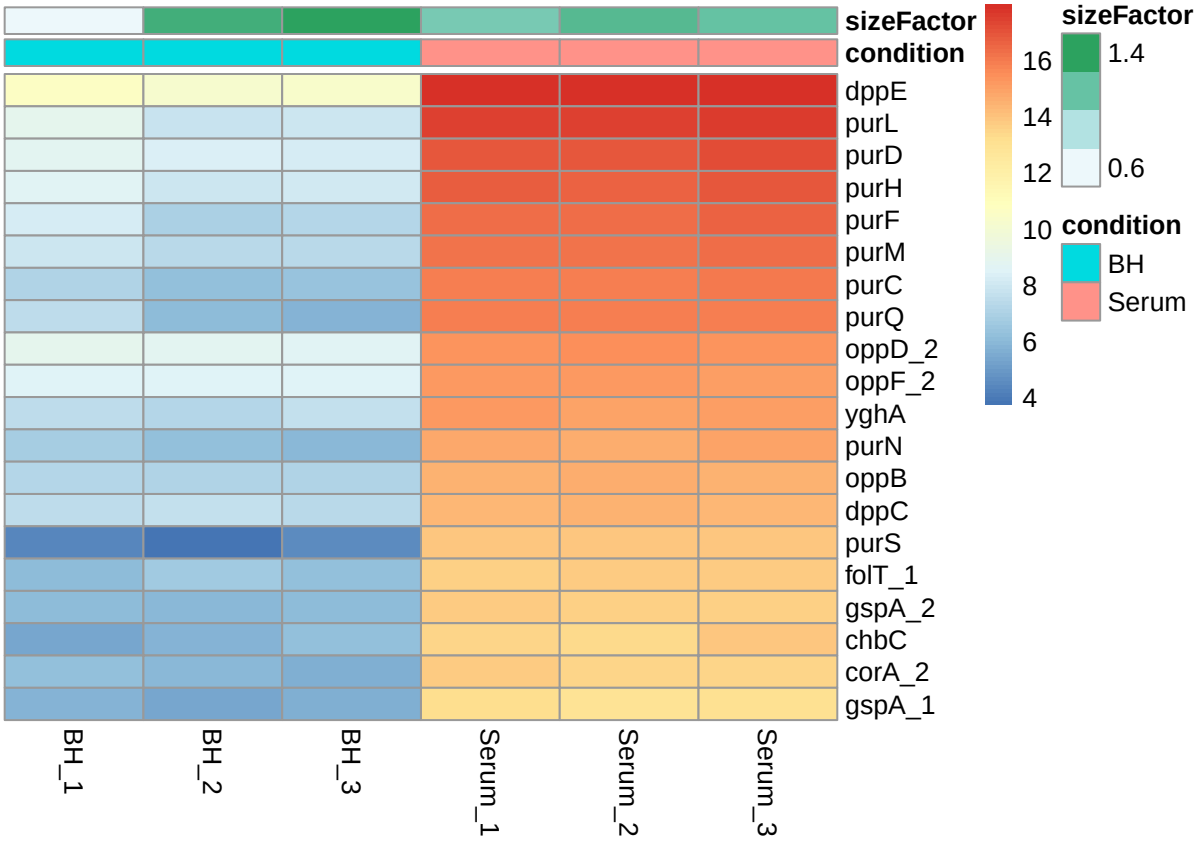
DE of upregulated genes

```
select <- order(rowMeans(counts(ddsHTSeq,normalized=TRUE)),
                decreasing=TRUE)

resNegOrdered <- res[order(-res$log2FoldChange),]

rowMeansCounts <- assay(ntd)[select,]
topVarNegGenes <- subset(rowMeansCounts, rownames(rowMeansCounts) %in% row.names(resNegOrdered)[1:20])


df2 <- as.data.frame(colData(ddsHTSeq)[,c("condition","sizeFactor")])
pheatmap(topVarNegGenes, cluster_rows=FALSE, show_rownames=TRUE,
         cluster_cols=FALSE, annotation_col=df2)
```



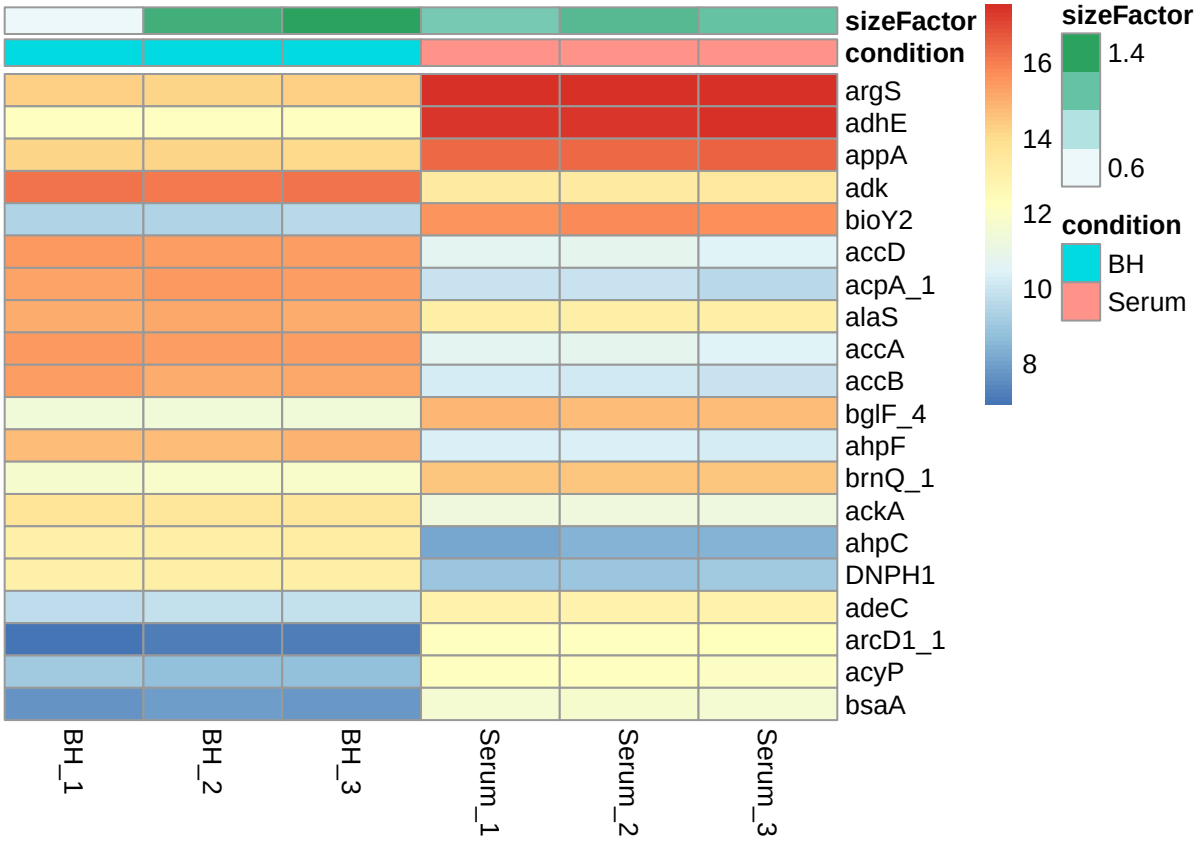
Ordered by p-value

```
select <- order(rowMeans(counts(ddsHTSeq,normalized=TRUE)),
                decreasing=TRUE)

resPvalOrdered <- res[order(res$pvalue),]

rowMeansCounts <- assay(ntd)[select,]
topVarPvalGenes <- subset(rowMeansCounts, rownames(rowMeansCounts) %in% row.names(resPvalOrdered)[1:20])


df3 <- as.data.frame(colData(ddsHTSeq)[,c("condition","sizeFactor")])
pheatmap(topVarPvalGenes, cluster_rows=FALSE, show_rownames=TRUE,
         cluster_cols=FALSE, annotation_col=df3)
```


sizeFactor
condition
argS
adhE
appA
adk
bioY2
accD
acpA_1
alaS
accA
accB
bglF_4
ahpF
brnQ_1
ackA
ahpC
DNPH1
adeC
arcD1_1
acyP
bsaA
BH_1
BH_2
BH_3
Serum_1
Serum_2
Serum_3
16
14
12
10
8
sizeFactor
1.4
0.6
condition
BH
Serum