ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

nexus
Personalized Health Technologies

PhD course 'Basic & Applied Cancer Biology'

Exercise

Michael Prummer, Nexus Personalized Health Technologies, ETH Zürich

November 23, 2015

Getting started

1. Install R
    - Download from http://stat.ethz.ch/CRAN/
2. Install Rstudio
    - Download from https://www.rstudio.com
3. Install packages

```r
install.packages("ggplot2", repos = "http://stat.ethz.ch/CRAN/")
source("https://bioconductor.org/biocLite.R")
biocLite(c("DESeq2","airway"))
```

4. Load the packages and start with a fresh workspace:

```r
library(DESeq2); library(ggplot2); library(airway)
rm(list=ls())
```

Data set description

- RNA-Seq experiment of airway smooth muscle cells
- Treatment: +/- 1 µM Dexamethasone for 18h; a glucocorticoid used in asthma patients to prevent or reduce inflammation of the airways.
- Cell lines: 4 primary human airway smooth muscle cell lines
- Reference

Himes BE, Jiang X, Wagner P, Hu R, Wang Q, Klanderman B, Whitaker RM, Duan Q, Lasky-Su J, Nikolos C, Jester W, Johnson M, Panettieri R Jr, Tantisira KG, Weiss ST, Lu Q. "RNA-Seq Transcriptome Profiling Identifies CRISPLD2 as a Glucocorticoid Responsive Gene that Modulates Cytokine Function in Airway Smooth Muscle Cells." PLoS One. 2014 Jun 13;9(6):e99625. PMID: 24926665. GEO: GSE52778.

Load the data

```
data("airway")
```

Inspect the data tables

Count table

```
assay(airway)[1:5,1:5]

##                 SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516
## ENSG00000000003        679        448        873        408       1138
## ENSG00000000005          0          0          0          0          0
## ENSG00000000419        467        515        621        365        587
## ENSG00000000457        260        211        263        164        245
## ENSG00000000460         60         55         40         35         78

dim(assay(airway))

## [1] 64102     8
```

64102 genes, 8 samples.

Total reads per sample:

```
colSums(assay(airway))

## SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516 SRR1039517
##   20637971   18809481   25348649   15163415   24448408   30818215
## SRR1039520 SRR1039521
##   19126151   21164133
```

Experimental meta data

```
colData(airway)

## DataFrame with 8 rows and 9 columns
##            SampleName     cell      dex    albut        Run avgLength
##              <factor> <factor> <factor> <factor>   <factor> <integer>
## SRR1039508 GSM1275862   N61311    untrt    untrt SRR1039508       126
## SRR1039509 GSM1275863   N61311      trt    untrt SRR1039509       126
## SRR1039512 GSM1275866  N052611    untrt    untrt SRR1039512       126
## SRR1039513 GSM1275867  N052611      trt    untrt SRR1039513        87
## SRR1039516 GSM1275870  N080611    untrt    untrt SRR1039516       120
## SRR1039517 GSM1275871  N080611      trt    untrt SRR1039517       126
## SRR1039520 GSM1275874  N061011    untrt    untrt SRR1039520       101
## SRR1039521 GSM1275875  N061011      trt    untrt SRR1039521        98
##            Experiment    Sample    BioSample
##              <factor>  <factor>     <factor>
## SRR1039508  SRX384345 SRS508568 SAMN02422669
## SRR1039509  SRX384346 SRS508567 SAMN02422675
## SRR1039512  SRX384349 SRS508571 SAMN02422678
## SRR1039513  SRX384350 SRS508572 SAMN02422670
## SRR1039516  SRX384353 SRS508575 SAMN02422682
## SRR1039517  SRX384354 SRS508576 SAMN02422673
## SRR1039520  SRX384357 SRS508579 SAMN02422683
## SRR1039521  SRX384358 SRS508580 SAMN02422677
```

2 conditions: cell and dex.

```
table(colData(airway)$cell, colData(airway)$dex)

##
##           trt untrt
##   N052611   1     1
##   N061011   1     1
##   N080611   1     1
##   N61311    1     1
```

One sample per cell-treatment combination, 4 cell types.

Generate the `DESeqDataSet`

```
dds <- DESeqDataSet(airway, design = ~ cell + dex)
```

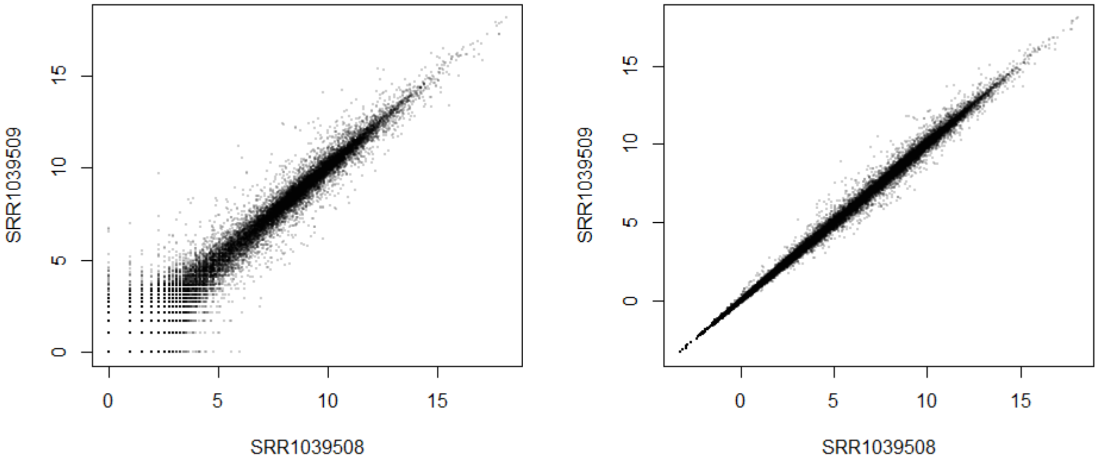
Visually exploring the dataset

Regularized log transform

```
rld <- rlog(dds)
head(assay(rld))

##                 SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516
## ENSG00000000003   9.399151   9.142478   9.501695   9.320796   9.757212
## ENSG00000000005   0.000000   0.000000   0.000000   0.000000   0.000000
## ENSG00000000419   8.901283   9.113976   9.032567   9.063925   8.981930
## ENSG00000000457   7.949897   7.882371   7.834273   7.916459   7.773819
## ENSG00000000460   5.849521   5.882363   5.486937   5.770334   5.940407
## ENSG00000000938  -1.638084  -1.637483  -1.558248  -1.636072  -1.597606
##                 SRR1039517 SRR1039520 SRR1039521
## ENSG00000000003   9.512183   9.617378   9.315309
## ENSG00000000005   0.000000   0.000000   0.000000
## ENSG00000000419   9.108531   8.894830   9.052303
## ENSG00000000457   7.886645   7.946411   7.908338
## ENSG00000000460   5.663847   6.107733   5.907824
## ENSG00000000938  -1.639362  -1.637608  -1.637724
```

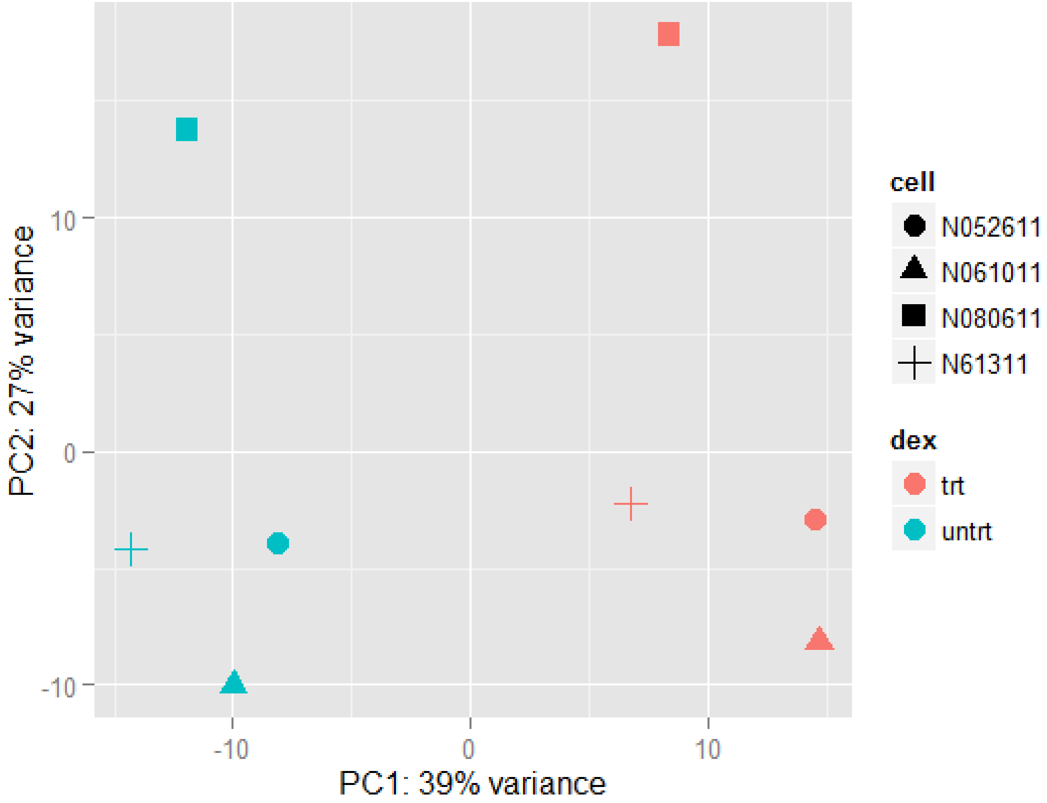
Comparison with unmodified log2

```
opar <- par( mfrow = c( 1, 2 ) )
dds <- estimateSizeFactors(dds)
plot( log2( 1 + counts(dds, normalized=TRUE)[ , 1:2] ),
     col=rgb(0,0,0,.2), pch=16, cex=0.3 )
plot( assay(rld)[ , 1:2],
     col=rgb(0,0,0,.2), pch=16, cex=0.3 )

par(opar)
```



Sample similarities: PCA plot

```
(data <- plotPCA(rld, intgroup = c("dex", "cell"), returnData=T))

##                   PC1        PC2           group   dex    cell       name
## SRR1039508 -14.331359  -4.208796  untrt : N61311 untrt  N61311 SRR1039508
## SRR1039509   6.754169  -2.245244    trt : N61311   trt  N61311 SRR1039509
## SRR1039512  -8.130393  -3.952904 untrt : N052611 untrt N052611 SRR1039512
## SRR1039513  14.505648  -2.941862   trt : N052611   trt N052611 SRR1039513
## SRR1039516 -11.891410  13.735002 untrt : N080611 untrt N080611 SRR1039516
## SRR1039517   8.373975  17.823844   trt : N080611   trt N080611 SRR1039517
## SRR1039520  -9.965898 -10.014674 untrt : N061011 untrt N061011 SRR1039520
## SRR1039521  14.685269  -8.195366   trt : N061011   trt N061011 SRR1039521

percentVar <- round(100 * attr(data, "percentVar"))
ggplot(data, aes(x=PC1, y=PC2, color=dex, shape=cell)) + geom_point(size=4) +
 xlab(paste0("PC1: ", percentVar[1], "% variance")) +
 ylab(paste0("PC2: ", percentVar[2], "% variance"))
```



66% of the total variance are visible in the projection.

Large cell specific differences along PC2 are present in parallel to treatment specific differences along PC1. Good quality data.

Differential expression analysis

Running the pipeline

Make sure the correct reference level "untrt" is chosen:

```
levels(dds$dex)

## [1] "trt"   "untrt"

dds$dex <- relevel(dds$dex, "untrt")
res <- DESeq(dds)

## using pre-existing size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing

d.res <- results(res)
```

Inspecting the results table

```
summary(d.res)

##
## out of 33469 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 2641, 7.9%
## LFC < 0 (down)    : 2242, 6.7%
## outliers [1]      : 0, 0%
## low counts [2]    : 15441, 46%
## (mean count < 5)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

mcols(d.res, use.names=TRUE)

## DataFrame with 6 rows and 2 columns
##                        type                               description
##                 <character>                               <character>
## baseMean       intermediate mean of normalized counts for all samples
## log2FoldChange      results  log2 fold change (MAP): dex trt vs untrt
## lfcSE               results          standard error: dex trt vs untrt
## stat                results          Wald statistic: dex trt vs untrt
## pvalue              results       Wald test p-value: dex trt vs untrt
## padj                results                      BH adjusted p-values
```

Automatic pre-filtering was enabled.

```
length(which(is.na(d.res$padj)))

## [1] 46074

round(min(d.res$baseMean[!is.na(d.res$padj)]))

## [1] 5
```

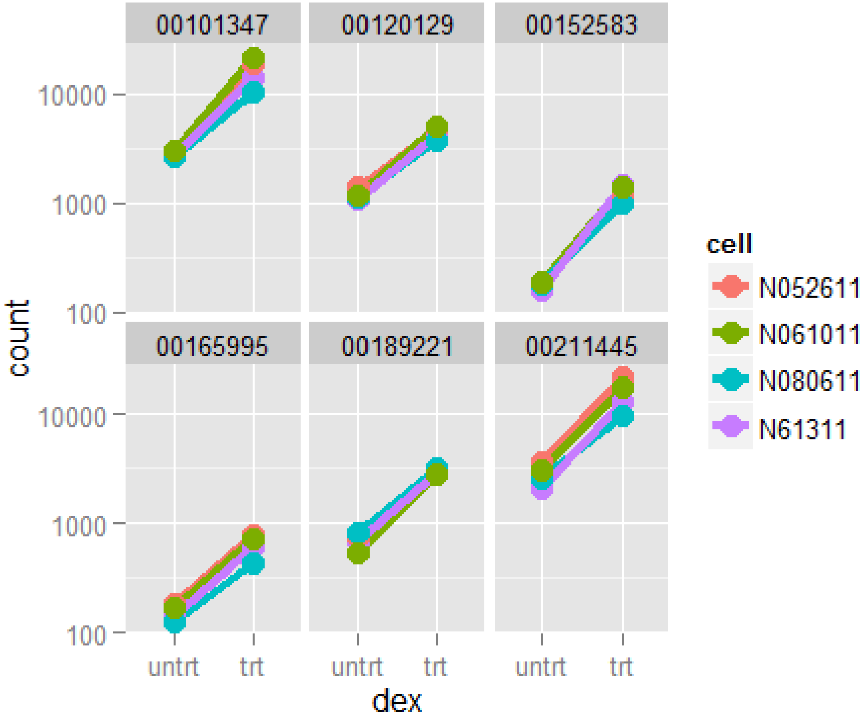
46074 genes with mean counts < 5 were excluded.

Low count reads contain comparably less information than high count reads. Excluding those should improve the quality of the results by reducing false positive findings.

Diagnostic plots

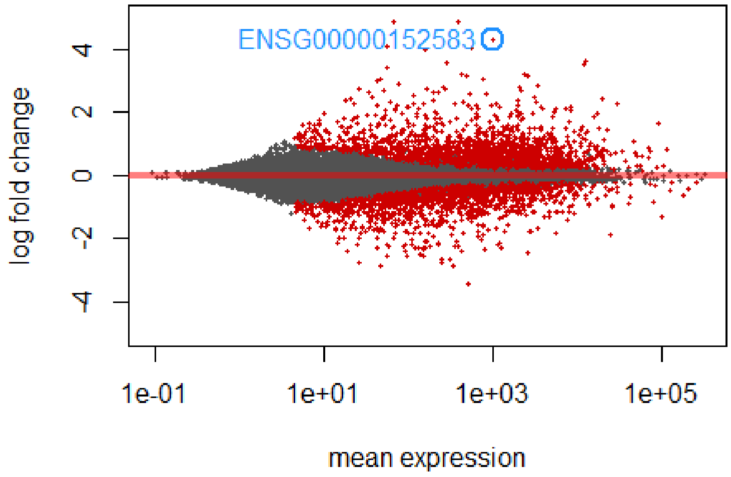
Top 6 DEG's

```
r.sig = d.res[which(d.res$padj < 0.05),]
r.sig[order(r.sig$padj), ][1:5,]

## log2 fold change (MAP): dex trt vs untrt
## Wald test p-value: dex trt vs untrt
## DataFrame with 5 rows and 6 columns
##                   baseMean log2FoldChange     lfcSE      stat
##                  <numeric>      <numeric> <numeric> <numeric>
## ENSG00000152583   997.4398       4.316100 0.1724127  25.03354
## ENSG00000165995   495.0929       3.188698 0.1277441  24.96160
## ENSG00000101347 12703.3871       3.618232 0.1499441  24.13054
## ENSG00000120129  3409.0294       2.871326 0.1190334  24.12201
## ENSG00000189221  2341.7673       3.230629 0.1373644  23.51868
##                        pvalue          padj
##                     <numeric>     <numeric>
## ENSG00000152583 2.637881e-138 4.755573e-134
## ENSG00000165995 1.597973e-137 1.440413e-133
## ENSG00000101347 1.195378e-128 6.620010e-125
## ENSG00000120129 1.468829e-128 6.620010e-125
## ENSG00000189221 2.627083e-122 9.472210e-119

topGenes = order(d.res$pad)[1:6]
data = data.frame(count = as.vector(2^assay(rld)[topGenes,]))
data$Gene = rep(rownames(assay(rld))[topGenes], ncol(assay(rld)))
data$Gene = substr(data$Gene, 8, 15)
data$sample = rep(colnames(assay(rld)), each=6)
data$cell = rep(colData(rld)$cell, each=6)
data$dex = rep(colData(rld)$dex, each=6)
data$dex = factor(data$dex, levels=c("untrt", "trt"))
ggplot(data, aes(x=dex, y=count, col=cell, group=cell)) +
  scale_y_log10() + geom_line(size=1.5) +
  geom_point(size=4) +
  facet_wrap(~Gene, nrow=2)
```


00101347
00120129
00152583
00165995
00189221
00211445
count
10000
1000
100
untrt
trt
dex
cell
N052611
N061011
N080611
N61311

MvA plot

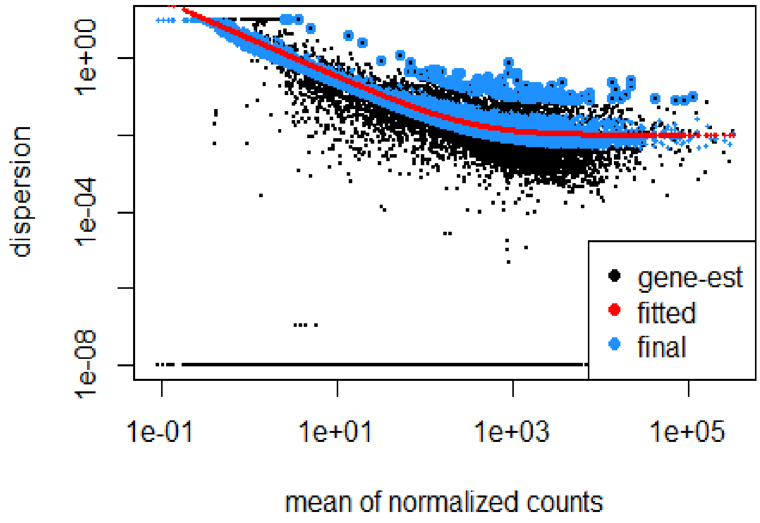
```r
plotMA(d.res, ylim=c(-5,5))
points(d.res$baseMean[topGenes[1]], d.res$log2FoldChange[topGenes[1]],
       col="dodgerblue", cex=2, lwd=2)
text(d.res$baseMean[topGenes[1]], d.res$log2FoldChange[topGenes[1]],
     rownames(d.res)[topGenes[1]], pos=2, col="dodgerblue")
```



Looks good, no comment.

Dispersion plot

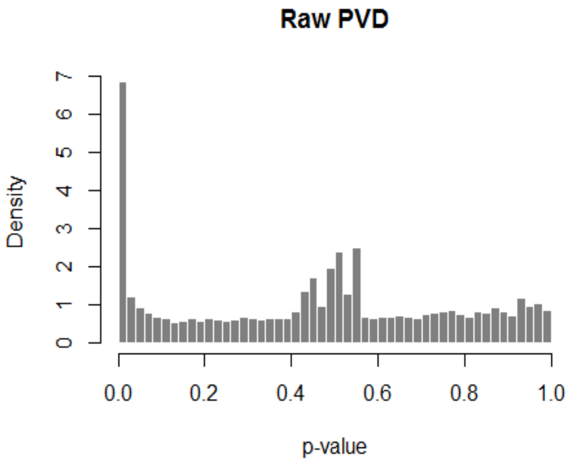
```r
plotDispEsts(res)
```



The gene group-wise dispersion estimate (red line) neems to fit well the general dispersion-mean-relation. Potentially underestimated gene dispersions (black dots at 10^{-8}) are shrinked towards the gene group-wise dispersion estimate. Genes with unusually high dispersion are not shrinked to avoid false positives (blue circle around black dots).

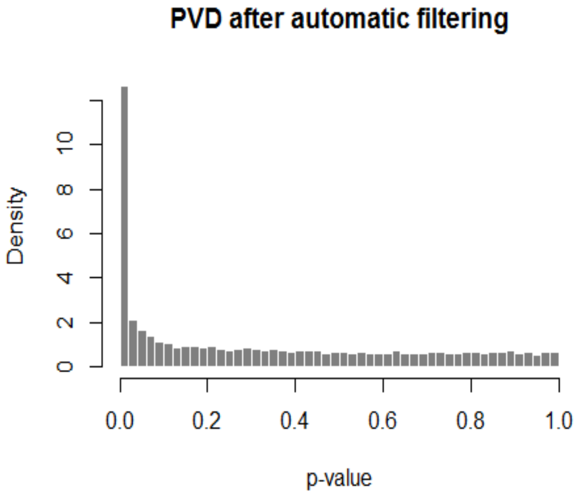
p-value distribution

```
hist(d.res$pvalue, breaks=50, freq = F, col="grey50", border="white", xlab="p-value
", main="Raw PVD")
```



Indications of an irregularity in the PVD: over-representation (bump) of p-values at around p = 0.5. Try to avoid this by additional independent filtering:

```
hist(d.res$pvalue[!is.na(d.res$padj)], breaks=50, freq=F, col="grey50", border="whi
te", xlab="p-value", main="PVD after automatic filtering")
```



Annotation: adding gene names

```
library(org.Hs.eg.db)

## Loading required package: AnnotationDbi
## Loading required package: DBI

columns(org.Hs.eg.db)

##  [1] "ACCNUM"       "ALIAS"        "ENSEMBL"      "ENSEMBLPROT" 
##  [5] "ENSEMBLTRANS" "ENTREZID"     "ENZYME"       "EVIDENCE"    
##  [9] "EVIDENCEALL"  "GENENAME"     "GO"           "GOALL"       
## [13] "IPI"          "MAP"          "OMIM"         "ONTOLOGY"    
## [17] "ONTOLOGYALL"  "PATH"         "PFAM"         "PMID"        
## [21] "PROSITE"      "REFSEQ"       "SYMBOL"       "UCSCKG"      
## [25] "UNIGENE"      "UNIPROT"

d.res$hgnc_symbol <-
    unname(mapIds(org.Hs.eg.db, rownames(d.res), "SYMBOL", "ENSEMBL"))
d.res$entrezgene <-
    unname(mapIds(org.Hs.eg.db, rownames(d.res), "ENTREZID", "ENSEMBL"))
```

Now the results have the desired external gene ids:

```
resOrdered <- d.res[order(d.res$pvalue),]
head(resOrdered)

## log2 fold change (MAP): dex trt vs untrt
## Wald test p-value: dex trt vs untrt
## DataFrame with 6 rows and 8 columns
##                    baseMean log2FoldChange     lfcSE      stat
##                   <numeric>      <numeric> <numeric> <numeric>
## ENSG00000152583   997.4398       4.316100 0.1724127  25.03354
## ENSG00000165995   495.0929       3.188698 0.1277441  24.96160
## ENSG00000101347 12703.3871       3.618232 0.1499441  24.13054
## ENSG00000120129  3409.0294       2.871326 0.1190334  24.12201
## ENSG00000189221  2341.7673       3.230629 0.1373644  23.51868
## ENSG00000211445 12285.6151       3.552999 0.1589971  22.34631
##                         pvalue          padj hgnc_symbol  entrezgene
##                      <numeric>     <numeric> <character> <character>
## ENSG00000152583 2.637881e-138 4.755573e-134     SPARCL1        8404
## ENSG00000165995 1.597973e-137 1.440413e-133      CACNB2         783
## ENSG00000101347 1.195378e-128 6.620010e-125      SAMHD1       25939
## ENSG00000120129 1.468829e-128 6.620010e-125       DUSP1        1843
## ENSG00000189221 2.627083e-122 9.472210e-119        MAOA        4128
## ENSG00000211445 1.311440e-110 3.940441e-107        GPX3        2878
```

Exporting results

```
write.csv(as.data.frame(resOrdered), file="results.csv")
```

Appendix

Other comparisons

```r
results(res, contrast=c("cell", "N061011", "N61311"))
```

```
## log2 fold change (MAP): cell N061011 vs N61311
## Wald test p-value: cell N061011 vs N61311
## DataFrame with 64102 rows and 6 columns
##                   baseMean log2FoldChange     lfcSE        stat    pvalue
##                  <numeric>      <numeric> <numeric>   <numeric> <numeric>
## ENSG00000000003 708.60217     0.29055775 0.1360076  2.13633388 0.03265221
## ENSG00000000005   0.00000             NA        NA          NA         NA
## ENSG00000000419 520.29790    -0.05069642 0.1491735 -0.33984871 0.73397047
## ENSG00000000457 237.16304     0.01474463 0.1816382  0.08117584 0.93530211
## ENSG00000000460  57.93263     0.20247610 0.2807312  0.72124547 0.47075850
## ...                    ...            ...       ...         ...        ...
## LRG_94                   0             NA        NA          NA         NA
## LRG_96                   0             NA        NA          NA         NA
## LRG_97                   0             NA        NA          NA         NA
## LRG_98                   0             NA        NA          NA         NA
## LRG_99                   0             NA        NA          NA         NA
##                       padj
##                  <numeric>
## ENSG00000000003 0.2115083
## ENSG00000000005        NA
## ENSG00000000419 0.9339283
## ENSG00000000457 0.9885943
## ENSG00000000460 0.8333258
## ...                    ...
## LRG_94                  NA
## LRG_96                  NA
## LRG_97                  NA
## LRG_98                  NA
## LRG_99                  NA
```