ETH zürich

PhD course Basic & Applied Cancer Biology

Differential Gene Expression Analysis

Michael Prummer
Biostatistician
NEXUS Personalized Health Technologies
ETH Zürich

nexus
Personalized Health Technologies

Outline

- Background & statistical aspects
- 1st Workflow demo
- 1st Hands-on exercise
- Break
- 2nd Workflow demo
- 2nd Hands-on exercise

Installation instruction

R and Rstudio

1. Install R
2. Install Rstudio

- Download latest versions from:
  - http://stat.ethz.ch/CRAN/
  - https://www.rstudio.com

- Installation of relevant packages

```
source("https://bioconductor.org/biocLite.R")
biocLite(c("DESeq2", "airway"))
install.packages("ggplot2", repos = "http://stat.ethz.ch/CRAN/")
library(DESeq2); library(ggplot2); library(airway)
print("hello world"); 2 + 3
```

- Other relevant web sites:
  - https://www.r-project.org/
  - http://www.bioconductor.org/
- Most important R commands:

```
help()  # or simply ?<command>
str()   # returns the "structure
        # of an object
dim()   # the dimensions
class() # the class of an object
```

RNAseq workflow

1. **Experimental design**
2. Wet-lab
3. Sequencing
4. Alignment
5. Reduction to ‘count tables’
6. **Analysis**
7. Comprehension

Experimental design

- **Keep it simple**
  - Classical experimental designs
    - 2 group comparison
    - multiple group comparisons
    - time series
  - Avoid missing values
  - Perform a sample size estimation (PMID 23961961)

- **Replicates (rule of thumb)**
  - 1 sample per treatment:
    - qualitative only, one-hit-wonder?
  - 3-5 replicates per treatment:
    - designed experiment
    - controlled manipulation
    - cell lines / well-defined entities
    - can detect > 2-fold change
  - 10-50 replicates per treatment:
    - human subjects
  - 1000's of replicates:
    - prospective studies, e.g., SNP discovery

Experimental design

- **Avoid confounding co-variates**
  - Treatment x flowcell/machine/lab
  - Treatment x run/technician
  - Treatment x cell passage
  - Treatment x anything

- **Record known co-variates**
  - include in linear model
    - Phenotypic covariates, e.g., age, gender, …
    - Experimental covariates, e.g., lab, date of processing, …
  - do not include in linear model
    - Colliding co-variates, i.e., variables describing consequences of the experiment

Analysis

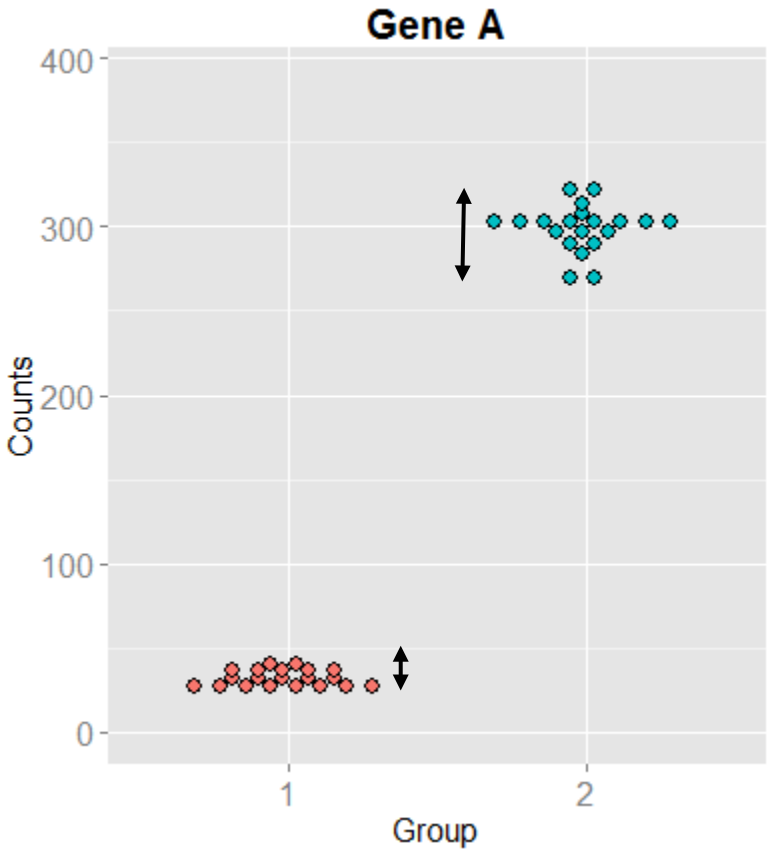
Unique statistical aspects

- Large data
  - few samples, many variables
- Univariate comparison of each gene between groups
- Each gene is analyzed by the *same* experimental design, under the *same* null hypothesis

Raw counts vs RPKM

- RPKM = Reads Per Kilobase of transcript per Million mapped reads
- Raw read counts are relevant for analysis, rather than a summary, such as, RPKM.
- Reason: for a given gene, more counts imply higher precision; RPKM etc., treat all estimates as equally informative.
- Comparison for each genomic region separately → different transcripts lengths don't matter

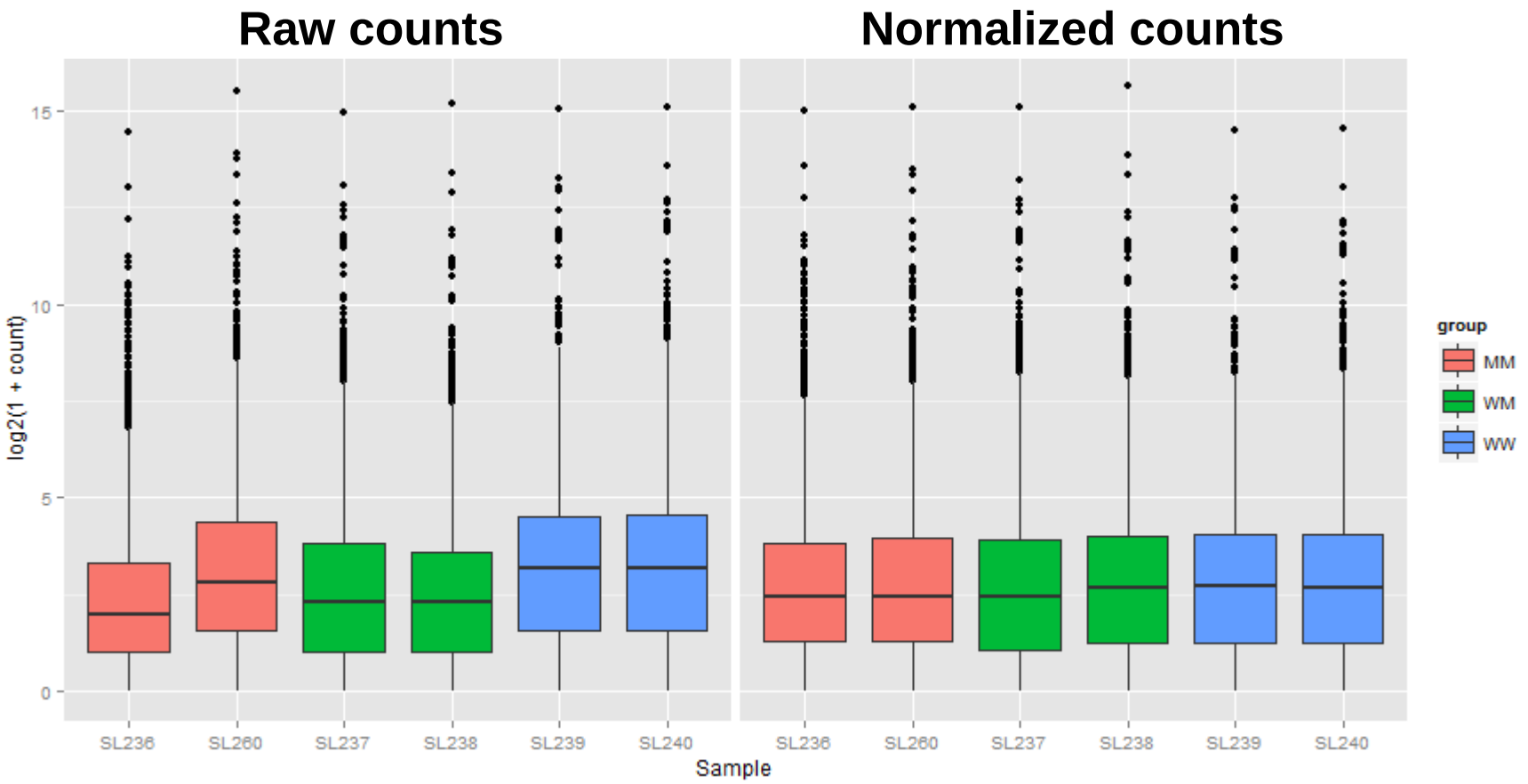
Modeling of count data



- Possible models
  - Poisson
    - Gene i, sample j, total counts per sample s_j, proportion of counts per gene & sample x_{ij}:
      Mean $\mu_{ij} = s_j x_{ij}$
      Variance $\sigma_{ij}^2 = \mu_{ij}$
    - correct for 1 sample
  - Negative binomial (NB)
    - $\sigma^2 = \mu + \mu^2 \alpha$,
      α – ***Dispersion***
    - better for capturing biological variation (multiple samples)

Normalization

- Total read counts per sample vary for uninteresting reasons



Normalization in R

- `DESeq2::estimateSizeFactors()`, Anders and Huber, 2010
  - For each gene: geometric mean of all samples.
  - For each sample: median ratio of the sample gene over the geometric mean of all samples
  - Functions other than the median can be used; control genes can be used
- `edgeR::calcNormFactors()`, TMM method of Robinson et al., 2010
  - Identify reference sample: library with upper quartile closest to the mean upper quartile of all libraries
  - Calculate M-value of each gene (log-fold change relative to reference)
  - Summarize library size as weighted trimmed mean of M-values.
- Further reading: useful comparison of normalization methods link

Dispersion estimation for NB distribution in R

- `DESeq2::estimateDispersions()`
  - Estimate per-gene dispersion
  - Fit a smoothed relationship between dispersion and abundance
  - Shrink per-gene dispersion towards the trend
- `edgeR::estimateDisp()`
  - *Common*: single dispersion for all genes;
    appropriate for small experiments (<10? samples)
  - *Tagwise*: different dispersion for all genes;
    appropriate for larger / well-behaved experiments
  - *Trended*: bin based on abundance, estimate common dispersion within bin, fit a loess-smoothed relationship between binned dispersion and abundance; appropriate for most (in-between) situations

Advanced topic: Comprehension

- Placing differentially expressed regions in context
- Gene names associated with genomic ranges
- Gene set enrichment and similar analysis
- Proximity to regulatory marks
- Integrate with other analyses, e.g., methylation, copy number, variants, …

Differential gene expression analysis

RNAseq Workshop

- RNA-Seq differential expression (DEA) workflow
- Use DESeq2 along with other *Bioconductor* packages:
  1. Start with read count table
     (the complete workflow starts from FASTQ files)
  2. Exploratory data analysis (EDA) & quality control (QC)
  3. DEA
  4. Inspection & interpretation of results
- A number of important other *Bioconductor* packages:
  Rsubread, edgeR, limma, BaySeq.

- Load required packages and start with a fresh workspace:

```
library(DESeq2)
library(ggplot2)
rm(list=ls())
```

From a matrix of counts to a DESeqDataSet

Load data from URL

- Starting point for DGE analysis: table of counts with 1 column per sample and 1 row per genetic feature, e.g., per gene or per miRNA.
- Example dataset (PMID: 20413459).
  - Matrix of counts (`mobData`) acquired for 3000 small RNA loci from Arabidopsis grafting experiments.

```
load(url("https://bios221.stanford.edu/data/mobData.RData"))
dim(mobData)
## [1] 3000 6
head(mobData)
##      SL236 SL260 SL237 SL238 SL239 SL240
## [1,]    21    52     4     4    86    68
## [2,]    18    21     1     5     1     1
## [3,]     1     2     2     3     0     0
## [4,]    68    87   270   184   396   368
## [5,]    68    87   270   183   396   368
## [6,]     1     0     6    10     6    12
```

From a matrix of counts to a DESeqDataSet

Assign group variable

MM = “triple mutant shoot grafted onto triple mutant root”
WM = “wild-type shoot grafted onto triple mutant root”
WW = “wild-type shoot grafted onto wild-type root”

```
mobDataGroups = data.frame(group = c("MM", "MM", "WM", "WM", "WW", "WW"),
                           sample = colnames(mobData))
rownames(mobDataGroups) = colnames(mobData)
mobDataGroups
##       group sample
## SL236    MM  SL236
## SL260    MM  SL260
## SL237    WM  SL237
## SL238    WM  SL238
## SL239    WW  SL239
## SL240    WW  SL240
```

From a matrix of counts to a DESeqDataSet

Make the DESeqDataSet

```
(dds <- DESeqDataSetFromMatrix(countData = mobData,
                               colData = mobDataGroups,
                               design = ~ group))
```

```
## class: DESeqDataSet
## dim: 3000 6
## metadata(0):
## assays(1): counts
## rownames: NULL
## rowRanges metadata column names(0):
## colnames(6): SL236 SL260 ... SL239 SL240
## colData names(2): group sample
```

- In case of multiple co-variates, the test is performed for the **last** one. For example, in `design = ~ month + day + treat`, the test is performed relative to the first factor level of `treat` while controlling for `month` and `day`.

Visually exploring the dataset

The rlog transformation

- Challenge: visual quality control of a huge-dimensional dataset (each gene is a dimension)
- Strategy: reduction of dimensions to 1-3
  - *Principal component analysis (PCA)*
  - *Multidimensional scaling (MDS)*
  - *Distance transform (& cluster)*
- Requirement: homoskedastic data (variance = const.)
  - *Log transform: low-count observations dominate*
- Solution: Regularized log transform
  - *Equal to log transform for high-count observations*
  - *Low-count observations are shrunk to global average*

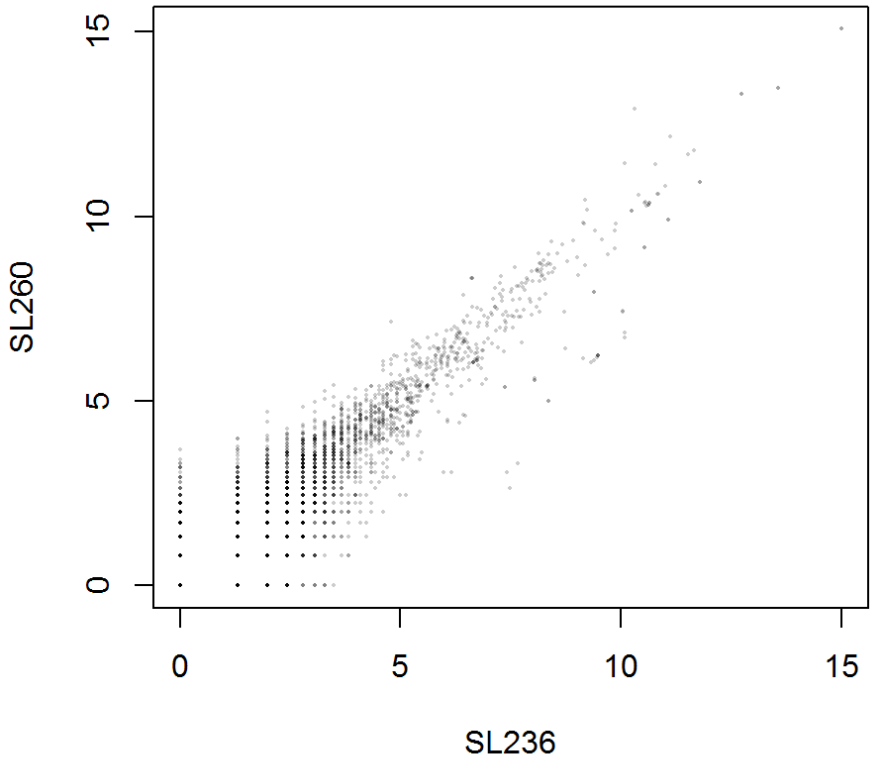
Regularized log transform

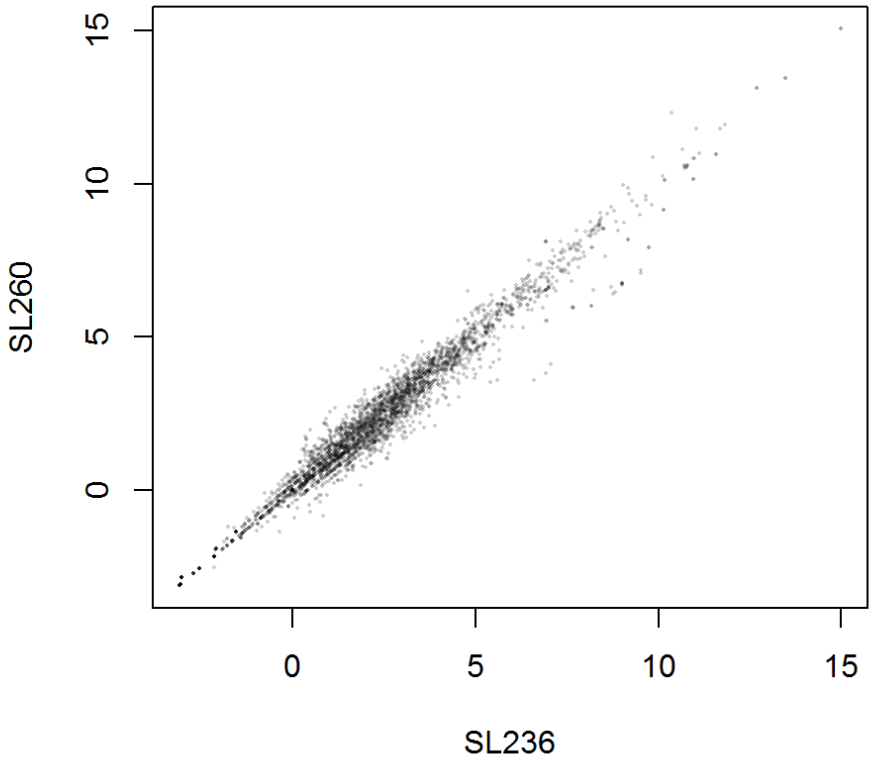
- *The function* `rlog()` *returns a* `SummarizedExperiment` *object which contains the rlog-transformed values in its* `assay()` *slot:*

```
rld <- rlog(dds)
head(assay(rld))
##          SL236     SL260     SL237     SL238       SL239      SL240
## [1,] 4.8078297 5.0331522 3.3604402 3.5026073  5.48661876  5.2108703
## [2,] 3.7532324 3.2852066 1.6670768 2.5622987  1.47413088  1.4596785
## [3,] 0.3928188 0.4039072 0.5463834 0.8386243 -0.02701533 -0.0304938
## [4,] 6.9276039 6.5271885 8.0184416 7.8416333  7.94676551  7.8334549
## [5,] 6.9271398 6.5266862 8.0180654 7.8356011  7.94638639  7.8330688
## [6,] 1.7114987 1.2230292 2.4289689 2.9891390  2.13521060  2.5960221
```

Regularized log transform

Comparison with ordinary log2

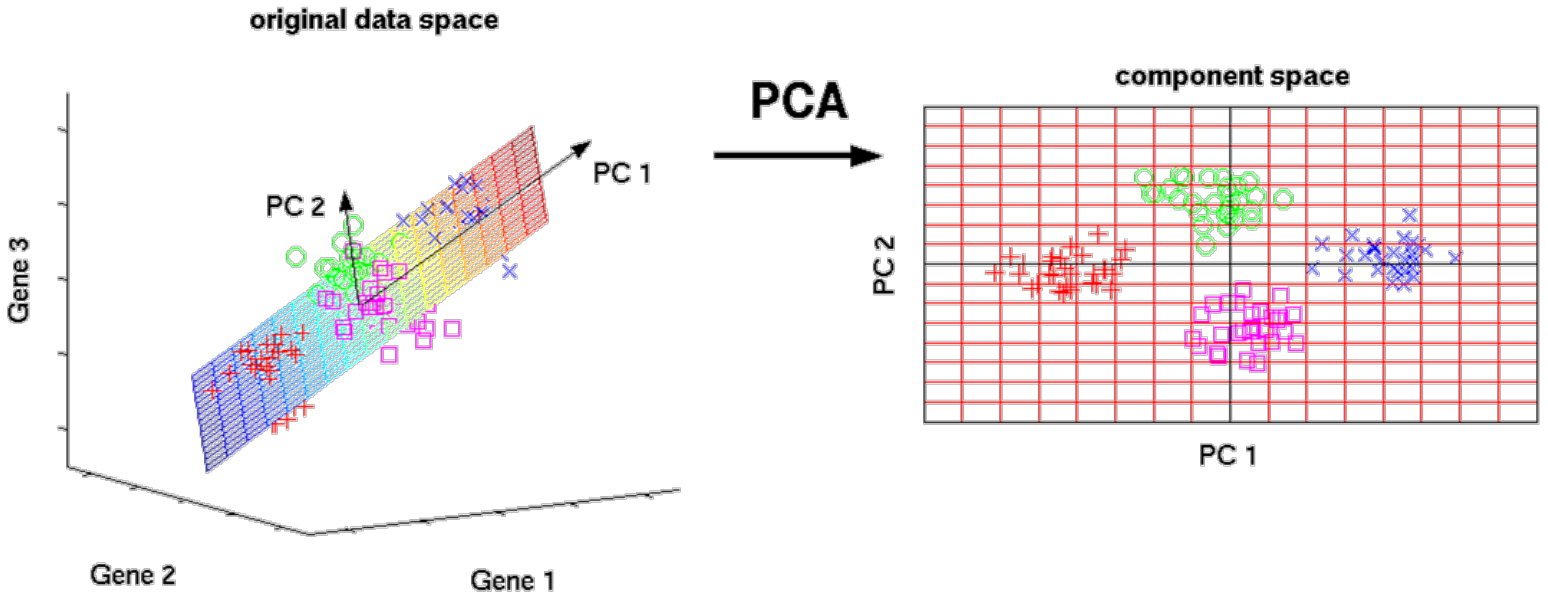




Dimension reduction by PCA

Visualizing high-dimensional data

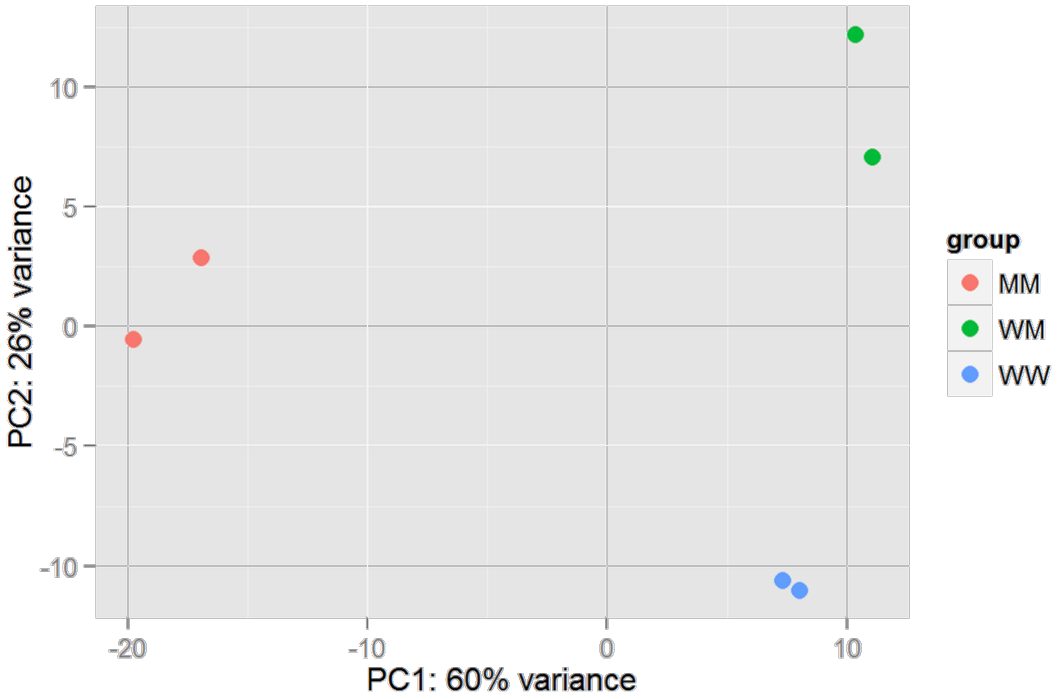
- PCA is a rotation in the n-dimensional space, followed by a projection to k<n dimensions.
- Principal component 1: direction of maximal variance
- Principal component 2: direction of 2nd largest variance, $PC2 \perp PC1$,
- …



Assessment of sample similarity

PCA plot

```
plotPCA(rld, intgroup = c( "group"))
```



- 86% of total variance are visible in the projection
- Between group differences are large compared to between sample differences
- Good quality data

Exercise 1

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.

Exercise 1

- Load data set

```
R> library(airway)
R> data("airway")
```

  To explore the data, use the functions `colData(), assay(), dim()`

- Question 1:
  a. How many samples, how many genes?
  b. Which and how many conditions?
  c. What is the experimental design formula?
- Question 2:
  a. Create a DESeqDataSet object using `DESeqDataSet( … )`
- Question 3:
  a. Describe and discuss the PCA plot.

Differential expression analysis

Running the pipeline

- Make sure the correct reference level “MM” is chosen:

```
levels(dds$group)
## [1] "MM" "WM" "WW"
# OK. if not, use relevel(); also: ?droplevels()
```

- Actual DESeq function:

```
res <- DESeq(dds) # further reading: ?DESeq()
## 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)
```

- The individual steps are described in detail in the manual page

Inspecting the results table

```
summary(d.res)
## out of 2945 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)     : 236, 8%
## LFC < 0 (down)   : 126, 4.3%
## outliers [1]     : 0, 0%
## low counts [2]   : 1084, 37%
## (mean count < 4)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

Note: DESeq tests by default the last level of the last variable in the design formula against the first level of this variable

```
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): group WW vs MM
## lfcSE               results          standard error: group WW vs MM
## stat                results          Wald statistic: group WW vs MM
## pvalue              results        Wald test p-value: group WW vs MM
## padj                results                      BH adjusted p-values
```

Default comparison: “WW” vs “MM”

```
d.res
## log2 fold change (MLE): group WW vs MM
## Wald test p-value: group WW vs MM
## DataFrame with 3000 rows and 6 columns
##       baseMean log2FoldChange     lfcSE       stat       pvalue         padj
##      <numeric>      <numeric> <numeric>  <numeric>    <numeric>    <numeric>
## 1     30.984327      0.5855865 0.5139228   1.139444 2.545178e-01 0.4848319210
## 2      8.557120     -3.8654316 0.9832647  -3.931222 8.451530e-05 0.0015572572
## 3      1.544659     -1.7452641 1.2538073  -1.391972 1.639310e-01           NA
## 4    207.063240      1.6605036 0.3766935   4.408102 1.042804e-05 0.0003093078
## 5    206.835135      1.6603646 0.3771657   4.402215 1.071512e-05 0.0003093078
## ...         ...            ...       ...        ...          ...          ...
## 2996   3.814474      0.3307716 1.0692853  0.3093389  0.757063719   0.87022581
## 2997   5.410225      1.0416062 0.9131658  1.1406540  0.254013935   0.48483192
## 2998   8.988868      1.5458153 0.7902434  1.9561256  0.050450354   0.18195370
## 2999   6.086980     -2.8066785 0.9461672 -2.9663663  0.003013414   0.02790031
## 3000   2.214956      1.3410856 1.1993575  1.1181700  0.263494394           NA
```

Specific comparison: “WM” vs “MM”

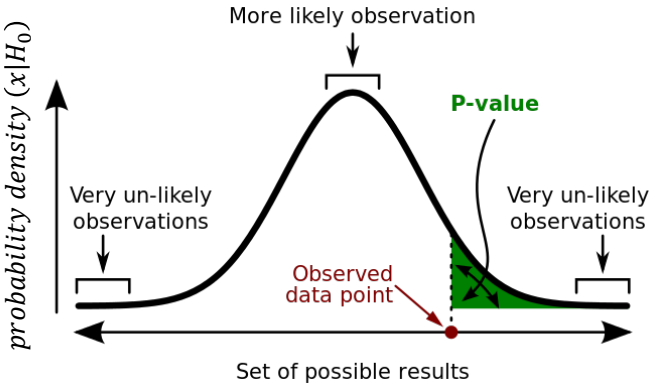
```
results(res, contrast=c("group", "WM", "MM"))
## log2 fold change (MAP): group WM vs MM
## Wald test p-value: group WM vs MM
## DataFrame with 3000 rows and 6 columns
##       baseMean log2FoldChange     lfcSE       stat       pvalue         padj
##      <numeric>      <numeric> <numeric>  <numeric>    <numeric>    <numeric>
## 1     30.984327     -2.5782426 0.6519701 -3.9545409 7.668176e-05 0.0013432528
## 2      8.557120     -2.1325605 0.9224584 -2.3118230 2.078744e-02 0.1027739198
## 3      1.544659      0.7495931 1.2321514  0.6083612 5.429480e-01           NA
## 4    207.063240      1.7189419 0.3790733  4.5345893 5.771567e-06 0.0001699685
## 5    206.835135      1.7153069 0.3795558  4.5192487 6.205948e-06 0.0001776125
## ...         ...            ...       ...        ...          ...          ...
## 2996   3.814474     -0.5352815 1.1122810 -0.4812466    0.6303412    0.8338889
## 2997   5.410225      0.4413423 0.9548843  0.4621946    0.6439418    0.8452776
## 2998   8.988868     -0.9605717 0.9135034 -1.0515250    0.2930175    0.5686835
## 2999   6.086980     -2.1644379 0.9588285 -2.2573775    0.0239845    0.1133407
## 3000   2.214956     -0.4408630 1.2384176 -0.3559890    0.7218488           NA
```

Multiple testing

p-value adjustment

- DEA is a massive multiple testing situation
- With a type I error of $\alpha = 0.05$, we expect a false positive rate of 5% in the absence of any real effect. Here, $3000 \cdot 0.05 = 150$ FPs!

- $P(\#FP = 0 | H_0\ true) = \quad (1-\alpha)^n$
- $P(\#FP \geq 1 | H_0\ true) = 1 - (1-\alpha)^n \cong 1 \text{ for } n > 100$
- Therefore, a p-value cutoff is not useful here.



- Better: controlling the False Discovery Rate:

$$FDR = \frac{FP}{FP + TP}$$

- In DESeq2, FDR is obtained using the Benjamini-Hochberg adjustment and is called `padj`.
- Number of significant DEG's:

```
sum(d.res$padj < 0.05, na.rm = T)
## [1] 246
```

Pre-filtering

- Question:
  - Can we reduce the number of tests to increase statistical power?
- Critical:
  - Need a smart choice without actually performing the test.
- Naïve and prohibited choice:
  - Omit cases with low fold-change.
- Better and allowed choice:
  - Omit cases with low mean (median) counts over all samples (`baseMean`).

Top hit list

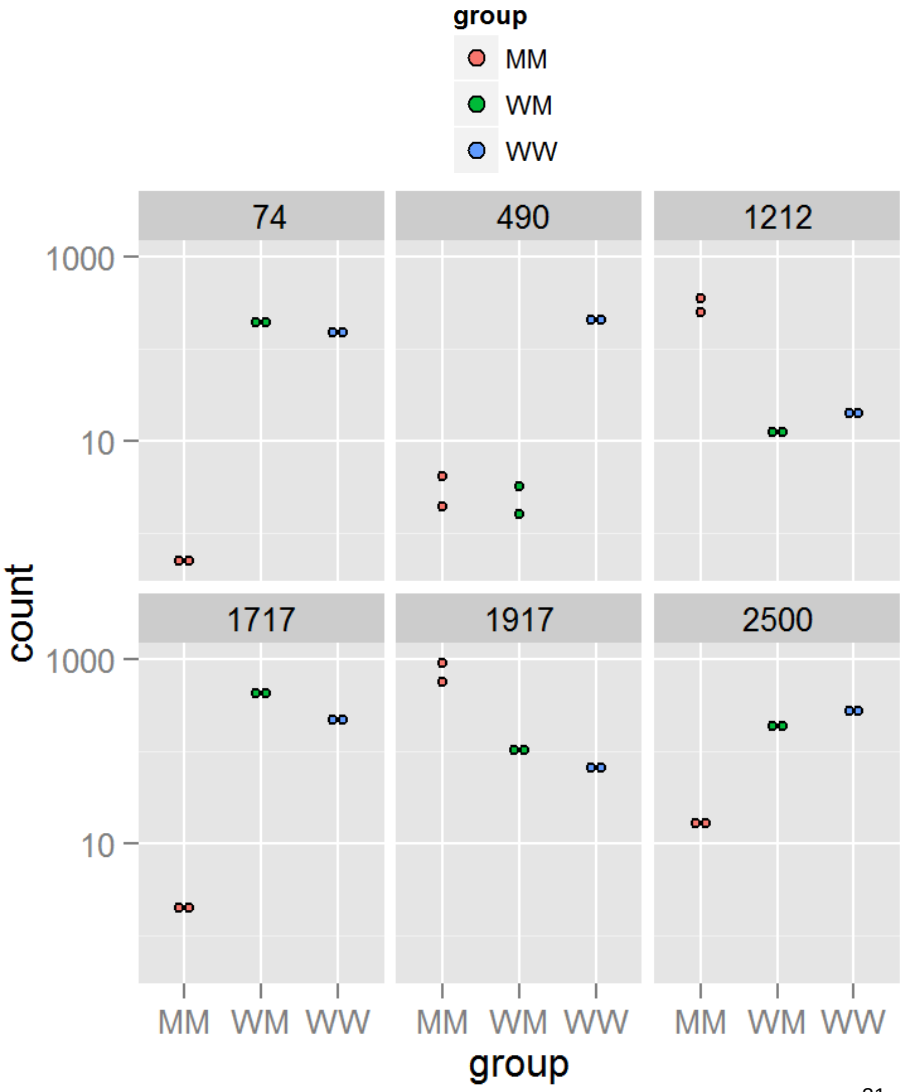
```
r.sig = d.res[which(d.res$padj < 0.05),]
head(r.sig[order(r.sig$padj), ])

## log2 fold change (MAP): group WW vs MM
## Wald test p-value: group WW vs MM
## DataFrame with 6 rows and 6 columns
##    baseMean log2FoldChange     lfcSE      stat       pvalue         padj
##   <numeric>      <numeric> <numeric> <numeric>    <numeric>    <numeric>
## 1 216.35016       6.358740 0.6513346  9.762631 1.628744e-22 3.031092e-19
## 2 159.77453       3.983793 0.4143577  9.614382 6.952528e-22 6.469327e-19
## 3  71.68716       5.821391 0.6533662  8.909844 5.110428e-19 3.170169e-16
## 4 112.71103      -3.852485 0.4346521 -8.863374 7.762848e-19 3.611665e-16
## 5 303.58163      -3.368202 0.3969663 -8.484855 2.159873e-17 8.039048e-15
## 6 116.67663       7.002300 0.8454583  8.282253 1.208661e-16 3.748864e-14
```

Diagnostic plots

Spot-check of raw counts

- Top 6 DEG's

```
topGenes = order(d.res$padj)[1:6]
data = data.frame(count =
       as.numeric(2^assay(rld)[topGenes,]))
data$Gene = rep(topGenes, ncol(assay(rld)))
data$sample = rep(colnames(assay(rld)),
                  each=6)
data$group = rep(colData(rld)$group,
                  each=6)
#str(data)
ggplot(data,
  aes(x=group, y=count, fill=group)) +
  scale_y_log10() +
  geom_dotplot(binaxis="y",
               stackdir="center") +
  facet_wrap(~Gene, nrow=2)
```

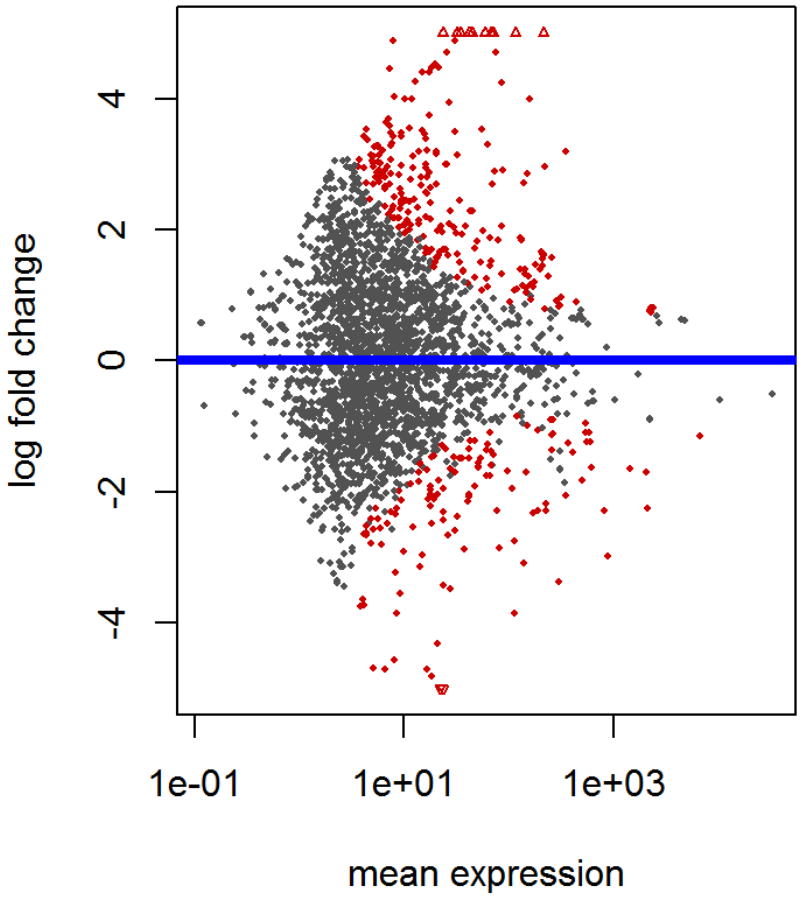


Diagnostic plots

MA plot

- Log2(foldchange between groups) vs Mean(all groups)
  - DEG's (red)
  - Rest (black)

```
plotMA(d.res)
```



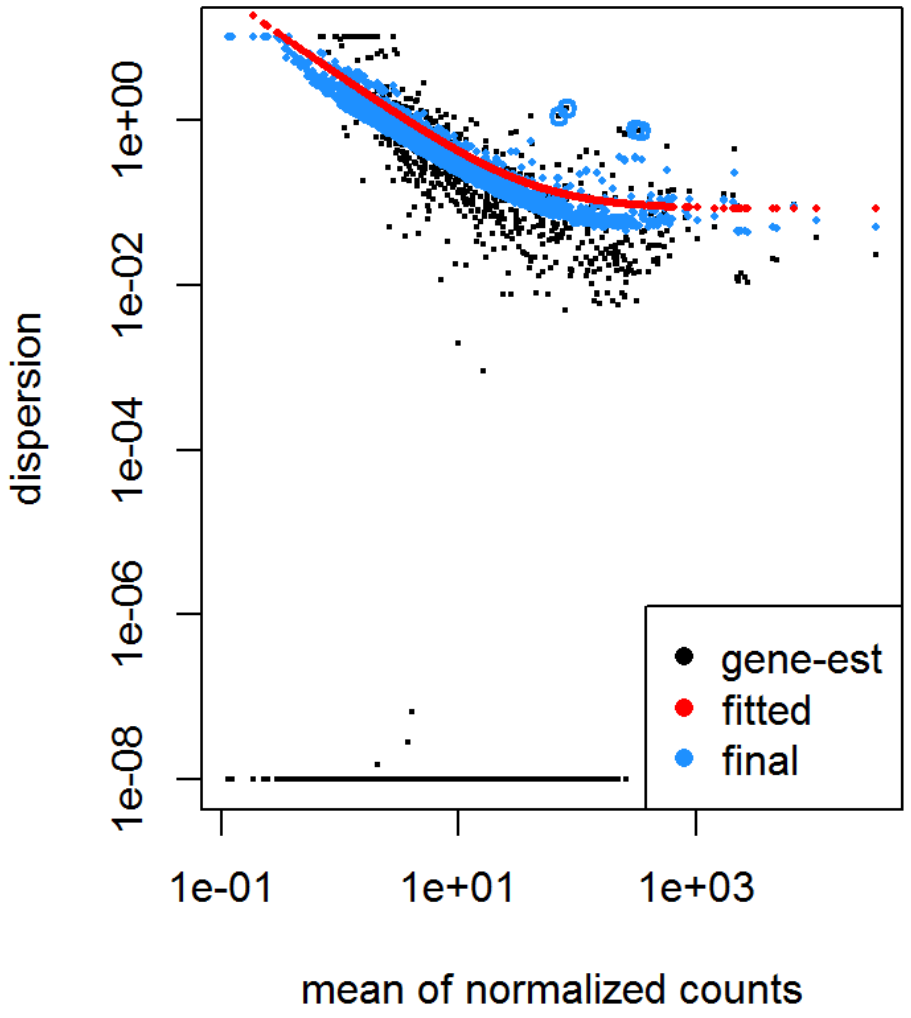
- Symmetric, no trend, all OK.

Diagnostic plots

Dispersion plot

- Comparison of dispersion estimation
  - *per gene (black)*
  - *per group of genes (fitted, red)*
  - *final value (blue)*

```
plotDispEsts(res)
```



- final values follow the trend of the gene-wise estimate
- moderation to avoid FP's (shrinkage towards group-wise estimate)

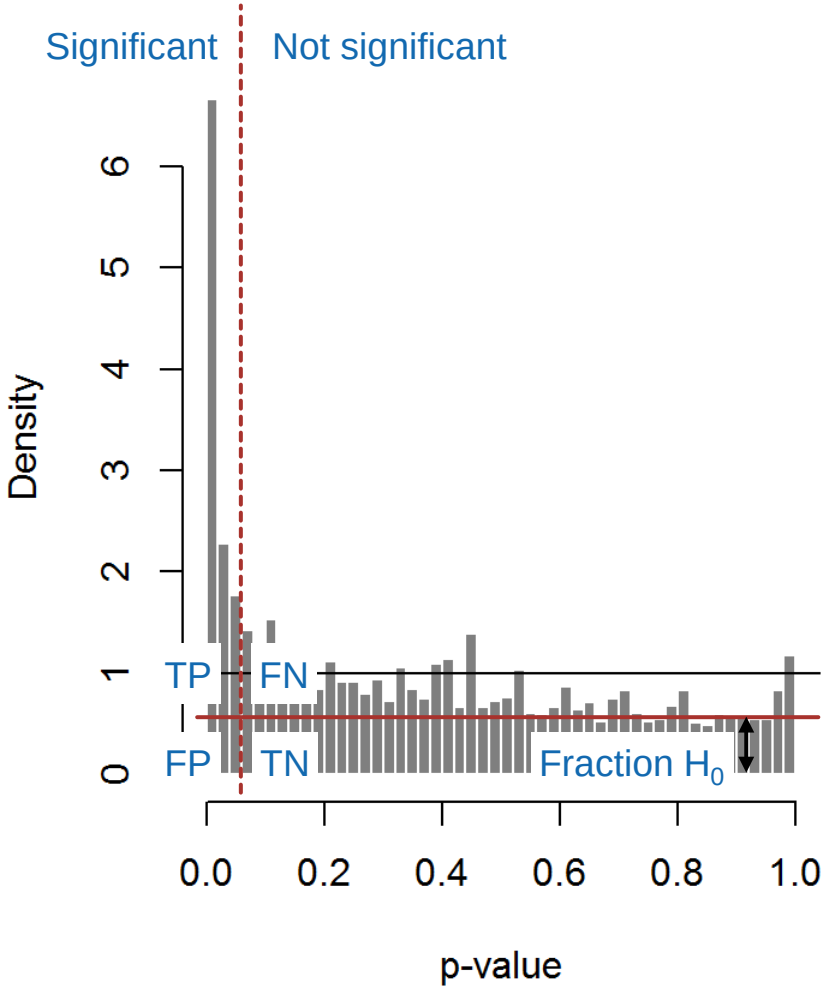
Diagnostic plots

p-value distribution (PVD)

- The PVD of H_0 samples is uniform.

```
hist(d.res$pvalue, breaks=50, freq = F,
     col="grey50", border="white",
     xlab="p-value", main = "")
abline(h=1)
```

- A regular PVD of mixed samples is continuously increasing towards p = 0 and is uniform on the right-hand side close to p = 1.
- The plateau value at p=1 corresponds to the fraction of H_0 samples.
- No irregularity can be seen.



Storey, Tibshirani 2003

Exercise 2

- Question 1:
  a. Execute the differential expression analysis pipeline.
  b. Did you choose to perform automatic independent filtering? Was it a good choice?
- Question 2:
  a. Show the 6 most significantly DEG's: result table and graph.
- Question 3:
  a. Describe and discuss the MvA plot.
  b. Describe and discuss the dispersion plot
  c. Describe and discuss the p-value distribution.

Export of results

Annotation: adding gene names

- Load the annotation package `org.Hs.eg.db`: (additional packages may be required)

```
library(org.Hs.eg.db)
columns(org.Hs.eg.db)
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"))
resOrdered <- d.res[order(d.res$pvalue),]
head(resOrdered)
```

- Export results

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