# Bi5444 Analysis of sequencing data
# Statistical analysis

Eva Budinska

budinska@recetox.muni.cz

# Aim

- To connect the information on molecular abundancies or processes from NGS with the conditions of the experiment
  - Comparing the molecular patterns between two or more groups (class comparison)
  - Discover new groups based on the molecular patterns (class discovery)
  - Predict existing groups (class prediction)
  - Analyze survival
  - Explore molecular events (pathway analysis…)

# The count table…

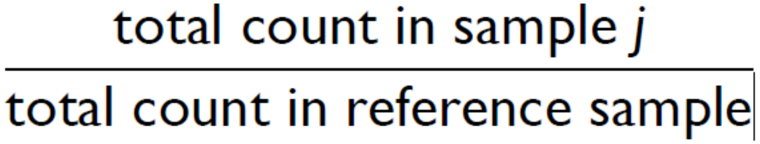
- The count table is a **quantitative representation** of the **abundance** of the sequence in the sample
- The problem of the count table is that the counts are **incomparable** due to **technical** and **biological** reasons:
  - we are unable to prepare libraries containing exactly the same amounts of DNA for each sample
  - Some features (genes) have longer target sequences than others, hence they will have more reads assigned!

# What to do? …. Normalize!

- **Goal**: compute a **normalization factor** for each sample, and **adjust** the **read counts** using this factor
- **After** the **adjustment**, the read counts for different samples (and different genes within sample) should be **comparable**
- Note that often, the normalization is not explicitly performed but rather **built into the existing analytical framework**

# Normalization approach I: total count

- **Define a reference sample** (either one of the observed samples or a "**pseudo-sample**") – this gives a "**target**" library **size**.
- The **normalization** factor for sample j is **defined** by



$$\frac{\text{total count in sample } j}{\text{total count in reference sample}}$$

- **RPKM/FPKM** is an **extension** of this **normalization** scheme, where we also normalize for the length of the gene

# RPKM/FPKM - outdated

- **FPKM** = Fragments Per Kilobase per Million mapped reads
- Similar to **RPKM**= Reads Per Kilobase per Million mapped reads
- **Not (anymore) recommended** for general use
- For some **plots** and **statistics** still **OK**
- **Accounts** for the **different lengths** of the features
- **Comparable within** the sample

# Normalization approach II: TMM

- **Trimmed Mean** of **M-values**
- M-values = log fold changes (compared to reference sample)
- A-values = average expression values
- Trim the genes with very small or very large M-or A-values
- Calculate the normalization factors based on a weighted M-value from the remaining genes
- Assumption: most genes are not differentially expressed
- Incorporated in **edgeR**

# Normalization approach II: TMM

- …

# Normalization approach III: RLE/DESeq

- Define the normalization factor for sample *j* as

$$\textit{median}_i \frac{\text{counts for gene } i \text{ in sample } j}{\text{counts for gene } i \text{ in reference sample}}$$

- Use a “pseudo-reference” sample with counts defined by the average of the individual sample counts
- Incorporated in **DESeq/DESeq2**

# Normalization approach IV: other summary measures

- Other measures can be used instead of the sum of the counts
  - Upper quartile
  - Median
  - Quantile normalization –adapted from microarrays

# Comparison of normalization approaches

- Nice evaluation, but only one of many http://www.ncbi.nlm.nih.gov/pubmed/22988256

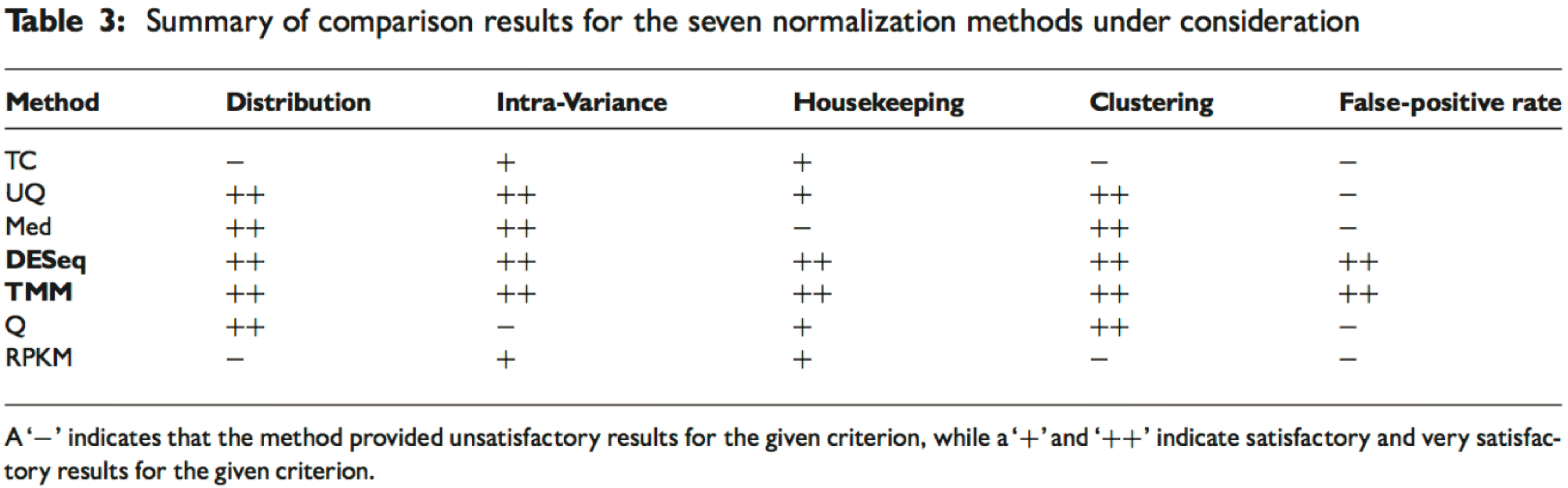


**Table 3:** Summary of comparison results for the seven normalization methods under consideration

| Method | Distribution | Intra-Variance | Housekeeping | Clustering | False-positive rate |
|---|---|---|---|---|---|
| TC | − | + | + | − | − |
| UQ | ++ | ++ | + | ++ | − |
| Med | ++ | ++ | − | ++ | − |
| **DESeq** | ++ | ++ | ++ | ++ | ++ |
| **TMM** | ++ | ++ | ++ | ++ | ++ |
| Q | ++ | − | + | ++ | − |
| RPKM | − | + | + | − | − |

A '−' indicates that the method provided unsatisfactory results for the given criterion, while a '+' and '++' indicate satisfactory and very satisfactory results for the given criterion.

# Other approaches

- **Spike-ins** with **known expression**
  - Very precise but more expensive
  - Getting more and more common
- Use “**housekeeping genes**”
  - Estimate normalization factor only from these
  - How do we know that the housekeeping genesare actually stable? Very often they are not!
  - If we do this, we have to choose at least several (>5) "housekeeping" genes

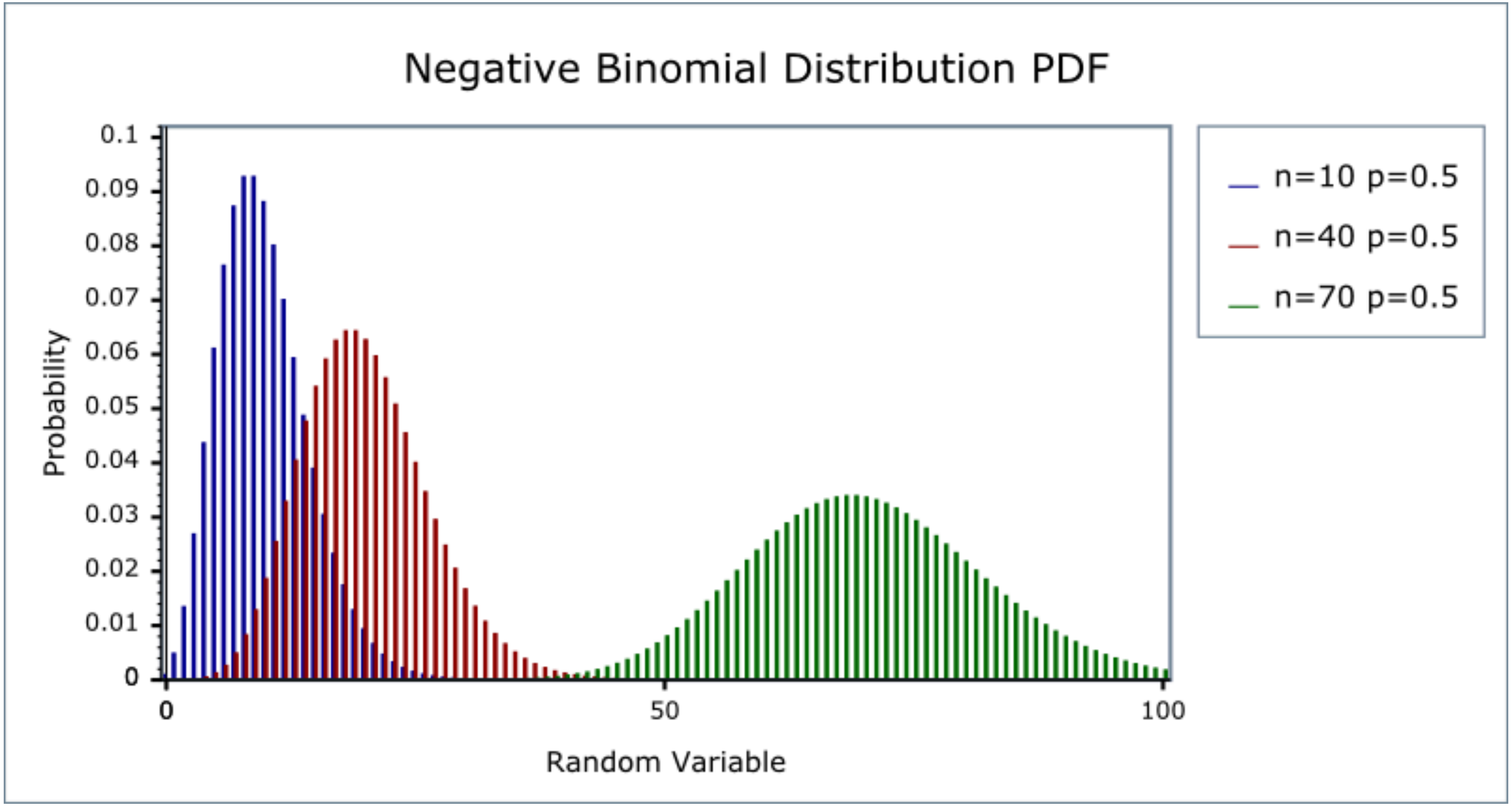
# Read count distribution –Negative Binomial distribution (?)

- The NB distribution extends the Poisson distribution by allowing for variability in the probability of a read mapping to a gene (λ).
- This implies the possibility of over-dispersion (i.e., variance exceeding the mean).
- The NB distribution has two parameters: the mean (μ) and the dispersion parameter (φ).

$$E[K]=\mu$$

$$var(K)=\mu+\varphi\mu 2$$

# Negative Binomial distribution



# From the differential gene expression

- Model the read count for the *i'*thgene in the *j'*thsample by

$$K_{ij} \sim NB(p_{ij}N_j, \varphi_{ij})$$

- Differential expression of a gene$_i$ is signified by differences in $p_{ij}$ between groups
- It is important to get dispersion estimates correct if we want to say something about the significance of the differences

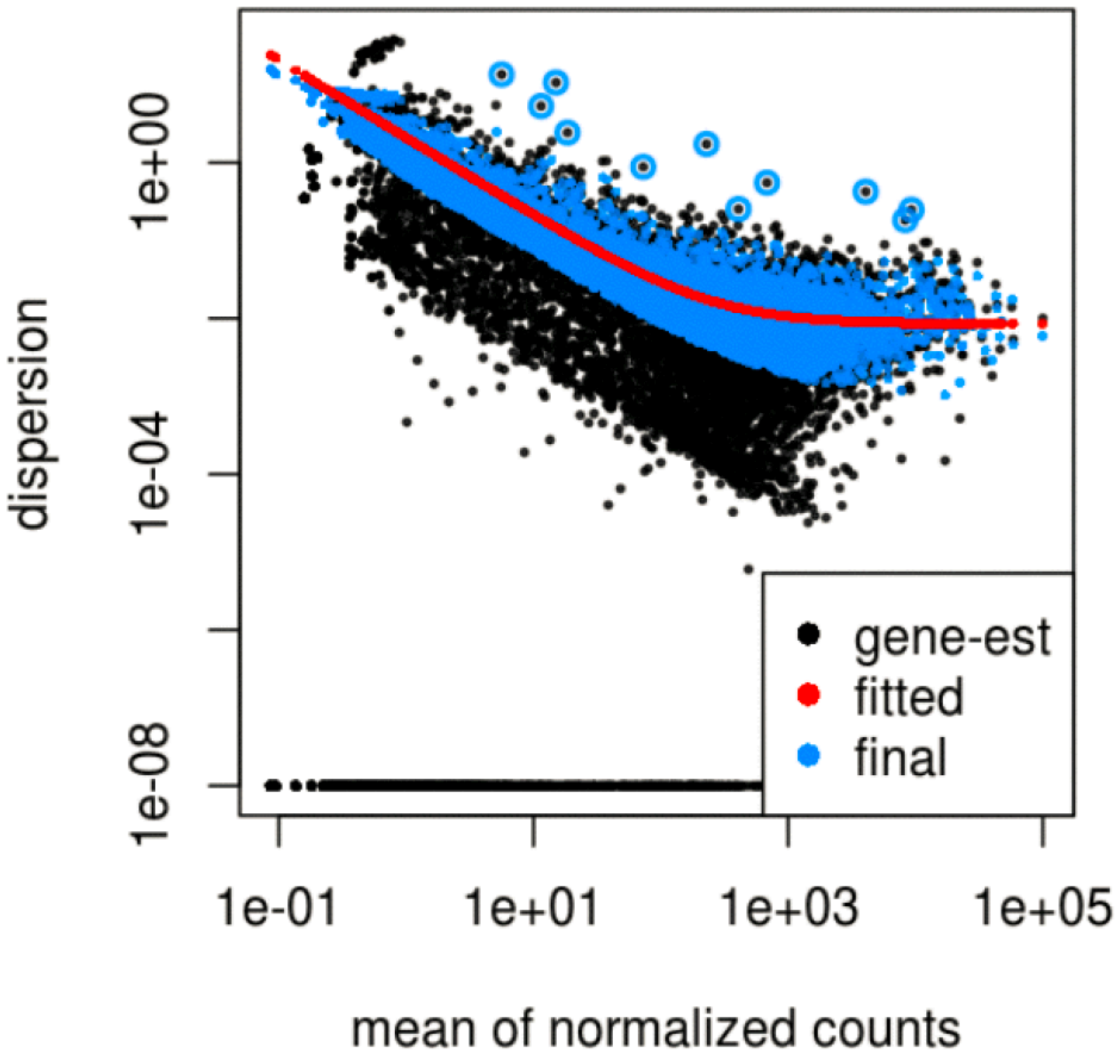
# Estimating the dispersion

- Due to the **small sample size**, **dispersion** (and hence the **variance**) **estimation** is **difficult**
- But we have **a lot of genes**!
- They **should not behave** completely **differently**
- Solution: **combine information across the genes** to **estimate** the **dispersion**!
- BUT the estimates from real data suggest that the dispersion may not be constant across genes

# Adjusting the dispersion

- Stabilize the individual estimates by squeezing them towards a common estimate or a trend (edgeR)
- Model the mean-dispersion (or meanvariance) relationship (DESeq)
- Bayesian approach using a prior based on empirical values (DESeq2)

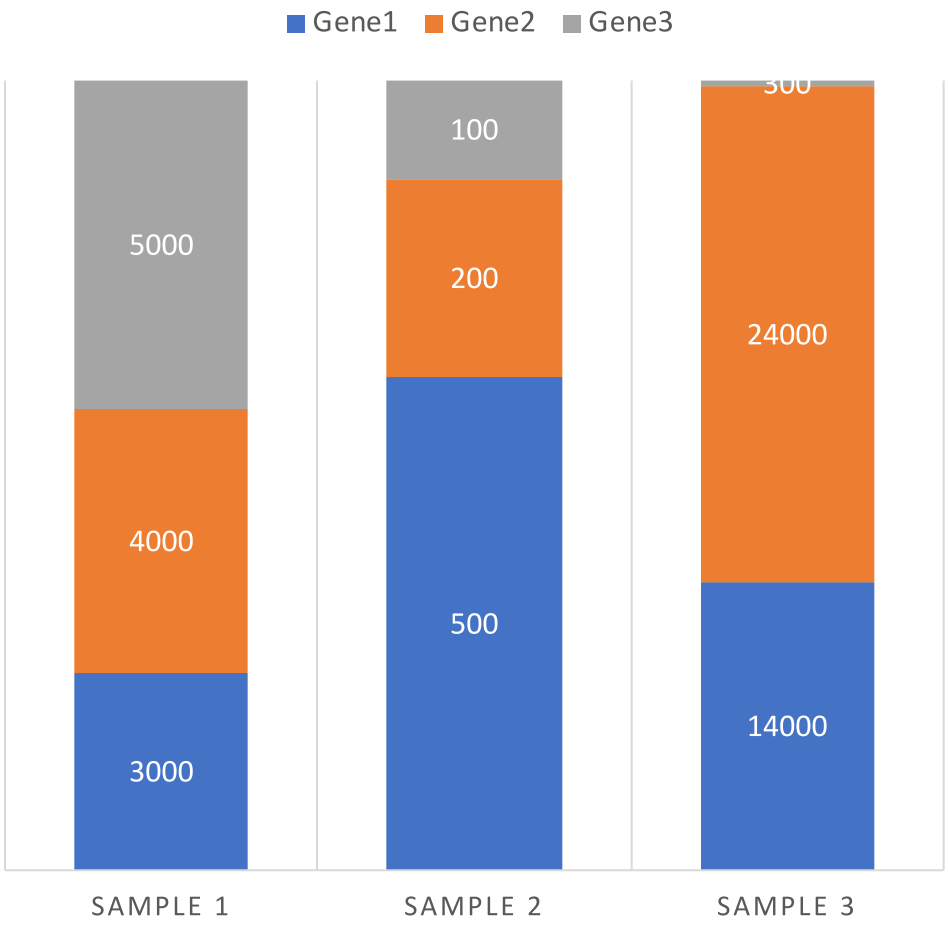
# Dispersion estimation example –DESeq2



# Compositional nature of the NGS data

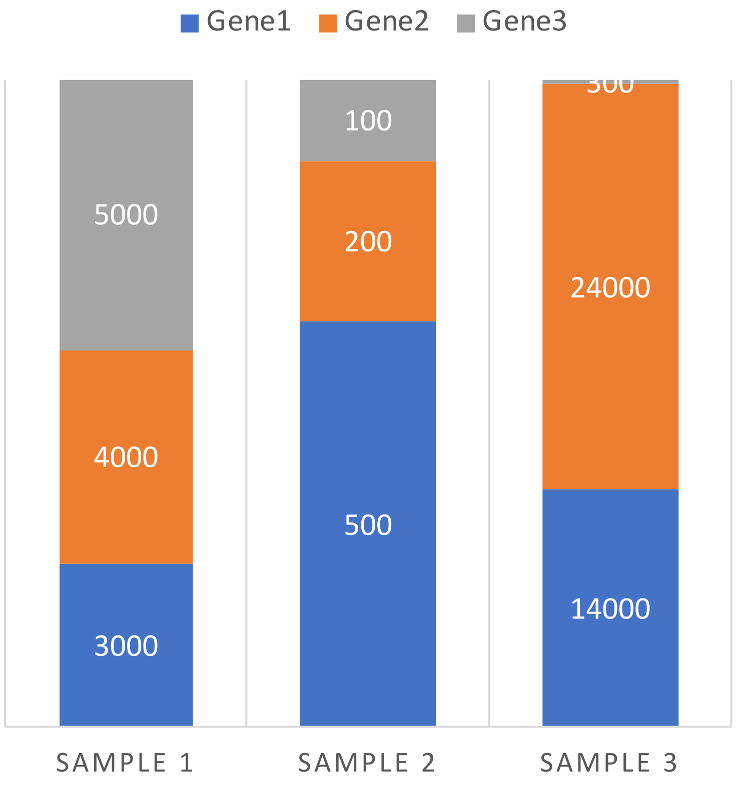
- The gene(transcript) abundancies (read counts) are **constrained by the maximum number of DNA reads** that the sequencer can provide (the total count constraint)

- Hence the data represents in fact a **proportion (composition) of genes**!

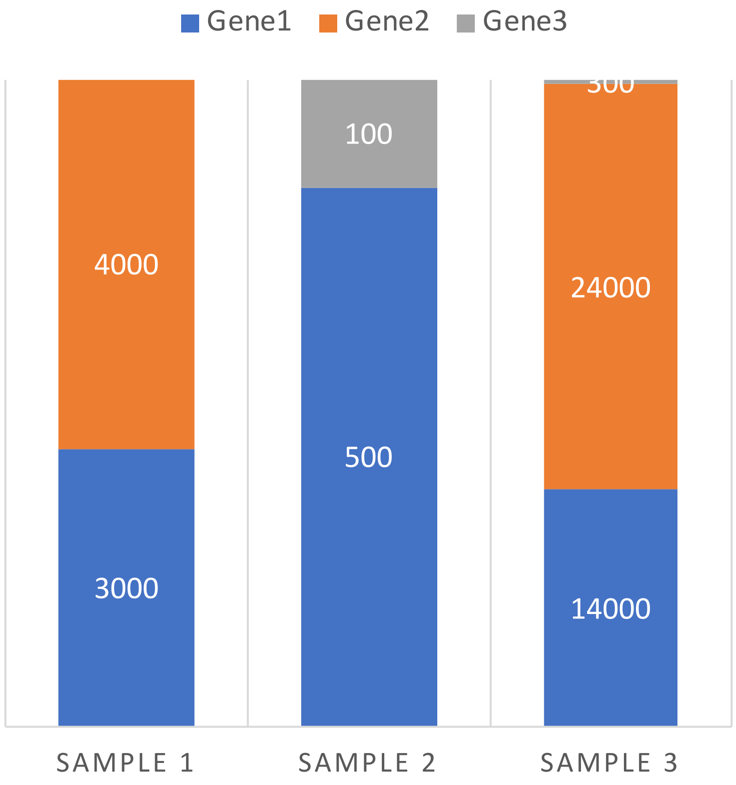


# The data is compositional – so what?

- The compositional nature of the data induces **strong dependencies** among the abundances of the different taxa:
    - an increase in the abundance of one gene implies the decrease of the observed number of counts – hence proportions - for other genes and vice versa



Before treatment



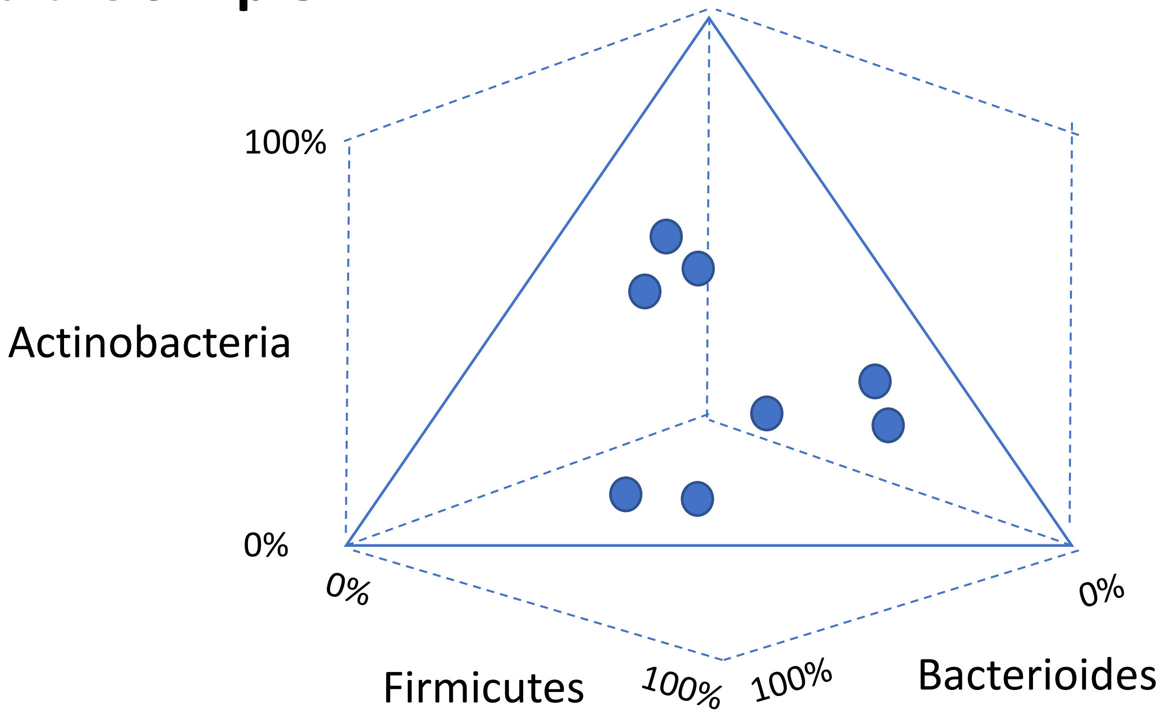
After treatment

# The data is compositional – so what?

In a composition the value of each component is not informative by itself and the **relevant information is contained in the ratios between the components.**

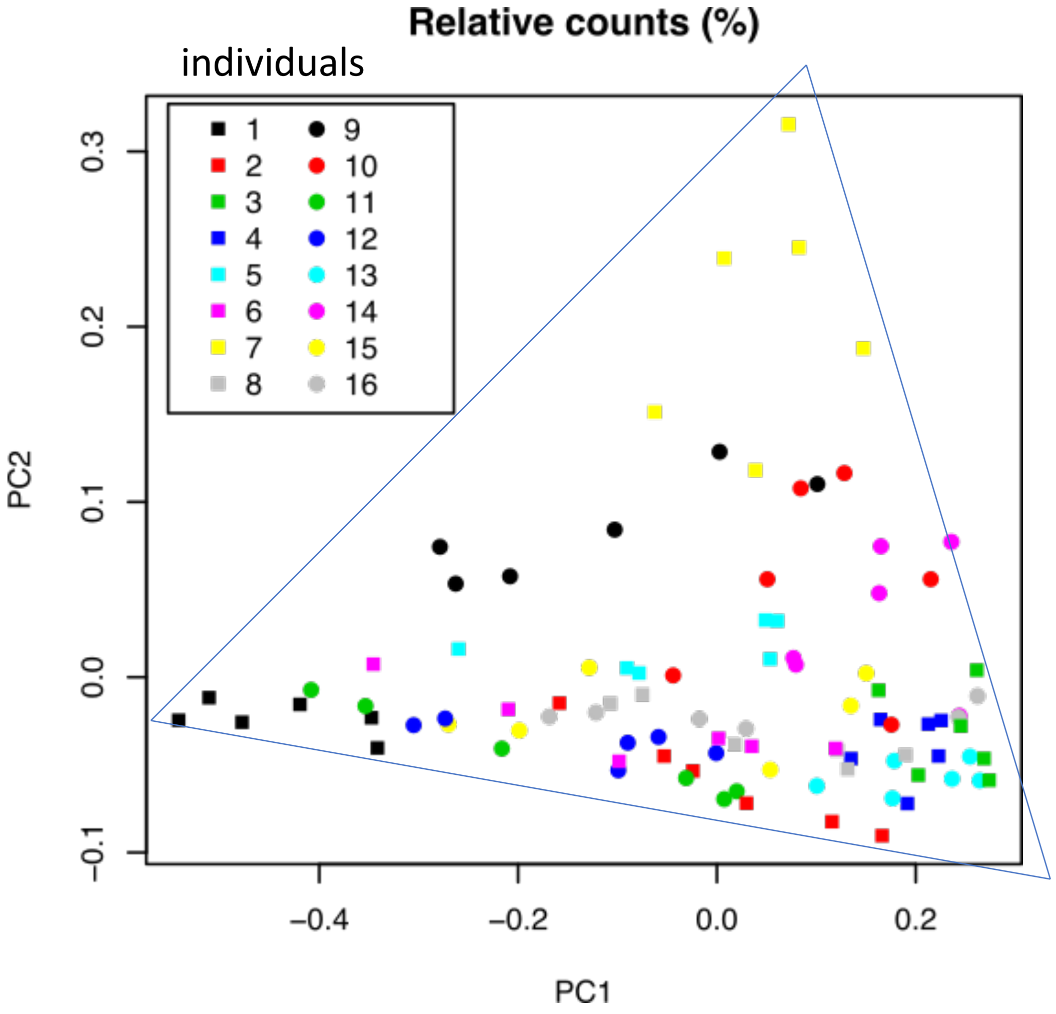
# The data is compositional – so what? / part 2

- Compositional data **do not exist in the Euclidean space**, but in a special constraint space called the **simplex**



- Hence it is incorrect to apply any multivariable techniques that are dependent on this space without proper transformation of data (e.g. PCA, clustering….)

# PCA on compositional data (without proper transformation)



# Statistical methods for analysis of compositional data need to fulfill these criteria:

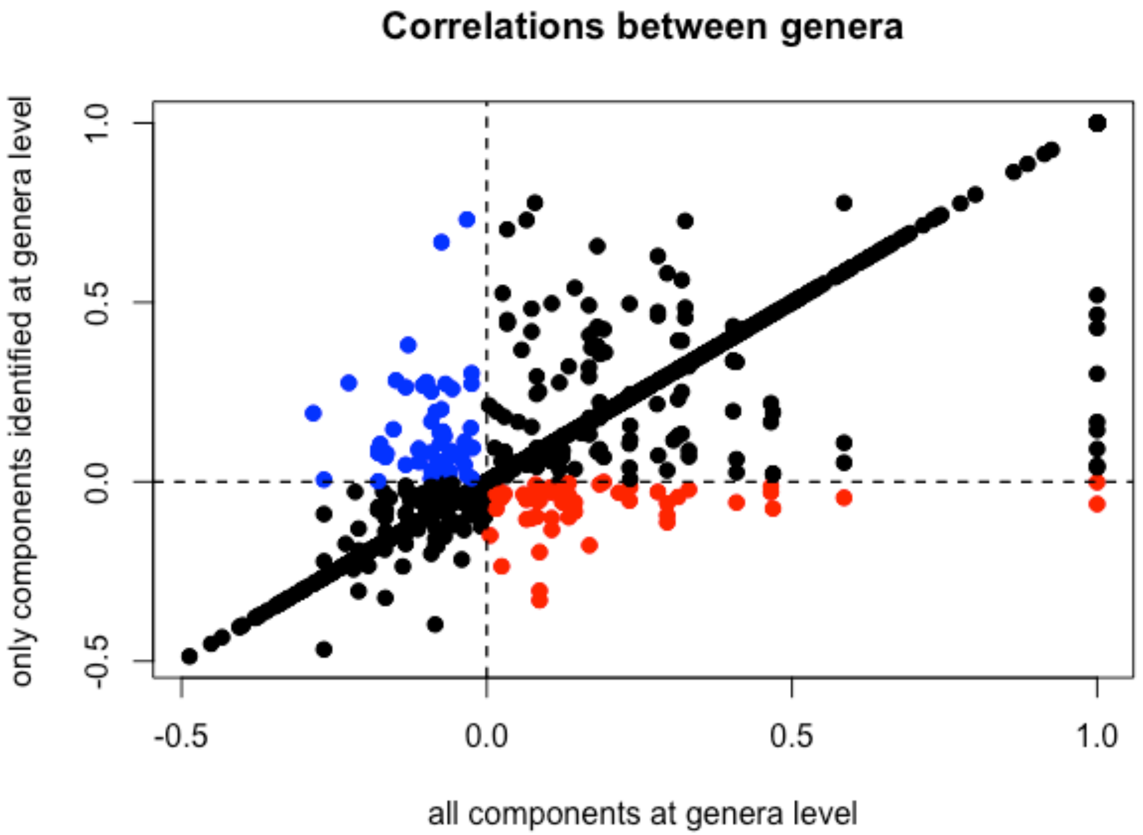
1. **Scale Invariance** (e.g. the result should be the same regardless of the scale of the measurement)
    - Example: how similar are these two samples?

| | % | | Absolute read counts | |
|---|---|---|---|---|
| | A | B | A | B |
| Fusobacteria | 10 | 11 | 700 | 11000 |
| Proteobacteria | 15 | 14 | 1050 | 14000 |
| Bacterioides | 25 | 20 | 1750 | 20000 |
| Firmicutes | 50 | 55 | 3500 | 55000 |
| Euclidean distance | 7.2 | | 57088.6 | |

# Statistical methods for analysis of compositional data need to fulfill these criteria:

**2. Subcompositional coherence** (e.g. the analyses should lead to the same conclusions regardless of which components of the data are included)

This is especially a problem for correlations between taxa, which tend to be more negative when we remove some taxa and recalculate the proportions.



# Statistical methods for analysis of compositional data need to fulfill these criteria:

2. **Subcompositional coherence** (e.g. the analyses should lead to the same conclusions regardless of which components of the data are included)

Alternative(s) to correlation:

$$VLR\left(\boldsymbol{x}_g, \boldsymbol{x}_h\right) = var\ (\ln \frac{x_g^1}{x_h^1} + \ln \frac{x_g^2}{x_h^2} + \ldots + \ln \frac{x_g^n}{x_h^n}\ )$$

1. phi (Φ) = var(Ai -Aj)/var(Ai)
2. rho (ρ) = var(Ai -Aj)/(var(Ai) + var(Aj))
3. phis (Φs) = var(Ai -Aj)/var(Ai +Aj)

Ai is the log-transformed values for a metagenomic component ‘i’ in the data

Aitchison, 1982, J.R.Statist. Soc.
Lovell et al., 2015, PLoS Comp Biol
Quinn et al, 2017, Scientific Reports 7

# Data transformation (normalization)

- Compositional data can be normalized in order to make them suitable for existing statistical techniques
- Aitchinson, 1982 - build a theory and concepts of analysis of compositional data and suggested normalizations
- Basic concept – make log-ratios between components

**ALR (additive log-ratio transformation)**

$$\operatorname{alr}(x) = \left[\log \frac{x_1}{x_D} \cdots \log \frac{x_{D-1}}{x_D}\right]$$

+ good for most statistical techniques  
– needs careful selection of one component, we are working with k-1 taxa, more difficult to interpret

**CLR (centered log-ratio transformation)**

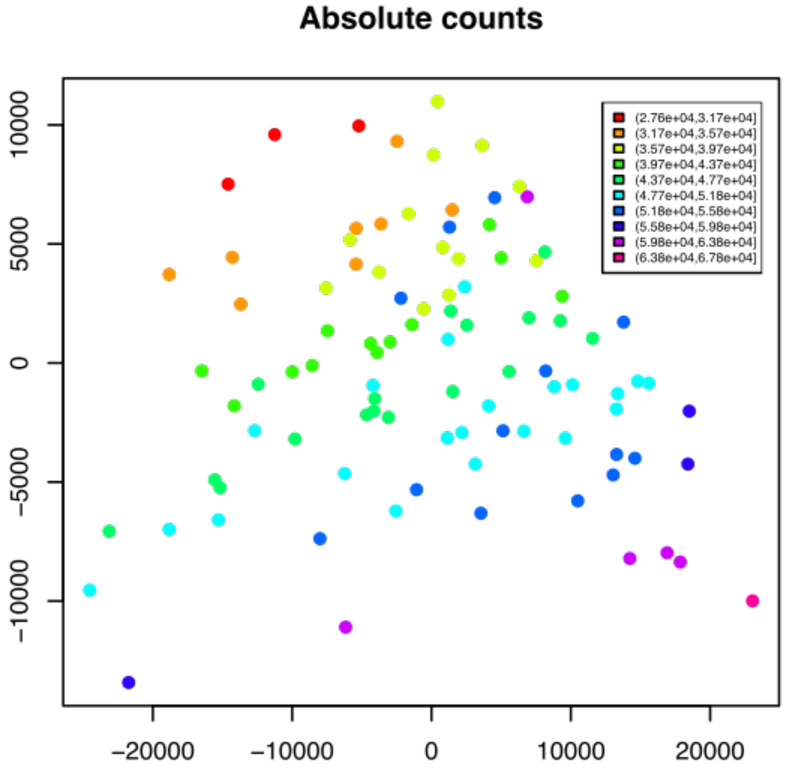
$$\operatorname{clr}(x) = \left[\log \frac{x_1}{g(x)} \cdots \log \frac{x_D}{g(x)}\right]$$

+ ratio to geometric mean, preserves all taxa, no need to select one  
– creates singular covariance matrix

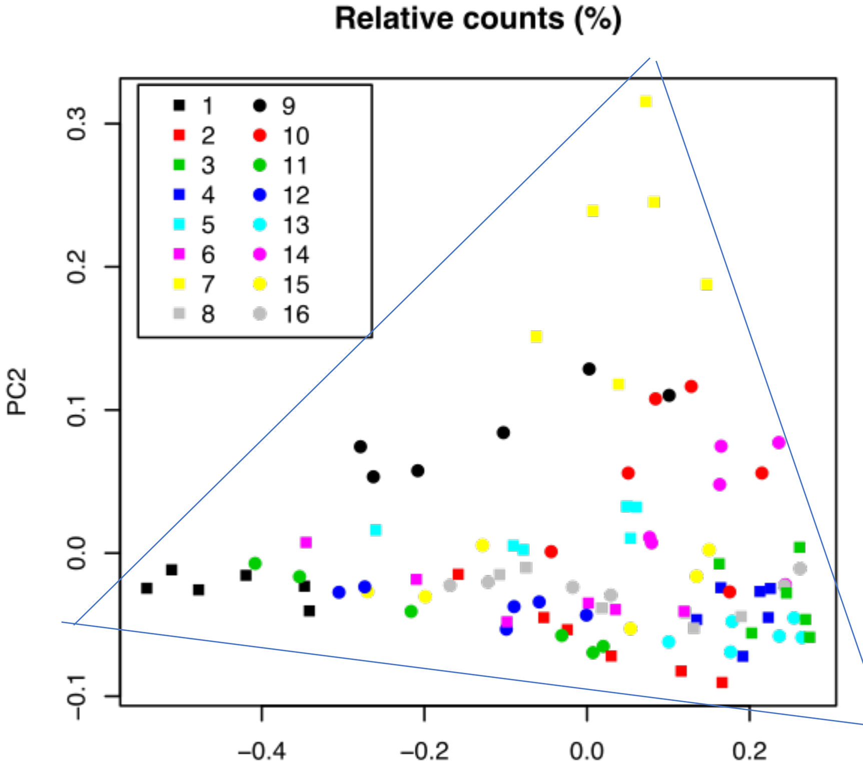
**ILR (isometric log-ratio transformation)** [Egozcque, 2003]

**PhILR (phylogenetic partitioning based ILR transform)** [Silverman et al, 2017]

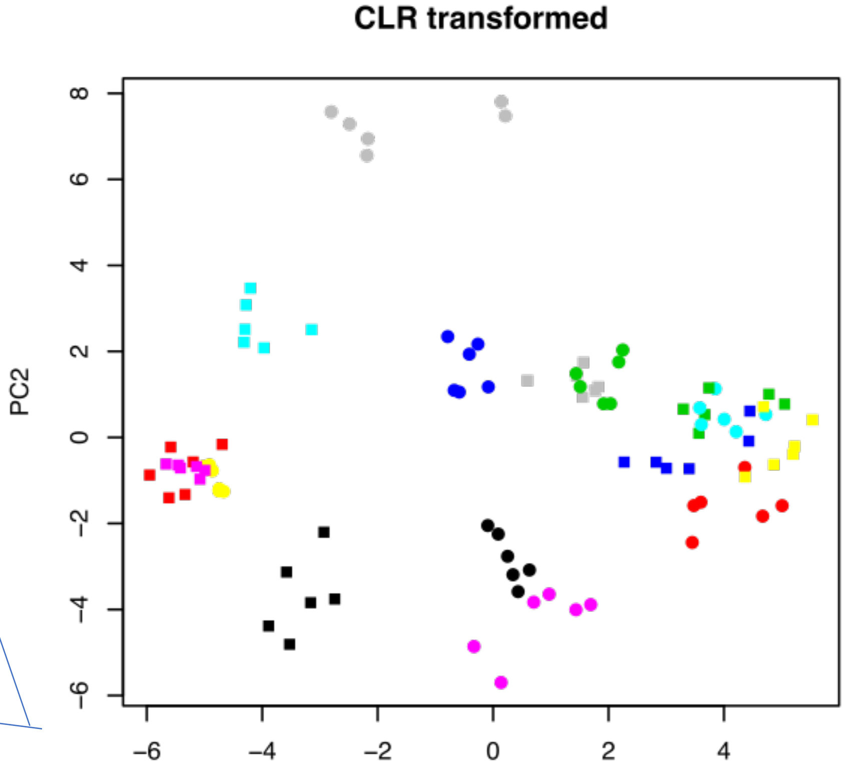
# Compositional data - PCA before and after normalization


Absolute counts

PCA on absolute counts – main variance lies in the sequencing depth


Relative counts (%)

Relative data – problem with simplex, colour by individual


CLR transformed

CLR transformed data – colour by individual

# The excess zero problem

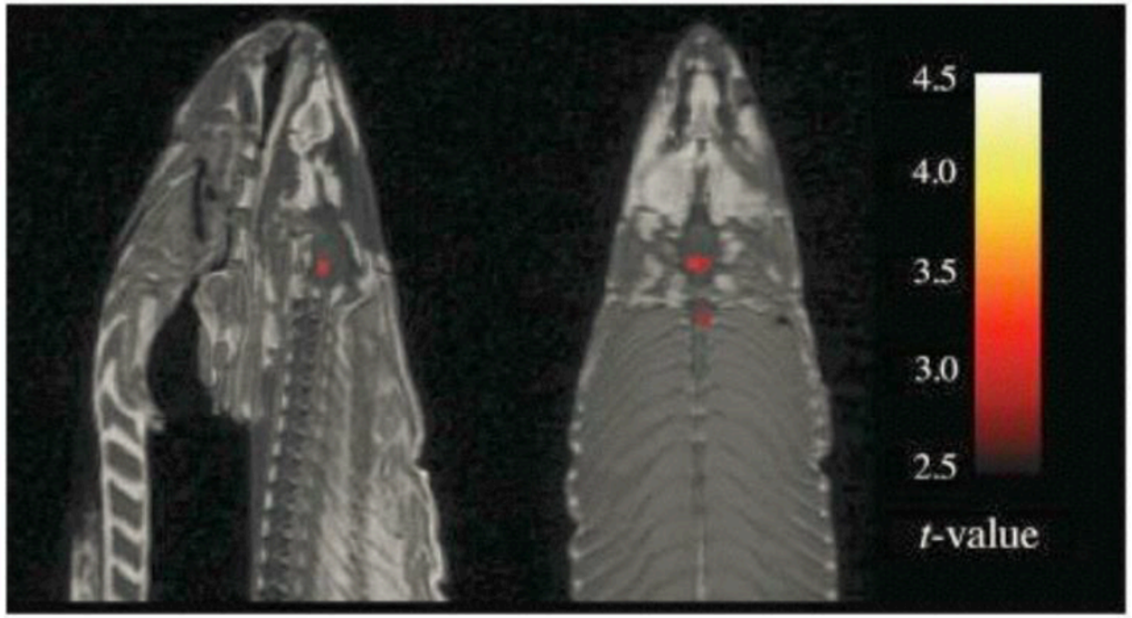
- Log-ratio transformations require data with **positive values**, any statistical analysis of count compositions must be preceded by a proper **replacement of the zeros**
- We do not know whether the zeros are real or just below threshold

- What to do?
  - E.g. Bayesian multiplicative treatment of count zeros [Martín-Fernandez,2014,Statistical Modelling]
  - Analysis and correction of compositional bias in sparse sequencing count data [Kumar et al., *BMC Genomics* **volume 19**, Article number: 799 (2018) ]

# Large number of genes –a lot of statistical tests

- **p-values** are suitable tools for inference when a **single hypothesis** is tested
- **p-value** = probability of obtaining a test statistic at least as extreme as the one observed, if the null hypothesis is true (that is, without any true signal in the data)
- But this means that even if the null hypothesis is true, there is a **non-zero probability** of obtaining such an **extreme test statistic**
- If we perform **many tests** (even with true null hypotheses), we will **get extreme** test statistics (and correspondingly low p-values) **every once in a while**

**NEUROSCIENCE PRIZE**: Craig Bennett, Abigail Baird, Michael Miller, and George Wolford [USA], for demonstrating that brain researchers, by using complicated instruments and simple statistics, can see meaningful brain activity anywhere — even in a dead salmon.



IMPROBABLE RESEARCH

## METHODS

Subject. One mature Atlantic Salmon (Salmo salar) participated in the fMRI study. The salmon was approximately 18 inches long, weighed 3.8 lbs, and was not alive at the time of scanning.

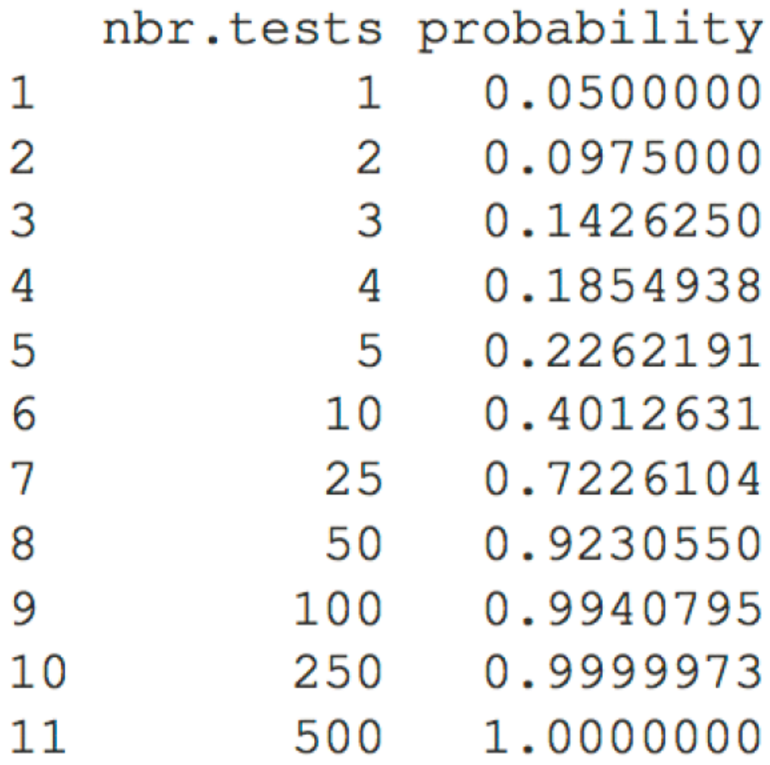
Task. The task administered to the salmon involved completing an open-ended mentalizing task. The salmon was shown a series of photographs depicting human individuals in social situations with a specified emotional valence. The salmon was asked to determine what emotion the individual in the photo must have been experiencing.

Design. Stimuli were presented in a block design with each photo presented for 10 seconds followed by 12 seconds of rest. A total of 15 photos were displayed. Total scan time was 5.5 minutes.

Craig M. Bennett, Abigail A. Baird, Michael B. Miller, and George L. Wolford, Journal of Serendipitous and Unexpected Results, vol. 1, no. 1, 2010, pp. 1-5

# What’s the problem?

As the **number** of **hypotheses increases**, the **probability** of obtaining at least **one low p-value**(e.g., <0.05) **increases, too**



```
   nbr.tests probability
1          1   0.0500000
2          2   0.0975000
3          3   0.1426250
4          4   0.1854938
5          5   0.2262191
6         10   0.4012631
7         25   0.7226104
8         50   0.9230550
9        100   0.9940795
10       250   0.9999973
11       500   1.0000000
```

# What can we do?

- The most common approach is to ***correct*** or ***adjust*** the observed **p-values**t o account for the **multiple testing** through correction of **family-wise error rate** or **false-discovery rate (FDR)**
- Typically, *multiply* each p-value with a number ≥ 1 to obtain ***adjusted p-values***
- Only if the **adjusted p-value** is **small** we call the result **significant**

# Bonferonni correction

- Divide the alpha level by the number of tests

- e.g. 1000 tests and alpha=0.05 => adjusted alpha level is 0.00005!

# Benjamini-Hochberg (FDR) correction

Idea: instead of focusing on the per-gene false positive probability, try to control the fraction of false positives among the genes that are considered significant.

- We can tolerate a few false positives if we simultaneously have a lot of true positive findings.
- After **FDR** (e.g. **Benjamini-Hochberg**) adjustment:
  - An adjusted p-value of (e.g.) 0.05 means that the smallest false discovery rate that we can get if we want to consider the given gene as significant, is 5%.
  - An adjusted p-value close to 1 means that we can not consider the corresponding gene to be significant without accepting that almost all our findings will be false.

# Benjamini-Hochberg (FDR) correction

Idea: instead of focusing on the per-gene false positive probability, try to control the fraction of false positives among the genes that are considered significant.

- We can tolerate a few false positives if we simultaneously have a lot of true positive findings.
- After **FDR** (e.g. **Benjamini-Hochberg**) adjustment:
  - An adjusted p-value of (e.g.) 0.05 means that the smallest false discovery rate that we can get if we want to consider the given gene as significant, is 5%.
  - An adjusted p-value close to 1 means that we can not consider the corresponding gene to be significant without accepting that almost all our findings will be false.

# DE calculation

## Differential expression

# Inputs to the calculation

Two main inputs

**1. Table** with **raw or normalized** gene counts – column per sample and row per gene

**2. Design table** –assignment of groups/conditions to the samples

- Additional input -design/model matrix
- "Normal" comparison~condition
- "Paired" comparison ~patient+condition

# edgeR

Implemented in R

- Count-based approach
- Assumes a NB distribution
- TMM normalizationby default (other alternatives available)
- Estimates dispersion by shrinking towards a common or trended estimate
- Allows a large variety of experimental designs through the use of a generalized linear model (GLM) framework

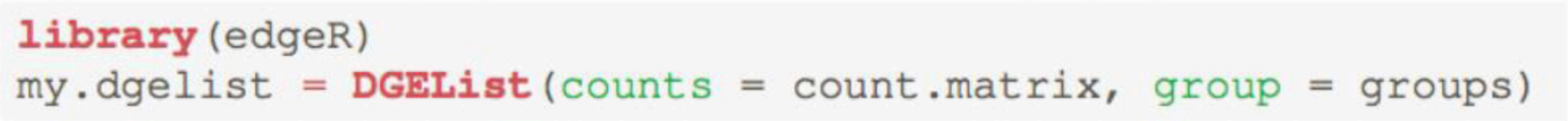
# DESeq2

Implemented in R

- Count-based approach
- Assumes a NB distribution
- **RLE normalization**
- Estimates **dispersion** by a **Bayesian** approach
- Implements **outlier detection**and **independent filtering**
- Allows a **large variety** of **experimental designs** through the use of a generalized linear model  (GLM) framework

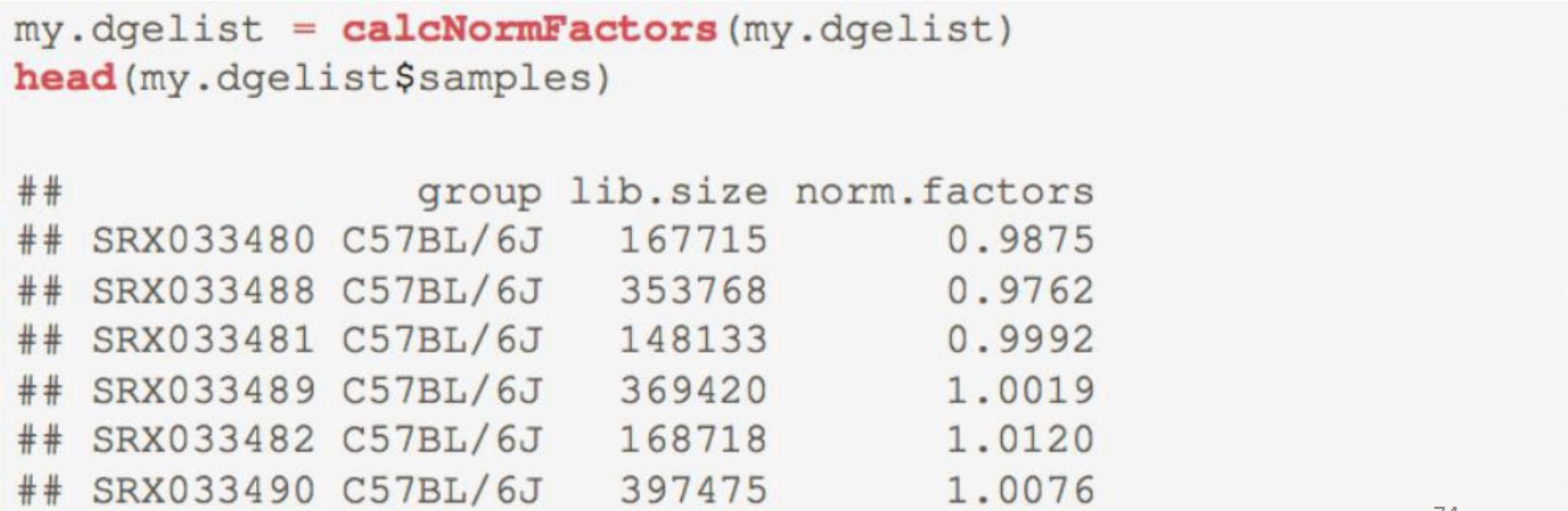
# edgeR – example

- Create a DGEList object from a count matrix and a vector of class labels.



```r
library(edgeR)
my.dgelist = DGEList(counts = count.matrix, group = groups)
```

- Calculate normalization factors

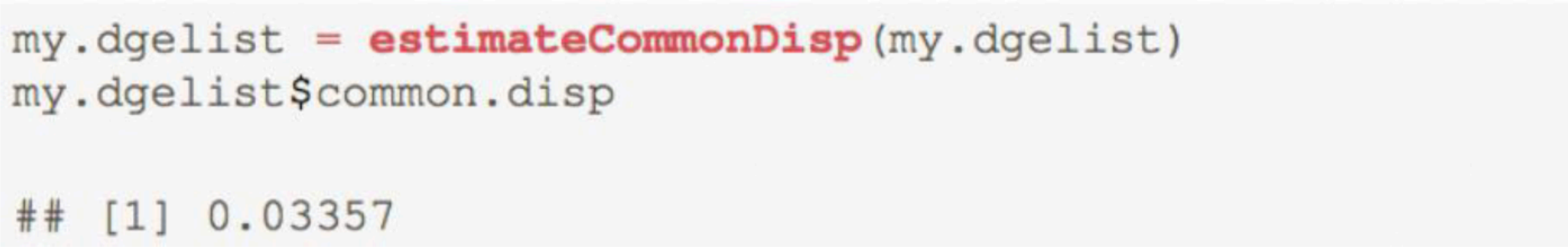


```r
my.dgelist = calcNormFactors(my.dgelist)
head(my.dgelist$samples)


##                 group lib.size norm.factors
## SRX033480 C57BL/6J   167715       0.9875
## SRX033488 C57BL/6J   353768       0.9762
## SRX033481 C57BL/6J   148133       0.9992
## SRX033489 C57BL/6J   369420       1.0019
## SRX033482 C57BL/6J   168718       1.0120
## SRX033490 C57BL/6J   397475       1.0076
```

# edgeR – example

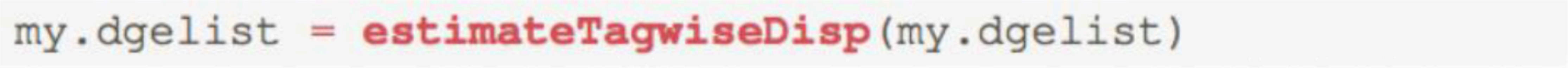
- Estimate the common dispersion



```
my.dgelist = estimateCommonDisp(my.dgelist)
my.dgelist$common.disp


## [1] 0.03357
```

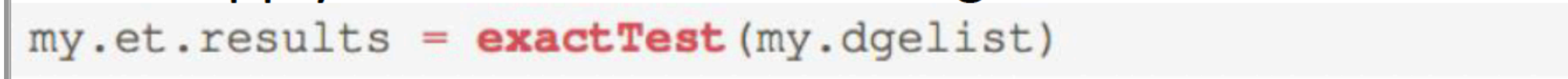
- Estimate the tagwise dispersions



```
my.dgelist = estimateTagwiseDisp(my.dgelist)
```

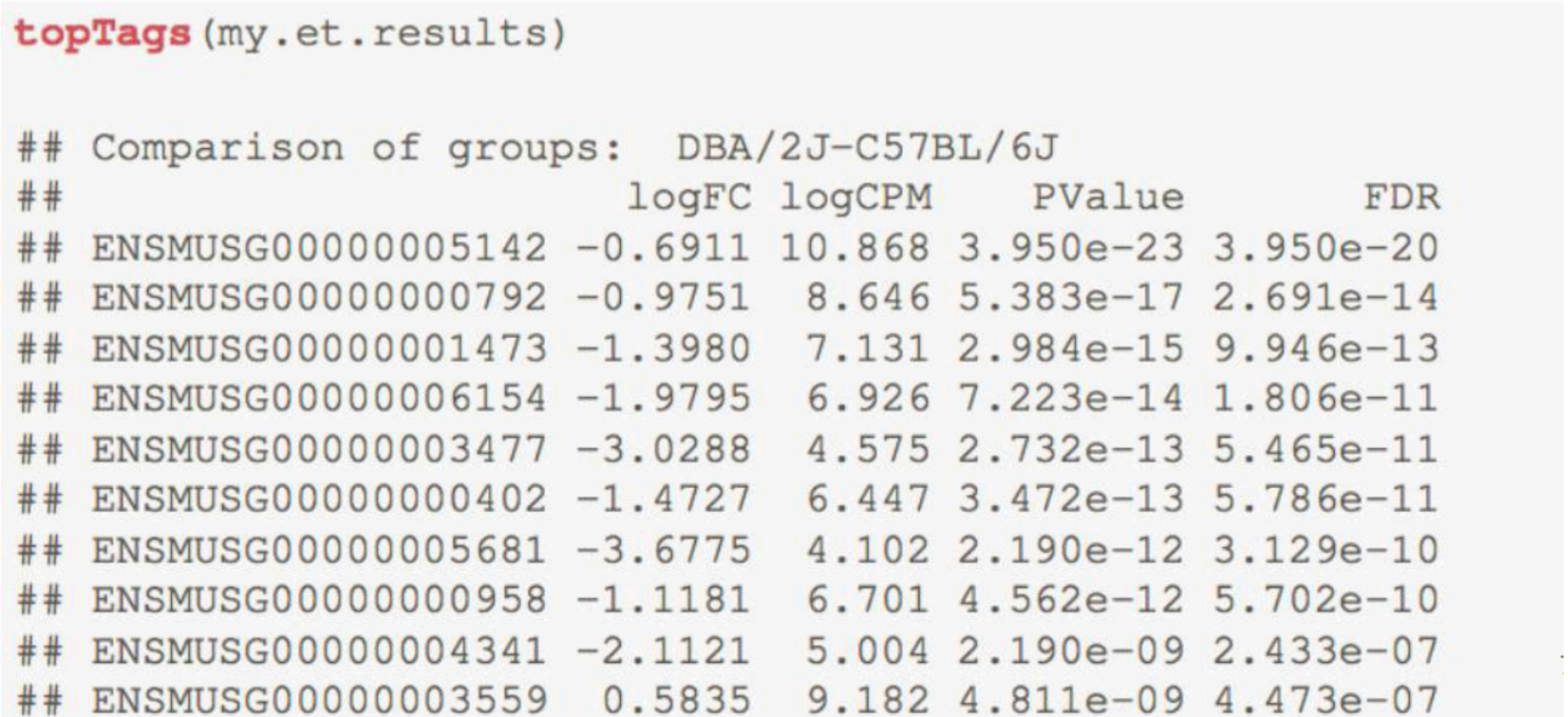
# edgeR – example

- Apply the exact test to each gene



```
my.et.results = exactTest(my.dgelist)
```

- Display the top-ranked genes

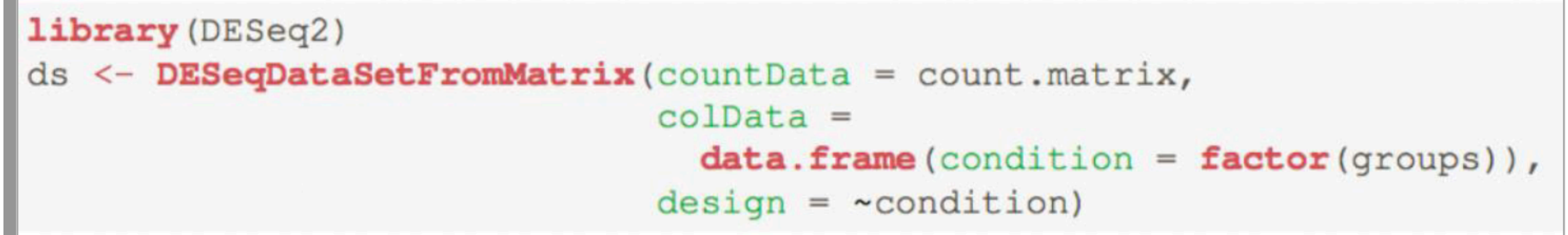


```
topTags(my.et.results)


## Comparison of groups:  DBA/2J-C57BL/6J
##                     logFC logCPM    PValue       FDR
## ENSMUSG00000005142 -0.6911 10.868 3.950e-23 3.950e-20
## ENSMUSG00000000792 -0.9751  8.646 5.383e-17 2.691e-14
## ENSMUSG00000001473 -1.3980  7.131 2.984e-15 9.946e-13
## ENSMUSG00000006154 -1.9795  6.926 7.223e-14 1.806e-11
## ENSMUSG00000003477 -3.0288  4.575 2.732e-13 5.465e-11
## ENSMUSG00000000402 -1.4727  6.447 3.472e-13 5.786e-11
## ENSMUSG00000005681 -3.6775  4.102 2.190e-12 3.129e-10
## ENSMUSG00000000958 -1.1181  6.701 4.562e-12 5.702e-10
## ENSMUSG00000004341 -2.1121  5.004 2.190e-09 2.433e-07
## ENSMUSG00000003559  0.5835  9.182 4.811e-09 4.473e-07
```

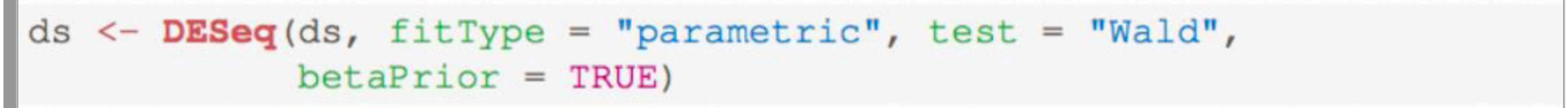
# DESeq2 – example

- Create a DESeqDataSet



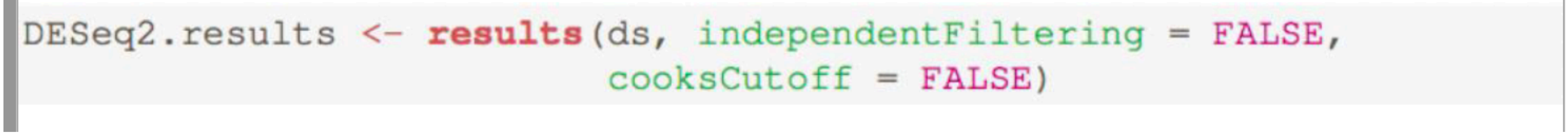
```
library(DESeq2)
ds <- DESeqDataSetFromMatrix(countData = count.matrix,
                             colData =
                               data.frame(condition = factor(groups)),
                             design = ~condition)
```

- Perform the differential expression analysis



```
ds <- DESeq(ds, fitType = "parametric", test = "Wald",
            betaPrior = TRUE)
```

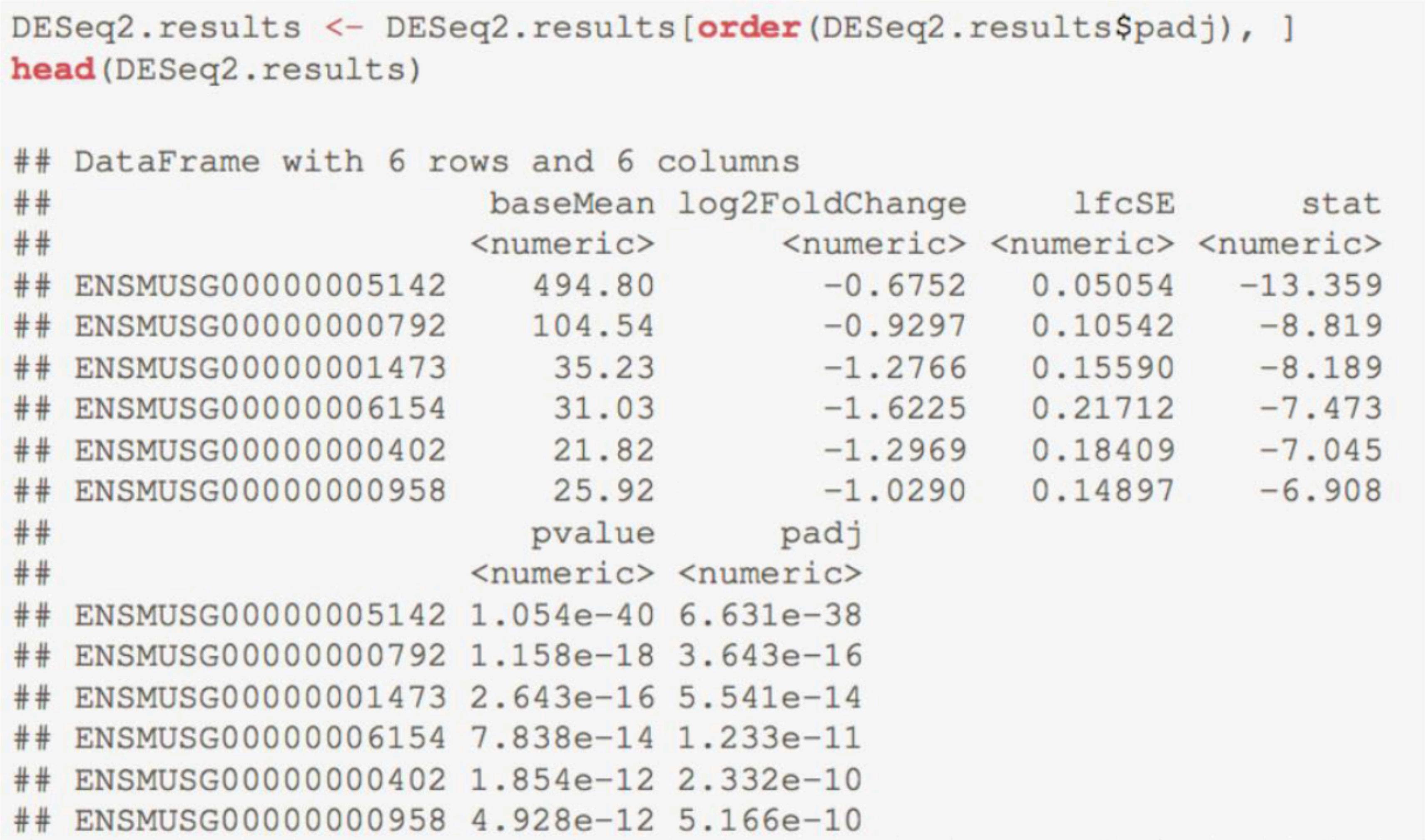
- Get the results



```
DESeq2.results <- results(ds, independentFiltering = FALSE,
                          cooksCutoff = FALSE)
```

# DESeq2 – example

- Order results by significance



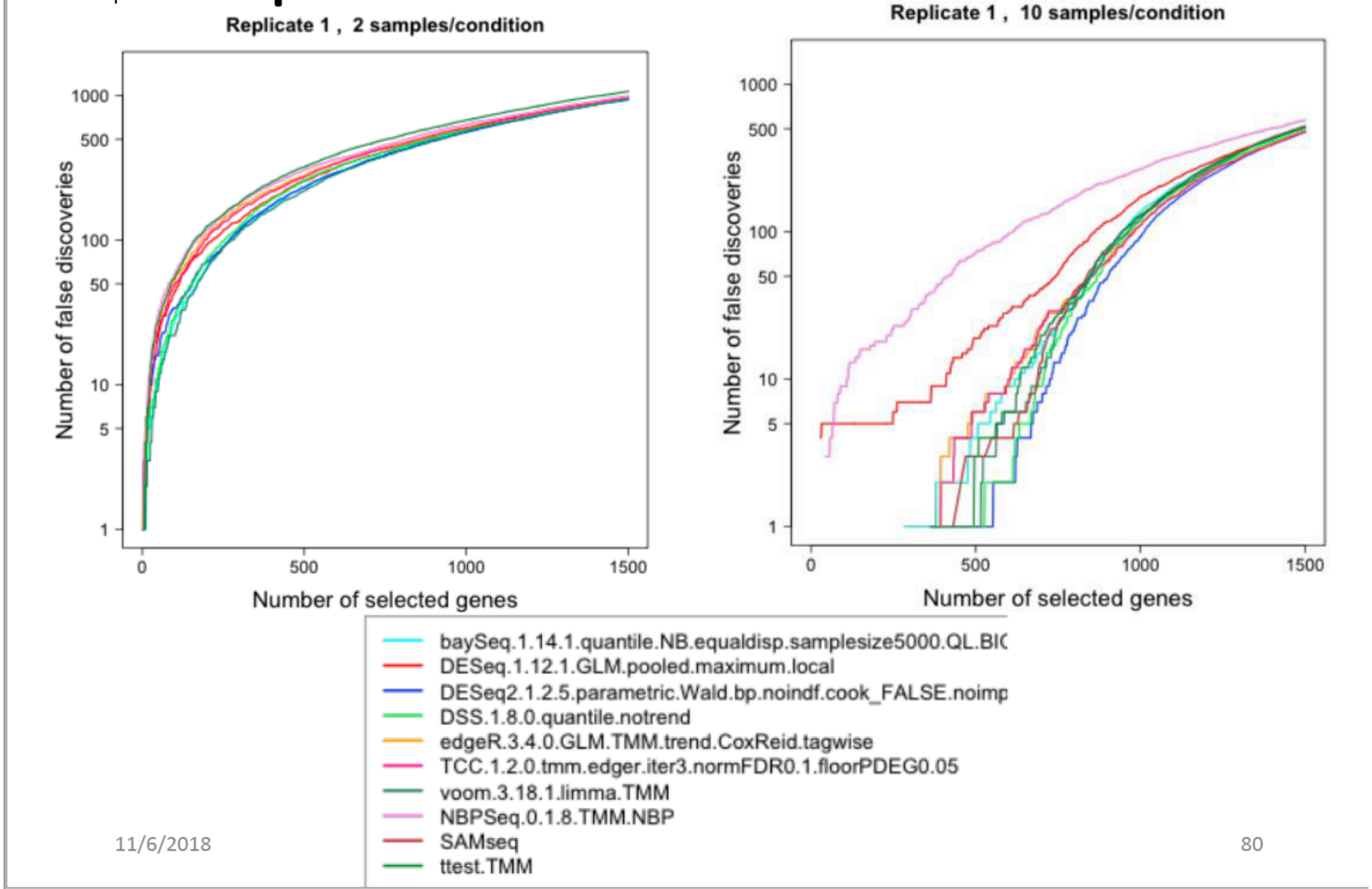
```
DESeq2.results <- DESeq2.results[order(DESeq2.results$padj), ]
head(DESeq2.results)

## DataFrame with 6 rows and 6 columns
##                     baseMean log2FoldChange     lfcSE      stat
##                    <numeric>      <numeric> <numeric> <numeric>
## ENSMUSG00000005142    494.80        -0.6752   0.05054   -13.359
## ENSMUSG00000000792    104.54        -0.9297   0.10542    -8.819
## ENSMUSG00000001473     35.23        -1.2766   0.15590    -8.189
## ENSMUSG00000006154     31.03        -1.6225   0.21712    -7.473
## ENSMUSG00000000402     21.82        -1.2969   0.18409    -7.045
## ENSMUSG00000000958     25.92        -1.0290   0.14897    -6.908
##                       pvalue      padj
##                    <numeric> <numeric>
## ENSMUSG00000005142 1.054e-40 6.631e-38
## ENSMUSG00000000792 1.158e-18 3.643e-16
## ENSMUSG00000001473 2.643e-16 5.541e-14
## ENSMUSG00000006154 7.838e-14 1.233e-11
## ENSMUSG00000000402 1.854e-12 2.332e-10
## ENSMUSG00000000958 4.928e-12 5.166e-10
```

# Other tools

- voom+limma
- baySeq
- Cuffdiff2 (+cummerbund)
- And many other R packages

# Comparison of DE techniques



Replicate 1 , 2 samples/condition

Replicate 1 , 10 samples/condition