Analyses biostatistiques de données RNA-seq

Nathalie Vialaneix (MIAT, INRAE)

en collaboration avec Ignacio Gonzàles et Annick Moisan

nathalie.vialaneix@inrae.fr

`http://www.nathalievialaneix.eu`

INRAE Genotoul Bioinfo

Toulouse, 16-17 mai 2024

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

INRAE

Outline

Exploratory analysis

- Introduction
- Experimental design
- Data exploration and quality assessment

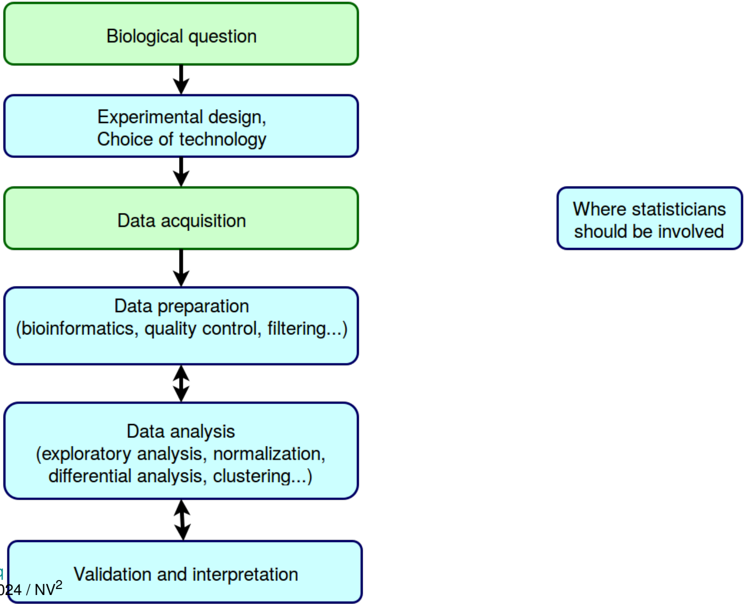
Normalization

- Raw data filtering
- Interpreting read counts

Differential Expression analysis

- Hypothesis testing and correction for multiple tests
- Differential expression analysis for RNAseq data
- Interpreting and improving the analysis

A typical transcriptomic experiment



Biological question

Experimental design,
Choice of technology

Data acquisition

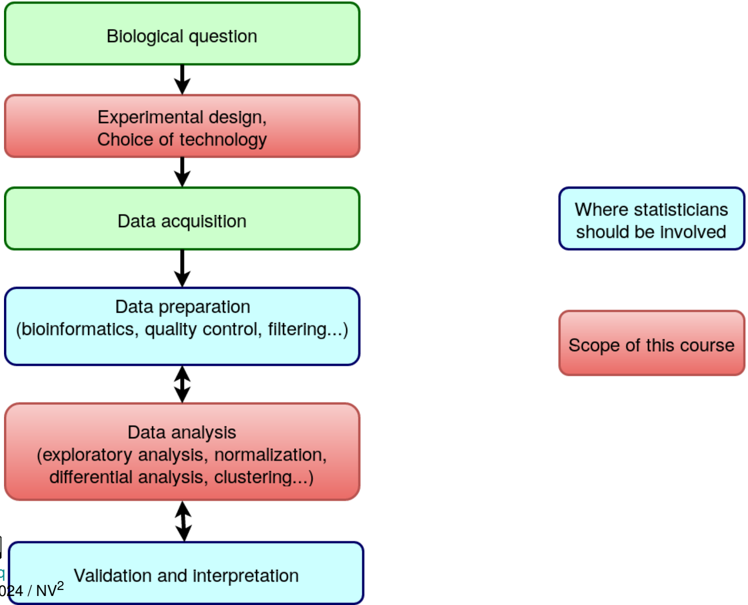
Data preparation
(bioinformatics, quality control, filtering...)

Data analysis
(exploratory analysis, normalization,
differential analysis, clustering...)

Validation and interpretation

Where statisticians
should be involved

A typical transcriptomic experiment



Biological question

Experimental design,
Choice of technology

Data acquisition

Data preparation
(bioinformatics, quality control, filtering...)

Data analysis
(exploratory analysis, normalization,
differential analysis, clustering...)

Validation and interpretation

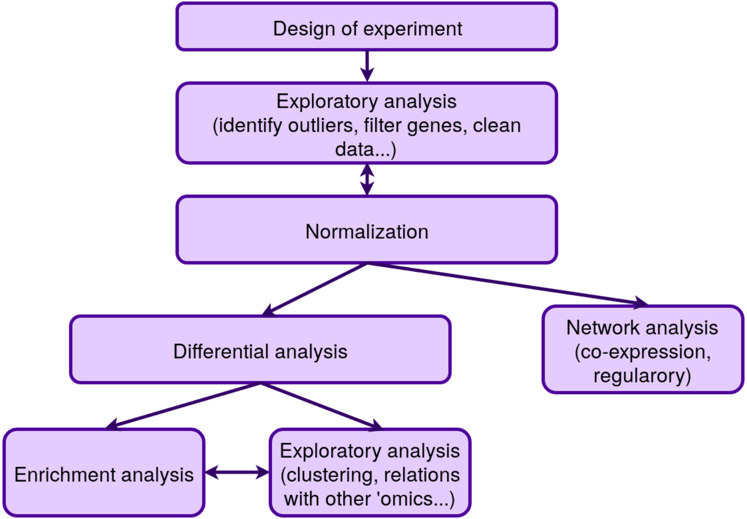
Where statisticians
should be involved

Scope of this course

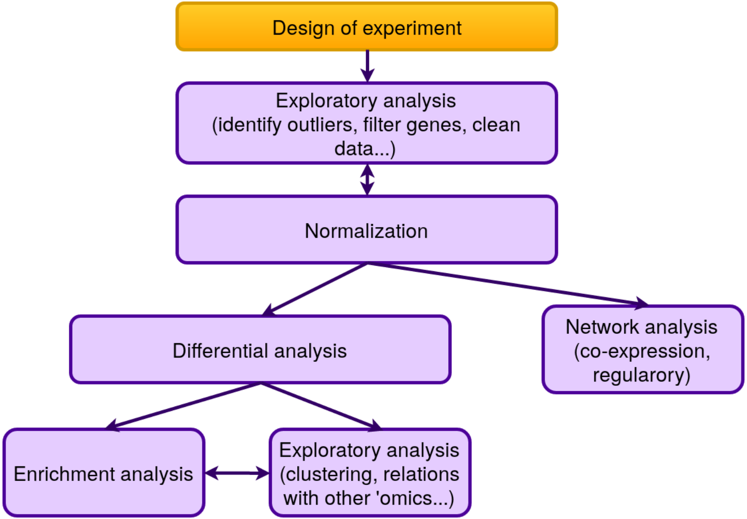
Outline

- Exploratory analysis
  - Introduction
  - Experimental design
  - Data exploration and quality assessment
- Normalization
  - Raw data filtering
  - Interpreting read counts
- Differential Expression analysis
  - Hypothesis testing and correction for multiple tests
  - Differential expression analysis for RNAseq data
  - Interpreting and improving the analysis

Steps in RNAseq data analysis


Design of experiment
Exploratory analysis (identify outliers, filter genes, clean data...)
Normalization
Differential analysis
Network analysis (co-expression, regularory)
Enrichment analysis
Exploratory analysis (clustering, relations with other 'omics...)

Part I: Experimental design



Design of experiment

Exploratory analysis (identify outliers, filter genes, clean data...)

Normalization

Differential analysis

Network analysis (co-expression, regularory)

Enrichment analysis

Exploratory analysis (clustering, relations with other 'omics...)

Confounded effects: a simple example

Basic experiment: find differences between control/treated plants

control group plant

treated group plant

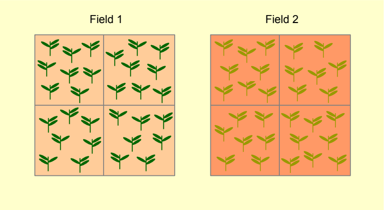
Confounded effects: a simple example

Basic experiment: find differences between control/treated plants

control group plant

treated group plant

A bad experimental design: grow all control group plants in one field and grow all treated group plants in another field



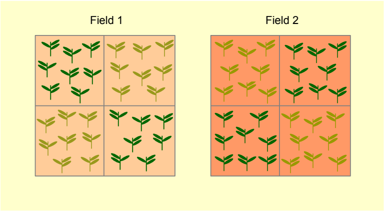
differences due to the field / the treatment can not be distinguished ⇒ confounded effects

Confounded effects: a simple example

Basic experiment: find differences between control/treated plants

control group plant treated group plant

A good experimental design: grow half control group plants (chosen at random) and half treated group plants in one field (and the rest in the other field)



differences due to the field / the treatment can be estimated separately

Confounded effects: a simple example

Basic experiment: find differences between control/treated plants

control group plant

treated group plant

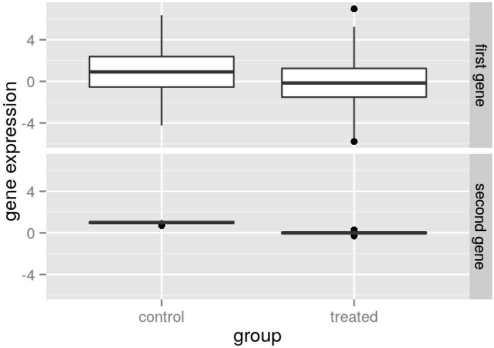
In summary, what is a good experimental design?

Experimental design are usually not as simple as this example: they can include multiple experimental factors (day of experiment, flow cell, . . .) and multiple covariates (sex, parents, . . .).

⇒ The experimental design must be carefully thought before starting the experiment and confounded effects must be searched for in a systematic manner.

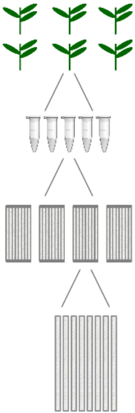
Effect & Variation

2 conditions, 2 genes whose expression distribution is:



- **first gene:** different median levels between the two groups but large variance: differences may be non significant
- **second gene:** different median levels between the two groups but very small variance: differences may be significant

Source of variation in RNA-seq experiments

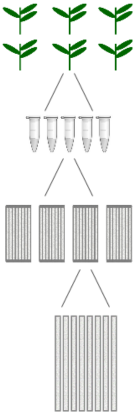


1. at the top layer: biological variations (*i.e.*, individual differences due to *e.g.*, environmental or genetic factors)

2. at the middle layer: technical variations (library preparation effect)

3. at the bottom layer: technical variations (lane and cell flow effects)

Source of variation in RNA-seq experiments



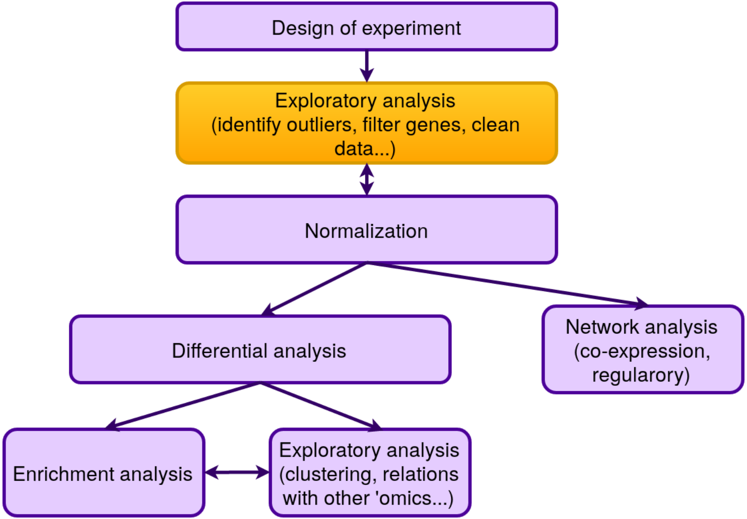
1. at the top layer: biological variations (*i.e.*, individual differences due to *e.g.*, environmental or genetic factors)

2. at the middle layer: technical variations (library preparation effect)

3. at the bottom layer: technical variations (lane and cell flow effects)

lane effect < cell flow effect < library preparation effect ≪ biological effect ⇒ 2 × 3 biological replicates at least **[Liu et al., 2014]**

Part II: Exploratory analysis



Design of experiment

Exploratory analysis (identify outliers, filter genes, clean data...)

Normalization

Differential analysis

Network analysis (co-expression, regularory)

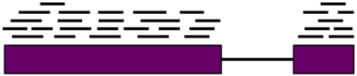
Enrichment analysis

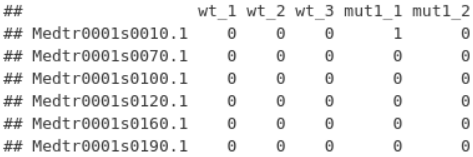
Exploratory analysis (clustering, relations with other 'omics...)

Some features of RNAseq data

What must be taken into account?

- discrete, non-negative data (total number of aligned reads)



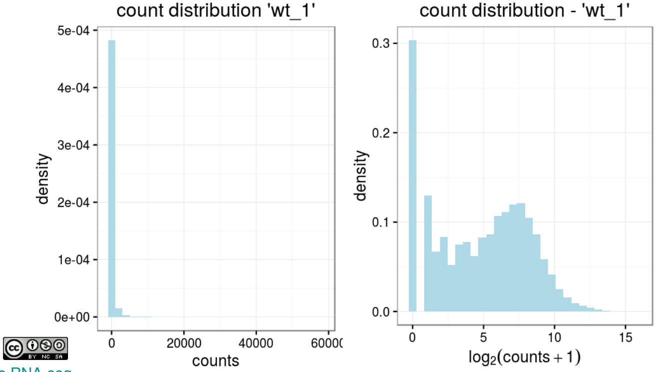


```
##                       wt_1 wt_2 wt_3 mut1_1 mut1_2
## Medtr0001s0010.1    0    0    0      1      0
## Medtr0001s0070.1    0    0    0      0      0
## Medtr0001s0100.1    0    0    0      0      0
## Medtr0001s0120.1    0    0    0      0      0
## Medtr0001s0160.1    0    0    0      0      0
## Medtr0001s0190.1    0    0    0      0      0
```

Some features of RNAseq data

What must be taken into account?

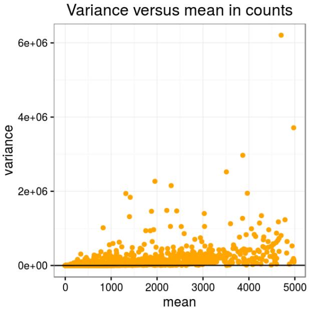
- discrete, non-negative data (total number of aligned reads)
- skewed data



Some features of RNAseq data

What must be taken into account?

- discrete, non-negative data (total number of aligned reads)
- skewed data
- overdispersion (variance $\gg$ mean)



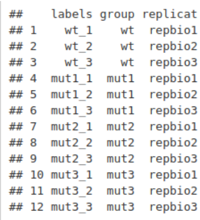
black line is “variance = mean”

Dataset used in the examples

Dataset provided by courtesy of the transcriptomic platform of IPS2

Three files:

- `D1-counts.txt` contains the raw counts of the experiment (13 columns: the first one contains the gene names, the others correspond to 12 different samples; gene names have been shuffled);
- `D1-genesLength.txt` contains information about gene lengths;
- `D1-targets.txt` contains information about the sample and the experimental design.



```
##    labels group replicat
## 1    wt_1    wt  repbio1
## 2    wt_2    wt  repbio2
## 3    wt_3    wt  repbio3
## 4  mut1_1  mut1  repbio1
## 5  mut1_2  mut1  repbio2
## 6  mut1_3  mut1  repbio3
## 7  mut2_1  mut2  repbio1
## 8  mut2_2  mut2  repbio2
## 9  mut2_3  mut2  repbio3
## 10 mut3_1  mut3  repbio1
## 11 mut3_2  mut3  repbio2
## 12 mut3_3  mut3  repbio3
```

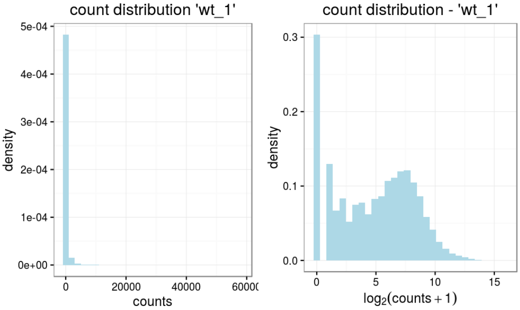
Dataset used in the examples

These text files are loaded with:

```
raw_counts <- read.table("D1-counts.txt", header = TRUE,
                         row.names = 1)
raw_counts <- as.matrix(raw_counts)
design <- read.table("D1-targets.txt", header = TRUE,
                     stringsAsFactors = FALSE)
gene_lengths <- scan("D1-genesLength.txt")
```

Count distribution

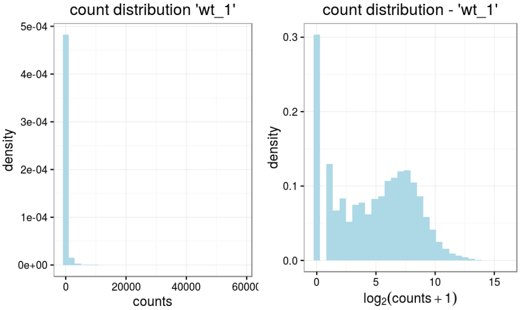
The count distribution (*i.e.*, the number of times a given count is obtained in the data) can be visualized with histograms (boxplots or violin plots can also be used):



This distribution is highly skewed and it is better visualized using a $\log_2$ transformation before it is displayed.

Count distribution

The count distribution (*i.e.*, the number of times a given count is obtained in the data) can be visualized with histograms (boxplots or violin plots can also be used):

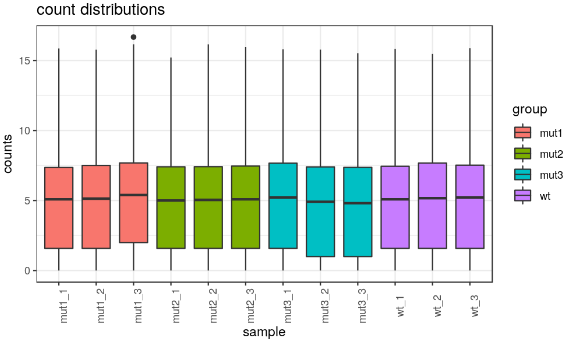


This distribution is highly skewed and it is better visualized using a $\log_2$ transformation before it is displayed.

The library size is the sum of all counts in a given sample.

Count distribution between samples

The count distribution between different samples can be compared with parallel boxplots or violin plots:



It is expected that, within a given condition (group), the count distributions are similar. The same is often also expected between conditions.

Check reproducibility between samples

MA plots can be used to visualize reproducibility between samples of an experiment (and thus check if normalization is needed). They plot the log-fold change (M-values) against the log-average (A-values):

M-values: log of ratio between counts between two samples:

$$M_g = \log_2(K_{gj}) - \log_2(K_{gj'})$$

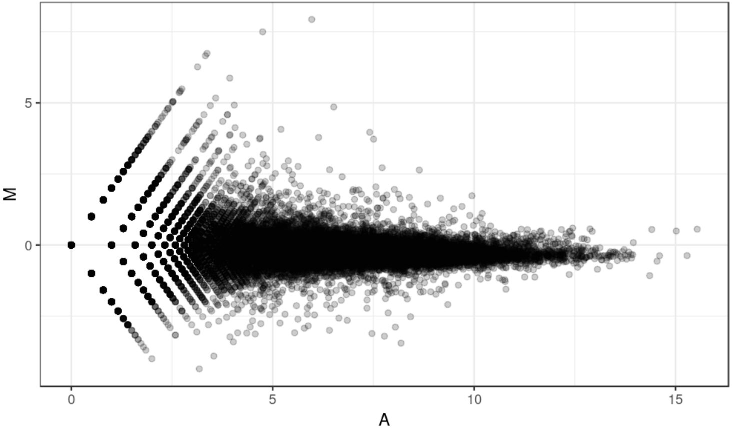
A-values: average log counts between two samples:

$$A_g = \frac{\log_2(K_{gj}) + \log_2(K_{gj'})}{2}$$

where K_{gj} stands for the counts for gene g in sample j.

Check reproducibility between samples

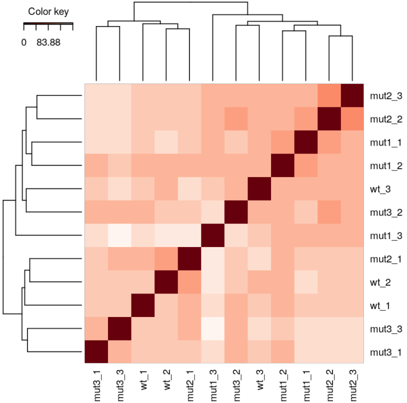
MA plots can be used to visualize reproducibility between samples of an experiment (and thus check if normalization is needed). They plot the log-fold change (M-values) against the log-average (A-values):



Check similarity between samples

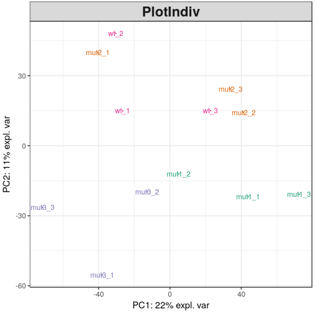
Similarities between samples can be visualized with a HAC and a heatmap:

- perform hierarchical ascending classification (HAC) using Euclidean distance between samples: $\delta(j, j') = \sum_g \left(\log_2(K_{gj}) - \log_2(K_{gj'})\right)^2$
- visualize the strength of the similarity with heatmap.



Search for the main structures in the data: PCA

PCA (on $\log_2$ counts) can be used to project data into a small dimensional space and search for unexpected experimental effects in the data.



(MDS is equivalent to PCA when used with the standard Euclidean distance)

Remark: In **DESeq**, the function `plotPCA` performs PCA on the top genes with the highest variance.

Outline

Exploratory analysis

- Introduction
- Experimental design
- Data exploration and quality assessment

Normalization

- Raw data filtering
- Interpreting read counts

Differential Expression analysis

- Hypothesis testing and correction for multiple tests
- Differential expression analysis for RNAseq data
- Interpreting and improving the analysis

Raw data filtering

Filtering consists in removing genes with low expression. Different strategies can be used:

- [Sultan et al., 2008]: filter out genes with a total read count smaller than a given threshold;
- [Bottomly et al., 2011]: filter out genes with zero count in an experimental condition;
- [Robinson and Oshlack, 2010]: filter out genes such that the number of samples with a CPM value (for this gene) smaller than a given threshold is larger than the smallest number of samples in a condition. With CPM: Count Per Million (*i.e.*, raw count divived by library size, this strategy takes into account differences in library sizes).

Raw data filtering

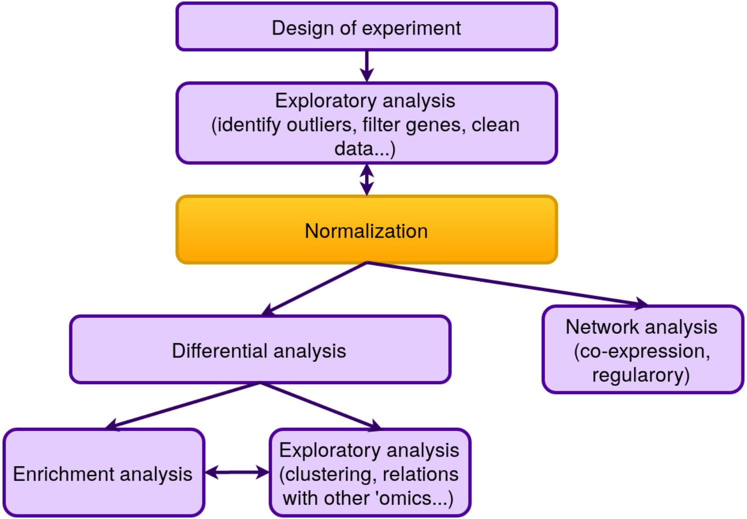
Filtering consists in removing genes with low expression. Different strategies can be used:

- [Sultan et al., 2008]: filter out genes with a total read count smaller than a given threshold;
- [Bottomly et al., 2011]: filter out genes with zero count in an experimental condition;
- [Robinson and Oshlack, 2010]: filter out genes such that the number of samples with a CPM value (for this gene) smaller than a given threshold is larger than the smallest number of samples in a condition. With CPM: Count Per Million (*i.e.*, raw count divived by library size, this strategy takes into account differences in library sizes).

More sophisticated filtering

To account for the fact that lowly expressed genes are almost never found differentially expressed, a more sophisticated filtering can be performed.

Part III: Normalization


Design of experiment
Exploratory analysis (identify outliers, filter genes, clean data...)
Normalization
Differential analysis
Network analysis (co-expression, regularory)
Enrichment analysis
Exploratory analysis (clustering, relations with other 'omics...)

Purpose of normalization

- identify and correct technical biases (due to sequencing process) to make counts comparable

- types of normalization: within sample normalization and between sample normalization

Within sample normalization

Example: (read counts)

	sample 1	sample 2	sample 3
gene A	752	615	1203
gene B	1507	1225	2455

counts for gene B are twice larger than counts for gene A because:

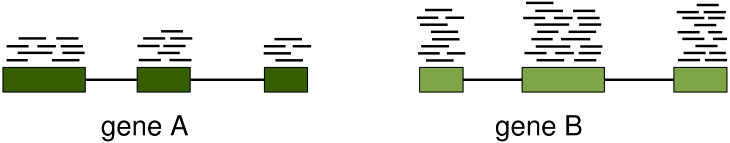
Within sample normalization

Example: (read counts)

	sample 1	sample 2	sample 3
gene A	752	615	1203
gene B	1507	1225	2455

counts for gene B are twice larger than counts for gene A because:

- gene B is expressed with a number of transcripts twice larger than gene A



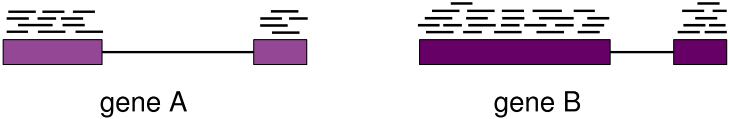
Within sample normalization

Example: (read counts)

	sample 1	sample 2	sample 3
gene A	752	615	1203
gene B	1507	1225	2455

counts for gene B are twice larger than counts for gene A because:

- both genes are expressed with the same number of transcripts but gene B is twice longer than gene A



Within sample normalization

- Purpose of within sample comparison: enabling comparisons of genes from a same sample
- Sources of variability: gene length, sequence composition (GC content)

These differences need not to be corrected for a differential analysis and are not really relevant for data interpretation.

Between sample normalization

Example: (read counts)

	sample 1	sample 2	sample 3
gene A	752	615	1203
gene B	1507	1225	2455

counts in sample 3 are much larger than counts in sample 2 because:

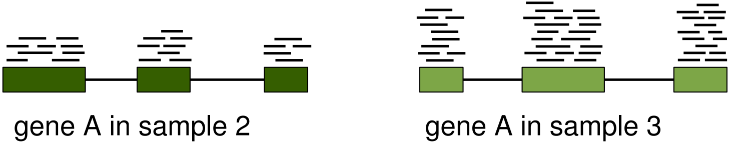
Between sample normalization

Example: (read counts)

	sample 1	sample 2	sample 3
gene A	752	615	1203
gene B	1507	1225	2455

counts in sample 3 are much larger than counts in sample 2 because:

- gene A is more expressed in sample 3 than in sample 2



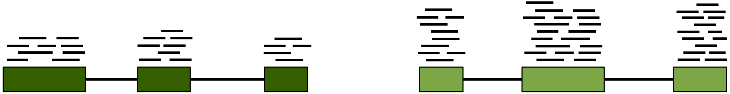
Between sample normalization

Example: (read counts)

	sample 1	sample 2	sample 3
gene A	752	615	1203
gene B	1507	1225	2455

counts in sample 3 are much larger than counts in sample 2 because:

- gene A is expressed similarly in the two samples but sequencing depth is larger in sample 3 than in sample 2 (*i.e.*, differences in library sizes)



Between sample normalization

- Purpose of between sample comparison: enabling comparisons of a gene in different samples
- Sources of variability: library size, ...

These differences must be corrected for a differential analysis and for data interpretation.

Principles for sequencing depth normalization

Basics

1. choose an appropriate baseline for each sample
2. for a given gene, compare counts relative to the baseline rather than raw counts

Principles for sequencing depth normalization

Basics

1. choose an appropriate baseline for each sample
2. for a given gene, compare counts relative to the baseline rather than raw counts

In practice: Raw counts correspond to different sequencing depths

	control				treated		
Gene 1	5	1	0	0	**4**	**0**	**0**
Gene 2	0	2	1	2	**1**	**0**	**0**
Gene 3	92	161	76	70	**140**	**88**	**70**
:	:				:		
:	:				:		
:	:				:		
Gene G	15	25	9	5	**20**	**14**	**17**

Principles for sequencing depth normalization

Basics

1. choose an appropriate baseline for each sample
2. for a given gene, compare counts relative to the baseline rather than raw counts

In practice: A co mple

	control				treated		
Gene 1	5	1	0	0	**4**	**0**	**0**
Gene 2	0	2	1	2	**1**	**0**	**0**
Gene 3	92	161	76	70	**140**	**88**	**70**
:		:			:		
:		:			:		
:		:			:		
Gene G	15	25	9	5	**20**	**14**	**17**
C_j	1.1	1.6	0.6	0.7	**1.4**	**0.7**	**0.8**

Principles for sequencing depth normalization

Basics

1. choose an appropriate baseline for each sample
2. for a given gene, compare counts relative to the baseline rather than raw counts

In practice: Every counts is multiplied by the correction factor corresponding to its sample

Gene 3	92	161	76	70	**140**	**88**	**70**	
C_j	1.1	1.6	0.6	0.7	**1.4**	**0.7**	**0.8**	
								x
Gene 3	101.2	257.6	45.6	49	**196**	**61.6**	**56**	

Principles for sequencing depth normalization

Basics

1. choose an appropriate baseline for each sample
2. for a given gene, compare counts relative to the baseline rather than raw counts

Consequences: Library sizes for normalized counts are roughly equal.

	control				treated			
Gene 1	5.5	1.6	0	0	5.6	0	0	
Gene 2	0	3.2	0.6	1.4	1.4	0	0	
Gene 3	101.2	257.6	45.6	49	196	61.6	56	
:		:				:		
:		:				:		
:		:				:		
Gene G	16.5	40	5.4	5.5	28	9.8	13.6	+
Lib. size	13.1	13.0	13.2	13.1	13.2	13.0	13.1	x 10^5

Principles for sequencing depth normalization

Definition

If K_{gj} is the raw count for gene g in sample j then, the normalized counts is defined as:

$$\widetilde{K}_{gj} = \frac{K_{gj}}{s_j \times D_j} \times 10^6$$

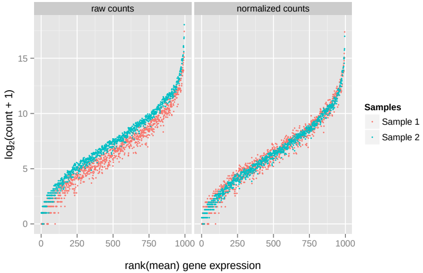
in which: $D_j = \sum_g K_{gj}$ is the library size of sample j, s_j is the correction factor of the library size for sample j and thus $C_j = \frac{10^6}{s_j D_j}$.

Distribution adjustment

- Total read count adjustment [Mortazavi et al., 2008]

$$s_j = 1 \qquad \text{and thus: } \widetilde{K}_{gj} = \frac{K_{gj}}{D_j} \times 10^6$$

(Counts Per Million).



edgeR:

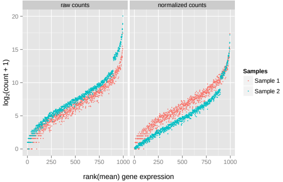
```
cpm(...,
    normalized.lib.sizes=FALSE)
```

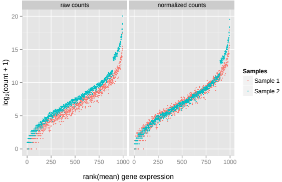
Distribution adjustment

- Total read count adjustment [Mortazavi et al., 2008]
- (Upper) Quartile normalization [Bullard et al., 2010]

$$s_j = \frac{Q_j^{(p)}}{\frac{1}{N}\sum_{l=1}^{N} Q_l^{(p)}}$$

N: number of samples, $Q_j^{(p)}$: quantile in sample j





edgeR:

```
calcNormFactors(..., method = "upperquartile",
                 p = 0.75)
```

Method using gene lengths (intra & inter sample normalization)

RPKM: Reads Per Kilobase per Million mapped Reads

Assumptions: read counts are proportional to expression level, transcript length and sequencing depth

$$s_j = \frac{D_j L_g}{10^3 \times 10^6}$$

in which L_g is gene length (bp).

edgeR:

```
rpkm(..., gene.length = ...)
```

Unbiaised estimation of number of reads but affect variability [Oshlack and Wakefield, 2009].

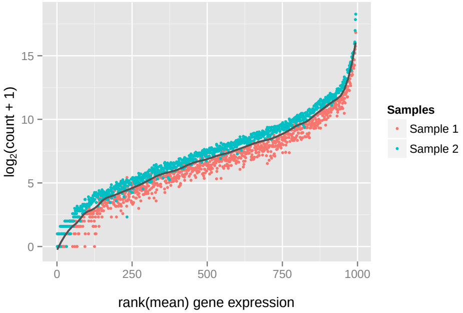
Relative Log Expression (RLE)

Method:

1. compute a pseudo-reference sample: geometric mean across samples

$$R_g = \left(\prod_{j=1}^{N} K_{gj} \right)^{1/N}$$

(geometric mean is less sensitive to extreme values than standard mean)

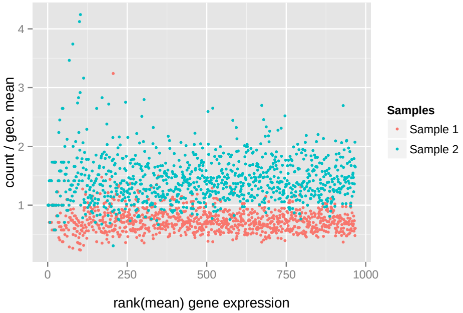


Relative Log Expression (RLE)

Method:

1. compute a pseudo-reference sample
2. center samples compared to the reference

$$\tilde{K}_{gj} = \frac{K_{gj}}{R_g} \qquad \text{with} \qquad R_g = \left(\prod_{j=1}^{N} K_{gj} \right)^{1/N}$$

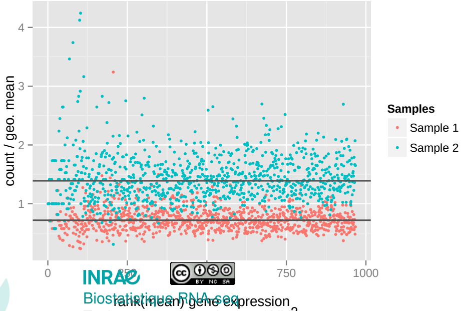


Relative Log Expression (RLE)

Method:

1. compute a pseudo-reference sample
2. center samples compared to the reference
3. compute normalization factor: median of centered counts over the genes

$$\tilde{s}_j = \underset{g}{\text{median}} \left\{ \tilde{K}_{gj} \right\} \quad \text{factors multiply to 1:} \quad s_j = \frac{\tilde{s}_j}{\exp\left(\frac{1}{N}\sum_{l=1}^{N} \log(\tilde{s}_l)\right)}$$



with

$$\tilde{K}_{gj} = \frac{K_{gj}}{R_g}$$

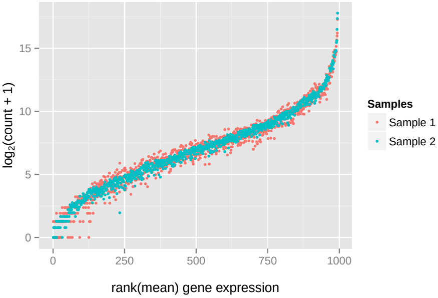
and

$$R_g = \left(\prod_{j=1}^{N} K_{gj} \right)^{1/N}$$

Relative Log Expression (RLE)

Method:

1. compute a pseudo-reference sample
2. center samples compared to the reference
3. compute normalization factor: median of centered counts over the genes



```
## with edgeR
calcNormFactors(...,
  method="RLE")

## with DESeq
estimateSizeFactors(...)
```

Trimmed Mean of M-values (TMM)

Assumptions behind the method

- the total read count strongly depends on a few highly expressed genes
- most genes are not differentially expressed

Trimmed Mean of M-values (TMM)

Assumptions behind the method

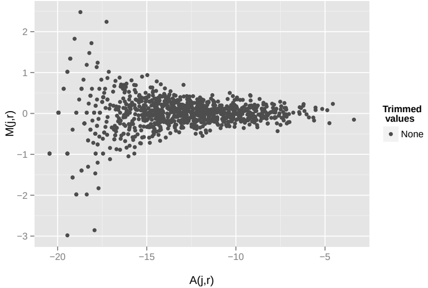
- the total read count strongly depends on a few highly expressed genes
- most genes are not differentially expressed

⇒ remove extreme data for fold-changed (M) and average intensity (A)

$$M_g(j,r) = \log_2\left(\frac{K_{gj}}{D_j}\right) - \log_2\left(\frac{K_{gr}}{D_r}\right) \qquad A_g(j,r) = \frac{1}{2}\left[\log_2\left(\frac{K_{gj}}{D_j}\right) + \log_2\left(\frac{K_{gr}}{D_r}\right)\right]$$

select as a reference sample, the sample *r* with the upper quartile closest to the average upper quartile

M- vs A-values



Trimmed Mean of M-values (TMM)

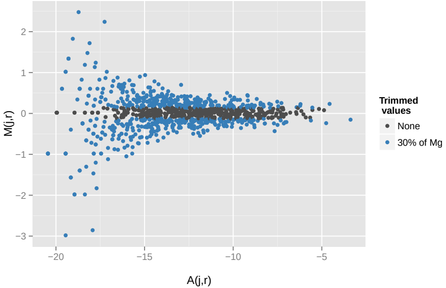
Assumptions behind the method

- the total read count strongly depends on a few highly expressed genes
- most genes are not differentially expressed

⇒ remove extreme data for fold-changed (M) and average intensity (A)

$$M_g(j,r) = \log_2\left(\frac{K_{gj}}{D_j}\right) - \log_2\left(\frac{K_{gr}}{D_r}\right) \qquad A_g(j,r) = \frac{1}{2}\left[\log_2\left(\frac{K_{gj}}{D_j}\right) + \log_2\left(\frac{K_{gr}}{D_r}\right)\right]$$

Trim 30% on M-values



Trimmed Mean of M-values (TMM)

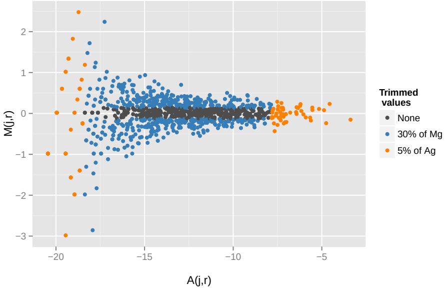
Assumptions behind the method

- the total read count strongly depends on a few highly expressed genes
- most genes are not differentially expressed

⇒ remove extreme data for fold-changed (M) and average intensity (A)

$$M_g(j,r) = \log_2\left(\frac{K_{gj}}{D_j}\right) - \log_2\left(\frac{K_{gr}}{D_r}\right) \qquad A_g(j,r) = \frac{1}{2}\left[\log_2\left(\frac{K_{gj}}{D_j}\right) + \log_2\left(\frac{K_{gr}}{D_r}\right)\right]$$

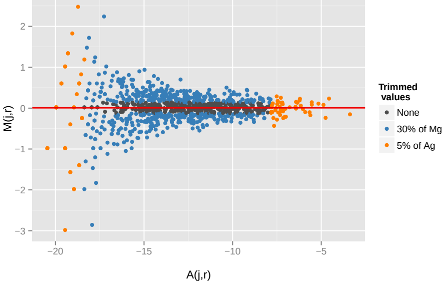
Trim 5% on A-values



Trimmed Mean of M-values (TMM)

Assumptions behind the method

- the total read count strongly depends on a few highly expressed genes
- most genes are not differentially expressed



On remaining data, compute the weighted mean of M-values:

$$\text{TMM}(j,r) = \frac{\sum_{g:\text{not trimmed}} w_g(j,r) M_g(j,r)}{\sum_{g:\text{not trimmed}} w_g(j,r)}$$

with $w_g(j,r) = \left(\frac{D_j - K_{gj}}{D_j K_{gj}} + \frac{D_r - K_{gr}}{D_r K_{gr}}\right)$.

Trimmed Mean of M-values (TMM)

Assumptions behind the method

- the total read count strongly depends on a few highly expressed genes
- most genes are not differentially expressed

Correction factors:

$$\tilde{s}_j = 2^{\mathrm{TMM}(j,r)} \quad \text{factors multiply to 1:} \quad s_j = \frac{\tilde{s}_j}{\exp\left(\frac{1}{N}\sum_{l=1}^{N} \log(\tilde{s}_l)\right)}$$

```
calcNormFactors(..., method="TMM")
```

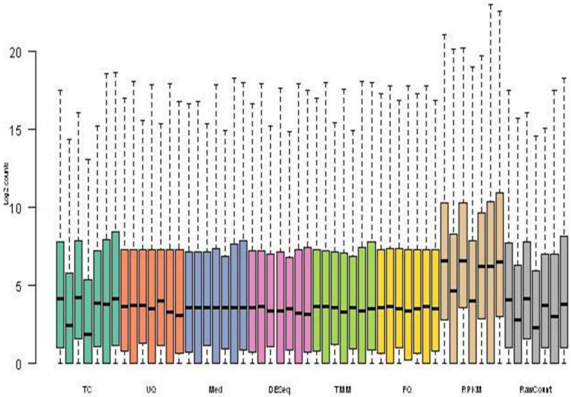
Comparison of the different approaches

Purpose of the comparison:

- finding the "best" method for all cases is not a realistic purpose
- find an approach which is robust enough to provide relevant results in all cases
- Method: comparison based on several criteria to select a method which is valid for multiple objectives

Comparison of the different approaches

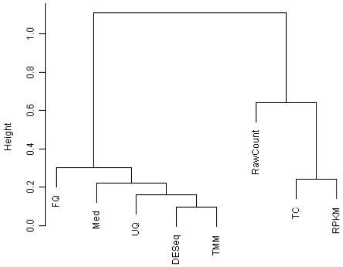
Effect on count distribution:

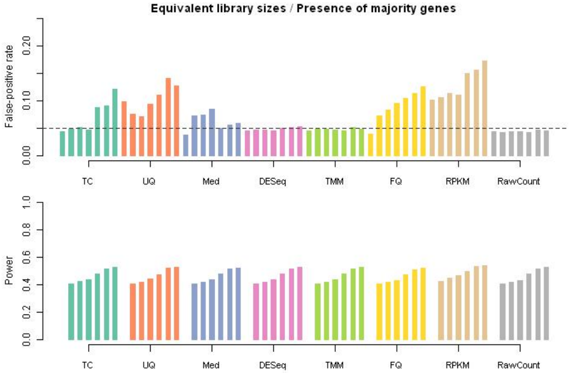


RPKM and TC are very similar to raw data.

Comparison of the different approaches

Effect on differential analysis (DESeq v. 1.6):

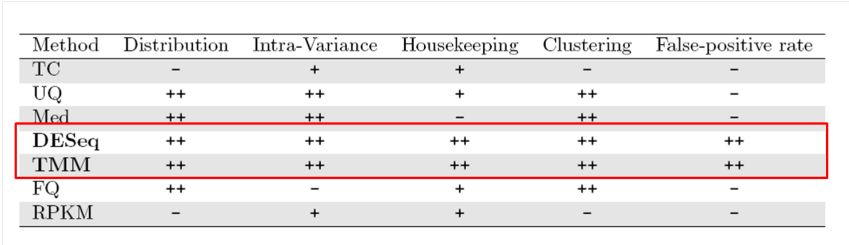




Inflated FPR for all methods except for TMM and DESeq (RLE).

Comparison of the different approaches

Conclusion: Differences appear based on data characteristics



Method	Distribution	Intra-Variance	Housekeeping	Clustering	False-positive rate
TC	–	+	+	–	–
UQ	++	++	+	++	–
Med	++	++	–	++	–
DESeq	++	++	++	++	++
TMM	++	++	++	++	++
FQ	++	–	+	++	–
RPKM	–	+	+	–	–

TMM and DESeq (RLE) are performant in a differential analysis context.

Outline

Exploratory analysis

- Introduction
- Experimental design
- Data exploration and quality assessment

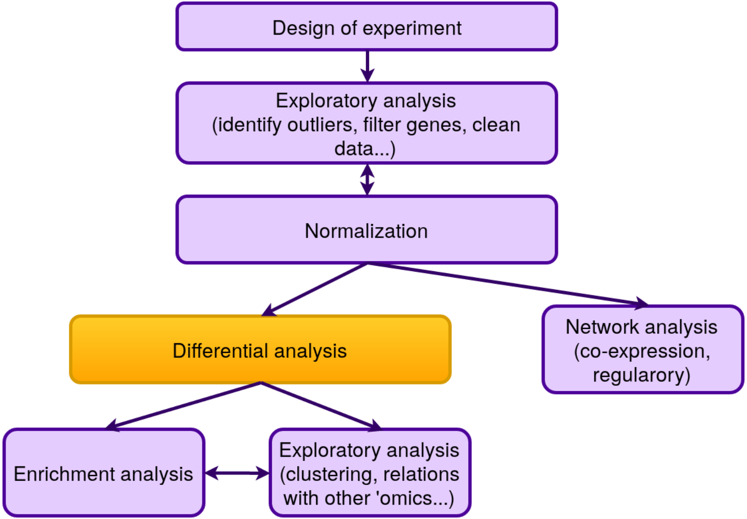
Normalization

- Raw data filtering
- Interpreting read counts

Differential Expression analysis

- Hypothesis testing and correction for multiple tests
- Differential expression analysis for RNAseq data
- Interpreting and improving the analysis

Part IV: Differential expression analysis



Design of experiment

Exploratory analysis (identify outliers, filter genes, clean data...)

Normalization

Differential analysis

Network analysis (co-expression, regularory)

Enrichment analysis

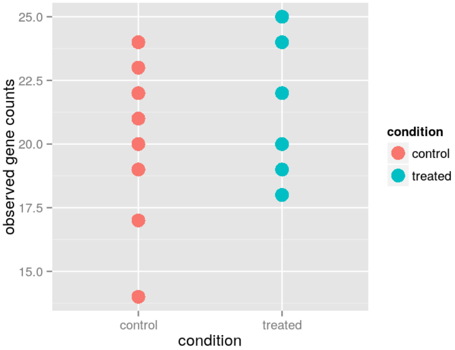
Exploratory analysis (clustering, relations with other 'omics...)

Different steps in hypothesis testing

1. formulate an hypothesis H_0:

   H_0: the average count for gene *g* in the control samples is the same that the average count in the treated samples

   which is tested against an alternative H_1: the average count for gene *g* in the control samples is different from the average count in the treated samples

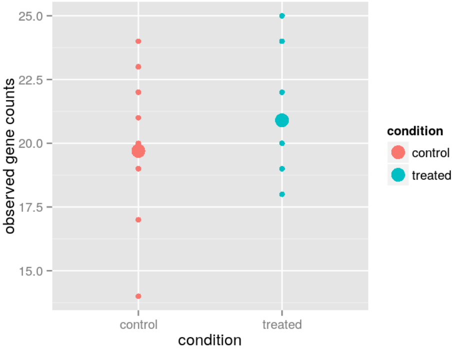


Different steps in hypothesis testing

1. formulate an hypothesis H_0:

   H_0: the average count for gene *g* in the control samples is the same that the average count in the treated samples

2. from observations, compute a test statistics (*e.g.*, the mean in the two samples)

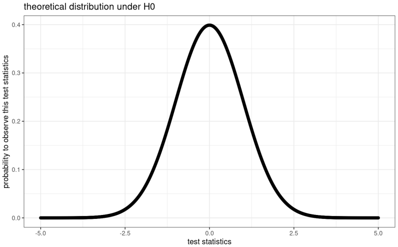


Different steps in hypothesis testing

1. formulate an hypothesis H_0:

   H_0: the average count for gene *g* in the control samples is the same that the average count in the treated samples

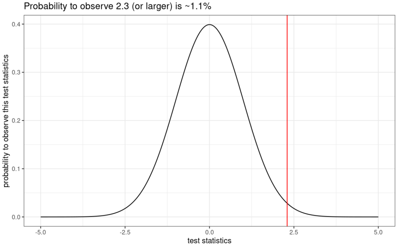
2. from observations, compute a test statistics (*e.g.*, the mean in the two samples)
3. find the theoretical distribution of the test statistics under H_0



Different steps in hypothesis testing

1. formulate an hypothesis H_0:

   H_0: the average count for gene *g* in the control samples is the same that the average count in the treated samples
2. from observations, compute a test statistics (*e.g.*, the mean in the two samples)
3. find the theoretical distribution of the test statistics under H_0
4. deduce the probability that the observations occur under H_0: this is called the p-value



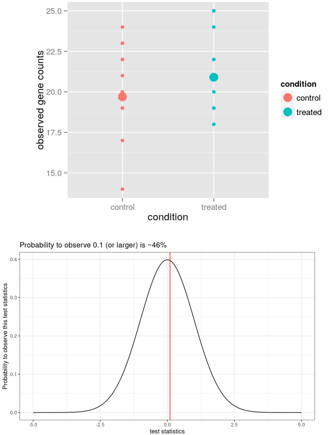
Different steps in hypothesis testing

1. formulate an hypothesis H_0:

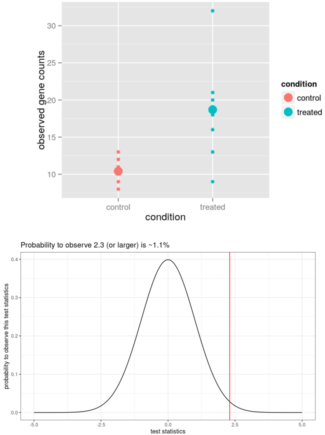
   H_0: the average count for gene *g* in the control samples is the
   same that the average count in the treated samples
2. from observations, compute a test statistics (*e.g.*, the mean in the two samples)
3. find the theoretical distribution of the test statistics under H_0
4. deduce the probability that the observations occur under H_0: this is called the p-value
5. conclude: if the p-value is low (usually below $\alpha = 5\%$ as a convention), H_0 is unlikely: we say that "H_0 is rejected".

   We have that: $\alpha = \mathbb{P}_{H_0}(H_0 \text{ is rejected})$.

Summary of the possible decisions



Do not reject H_0



Reject H_0

Types of errors in tests

		Reality	
		H_0 is true	H_0 is false
Decision	Do not reject H_0	Correct decision ☺ **(True Negative)**	Type II error ☹ **(False Negative)**
	Reject H_0	Type I error ☹ **(False Positive)**	Correct decision ☺ **(True Positive)**

$$\mathbb{P}(\text{Type I error}) = \alpha \text{ (risk)}$$

$$\mathbb{P}(\text{Type II error}) = 1 - \beta \text{ } (\beta\text{: power})$$

Why performing a large number of tests might be a problem?

Framework: Suppose you are performing G tests at level α.

$$\mathbb{P}(\text{at least one FP if H}_0\text{ is always true}) = 1 - (1 - \alpha)^G$$

Ex: for $\alpha = 5\%$ and $G = 20$, $\mathbb{P}(\text{at least one FP if H}_0\text{ is always true}) \simeq 64\%$!!!

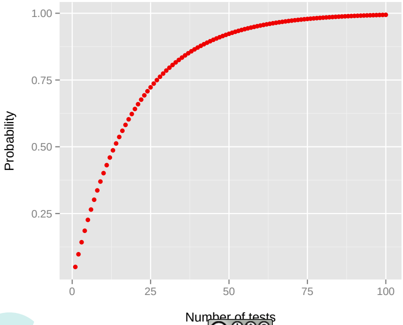
Why performing a large number of tests might be a problem?

Framework: Suppose you are performing G tests at level α.

$$\mathbb{P}(\text{at least one FP if } H_0 \text{ is always true}) = 1 - (1 - \alpha)^G$$

Ex: for $\alpha = 5\%$ and $G = 20$, $\mathbb{P}(\text{at least one FP if } H_0 \text{ is always true}) \simeq 64\%$!!!

Probability to have at least one false positive versus the number of tests performed when H_0 is true for all G tests



For more than 75 tests and if H_0 is always true, the probability to have at least one false positive is very close to 100%!

Notations for multiple tests

Number of decisions for G independent tests:

	True null hypotheses	False null hypotheses	Total
Not rejected	$G_0 - U$	$G_1 - V$	$G - R$
Rejected	U	V	R
Total	G_0	G_1	G

Notations for multiple tests

Number of decisions for G independent tests:

	True null hypotheses	False null hypotheses	Total
Not rejected	$G_0 - U$	$G_1 - V$	$G - R$
Rejected	U	V	R
Total	G_0	G_1	G

Instead of the risk α, control:

- familywise error rate (FWER): FWER $= \mathbb{P}(U > 0)$ (*i.e.*, probability to have at least one false positive decision)
- false discovery rate (FDR): FDR $= \mathbb{E}(Q)$ with

$$Q = \begin{cases} U/R & \text{if } R > 0 \\ 0 & \text{otherwise} \end{cases}$$

Adjusted p-values

Settings: p-values $p_1, \dots, p_G$ (*e.g.*, corresponding to G tests on G different genes)

Adjusted p-values

adjusted p-values are $\tilde{p}_1, \dots, \tilde{p}_G$ such that

$$\text{Rejecting tests such that } \tilde{p}_g < \alpha \quad \Longleftrightarrow \quad \mathbb{P}(U > 0) \leq \alpha \text{ or } \mathbb{E}(Q) \leq \alpha$$

Adjusted p-values

Settings: p-values $p_1, \dots, p_G$ (*e.g.*, corresponding to G tests on G different genes)

Adjusted p-values

adjusted p-values are $\tilde{p}_1, \dots, \tilde{p}_G$ such that

$$\text{Rejecting tests such that } \tilde{p}_g < \alpha \quad \Longleftrightarrow \quad \mathbb{P}(U > 0) \leq \alpha \text{ or } \mathbb{E}(Q) \leq \alpha$$

Computing p-values

1. order the p-values $p_{(1)} \leq p_{(2)} \leq \dots \leq p_{(G)}$

Adjusted p-values

Settings: p-values $p_1, \dots, p_G$ (*e.g.*, corresponding to G tests on G different genes)

Adjusted p-values

adjusted p-values are $\tilde{p}_1, \dots, \tilde{p}_G$ such that

$$\text{Rejecting tests such that } \tilde{p}_g < \alpha \quad \Longleftrightarrow \quad \mathbb{P}(U > 0) \leq \alpha \text{ or } \mathbb{E}(Q) \leq \alpha$$

Computing p-values

1. order the p-values $p_{(1)} \leq p_{(2)} \leq \dots \leq p_{(G)}$
2. compute $\tilde{p}_{(g)} = a_g p_{(g)}$
   - with Bonferroni method: $a_g = G$ (FWER)
   - with Benjamini & Hochberg method: $a_g = G/g$ (FDR)

Adjusted p-values

Settings: p-values $p_1, \dots, p_G$ (*e.g.*, corresponding to G tests on G different genes)

Adjusted p-values

adjusted p-values are $\tilde{p}_1, \dots, \tilde{p}_G$ such that

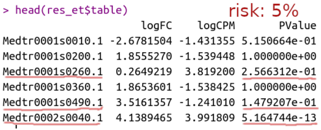
$$\text{Rejecting tests such that } \tilde{p}_g < \alpha \quad \Longleftrightarrow \quad \mathbb{P}(U > 0) \leq \alpha \text{ or } \mathbb{E}(Q) \leq \alpha$$

Computing p-values

1. order the p-values $p_{(1)} \leq p_{(2)} \leq \dots \leq p_{(G)}$
2. compute $\tilde{p}_{(g)} = a_g p_{(g)}$
   - with Bonferroni method: $a_g = G$ (FWER)
   - with Benjamini & Hochberg method: $a_g = G/g$ (FDR)
3. if adjusted p-values $\tilde{p}_{(g)}$ are larger than 1, correct $\tilde{p}_{(g)} \leftarrow \min\{\tilde{p}_{(g)}, 1\}$

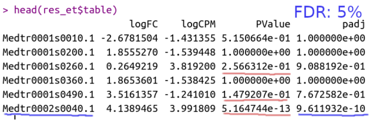
Adjusting p-values in practice

- compute adjusted p-values (Bonferroni or BH procedures for instance)
- select all genes for which this adjusted p-values is below 5% (for instance)
- this is equivalent to controlling either the probability to have at least one FP (FWER) or the average proportion of FP (FDR)



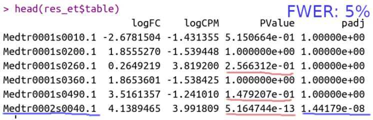
risk: 5%

```
> head(res_et$table)
                     logFC    logCPM       PValue
Medtr0001s0010.1 -2.6781504 -1.431355 5.150664e-01
Medtr0001s0200.1  1.8555270 -1.539448 1.000000e+00
Medtr0001s0260.1  0.2649219  3.819200 2.566312e-01
Medtr0001s0360.1  1.8653601 -1.538425 1.000000e+00
Medtr0001s0490.1  3.5161357 -1.241010 1.479207e-01
Medtr0002s0040.1  4.1389465  3.991809 5.164744e-13
```



FDR: 5%

```
> head(res_et$table)
                     logFC    logCPM       PValue         padj
Medtr0001s0010.1 -2.6781504 -1.431355 5.150664e-01 1.000000e+00
Medtr0001s0200.1  1.8555270 -1.539448 1.000000e+00 1.000000e+00
Medtr0001s0260.1  0.2649219  3.819200 2.566312e-01 9.088192e-01
Medtr0001s0360.1  1.8653601 -1.538425 1.000000e+00 1.000000e+00
Medtr0001s0490.1  3.5161357 -1.241010 1.479207e-01 7.672582e-01
Medtr0002s0040.1  4.1389465  3.991809 5.164744e-13 9.611932e-10
```



FWER: 5%

```
> head(res_et$table)
                     logFC    logCPM       PValue        padj
Medtr0001s0010.1 -2.6781504 -1.431355 5.150664e-01 1.00000e+00
Medtr0001s0200.1  1.8555270 -1.539448 1.000000e+00 1.00000e+00
Medtr0001s0260.1  0.2649219  3.819200 2.566312e-01 1.00000e+00
Medtr0001s0360.1  1.8653601 -1.538425 1.000000e+00 1.00000e+00
Medtr0001s0490.1  3.5161357 -1.241010 1.479207e-01 1.00000e+00
Medtr0002s0040.1  4.1389465  3.991809 5.164744e-13 1.44179e-08
```

Fisher's exact test for contingency tables

After normalization, one may build a contingency table like this one:

	treated	control	Total
gene g	n_{gA}	n_{gB}	n_g
other genes	$N_A - n_{gA}$	$N_B - n_{gB}$	$N - n_g$
Total	N_A	N_B	N

Question: is the number of reads of gene g in the treated sample significatively different than in the control sample?

Fisher's exact test for contingency tables

After normalization, one may build a contingency table like this one:

	treated	control	Total
gene g	n_{gA}	n_{gB}	n_g
other genes	$N_A - n_{gA}$	$N_B - n_{gB}$	$N - n_g$
Total	N_A	N_B	N

Question: is the number of reads of gene g in the treated sample significatively different than in the control sample?

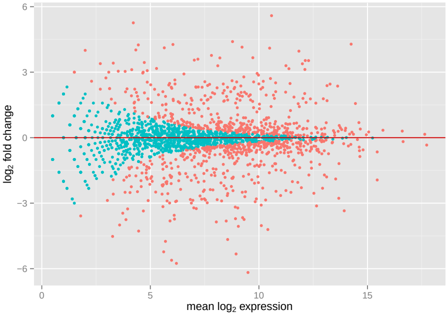
Method

Direct computation of the probability to obtain such a contingency table (or a "more extreme" contingency table) with:

- independency between the two columns of the contingency tables;
- the same marginals ("Total").

Example of results obtained with the Fisher test

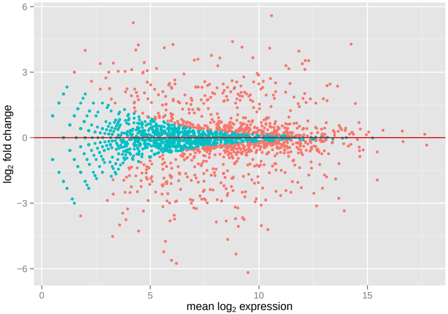
Genes declared significantly differentially expressed are in pink:



Main remark: more conservative for genes with a low expression

Example of results obtained with the Fisher test

Genes declared significantly differentially expressed are in pink:



Main remark: more conservative for genes with a low expression

Limitation of Fisher test

Highly expressed genes have a very large variance! As Fisher test does not estimate variance, it tends to detect false positives among highly expressed genes ⇒ do not use it!

Basic principles of tests for count data: 2 conditions and replicates

Notations: for gene g, K^1_{g1}, ..., $K^1_{gn_1}$ (condition 1) and K^2_{g1}, ..., $K^2_{gn_2}$ (condition 2)

- choose an appropriate distribution to model count data (discrete data, overdispersion)
- estimate its parameters for both conditions
- conclude by computing p-value

Basic principles of tests for count data: 2 conditions and replicates

Notations: for gene g, K^1_{g1}, ..., $K^1_{gn_1}$ (condition 1) and K^2_{g1}, ..., $K^2_{gn_2}$ (condition 2)

- choose an appropriate distribution to model count data (discrete data, overdispersion)

  $$K^k_{gj} \sim \mathrm{NB}(s^k_j \lambda_{gk}, \phi_g)$$

  in which:
  - s^k_j is library correction factor of sample j in condition k
  - λ_{gk} is the proportion of counts for gene g in condition k
  - ϕ_g is the (over)dispersion (parameter) of gene g (supposed to be identical for all samples)
- estimate its parameters for both conditions
- conclude by computing p-value

Basic principles of tests for count data: 2 conditions and replicates

Notations: for gene g, K^1_{g1}, ..., $K^1_{gn_1}$ (condition 1) and K^2_{g1}, ..., $K^2_{gn_2}$ (condition 2)

- choose an appropriate distribution to model count data (discrete data, overdispersion)

  $$K^k_{gj} \sim \text{NB}(s^k_j \lambda_{gk}, \phi_g)$$

  in which:

  - s^k_j is library correction factor of sample j in condition k
  - λ_{gk} is the proportion of counts for gene g in condition k
  - ϕ_g is the (over)dispersion (parameter) of gene g (supposed to be identical for all samples)
- estimate its parameters for both conditions

  λ_{g1} λ_{g2} ϕ_g
- conclude by computing p-value

Basic principles of tests for count data: 2 conditions and replicates

Notations: for gene g, K^1_{g1}, ..., $K^1_{gn_1}$ (condition 1) and K^2_{g1}, ..., $K^2_{gn_2}$ (condition 2)

- choose an appropriate distribution to model count data (discrete data, overdispersion)

  $$K^k_{gj} \sim \text{NB}(s^k_j \lambda_{gk}, \phi_g)$$

  in which:

  - s^k_j is library correction factor of sample j in condition k
  - λ_{gk} is the proportion of counts for gene g in condition k
  - ϕ_g is the (over)dispersion (parameter) of gene g (supposed to be identical for all samples)

- estimate its parameters for both conditions

  λ_{g1} λ_{g2} ϕ_g

- conclude by computing p-value ⇒ Test

  $$\text{H0} : \{\lambda_{g1} = \lambda_{g2}\}$$

First method: Exact Negative Binomial test

2 conditions only

First method: Exact Negative Binomial test

2 conditions only

Normalization is performed to get equal size librairies ⇒ s

First method: Exact Negative Binomial test

2 conditions only

Normalization is performed to get equal size librairies ⇒ s

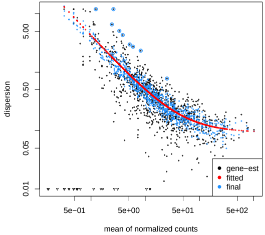
The test is performed similarly as for Fisher test (exact probability is computed according to NB distribution after parameters have been estimated)

Estimating the dispersion parameter ϕ_g

Some methods:

- **DESeq**, **DESeq2**: ϕ_g is a smooth function of $\lambda_g = \lambda_{g1} = \lambda_{g2}$

```r
dge <- estimateDispersion(dge)
```



- **edgeR**: estimate a common dispersion parameter for all genes and use it as a prior in a Bayesian approach to estimate a gene specific dispersion parameter by log-likelihood maximization

```r
dge <- estimateDisp(dge)
```

Perform the test

Some methods:

- **DESeq**, **DESeq2**: exact (**DESeq**) or approximate (Wald and LR in **DESeq2**) tests

```
res <- nbinomWaldTest(dge)
results(res)
```

```
res <- nbinomLR(dge)
results(res)
```

- **edgeR**: exact tests

```
res <- exactTest(dge)
topTags(res)
```

(comparison between methods in [Zhang et al., 2014])

More complex experiments: GLM

Framework:

$$K_{gj} \sim \text{NB}(\mu_{gj}, \phi_g) \qquad \text{with} \qquad \log(\mu_{gj}) = \log(s_j) + \log(\lambda_{gj})$$

in which:

- s_j is the library size correction for sample j;

More complex experiments: GLM

Framework:

$$K_{gj} \sim \text{NB}(\mu_{gj}, \phi_g) \qquad \text{with} \qquad \log(\mu_{gj}) = \log(s_j) + \log(\lambda_{gj})$$

in which:

- s_j is the library size correction for sample j;
- $\log(\lambda_{gj})$ is estimated (for instance) by a Generalized Linear Model (GLM):

$$\log(\lambda_{gj}) = \lambda_0 + \mathbf{x}_j^\top \beta_g$$

in which $\mathbf{x}_j$ is a vector of covariates.

More complex experiments: GLM

Framework:

$$K_{gj} \sim \text{NB}(\mu_{gj}, \phi_g) \qquad \text{with} \qquad \log(\mu_{gj}) = \log(s_j) + \log(\lambda_{gj})$$

in which:

- s_j is the library size correction for sample j;
- $\log(\lambda_{gj})$ is estimated (for instance) by a Generalized Linear Model (GLM):

$$\log(\lambda_{gj}) = \lambda_0 + \mathbf{x}_j^\top \beta_g$$

  in which $\mathbf{x}_j$ is a vector of covariates.

GLM allows to decompose the effects on the mean of

- different factors
- their interactions

More complex experiments: GLM in practice

edgeR

```
dge <- estimateDisp(dge, design) # estimation of dispersion
fit <- glmFit(dge, design) # estimation of parameters
res <- glmLRT(fit, ...) # tests (likelihood ratio)
topTags(res)
```

DESeq, DESeq2

```
dge <- estimateDispersions(dge)
fit <- fitNbinomGLMs(dge, count ~ ...)
fit0 <- fitNbinomGLMs(dge, count ~ 1)
res <- nbinomGLMTest(fit, fit0)
p.adjust(res, method = "BH")
```

Example

In an experiment, gene expression is influenced by:

- diets: A (reference diet) and B (another diet)
- genotypes: G (reference genotype), H (mutant 1), K (mutant 2)

Example

In an experiment, gene expression is influenced by:

- diets: A (reference diet) and B (another diet)
- genotypes: G (reference genotype), H (mutant 1), K (mutant 2)

The model with two additional effects writes:

$$\log(\lambda) = \underbrace{\beta_0}_{\text{basal level for reference}} + \underbrace{\beta_1 \mathbf{1}_{\text{diet B}}}_{\text{additional expression for diet B}} + \underbrace{\beta_2 \mathbf{1}_{\text{genotype H}}}_{\text{additional expression for mutant 1}} + \underbrace{\beta_3 \mathbf{1}_{\text{genotype K}}}_{\text{additional expression for mutant 2}}$$

Tests:

Example

In an an experiment, gene expression is influenced by:

- diets: A (reference diet) and B (another diet)
- genotypes: G (reference genotype), H (mutant 1), K (mutant 2)

The model with two additional effects writes:

$$\log(\lambda) = \underbrace{\beta_0}_{\text{basal level for reference}} + \underbrace{\beta_1 \mathbf{1}_{\text{diet B}}}_{\text{additional expression for diet B}} + \underbrace{\beta_2 \mathbf{1}_{\text{genotype H}}}_{\text{additional expression for mutant 1}} + \underbrace{\beta_3 \mathbf{1}_{\text{genotype K}}}_{\text{additional expression for mutant 2}}$$

Tests:

- Testing if the diet as an effet is equivalent to testing "$\beta_1 = 0$" `coef = 2` in `glmLRT` of **edgeR**

Example

In an experiment, gene expression is influenced by:

- diets: A (reference diet) and B (another diet)
- genotypes: G (reference genotype), H (mutant 1), K (mutant 2)

The model with two additional effects writes:

$$\log(\lambda) = \underbrace{\beta_0}_{\text{basal level for reference}} + \underbrace{\beta_1 \mathbf{1}_{\text{diet B}}}_{\text{additional expression for diet B}} + \underbrace{\beta_2 \mathbf{1}_{\text{genotype H}}}_{\text{additional expression for mutant 1}} + \underbrace{\beta_3 \mathbf{1}_{\text{genotype K}}}_{\text{additional expression for mutant 2}}$$

Tests:

- Testing if genotype K has an expression different to genotype H is equivalent to testing "$\beta_2 = \beta_3$" `contrast = c(0,0,1,-1)` in `glmLRT` of **edgeR**

Example

In an experiment, gene expression is influenced by:

- diets: A (reference diet) and B (another diet)
- genotypes: G (reference genotype), H (mutant 1), K (mutant 2)

The model with two additional effects writes:

$$\log(\lambda) = \underbrace{\beta_0}_{\text{basal level for reference}} + \underbrace{\beta_1 \mathbf{1}_{\text{diet B}}}_{\text{additional expression for diet B}} + \underbrace{\beta_2 \mathbf{1}_{\text{genotype H}}}_{\text{additional expression for mutant 1}} + \underbrace{\beta_3 \mathbf{1}_{\text{genotype K}}}_{\text{additional expression for mutant 2}}$$

Tests:

- Testing if the genotype has an effect is equivalent to testing the full model above against the model $\log(\lambda) = \beta_0 + \beta_1 \mathbf{1}_{\text{diet B}}$ or testing $\beta_2 = \beta_3 = 0$ (`coef = 3:4` `glmLRT` of **edgeR**)

Contrasts

$$\log(\lambda) = \underbrace{\beta_0}_{\text{basal level for reference}} + \underbrace{\beta_1 \mathbf{1}_{\text{diet B}}}_{\text{additional expression for diet B}} + \underbrace{\beta_2 \mathbf{1}_{\text{genotype H}}}_{\text{additional expression for mutant 1}} + \underbrace{\beta_3 \mathbf{1}_{\text{genotype K}}}_{\text{additional expression for mutant 2}}$$

testing if genotype K has an expression different to genotype H:

$$\begin{array}{rcccc} & \beta_0 & \beta_1 & \beta_2 & \beta_3 \\ \text{genotype K} & 1 & 0 & 0 & 1 \\ -\ \text{genotype H} & 1 & 0 & 1 & 0 \end{array}$$

Contrasts

$$\log(\lambda) = \underbrace{\beta_0}_{\text{basal level for reference}} + \underbrace{\beta_1 \mathbf{1}_{\text{diet B}}}_{\text{additional expression for diet B}} + \underbrace{\beta_2 \mathbf{1}_{\text{genotype H}}}_{\text{additional expression for mutant 1}} + \underbrace{\beta_3 \mathbf{1}_{\text{genotype K}}}_{\text{additional expression for mutant 2}}$$

testing if genotype K has an expression different to genotype H:

$$\begin{array}{rcccc}
 & \beta_0 & \beta_1 & \beta_2 & \beta_3 \\
\text{genotype K} & 1 & 0 & 0 & 1 \\
-\ \text{genotype H} & 1 & 0 & 1 & 0 \\
\Rightarrow\ \text{contrast:} & 0 & 0 & -1 & 1
\end{array}$$

Example

In an experiment, gene expression is influenced by:

- leg: L1, L2, L3, L4
- type: pull, push

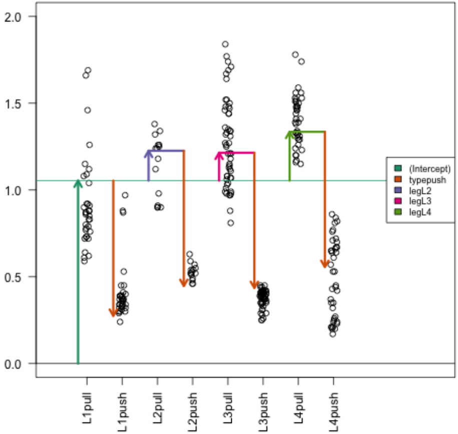
Example

In an experiment, gene expression is influenced by:

- leg: L1, L2, L3, L4
- type: pull, push

```
model.matrix(~ type + leg)
```

$$\beta_0 + \beta_1 \mathbf{1}_{\mathrm{L2}} + \beta_2 \mathbf{1}_{\mathrm{L3}} + \beta_3 \mathbf{1}_{\mathrm{L4}} + \gamma \mathbf{1}_{\mathrm{push}}$$



Example

In an experiment, gene expression is influenced by:

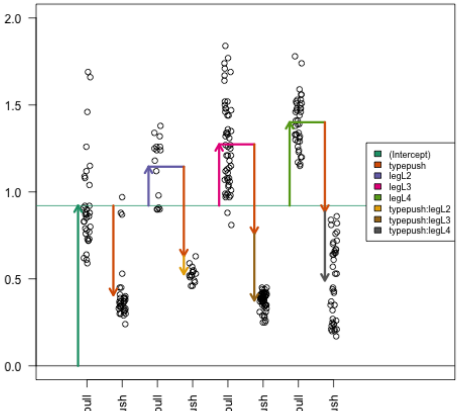
- leg: L1, L2, L3, L4
- type: pull, push

with the interaction term

```
model.matrix(~ type + leg +
type:leg)
```

$$\beta_0 + \beta_1 \mathbf{1}_{\text{push}} + \beta_2 \mathbf{1}_{\text{L2}} + \beta_3 \mathbf{1}_{\text{L3}} + \beta_4 \mathbf{1}_{\text{L4}} + \gamma_1 \mathbf{1}_{\text{push \& L2}} + \gamma_2 \mathbf{1}_{\text{push \& L3}} \gamma_2 \mathbf{1}_{\text{push \& L4}}$$

Testing interaction: `coef = 6:8`



Example

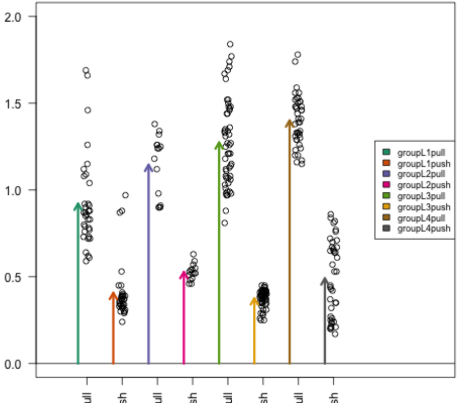
In an experiment, gene expression is influenced by:

- leg: L1, L2, L3, L4
- type: pull, push

equivalently, with group = leg × type

```
model.matrix(~ 0 + group)
```

$$
\beta_1 \mathbf{1}_{\text{L1 \& pull}} + \beta_2 \mathbf{1}_{\text{L1 \& push}} + \\
\beta_3 \mathbf{1}_{\text{L2 \& pull}} + \beta_4 \mathbf{1}_{\text{L2 \& push}} + \\
\beta_5 \mathbf{1}_{\text{L3 \& pull}} + \beta_6 \mathbf{1}_{\text{L3 \& push}} + \\
\beta_7 \mathbf{1}_{\text{L4 \& pull}} + \beta_8 \mathbf{1}_{\text{L4 \& push}}
$$



Alternative approach: linear model for count data

Basic idea:

1. data are transformed so that they are approximately normally distributed

```
tcount <- voom(counts, design)
```

2. a linear (Gaussian) model is fitted (with a Bayesian approach to improve FDR [McCarthy and Smyth, 2009]):

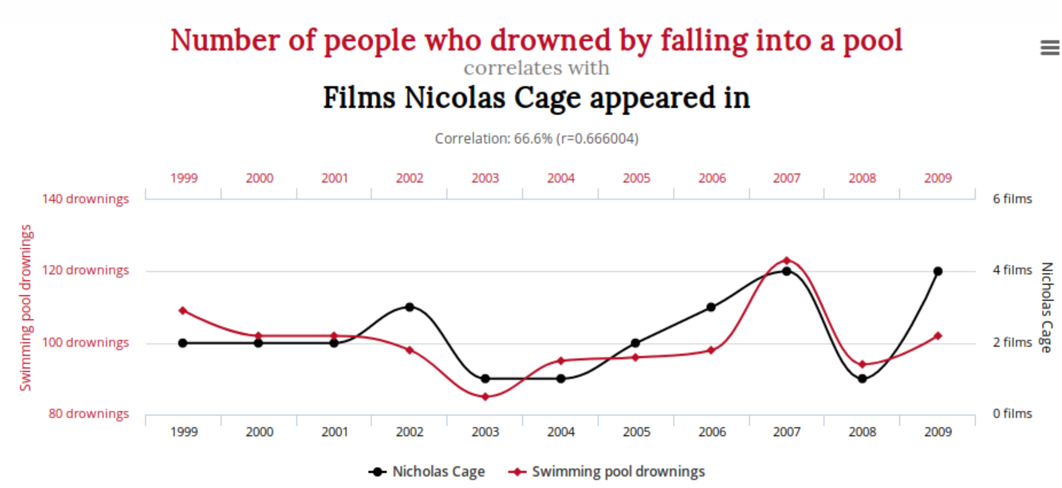
$$\widetilde{K}_{gj} \sim \mathcal{N}(\mu_{gj}, \sigma_g^2)$$

with

$$\mathbb{E}(\widetilde{K}_{gj}) = \beta_0 + \mathbf{x}_j^\top \beta_g$$

```
fit <- lmFit(tcount, design)
fit <- eBayes(fit)
topTables(fit, ...)
```

But never forget: correlation is not causality!

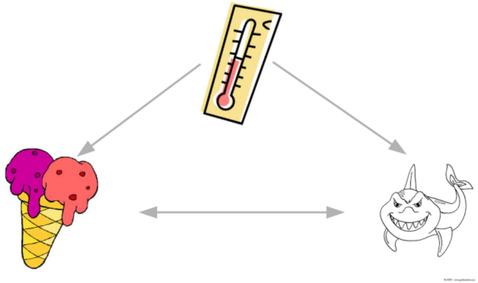


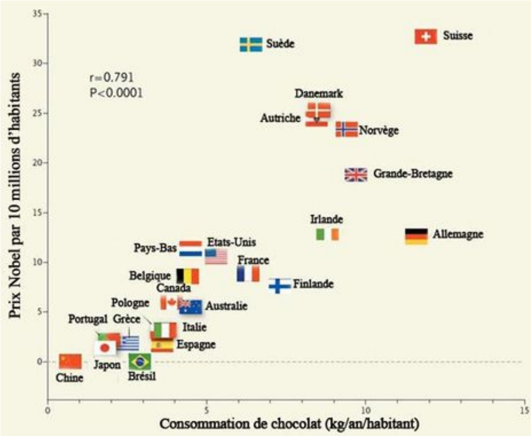
Number of people who drowned by falling into a pool
correlates with
Films Nicolas Cage appeared in

Correlation: 66.6% (r=0.666004)

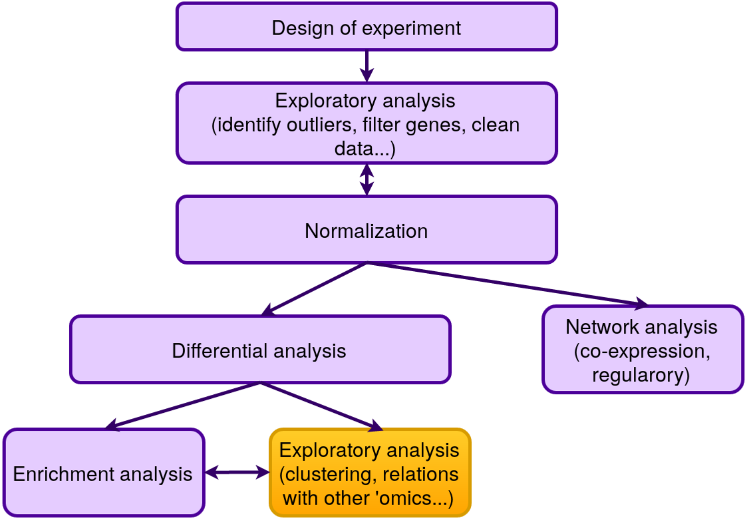
Spurious correlations: http://www.tylervigen.com/spurious-correlations

... and be aware of the Simpson’s effect!





Part V: Interpreting differential analysis results



Design of experiment

Exploratory analysis (identify outliers, filter genes, clean data...)

Normalization

Differential analysis

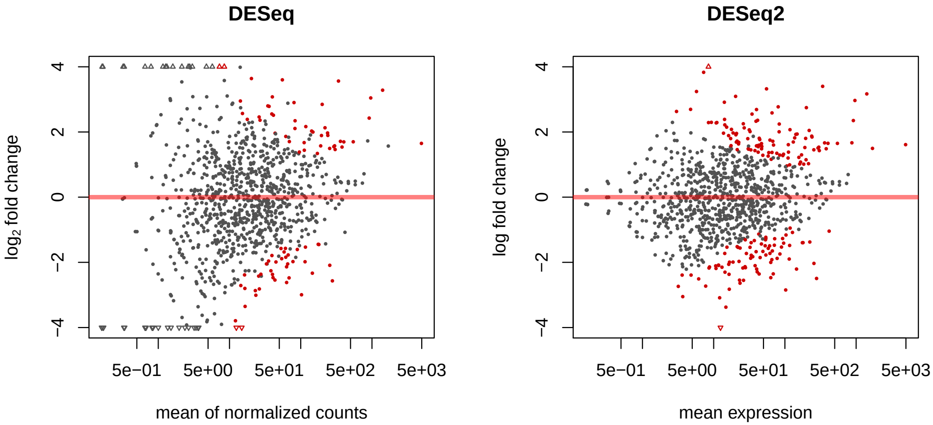
Network analysis (co-expression, regularory)

Enrichment analysis

Exploratory analysis (clustering, relations with other 'omics...)

Overview of the results: MA-plot

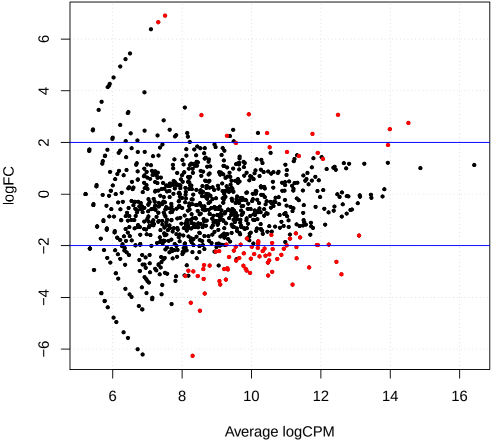
```
plotMA(..., main="DESeq", ylim=c(-4,4))
plotMA(..., main="DESeq2", ylim=c(-4,4))
```



(the last one includes a prior on $\log_2$ fold change which results in more moderated estimates for low count genes)

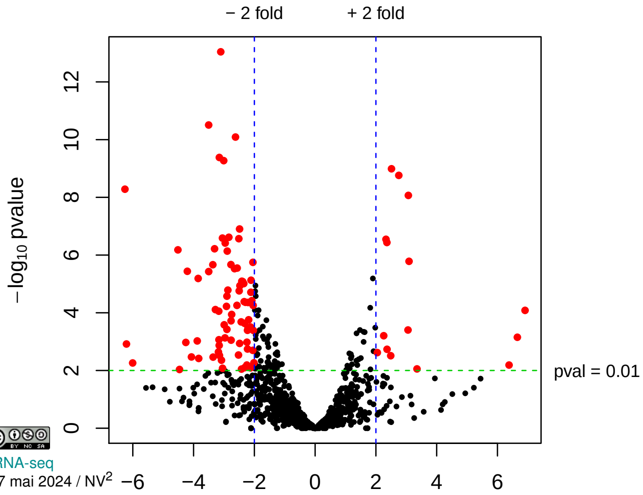
Overview of the results: MA-plot

```
plotSmear(..., de.tags = ...)
```



Fold change and p-value: the Volcano plot

p-value versus fold change (both log scaled) scatterplot. Significant genes are in red:



Gene clustering

Prior clustering: transform data to obtain counts with similar variance

- **DESeq**, **DESeq2**

```
varianceStabilizingTransformation(...)
```

- **DESeq2**

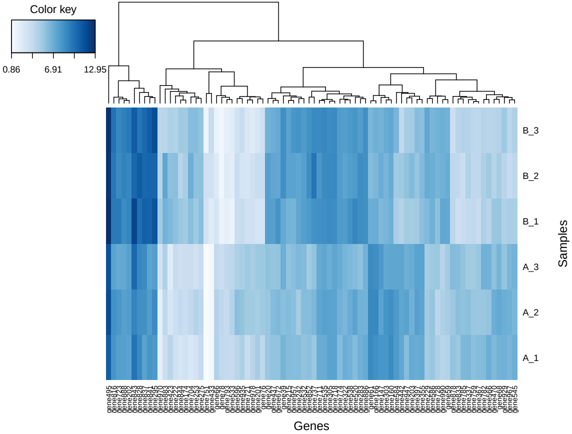
```
rlog(...)
```

- **edgeR**

```
cpm(..., prior.count=2, log=TRUE)
```

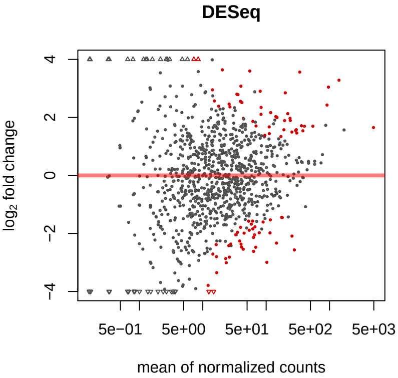
Gene clustering

On transformed data, use *e.g.*, heatmap:



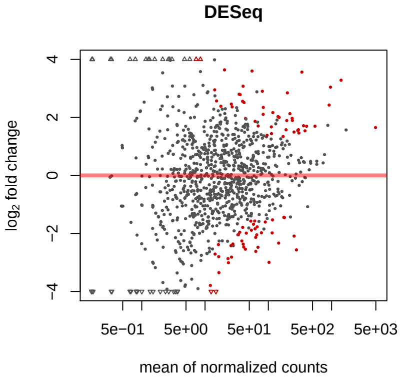
which is useful to visualize which genes are over/under-expressed in one condition.

Standard property of usual DE analyses



Remark: low read counts have a too large variance to be found differentially expressed.

Standard property of usual DE analyses

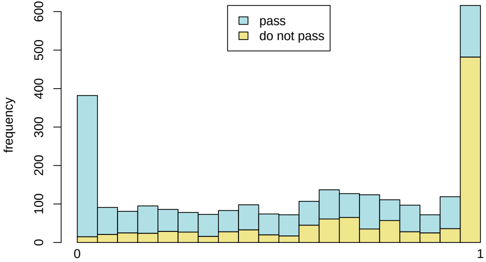


Remark: low read counts have a too large variance to be found differentially expressed.

Consequence: filtering out these genes before the DE analysis because it improves the power of the test because of multiple test correction.

Example

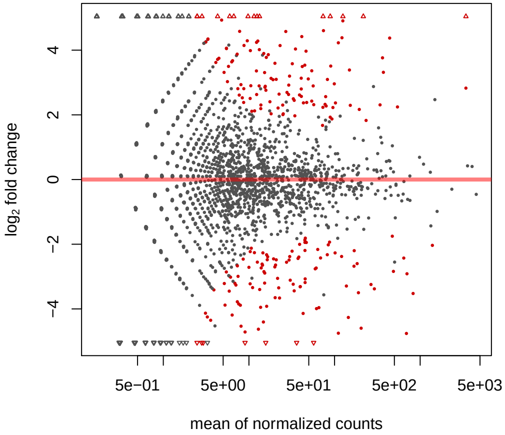
Filtering out the 40% genes that have the lowest overall counts does not affect much low p-values:



but leads to find new DE genes that were previously discarded by multiple test correction.

Filtering in practice

```
cdsFilt <- HTSFilter(..., plot=FALSE)$filteredData
res <- exactTest(cdsFilt)
```



In summary... (with **edgeR**)

preparation of the design of the experiment

sequencing

count data

exploratory analysis (`hist`, `boxplot`...)

creating an R object with count data (`DGEList`)

a **DGEList** object

normalization (`calcNormFactors`)

a **DGEList** object with normalization factors

fitting the model (`estimateDisp`)

a **DGEList** object with dispersion estimates

filtering low count genes (`HTSFilter`)

a **DGEList** object without filtered genes

test (`exactTest` or `glmFit`/`glmLRT`)

a **DGEExact** or **DGELRT** object

exploratory analysis (`topTags`, `plotSmear`...)

References

Bottomly, D., Walter, N., Ezzell Hunter, J., Darakjian, P., Kawane, S., Buck, K., Searles, R., Mooney, M., McWeeney, S., and Hitzemann, R. (2011).
Evaluating gene expression in C57BL/6J and DBA/2J mouse striatum using RNA-Seq and microarrays.
PLoS ONE, 6(3):e17820.

Bullard, J., Purdom, E., Hansen, K., and Dudoit, S. (2010).
Evaluation of statistical methods for normalization and differential expression in mRNA-seq experiments.
BMC Bioinformatics, 11(1):94.

Liu, Y., Zhou, J., and White, K. (2014).
RNA-seq differential expression studies: more sequence or more replication?
Bioinformatics, 30(3):301–304.

McCarthy, D. and Smyth, G. (2009).
Testing significance relative to a fold-change threshold is a TREAT.
Bioinformatics, 25:765–771.

Mortazavi, A., Williams, B., McCue, K., Schaeffer, L., and Wold, B. (2008).
Mapping and quantifying mammalian transcriptomes by RNA-Seq.
Nature Methods, 5:621–628.

Oshlack, A. and Wakefield, M. (2009).
Transcript length bias in RNA-seq data confounds systems biology.
Biology Direct, 4(14).

Robinson, M. and Oshlack, A. (2010).
A scaling normalization method for differential expression analysis of RNA-seq data.
Genome Biology, 11:R25.

Sultan, M., Schulz, M., Richard, H., Magen, A., Klingenhoff, A., Scherf, M., Seifert, M., Borodina, T., Soldatov, A., Parkhomchuk, D., Schmidt, D., O'Keeffe, S., Haas, S., Vingron, M., Lehrach, H., and Yaspo, M. (2008).
A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome.
Science, 321(5891).

Zhang, Z., Jhaveri, D., Marshall, V., Bauer, D., Edson, J., Narayanan, R., Robinson, G., Lundberg, A., Bartlett, P., Wray, N., and Zhao, Q. (2014).
A comparative study of techniques for differential expression analysis on RNA-seq data.
PLoS ONE, 9(8):e103207.