Statistical analysis of RNA-seq data

Andrea Rau

Statistical analysis of RNA-seq data

Andrea Rau

February 6-7, 2020 @ Agrocampus, Rennes

Outline

Measuring gene expression: Microarrays and RNA-seq

High-throughput biological assays can measure the abundance of DNA or RNA sequences for **tens of thousands of genes simultaneously**

Microarrays (1995 -)

- Sequence abundance is a function of the flouresence level recovered after a hybridization process: **continuous data** $\Rightarrow$ we typically model $\log_2$ intensities using normal distributions

RNA sequencing ($\sim$2008 -)

- Number of sequenced reads (counts) mapping to a gene is considered to be linearly related to the abundance of the target feature: **count data**

Overview of RNA-seq experiment and analysis

RNA-seq experiment and analysis

1. **Experimental design**
2. **Experiment**: sequencing (library) preparation, manufacturer protocols, etc.
3. **Pre-processing steps**: alignment/assembly, quality assignment
4. **Expression quantification**: per-gene or per-transcript
5. **Analysis**: normalization, differential analysis (estimation, testing), clustering, prediction, ...

RNA-seq experimental design

- **Randomization** and **Control**
- How many experimental conditions to be compared? (pairwise vs. multiple comparisons)
- How many **biological** or **technical** replicates can we afford to have? ($$)
- How many sequencing runs/flowcells (with 8 lanes per flowcell)?
- Type of sequencer (Solexa/Illumina, 454, SOLiD)
- Multiplexing?

Technical versus biological replicates

Technical replicates:

- Multiple sequencing lanes for the same individual
- Enables an estimation of technical effects
- $\Rightarrow$ Inference about a particular RNA sample

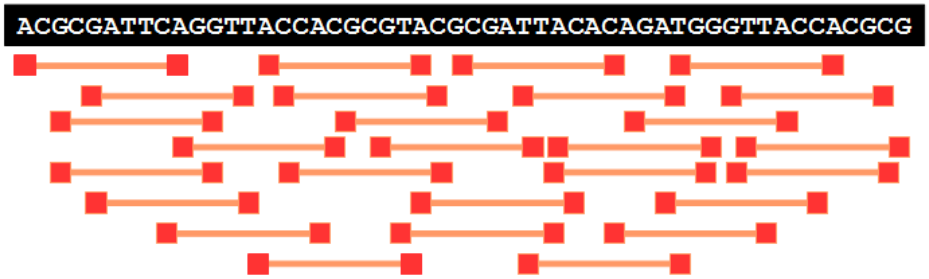
Biological replicates:

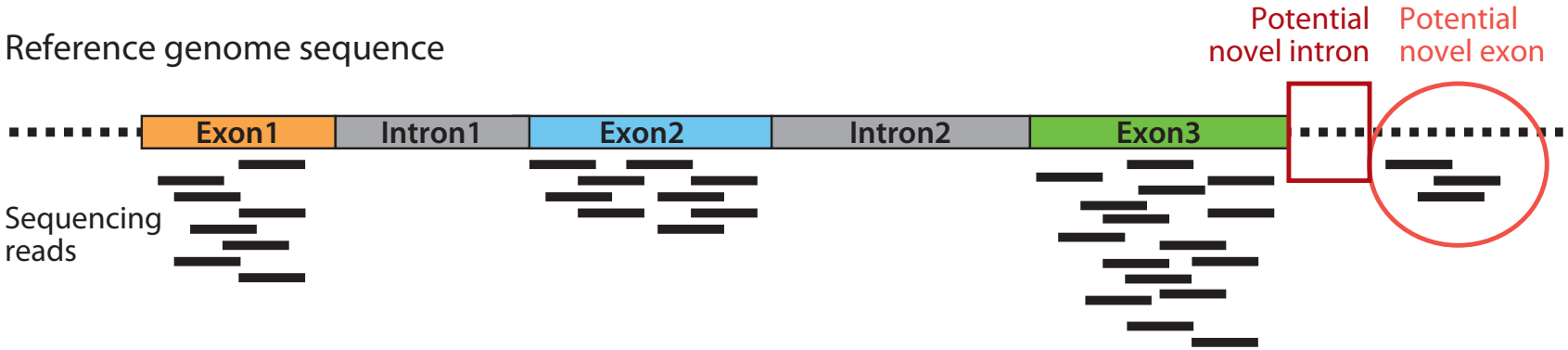
- mRNA extractions from separate organisms or cell lines under the same experimental condition
- More variable: technical + biological variation
- $\Rightarrow$ Inference about a biological population

RNA-seq pre-processing steps

After performing the RNA-seq experiment, bioinformatics takes over:

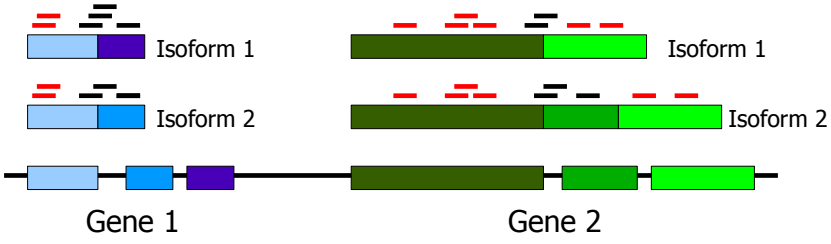
- Calling base pairs (A, C, G, or T?)
- **Alignment**: figuring out which sequencing reads belong to which gene by aligning them to a **reference sequence** using annotation





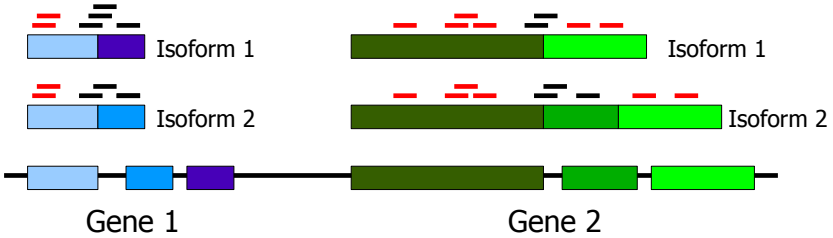
Quantifying gene expression

- Ambiguity in reads (multireads, align to more than one isoform)

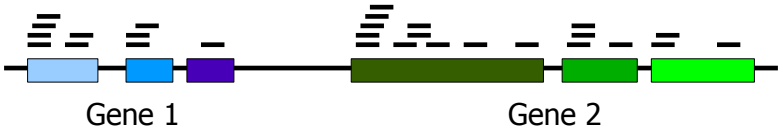


Quantifying gene expression

- Ambiguity in reads (multireads, align to more than one isoform)

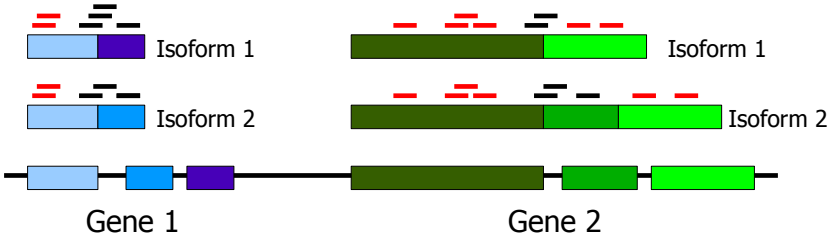


- Longer genes yield more reads (as they have a higher sampling rate)

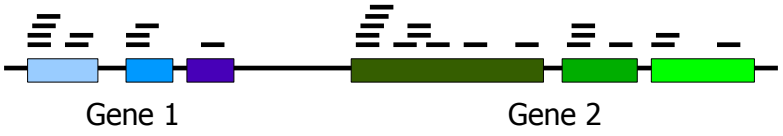


Quantifying gene expression

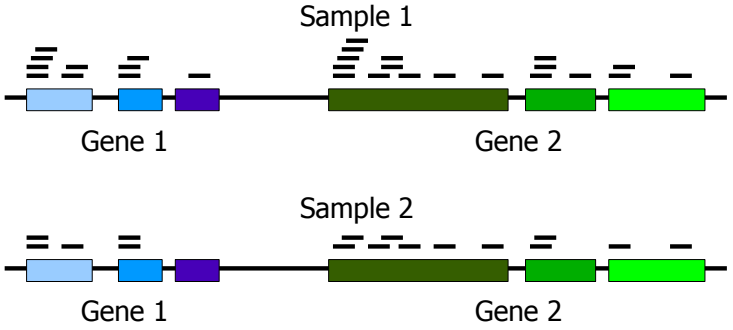
- Ambiguity in reads (multireads, align to more than one isoform)



- Longer genes yield more reads (as they have a higher sampling rate)

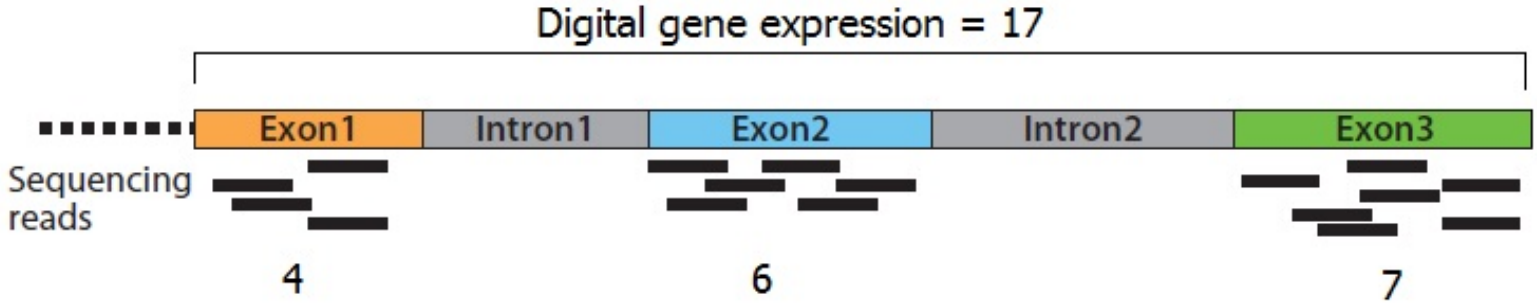


- Gene counts depend on total number of sequences (= "**library size**")



Quantifying gene expression

- The **quantification of gene expression** is still an open and active area of research: isoform-specific expression, strand-specific expression, ambiguity in mapping, ...
- Generally, focus on analysis of **gene-level count-based measures of expression**



Data For Statistical Analysis (Raw Counts)

	-Group A-			-Group B-		
Gene	**1**	**2**	**3**	**1**	**2**	**3**
13CDNA3	4	0	6	1	0	5
A2BP1	19	18	20	7	1	8
A2M	2724	2209	13	49	193	548
A4GALT	0	0	48	0	0	0
AAAS	57	29	224	49	202	92
AACS	1904	129	4	507	3	965
AADACL1	3	13	239	683	158	40
[...]						

Data For Statistical Analysis (Raw Counts)

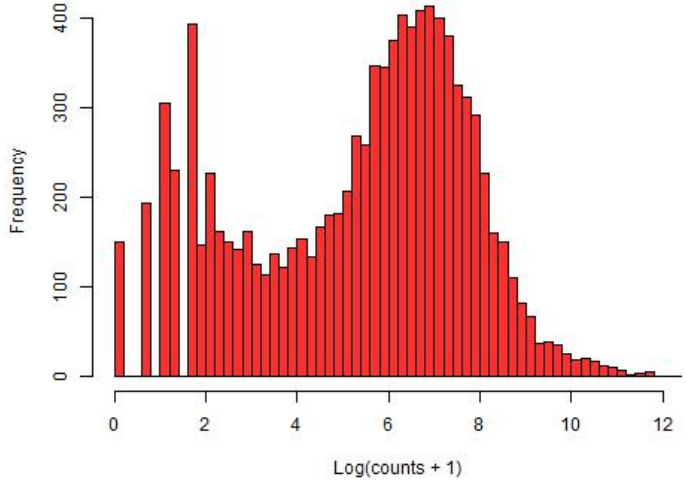
	-Group A-			-Group B-		
Gene	**1**	**2**	**3**	**1**	**2**	**3**
13CDNA3	4	0	6	1	0	5
A2BP1	19	18	20	7	1	8
A2M	2724	2209	13	49	193	548
A4GALT	0	0	48	0	0	0
AAAS	57	29	224	49	202	92
AACS	1904	129	4	507	3	965
AADACL1	3	13	239	683	158	40
[...]						

Data For Statistical Analysis (Raw Counts)

	-Group A-			-Group B-		
Gene	**1**	**2**	**3**	**1**	**2**	**3**
13CDNA3	4	0	6	1	0	5
A2BP1	19	18	20	7	1	8
A2M	2724	2209	13	49	193	548
A4GALT	0	0	48	0	0	0
AAAS	57	29	224	49	202	92
AACS	1904	129	4	507	3	965
AADACL1	3	13	239	683	158	40
[...]						

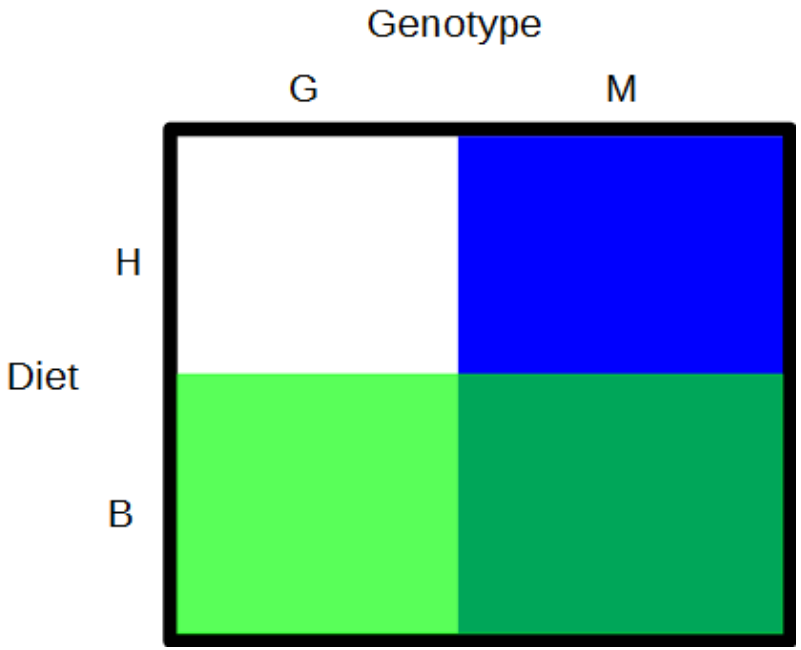
Some statistical challenges for RNA-seq data

- High dimensionality (large number of genes, few replicates)
- Discrete, positive, and skewed data
- Large dynamic range among genes (10^6 orders of magnitude), presence of 0 counts
  - Typically remove absent genes (those with 0 counts for all samples)
- Sequencing depth (= "library size") varies among samples



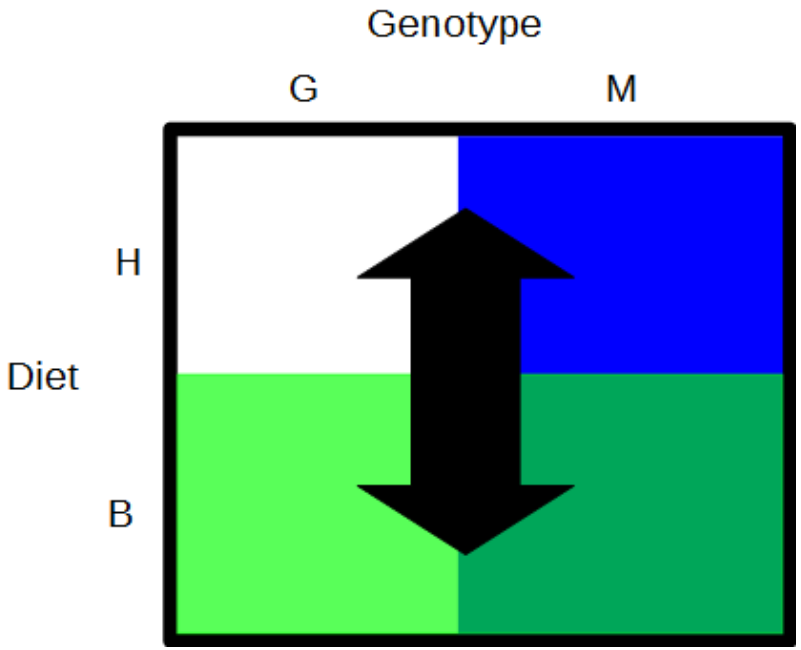
Example data: 2 ×2 factorial experiment in chickens

- Genotype (G, M) × diet (H, B) in two tissues (liver & adipose tissue to be analyzed independently)
- 4 replicates per combination of factors
- **Goal** is to identify genes that are differentially expressed:
  - Between genotypes, between diets
  - Interaction effect (genotype × diet)



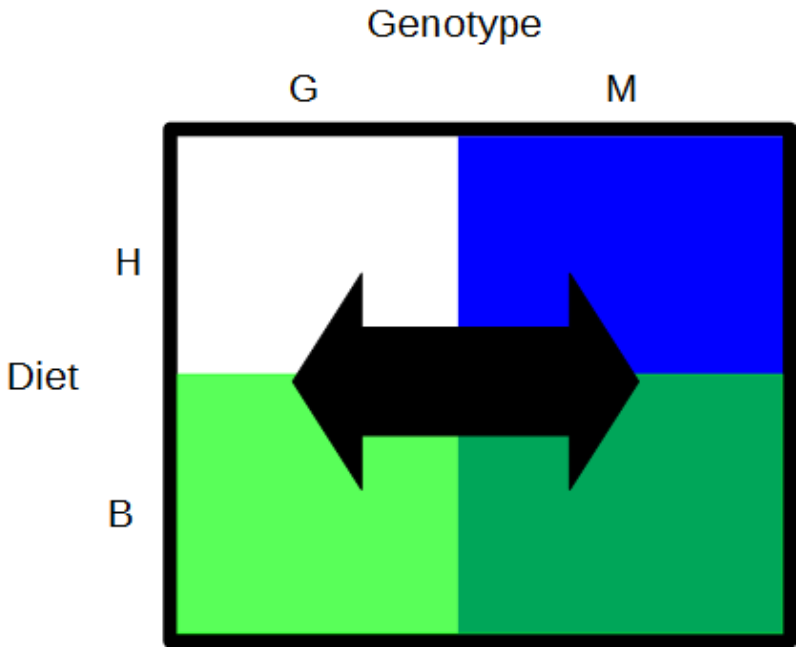
Example data: 2 ×2 factorial experiment in chickens

- Genotype (G, M) × diet (H, B) in two tissues (liver & adipose tissue to be analyzed independently)
- 4 replicates per combination of factors
- **Goal** is to identify genes that are differentially expressed:
  - Between genotypes, between diets
  - Interaction effect (genotype × diet)



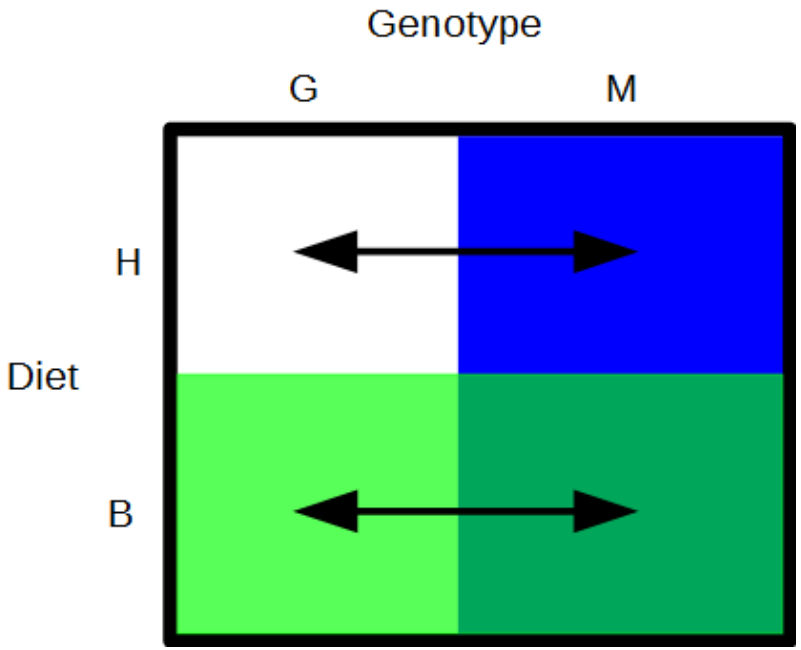
Example data: 2 ×2 factorial experiment in chickens

- Genotype (G, M) × diet (H, B) in two tissues (liver & adipose tissue to be analyzed independently)
- 4 replicates per combination of factors
- **Goal** is to identify genes that are differentially expressed:
  - Between genotypes, between diets
  - Interaction effect (genotype × diet)



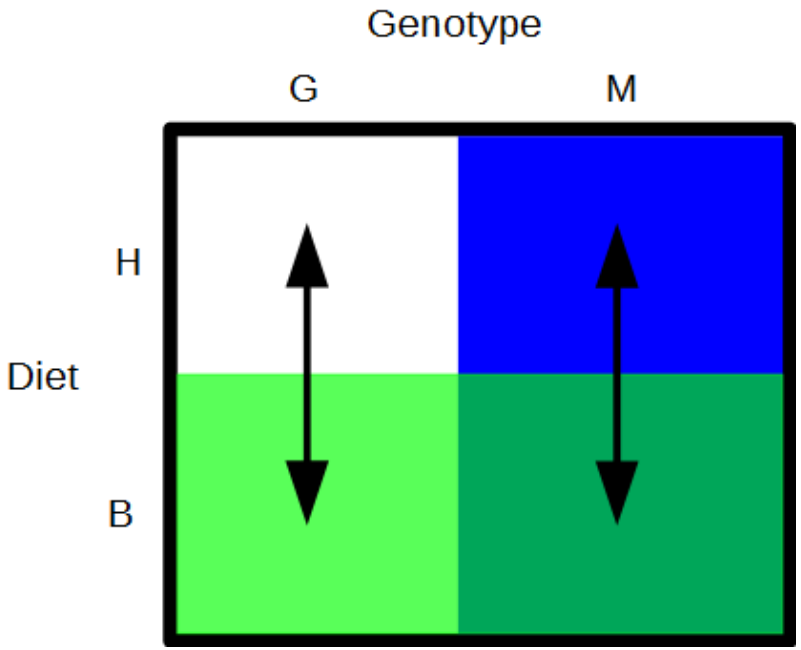
Example data: 2 ×2 factorial experiment in chickens

- Genotype (G, M) × diet (H, B) in two tissues (liver & adipose tissue to be analyzed independently)
- 4 replicates per combination of factors
- **Goal** is to identify genes that are differentially expressed:
  - Between genotypes, between diets
  - Interaction effect (genotype × diet)



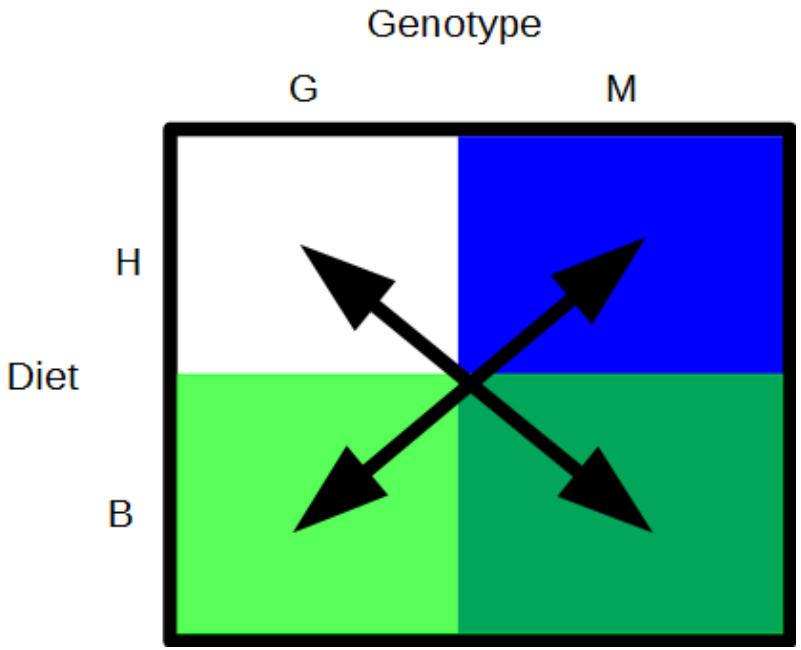
Example data: 2 ×2 factorial experiment in chickens

- Genotype (G, M) × diet (H, B) in two tissues (liver & adipose tissue to be analyzed independently)
- 4 replicates per combination of factors
- **Goal** is to identify genes that are differentially expressed:
  - Between genotypes, between diets
  - Interaction effect (genotype × diet)



Example data: 2 ×2 factorial experiment in chickens

- Genotype (G, M) × diet (H, B) in two tissues (liver & adipose tissue to be analyzed independently)
- 4 replicates per combination of factors
- **Goal** is to identify genes that are differentially expressed:
  - Between genotypes, between diets
  - Interaction effect (genotype × diet)



R and Bioconductor

What is Bioconductor? (http://www.bioconductor.org)

- Open-source and open-development software package implemented in R for analysis of genomic data
- Installing Bioconductor in R:

```
> source("http://www.bioconductor.org/biocLite.R")
> biocLite("DESeq2")
> library(DESeq2)        ## RNA-seq differential expression
> vignette("DESeq2")
```

Take an initial look at the chicken data

We will begin by loading necessary packages and data into R...

- Create a matrix containing counts with gene IDs as row names
- Create a design matrix identifiying the combination of factors for each sample

- Look at the first few rows of `dataTP` and `designTP`. What are the dimensions of `dataTP`?
- How many genes have zero counts for all samples? Remove them now. How many genes remain?
- What are the minimum and maximum counts?

Exploratory data analysis

- Common to find biases, systematic errors (e.g., sample labels accidentally switched), and unexpected variability (e.g., samples run in two different batches) in genomic data
- Graphing data may help detect such problems:
  - Histograms of log(counts+1)
  - Boxplots log(counts+1)
  - Barplots of library sizes
  - Scatterplots
- After transforming data (regularized log transformation):
  - Principal components analysis or multidimensional scaling
  - Hierarchical clustering of samples
  - NOTE: transformations only used for exploratory data analysis!

A note on principal components analysis for RNA-seq data

- Many exploratory analyses work best for (approximately) homoskedastic data (= variance does not depend on the mean)
- Check whether this appears to be the case for the chicken data

- PCA directly on read counts will typically depend only on the few most strongly expressed genes
- ⇒ typically transform (log, regularized log from `DESeq2`) normalized count values before performing PCA

Outline

Differential gene expression

What is differential gene expression?

A gene is declared **differentially expressed** (DE) if an observed difference or change in expression between two experimental conditions is statistically significant, i.e., greater than expected just due to natural random variation.

$\Rightarrow$ Statistical tools are required to make such a decision

Often used to compare transcript levels in different types of cells:

- Tissue: liver vs. brain
- Treatment: drugs A, B, and C
- State: tumor vs. nontumor
- Across time

Differential gene expression

Differential expression gene-by-gene

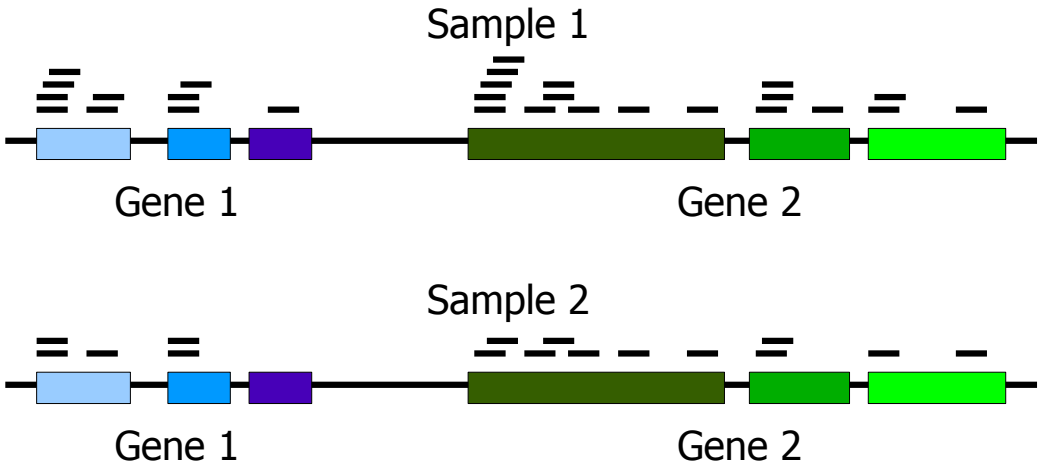
For each gene i, is there a significant difference in expression between groups A and B?

- Normalization for differences in library size
- Statistical model (definition and parameter estimation)
- Per-gene testing for differential expression:

$$H_{0i} : \mu_{i,\mathrm{A}} = \mu_{i,\mathrm{B}}$$

Differences in library size

Recall: Gene counts depend on total number of sequences (= "**library size**")



- Comparison of a fixed gene between two samples must account for differences in library size

Note: estimated normalization factors will be included directly within the model (more on this later)...

Some initial optimism for RNA-seq normalization...

"One particularly powerful advantage of RNA-seq is that it can capture transcriptome dynamics across different tissues or conditions without sophisticated normalization of data sets."
– Wang et al. (2009)

INNOVATION

RNA-Seq: a revolutionary tool for transcriptomics

Zhong Wang, Mark Gerstein and Michael Snyder

... but many technical biases inherent to RNA-seq data!

Technical variability within and between samples:

- Total number of mapped reads per sample ("library size" or sequencing depth)
- RNA composition bias
- GC-content
- Gene length
- ...

Normalization

Process to identify and correct systematic **technical biases** removing the least possible biological signal

- Normalization is needed and has a large impact on the results of a DE analysis (Bullard et al. 2010)
- Technology and platform-dependant

Global normalization methods

Divide counts by a scaling factor for each sample

Adjustment of count distributions

- Total number of reads (Marioni et al. 2008)
- Upper Quartile (Bullard et al. 2010)
- Median Med

- Quantile (as with microarray data)

Global normalization methods (continued)

Adjustment for library size and length

Assumption: read counts are proportional to expression level, sequencing depth, and gene length

Reads Per Kilobase Per Million mapped reads (Mortazavi et al. 2008):

$$\text{RPKM} = \frac{\text{number of mapped reads in the region}}{\frac{\text{total reads}}{1\times10^6} \times \frac{\text{region length}}{1000}}$$

- Originally introduced for comparisons between genes within a sample (correct bias due to gene length)
- Note: Oshlack and Wakefield (2009) showed that correcting for gene length in a differential analysis introduces a bias in per-gene variances

Effective library size normalization methods

Motivation

Different biological conditions express different RNA repertoires, leading to different total amounts of RNA
$\Rightarrow$ Strongly DE genes may distort ratio of total reads!

Assumption

Most genes are not differentially expressed

Aim

Minimize the effect of (very) high-count genes

Effective library size normalization methods

Trimmed Mean of M-values (TMM): Robinson and Oshlack (2010)

- Idea: Estimate global expression change between two conditions from non-extreme genes
- Filter genes with null counts, genes with very large expression (most extreme 5%), and genes with large log ratios between conditions (most extreme 30%)
- For each sample, among remaining genes the TMM is the weighted mean of log ratios between sample and a reference sample $\Rightarrow$ Under hypothesis of little DE, TMM should be close to 1
- TMM correction applied to library sizes and NOT directly to counts in the model

Effective library size normalization methods (continued)

DESeq: Anders and Huber (2010)

- Idea: Non-DE genes should have similar read counts across samples
- For each gene, the median of the ratio of its read counts to a pseudo-reference sample (its geometric mean across all samples) $\Rightarrow$ Under hypothesis of little DE, DESeq scaling factors should be close to 1
- DESeq correction applied directly to counts in the model

Conclusions of Statomique evaluation (Dillies et al. 2013)

Normalization for RNA-seq data is necessary and not trivial to account for systematic variation between samples and differences in library composition

- Hypothesis : the majority of genes is invariant between two samples
- TMM and DESeq are most robust and lead to best performance in per-gene DE analyses ⇒ these are scaling factors that are inserted directly into the model

A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis

Marie-Agnès Dillies, Andrea Rau*, Julie Aubert*, Christelle Hennequet-Antier*, Marine Jeanmougin*, Nicolas Servant*, Céline Keime*, Guillemette Marot, David Castel, Jordi Estelle, Gregory Guernec, Bernd Jagla, Luc Jouneau, Denis Laloë, Caroline Le Gall, Brigitte Schaëffer, Stéphane Le Crom*, Mickaël Guedj*, Florence Jaffrézic* and on behalf of The French StatOmique Consortium*

Outline

Some notation

Notation

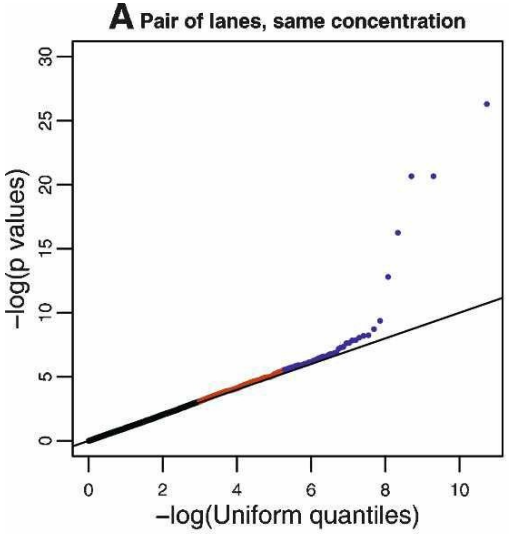
Let Y_{ij} be the count (expression measure) for gene i in sample j, with corresponding observed value y_{ij}.

- $i = 1, \ldots, n$ genes
- $\mathcal{C}(j)$ corresponds to the experimental condition of sample j
- Typically, $\mathcal{C}(j) \in \{1, 2\}$ (two group comparison) but here we have $\mathcal{C}(j) \in \{\mathrm{GB}, \mathrm{GH}, \mathrm{MB}, \mathrm{MH}\}$

Poisson model in RNA-seq data

Is the Poisson model really appropriate for RNA-seq counts?

- Nagalakshmi et al. (2008) and Marioni et al. (2008) found that genes from different **technical** replicates have a variance equal to the mean (= Poisson)
- Generally, technical replicates are summed (as the sum of two Poisson random variables is also Poisson)



Marioni et al. (2008), Fig 1 (hypergeometric test statistic to compare tech reps)

Overdispersion in RNA-seq data

Counts from **biological** replicates tend to have variance exceeding the mean (= **overdispersion**)...

What causes this overdispersion?

- **Shot noise**: unavoidable noise inherent in counting process (dominant for weakly expressed genes)

+

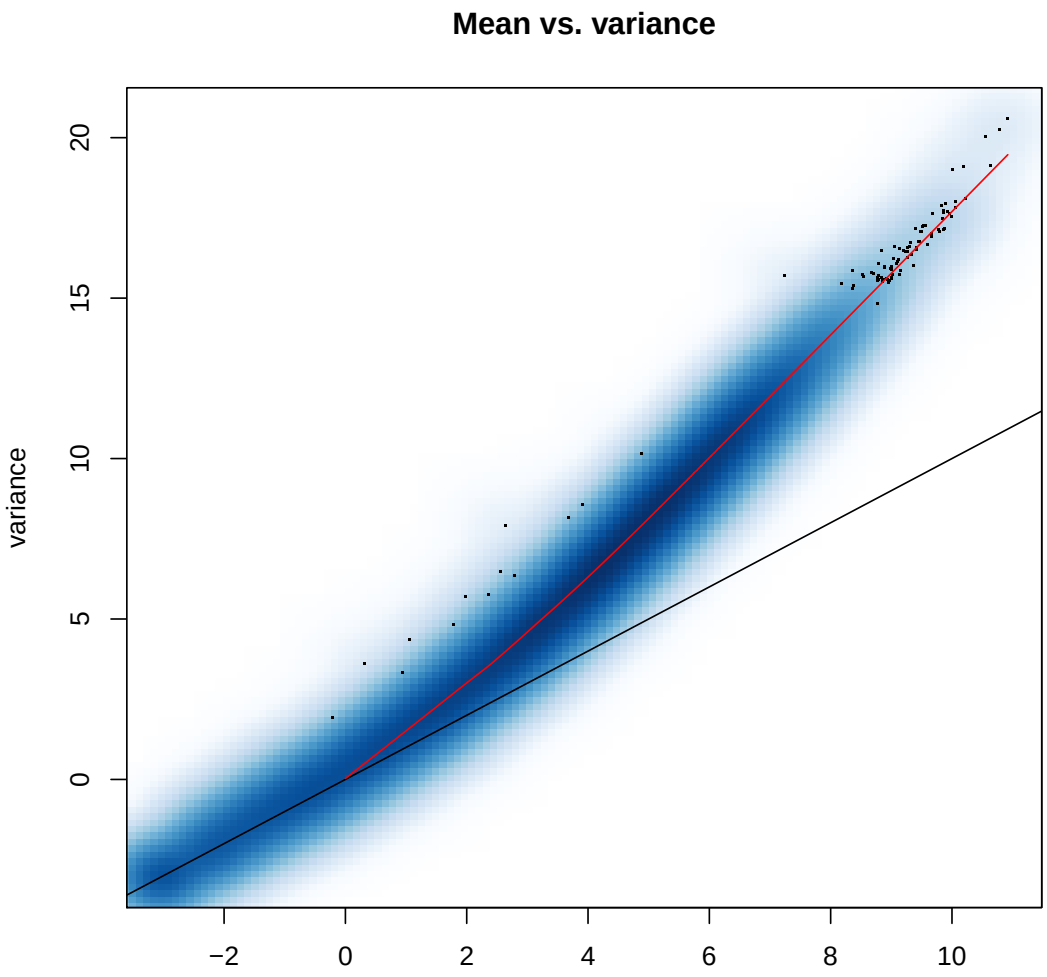
- **Technical noise**: from sample preparation and sequencing, hopefully negligable

+

- **Biological noise**: unaccounted for differences between samples (dominant for strongly expressed genes)

Overdispersion in RNA-seq data

Check for the presence of overdispersion in the data. Would a Poisson model be appropriate?



Negative binomial models

Negative binomial model

$$\Pr(Y_{ij} = y_{ij}) = \text{Negative binomial}(\mu_{ij}, \phi_i)$$

- $\mathrm{E}(Y_{ij}) = \mu_{ij}$
- $\mathrm{Var}(Y_{ij}) = \mu_{ij} + \phi_i \mu_{ij}^2$

We could consider ϕ (common dispersion parameter: easier to estimate but unrealistic) or ϕ_i (per-gene dispersion parameter)...

Negative binomial models

- Many genes, relatively few biological samples – so difficult to estimate ϕ_i on a gene-by-gene basis
- How to obtain estimates of overdispersion parameter for each gene?

Several proposed solutions

- edgeR: borrow information across genes for stable estimates of ϕ_i
- DESeq2: estimate the mean / variance relationship using parametric regression

DESeq2 Bioconductor package: Love et al. (2014)

DESeq2 basics:

- Most recent version of the original DESeq package
- Negative binomial error model
- Data-driven relationships of variance and mean estimated using parametric regression for robust and moderated fit across genes
  $\Rightarrow$ genes with similar expression strength have similar dispersion
- Wald's test (or likelihood ratio test) for null hypothesis

DESeq2 Bioconductor package: Love et al. (2014)

Assumptions:

1. $Y_{ij} \sim \text{NB}(\mu_{ij}, \sigma^2_{ij})$, where μ_{ij} is the mean, and σ^2_{ij} is the variance, and

$$\sigma^2_{ij} = \mu_{ij} + \phi_i \mu^2_{ij}$$

with dispersion ϕ_i

DESeq2 Bioconductor package: Love et al. (2014)

Assumptions:

1. $Y_{ij} \sim \mathrm{NB}(\mu_{ij}, \sigma_{ij}^2)$, where μ_{ij} is the mean, and σ_{ij}^2 is the variance, and

$$\sigma_{ij}^2 = \mu_{ij} + \phi_i \mu_{ij}^2$$

   with dispersion ϕ_i

2. The mean μ_{ij} is the product of a condition-dependent per-gene value q_{ij} and a **size factor** (library size) m_j:

$$\mu_{ij} = q_{ij} m_j$$

$$\log q_{ij} = X_j \boldsymbol{\alpha}_i = \sum_r x_{jr} \alpha_{ir}$$

   where X_j is the design matrix and $\boldsymbol{\alpha}_i$ the vector of coefficients

DESeq2 Bioconductor package: Love et al. (2014)

Assumptions:

1. $Y_{ij} \sim \mathrm{NB}(\mu_{ij}, \sigma_{ij}^2)$, where μ_{ij} is the mean, and σ_{ij}^2 is the variance, and

$$\sigma_{ij}^2 = \mu_{ij} + \phi_i \mu_{ij}^2$$

   with dispersion ϕ_i
2. The mean μ_{ij} is the product of a condition-dependent per-gene value q_{ij} and a **size factor** (library size) m_j:

$$\mu_{ij} = q_{ij} m_j$$

$$\log q_{ij} = X_j \boldsymbol{\alpha}_i = \sum_r x_{jr} \alpha_{ir}$$

   where X_j is the design matrix and $\boldsymbol{\alpha}_i$ the vector of coefficients
3. Per-gene variance is a smooth function of the mean:

$$\sigma_{ij} = f(q_{ij})$$

DESeq2 Bioconductor package: Love et al. (2014)

Three sets of parameters need to be estimated:

1. Size factors m_{jk} (normalization factors, cf earlier slides)
2. The smooth function $f : \mathbb{R}^+ \to \mathbb{R}^+$ to model dependence of σ_{ij} on the expected mean q_{ij}
3. For each sample, n expression strength parameters q_{ij}

DESeq2 dispersion parameter estimation

Empirical Bayes estimation of gene-wise dispersions:

1. Estimate gene-wise dispersions using MLE: $\hat{\phi}_i^{\text{MLE}}$
2. Fit a Gamma GLM to per-gene means and dispersions:

$$\hat{\phi}_i = \alpha_0 + \alpha_1 / q_{ij}$$

   and obtain fitted dispersion estimates: $\hat{\phi}_i$
3. Shrink gene-wise dispersion estimates towards estimates $\tilde{\phi}_i$

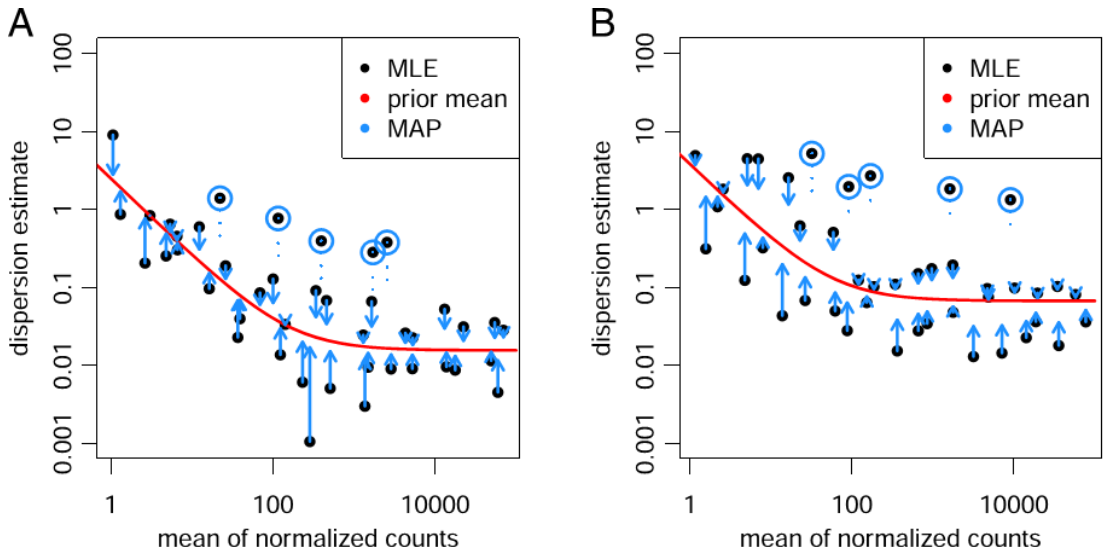


Fig. 1 from Love et al. (2014)

DESeq2 parameter estimation (continued)

For each sample j, n expression strength parameters q_{ij}:

- Average of counts from the replicates for each condition, transformed to the normalized scale: $\hat{q}_{ij}$
- Fit negative binomial GLM:

$$\log \hat{q}_{ij} \sim \mathrm{NB}\left(\sum_r x_{jr}\alpha_{ir}, \tilde{\phi}_i\right)$$

- Test the null hypothesis $H_{0ir} : \alpha_{ir} = 0$ with Wald's test:

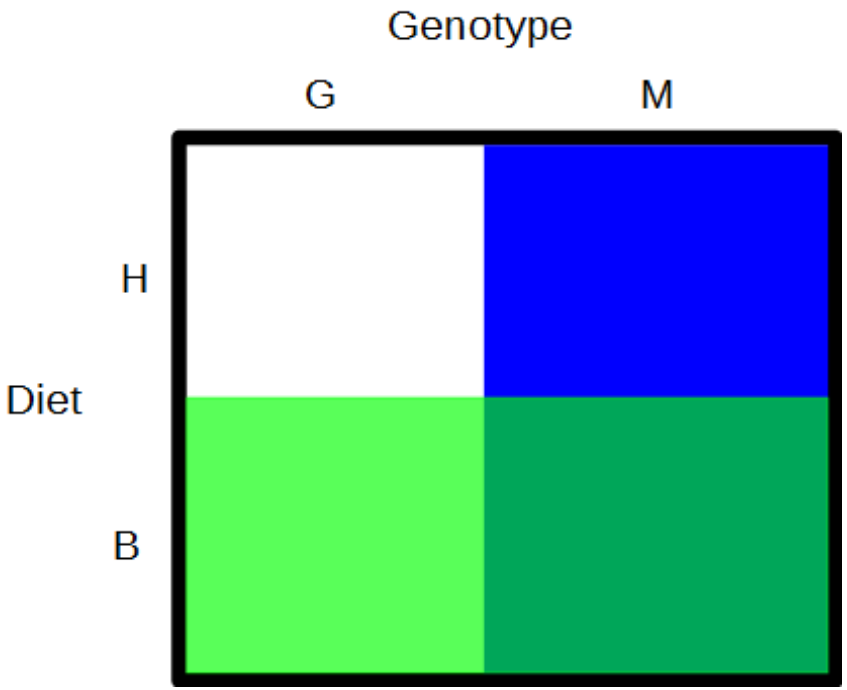
$$\mathrm{Wald}_{ir} = \hat{\alpha}_{ir}/\mathrm{SE}(\hat{\alpha}_{ir}) \sim \mathcal{N}(0, 1)$$

DESeq2: Some practical considerations

- Various default calculations not described here:
  - Shrinkage of log-fold changes for low-count genes
  - Automatic filtering of weakly expressed genes
  - Outlier detection via Cook's distance
- Data must be input as raw counts; normalization offsets are directly included in the model
- Each column should be an independent biological replicate
- Check out the DESeq2 Users' Guide for examples

- Latest version: DESeq2 version 1.26.0 (Bioconductor 3.10)

Analyzing a 2×2 factorial experiment with DESeq2



Recall: we have 2 factors (genotype, diet) that each have 2 levels, and all 4 combinations are observed = factorial design

What are the comparisons of interest?

Analyzing a 2×2 factorial experiment as a single factor

Easiest way to set up analysis is to analyze the data as a single factor with 4 levels { GH, GB, MH, MB }:

```
designTP$group <- factor(paste0(designTP$genotype,
                                designTP$regime))
dds <- DESeqDataSetFromMatrix(countData = dataTP,
                              colData = designTP,
                              design = ~0 + group)
```

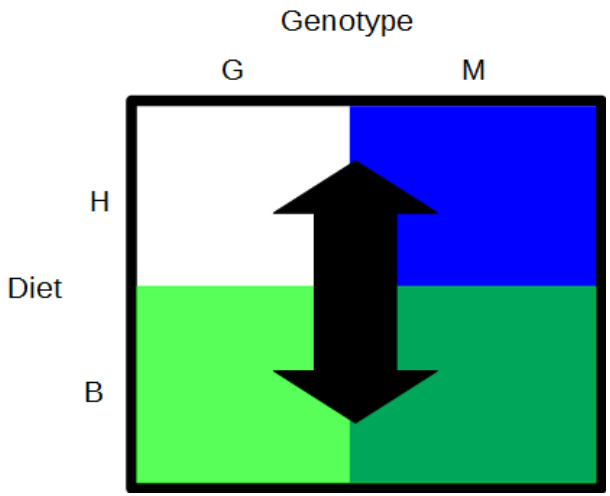
The model thus has four coefficients:

1. GB
2. GH
3. MB
4. MH

Analyzing a 2×2 factorial experiment as a single factor

Now we can extract the comparisons of interest by testing contrasts $H_0 : \mathbf{C}'\boldsymbol{\alpha} = 0$.

Average diet effect:



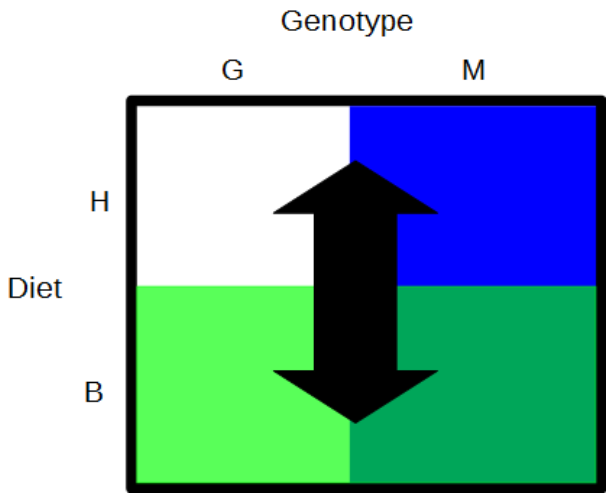
$$\begin{pmatrix} 1 & 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} GH \\ MH \\ GB \\ MB \end{pmatrix} =$$

$$(GH + MH) - (GB + MB) = 0$$

Analyzing a 2×2 factorial experiment as a single factor

Now we can extract the comparisons of interest by testing contrasts $H_0 : \mathbf{C}'\boldsymbol{\alpha} = 0$.

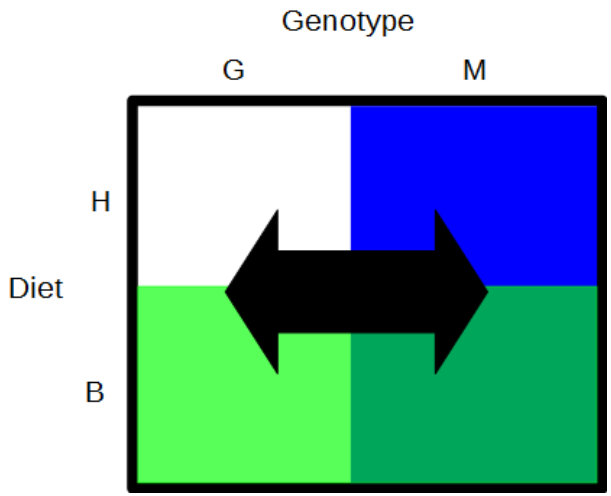
Average diet effect:



$$\begin{pmatrix} 1 & 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} GH \\ MH \\ GB \\ MB \end{pmatrix} =$$

$$(GH + MH) - (GB + MB) = 0$$

Average genotype effect:

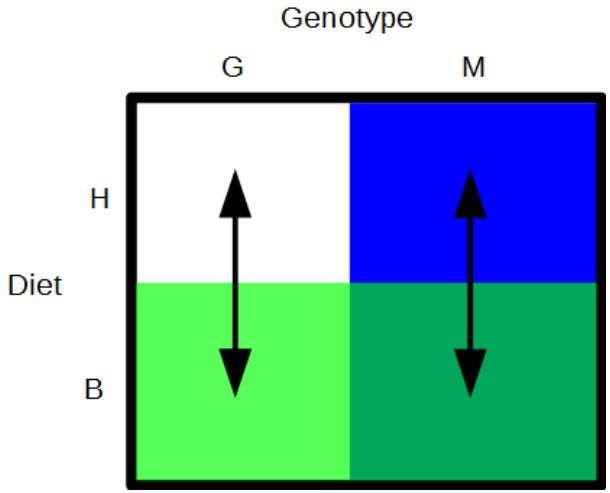


$$\begin{pmatrix} 1 & -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} GH \\ MH \\ GB \\ MB \end{pmatrix} =$$

$$(GH + GB) - (MH + MB) = 0$$

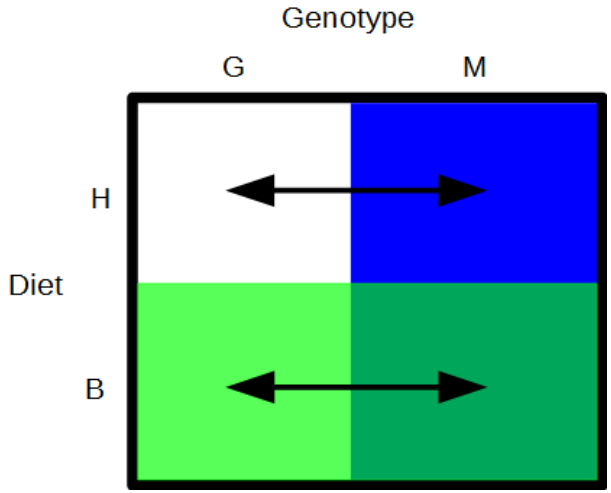
Analyzing a 2×2 factorial experiment as a single factor

Diet effect in each genotype:



$$GH - GB = 0$$
$$MH - MB = 0$$

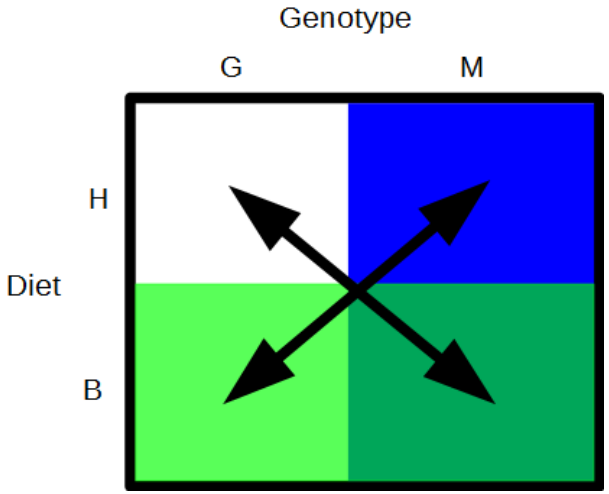
Genotype effect in each diet:



$$GH - MH = 0$$
$$GB - MB = 0$$

Analyzing a 2×2 factorial experiment as a single factor

Genotype × diet interaction effect:



- Is the effect of genotype different for each diet? Is the effect of diet different for each genotype?

$$(GH - GB) - (MH - MB) =$$
$$(GH - MH) - (GB - MB) =$$
$$(GH + MB) - (GB + MH) = 0$$

2×2 factorial experiment as classic interaction model

Also possible to analyze data as a classic model with two fixed effects and an interaction effect:

```
dds2 <- DESeqDataSetFromMatrix(countData = dataTP,
        colData = designTP,
        design = ~ genotype + regime + genotype:regime)
```

The model thus has four coefficients that are NOT INTERPRETED AS BEFORE:

1. Intercept
2. genotype_M_vs_G
3. regime_H_vs_B
4. genotypeM.regimeH

2×2 factorial experiment as classic interaction model[1]

Important to check the reference level of each factor (by default, first in alphabetical order):

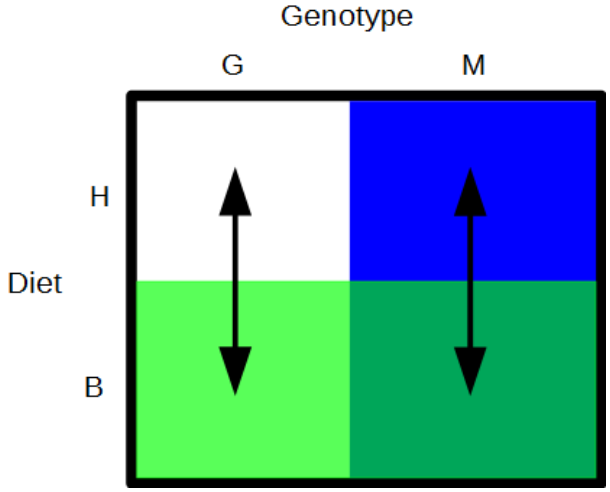
```
levels(designTP$genotype)
levels(designTP$regime)
```

Coefficient	Interpretation
Intercept	Baseline level of ref genotype (G) for ref diet (B)
genotype_M_vs_G	Difference between M and G for ref diet (B)
regime_H_vs_B	Difference between B and H for ref genotype (G)
genotypeM.regimeH	Interaction between genotype and diet

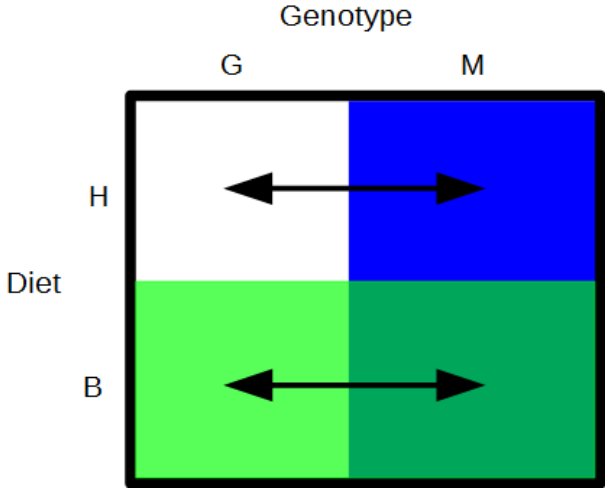
[1] NOTE: depending on the version of DESeq2, the results for the single factor vs. classic interaction model may not be exactly the same for the main effects due to slight differences in shrinkage procedures

2×2 factorial experiment as classic interaction model

Diet effect in each genotype:



Genotype effect in each diet:



Reference level (G):

$$\text{regime\_H\_vs\_B} = 0$$

Non-reference level (M):

$$\text{regime\_H\_vs\_B} + \text{genotypeM.regimeH} = 0$$

Reference level (B):

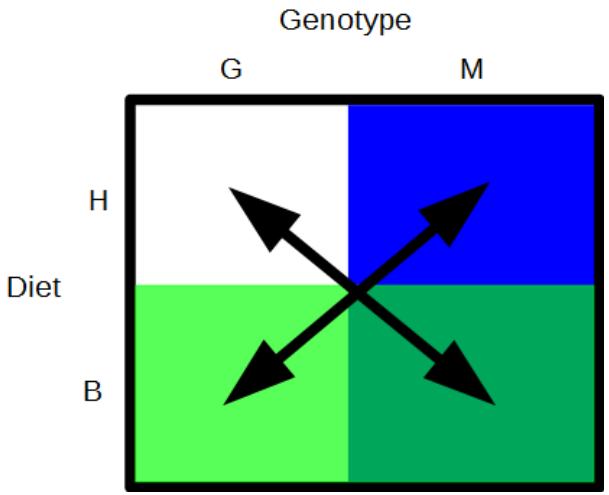
$$\text{genotype\_M\_vs\_G} = 0$$

Non-reference level (H):

$$\text{genotype\_M\_vs\_G} + \text{genotypeM.regimeH} = 0$$

2×2 factorial experiment as classic interaction model

Genotype × diet interaction effect:



- Is the effect of genotype different for each diet? Is the effect of diet different for each genotype?

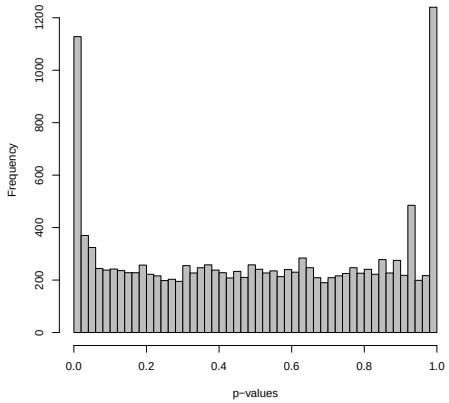
$$\text{genotypeM.regimeH} = 0$$

Outline

Filtering in differential expression analysis

Differential analyses performed gene-by-gene, requiring a correction for multiple testing (e.g., FDR control):

- Stringent correction due to large number of hypothesis tests
- Usually assume p-values are uniformly distributed under H_0



Filtering for RNA-seq data

- Identify and remove genes that generate an uninformative signal
- Only test hypotheses for genes passing filter $\Rightarrow$ tempered correction for multiple testing
- Usually little discussion about appropriate filter & threshold

Defining a data-based filter for HTS data[2]

Let $\mathbf{y}_j$ be the full vector of normalized read counts (e.g. after scaling raw counts by effective library size) in a given sample j.

Idea:

Find the threshold s that maximizes the filtering similarity among replicates in the same condition ($\mathcal{C}(j) = \mathcal{C}(j')$) using the Jaccard index:

$$J_s(\mathbf{y}_j, \mathbf{y}_{j'}) = \frac{a}{a+b+c}$$

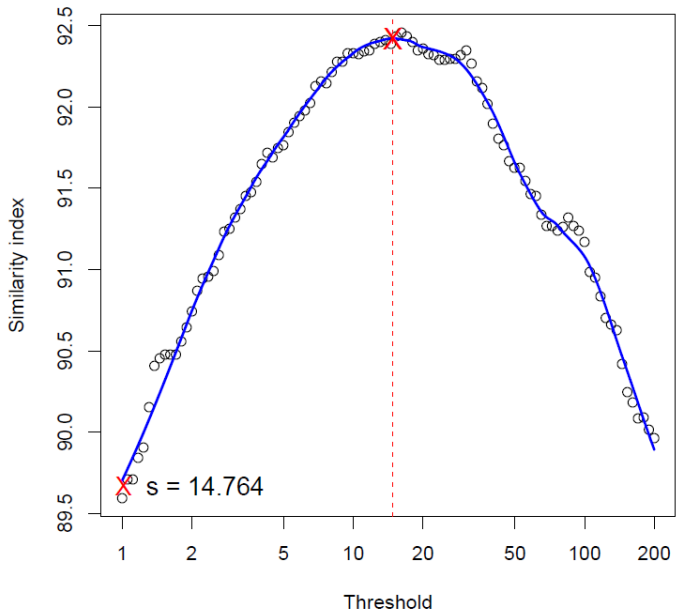
		Sample j	
		Normalized counts $> s$	Normalized counts $\leq s$
Sample j'	Normalized counts $> s$	a	b
	Normalized counts $\leq s$	c	d

[2] Rau et al. (2013)

HTSFilter: Data-driven filtering threshold for HTS data

- Multiple replicates/conditions typically available ⇒ define a global filtering similarity by averaging the pairwise Jaccard indices within each condition:

$$J_s^\star(\mathbf{y}) = \text{mean}\{J_s(\mathbf{y}_j, \mathbf{y}_{j'}) : j < j' \text{ and } \mathcal{C}(j) = \mathcal{C}(j'$$

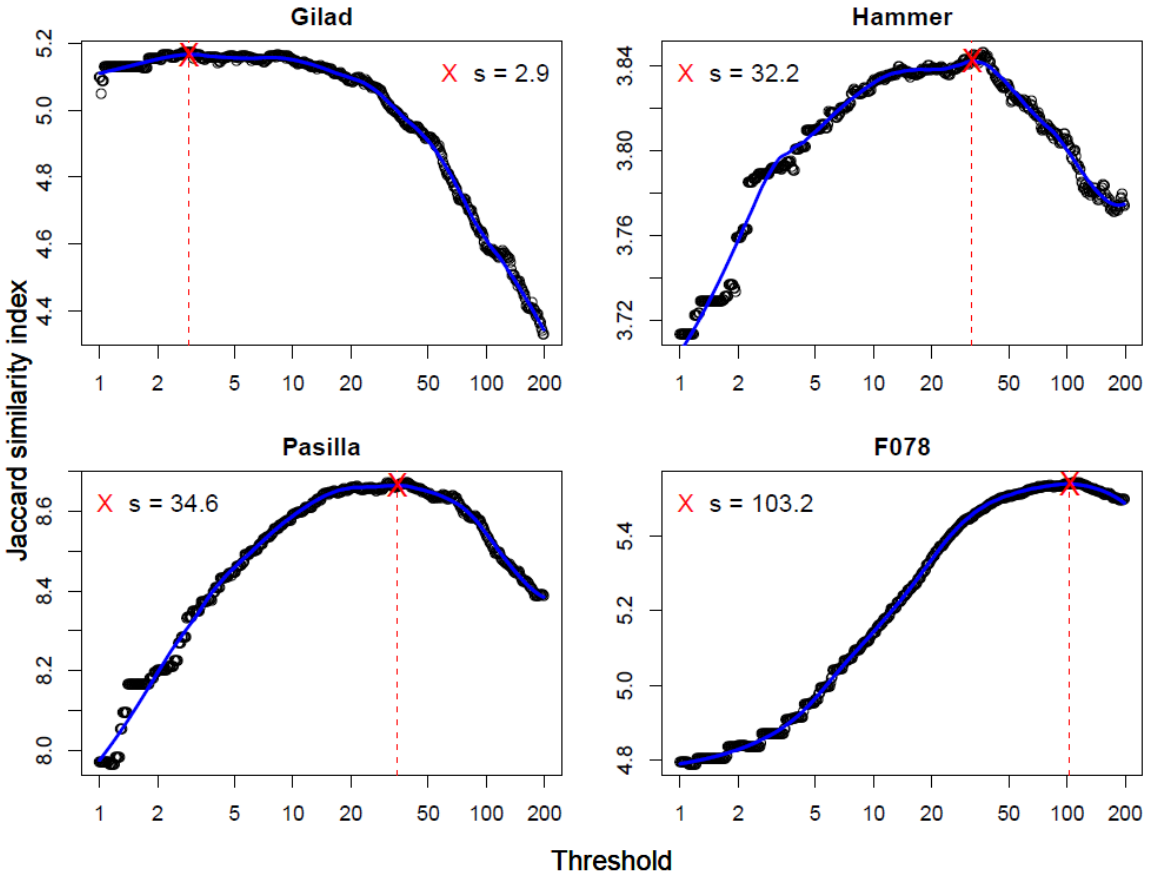


- Data-based filter threshold $s^\star = \arg\max_s J_s^\star(\mathbf{y})$

Proposed data-based Jaccard filter

Filter genes with normalized read counts $\leq s^\star$ in all samples

A word on data-driven threshold values...



Filtering threshold is specific to each dataset (tissue, organism, sequencing depth, intra-condition variability ...)

Implementation of HTSFilter in the DESeq2 pipeline

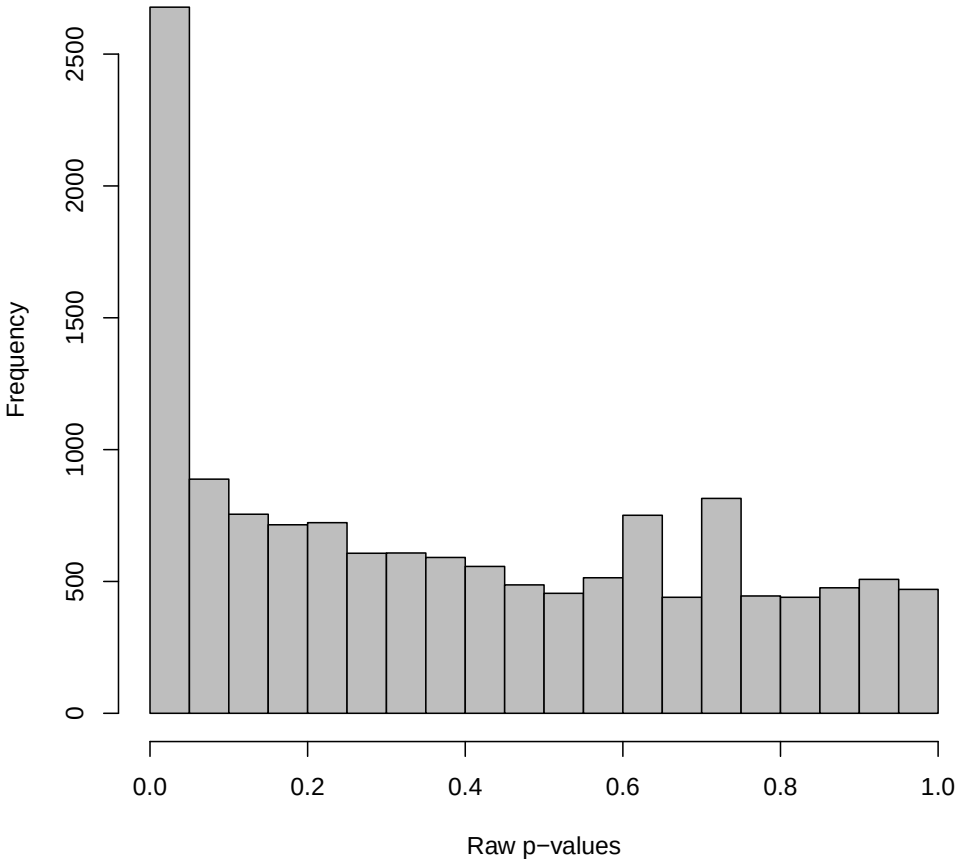
```
> library(DESeq2)
> library(HTSFilter2)
...
> ## DESeq commands
> dds <- DESeqDataSetFromMatrix(...)
> dds <- DESeq(dds)
> ## HTSFilter
> ddsFilter <- HTSFilter(dds)$filteredData
...
```

... and if using HTSFilter, remember to use independentFiltering=FALSE in the DESeq2 results function! (TRUE by default)

Outline

DESeq2: Testing and distribution of raw *p*-values

- Under H_0, *p*-values are uniformly distributed...



Correction for multiple testing

- Reminder: Thousands of genes are analyzed simultaneously!

		Decision	
		Declared DE	Declared NDE
Truth	m_1 DE	TP	FN
	m_0 NDE	FP	TN
	m		

DE & NDE: differentially and non-differentially expressed
TP: true positives
FP: false positives
TN: true negatives
FN: false negatives

Correction for multiple testing: An example

Suppose the following:

- NO genes are differentially expressed ($m = m_0$)
- Each individual test is performed with significance level α
- If $m = 10,000$ and $\alpha = 0.05$, 500 genes will be declared DE although they are NDE

Correction for multiple testing

Control the global risk of having a false positive

Definition of global risk

Family-wise Error Rate (FWER)

- Probability of having at least one false positive
- Bonferroni procedure: each test performed with significance level α/m to control FWER at level α
- VERY conservative, lacks power for large m

Definition of global risk

False discovery rate (FDR)

Proportion of false discoveries (FP) expected among all discoveries (TP + FP):

$$FDR = E\left(\frac{FP}{TP + FP}\right) \text{ if } TP + FP > 0$$
$$= 0 \text{ otherwise}$$

$\Rightarrow$ We are willing to accept a few Type I errors (FP) if their # remains sufficiently small compared to total # of rejected hypotheses

- Increases detection power compared to Bonferroni
- Most commonly used approach: Benjamini and Hochberg (1995) via `p.adjust()` function

Outline

limma Bioconductor package: Smyth (2004)

A brief introduction (or reminder):

- $\mathrm{E}(Y_{ij}) = X_j \boldsymbol{\alpha}_i$, where X_j is the design matrix and $\boldsymbol{\alpha}_i$ the vector of coefficients (no assumption of normality)
- $\mathrm{Var}(Y_{ij}) = w_i \sigma_i^2$ with sample value s_i^2, degrees of freedom f_i, and (known) weights w_i
- Contrasts of interest are $\boldsymbol{\beta}_i = \mathbf{C}' \boldsymbol{\alpha}_i$ where $\mathbf{C}$ is the contrasts matrix

limma Bioconductor package: Smyth (2004)

A brief introduction (or reminder):

- $\mathrm{E}(Y_{ij}) = X_j \boldsymbol{\alpha}_i$, where X_j is the design matrix and $\boldsymbol{\alpha}_i$ the vector of coefficients (no assumption of normality)
- $\mathrm{Var}(Y_{ij}) = w_i \sigma_i^2$ with sample value s_i^2, degrees of freedom f_i, and (known) weights w_i
- Contrasts of interest are $\boldsymbol{\beta}_i = \mathbf{C}' \boldsymbol{\alpha}_i$ where $\mathbf{C}$ is the contrasts matrix

Ordinary test-statistic for rth contrast for gene i is

$$t_{ir} = \frac{\hat{\beta}_{ik}}{u_{ir} s_i}$$

where u_{ir} is the unscaled standard deviation of contrast r

limma Bioconductor package: Smyth (2004)

Empirical Bayes shrinkage estimator:

- Borrow information across all genes for more stable per-gene variance estimates
- Assume an inverse χ^2 prior for σ_i^2 with mean s_0 and f_0 df
- The posterior mean for the residual variance is

$$\tilde{s}_i^2 = \frac{f_0 s_0^2 + f_i s_i^2}{f_0 + f_i}$$

limma Bioconductor package: Smyth (2004)

Empirical Bayes shrinkage estimator:

- Borrow information across all genes for more stable per-gene variance estimates
- Assume an inverse χ^2 prior for σ_i^2 with mean s_0 and f_0 df
- The posterior mean for the residual variance is

$$\tilde{s}_i^2 = \frac{f_0 s_0^2 + f_i s_i^2}{f_0 + f_i}$$

Then use the moderated test-statisic for rth contrast for gene i:

$$t_{ir} = \frac{\hat{\beta}_{ik}}{u_{ir}\tilde{s}_i}$$

where u_{ir} is the unscaled standard deviation of contrast r

limma-voom approach: Law et al. (2014)

To use the limma pipeline with RNA-seq, two steps are needed:

1. Transform data to log-cpm values:

$$\text{log-cpm} = \log_2 \left(\frac{y_{ij} + 0.5}{y_{j\cdot} + 1} \times 10^6 \right)$$

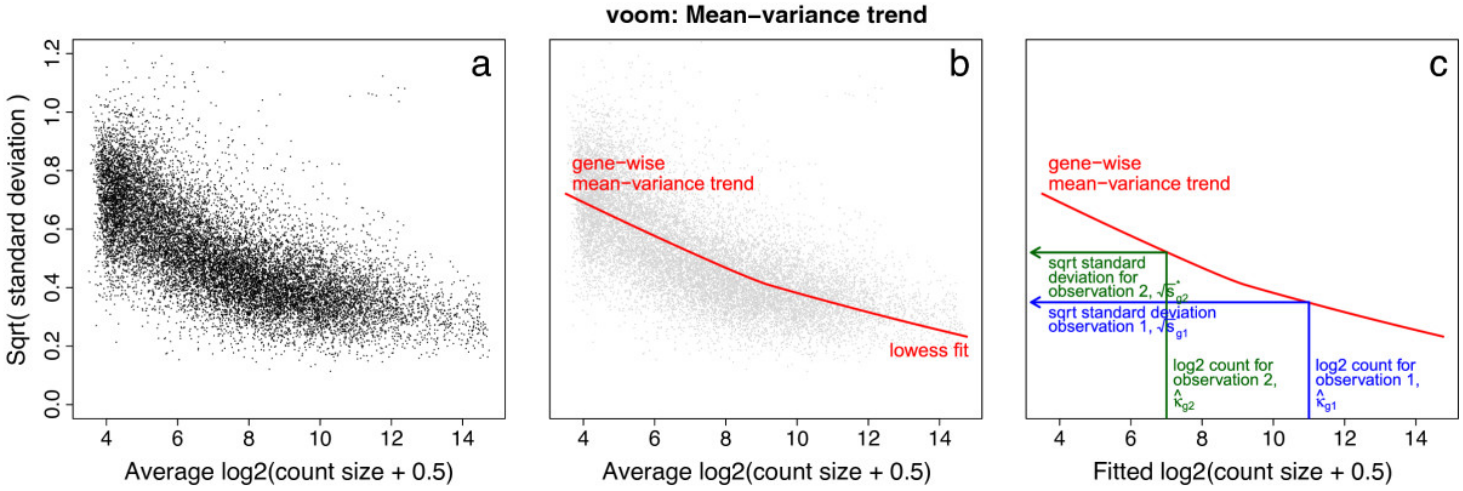
limma-voom approach: Law et al. (2014)

To use the limma pipeline with RNA-seq, two steps are needed:

1. Transform data to log-cpm values:

$$\text{log-cpm} = \log_2 \left(\frac{y_{ij} + 0.5}{y_{j\cdot} + 1} \times 10^6 \right)$$

2. Voom variance modeling for precision weights:
   - Plot log-mean counts versus $\sqrt{s_i}$, and fit a loess curve to
   - Precision weights $w_i = \text{loess}(\text{log-mean counts})^{-4}$ used in standard limma pipeline



limma-voom approach: Law et al. (2014)

Similarities with DESeq2 approach:

- Correction for multiple testing needed
- 2 $\times$ factorial experiment may be analysed as a single factor or two factors
- Definition of contrasts as before
- Data filtering for weakly expressed genes

In general, limma-voom performs very well when a large number of replicates are available...

Next steps

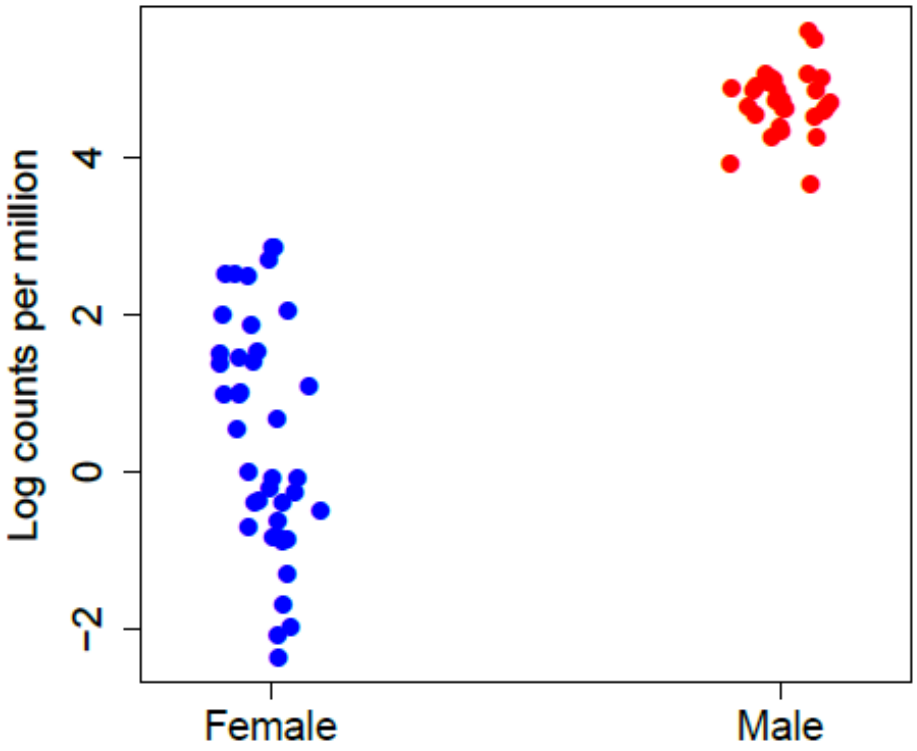
What happens after a differential analysis?

- Further analysis
  - Differential variability analysis using `DiffVar`
  - Gene co-expression analysis using `coseq`
  - Test for enriched functional categories using `goseq` (i.e., do differentially expressed genes tend to share the same function?)
  - Inference of gene networks
  - Integration with other data (epigenomic, metabolomic, proteomic, ...)
- Biological validation
  - Gene knock-down experiments
  - qPCR validation

Outline

Differential variability

- Although focus is typically on differential mean expression among groups, it may be of interest to identify genes with differing variability between groups
- Originally proposed for DNA methylation data (`DiffVar` function in the `missMethyl` package)



DiffVar approach[3]

- Intuitively, variability may be thought of as a distance from each point in a group to the group mean $d_{ij} = y_{ij} - \bar{y}_{i,\mathcal{C}(j)}$
  - Highly variable groups = consistently large deviations from mean
  - Low variability groups = consistently small deviations from mean
- Perform moderated t-test using `limma` pipeline on $|d_{ij}|$ or d_{ij}^2 values to test null hypothesis of equal variance between groups
- Multiple testing correction as before

[3] Phipson and Oshlack (2014)

Outline

From gene co-expression to gene function prediction

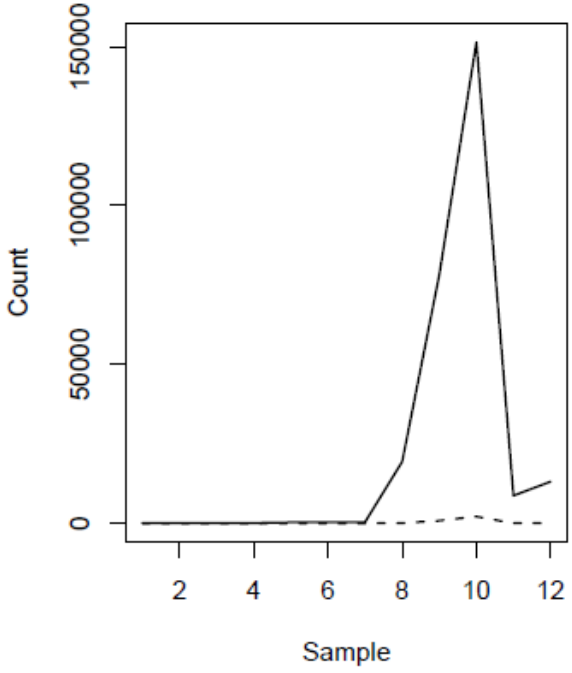
Co-expression (clustering) analysis

- Study patterns of relative gene expression (*profiles*) across several conditions
- $\Rightarrow$ Co-expression is a tool to study genes without known or predicted function (orphan genes)
- Exploratory tool to identify expression trends from the data ($\neq$ sample classification, identification of differential expression)

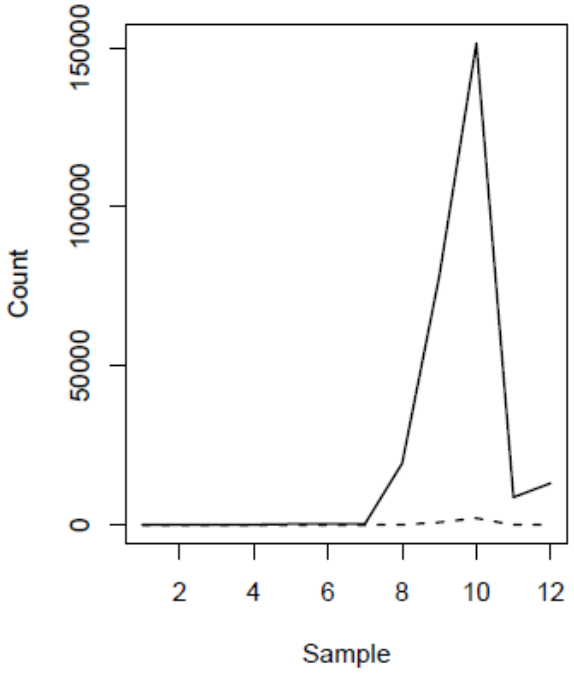
In practice, prior to co-expression analysis:

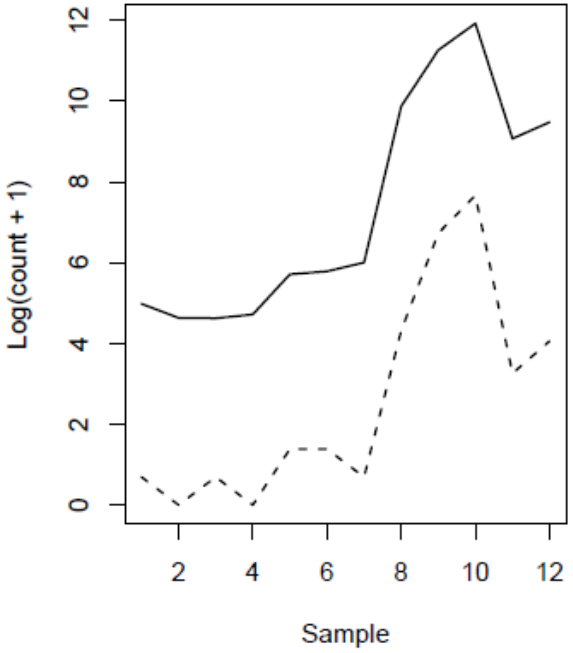
- Perform differential analysis and filter out genes that are declared non-differentially expressed
- Filter out weakly expressed genes

RNA-seq profiles for co-expression

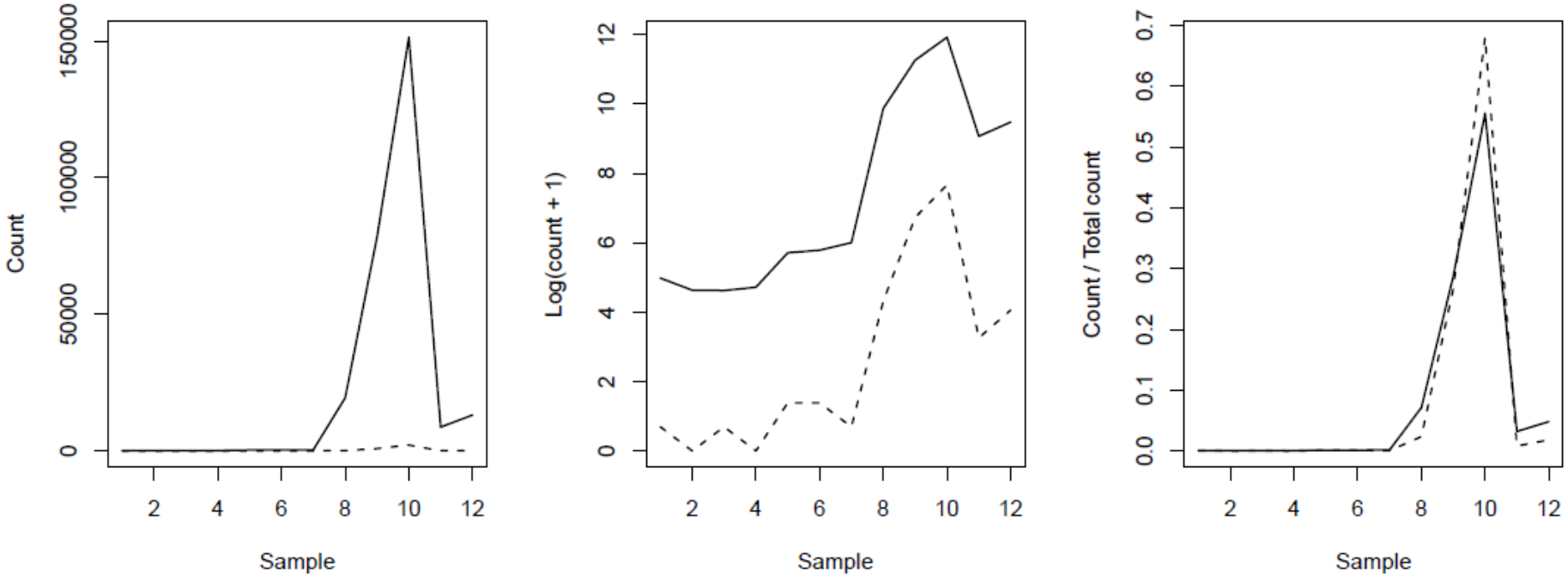


RNA-seq profiles for co-expression



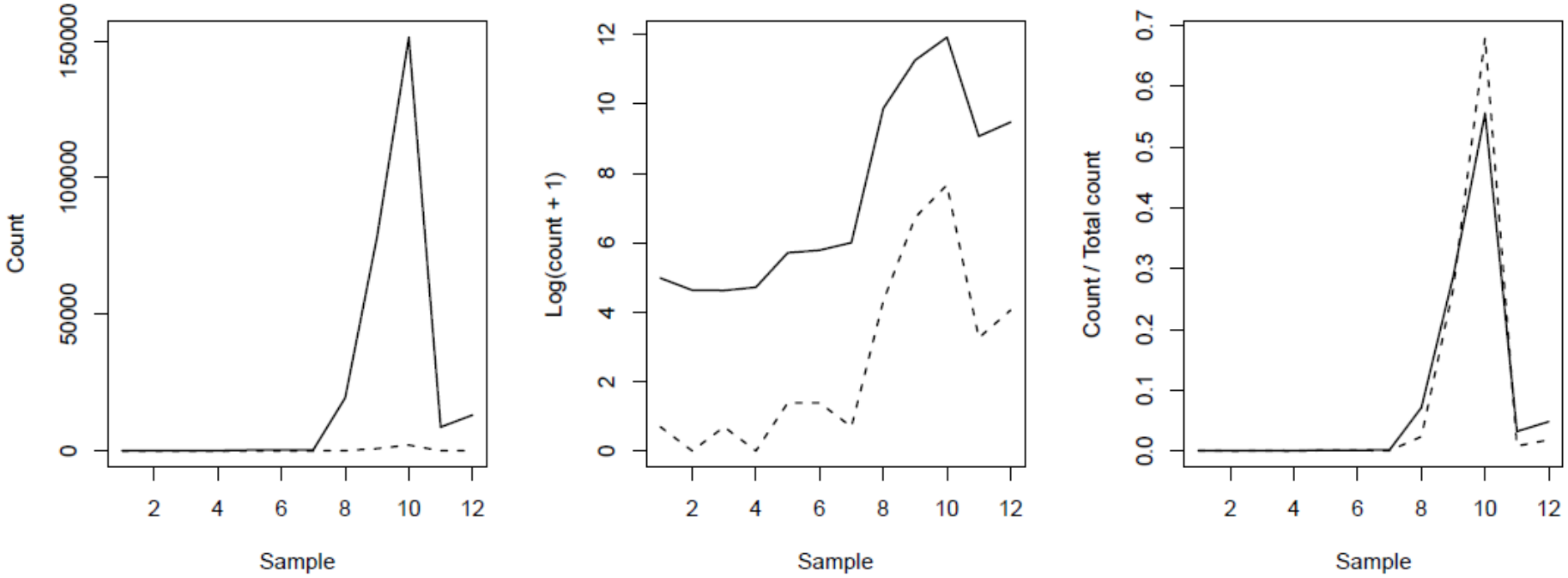


RNA-seq profiles for co-expression



- Let y_{ij} be the raw count for gene i in sample j, with library size s_j
- Profile for gene i: $p_{ij} = \frac{y_{ij}}{\sum_\ell y_{i\ell}}$

RNA-seq profiles for co-expression



- Let y_{ij} be the raw count for gene i in sample j, with library size s_j
- Profile for gene i: $p_{ij} = \frac{y_{ij}}{\sum_\ell y_{i\ell}}$
- Normalized profile for gene i: $p_{ij} = \frac{y_{ij}/s_j}{\sum_\ell y_{i\ell}/s_j}$

Unsupervised clustering

Objective

Define homogeneous and well-separated groups of genes from transcriptomic data

What does it mean for a pair of genes to be close?
Given this, how do we define groups?

Unsupervised clustering

Objective

Define homogeneous and well-separated groups of genes from transcriptomic data

What does it mean for a pair of genes to be close?
Given this, how do we define groups?

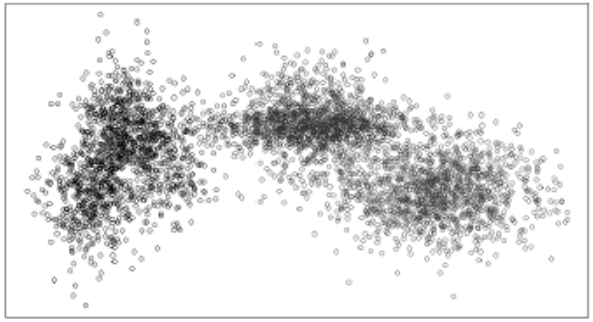
Two broad classes of methods typically used:

1. Centroid-based clustering (K-means and hierarchical clustering)
2. Model-based clustering (mixture models)

Model-based clustering

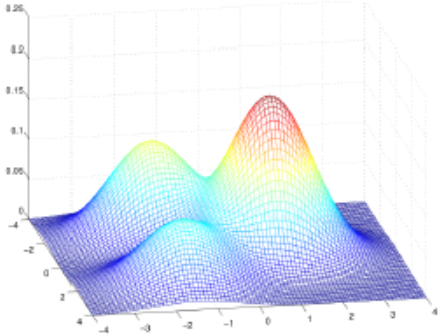
- Probabilistic clustering models : data are assumed to come from distinct subpopulations, each modeled separately
- Rigourous framework for parameter estimation and model selection
- **Output**: each gene assigned a probability of cluster membership

what we observe

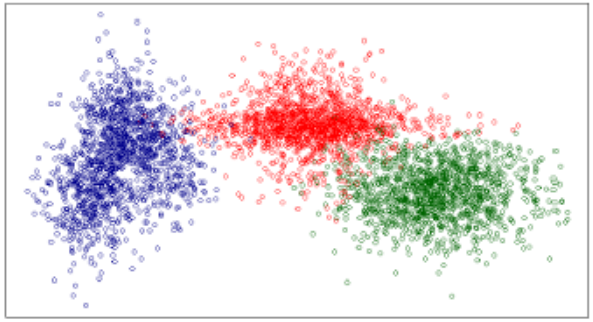


$Z = ?$

the model



the expected results



$Z : 1 = \bullet, 2 = \bullet, 3 = \bullet$

Key ingredients of a mixture model

- Let $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$ denote the observations with $\mathbf{y}_i \in \mathbb{R}^p$
- We introduce a latent variable to indicate the group from which each observation arises:

$$Z_i \sim \mathcal{M}(n; \pi_1, \ldots, \pi_K),$$
$$P(Z_i = k) = \pi_k$$

- Assume that $\mathbf{y}_i$ are conditionally independent given Z_i
- Model the distribution of $\mathbf{y}_i | Z_i$ using a parametric distribution:

$$(\mathbf{y}_i | Z_i = k) \sim f(\cdot; \theta_k)$$

Questions around the mixtures

- Model: what distribution to use for each component ?
  $\rightsquigarrow$ depends on the observed data.

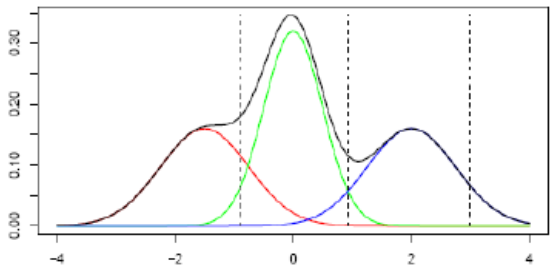
- Inference: how to estimate the parameters ?
  $\rightsquigarrow$ usually done with an EM-like algorithm (Dempster *et al.*, 1977)

- Model selection: how to choose the number of components ?
  - A collection of mixtures with **a varying number of components** is usually considered
  - A penalized criterion (e.g., BIC, ICL) is used to select the best model from the collection

Clustering data into components

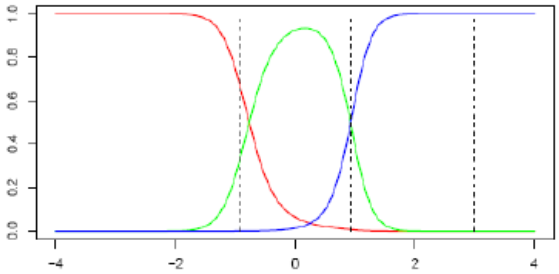
Distributions:

$$g(x) = \pi_1 f_1(x) + \pi_2 f_2(x) + \pi_3 f_3(x)$$



Conditional probabilities:

$$\tau_{ik} = \frac{\pi_k f_k(x_i)}{g(x_i)}$$



Maximum a posteriori (MAP) rule: Assign genes to the component with highest conditional probability τ_{ik}:

τ_{ik} (%)	$k = 1$	$k = 2$	$k = 3$
$i = 1$	65.8	34.2	0.0
$i = 2$	0.7	47.8	51.5
$i = 3$	0.0	0.0	100
...	...	...	...

Finite mixture models for RNA-seq

Assume data $\mathbf{y}$ come from K distinct subpopulations, each modeled separately:

$$f(\mathbf{y}|K, \boldsymbol{\Psi}_K) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k f_k(\mathbf{y}_i; \boldsymbol{\theta}_k)$$

- $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)'$ are the mixing proportions, where $\sum_{k=1}^{K} \pi_k = 1$
- f_k are the densities of each of the components

Finite mixture models for RNA-seq

Assume data $\mathbf{y}$ come from K distinct subpopulations, each modeled separately:

$$f(\mathbf{y}|K, \boldsymbol{\Psi}_K) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k f_k(\mathbf{y}_i; \boldsymbol{\theta}_k)$$

- $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)'$ are the mixing proportions, where $\sum_{k=1}^{K} \pi_k = 1$
- f_k are the densities of each of the components

- For microarray data, we often assume $\mathbf{y}_i|k \sim \text{MVN}(\boldsymbol{\mu}_k, \Sigma_k)$
- What about RNA-seq data?

Finite mixture models for RNA-seq data

$$f(\mathbf{y}|K, \boldsymbol{\Psi}_K) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k f_k(\mathbf{y}_i|\boldsymbol{\theta}_k)$$

For RNA-seq data, we must choose the family & parameterization of $f_k(\cdot)$:

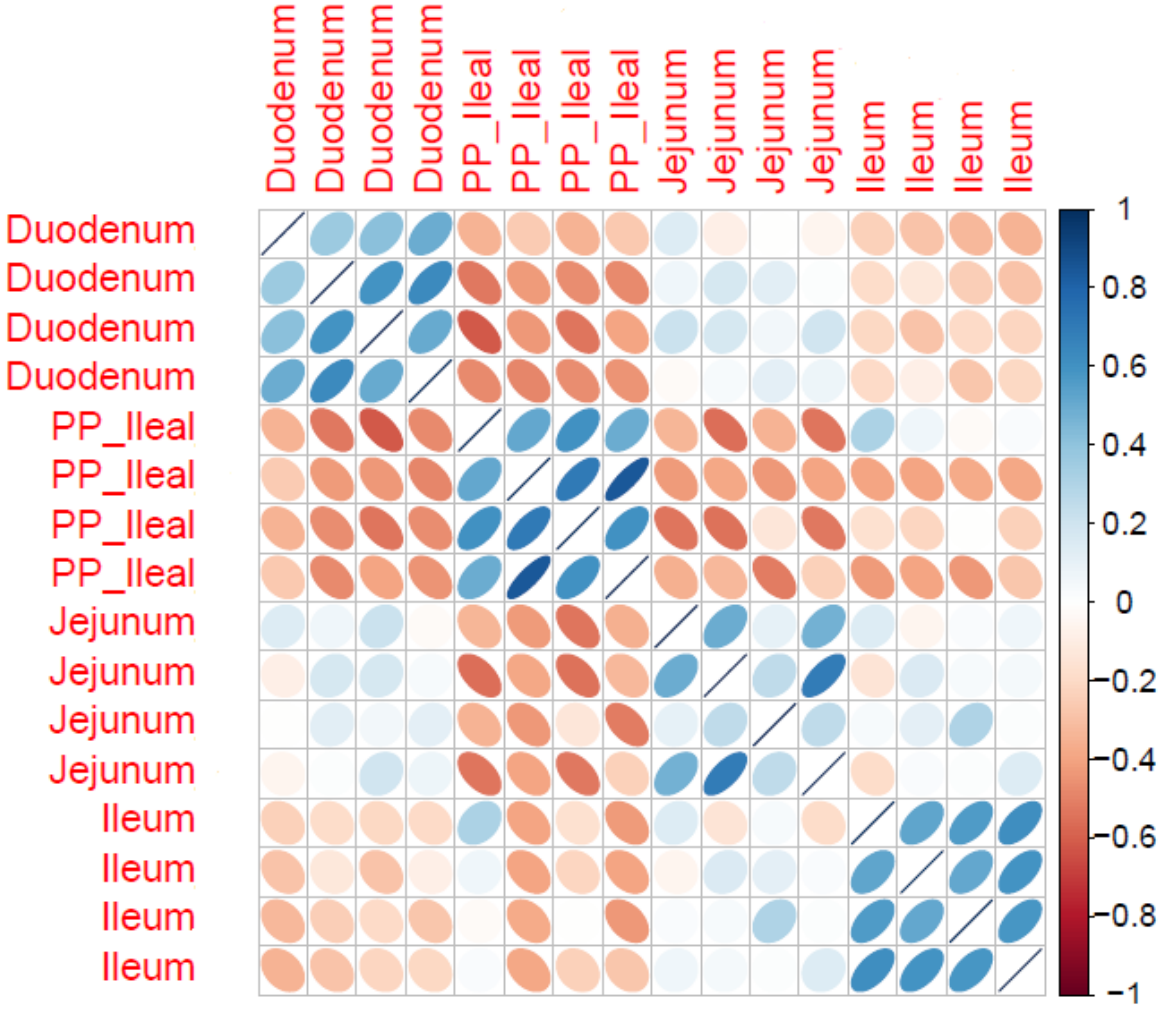
1. Directly model read counts (`HTSCluster`): Rau et al. (2015)

$$\mathbf{y}_i|Z_i = k \sim \prod_{j=1}^{J} \text{Poisson}(y_{ij}|\mu_{ijk})$$

2. Apply appropriately chosen data transformation (`coseq`): Rau and Maugis-Rabusseau (2017)

$$g(\mathbf{y}_i)|Z_i = k \sim \text{MVN}(\mu_k, \Sigma_k)$$

Correlation structures in RNA-seq data



Example: data from Mach *et al.* (2014) on site-specific gene expression along the gastrointestinal tract of 4 healthy piglets

Gaussian mixture models for RNA-seq

Idea: Transform RNA-seq data, then apply Gaussian mixture models

Several data transformations have been proposed for RNA-seq to render the data approximately homoskedastic:

- $\log_2(y_{ij} + c)$
- Variance stabilizing transformation (`DESeq`)
- Moderated log counts per million (`edgeR`)
- Regularized log-transformation (`DESeq2`)

... but recall that we wish to cluster the normalized profiles $p_{ij} = \frac{y_{ij}/s_j}{\sum_\ell y_{i\ell}/s_j}$

Remark: transformation needed for normalized profiles

- Note that the normalized profiles are *compositional data*, i.e. the sum for each gene $p_{i\cdot} = 1$
- This implies that the vector $\mathbf{p}_i$ is linearly dependent $\Rightarrow$ imposes constraints on the covariance matrices Σ_k that are problematic for the general GMM

- As such, we consider a transformation on the normalized profiles to break the sum constraint:

$$\tilde{p}_{ij} = g(p_{ij}) = \arcsin\left(\sqrt{p_{ij}}\right)$$

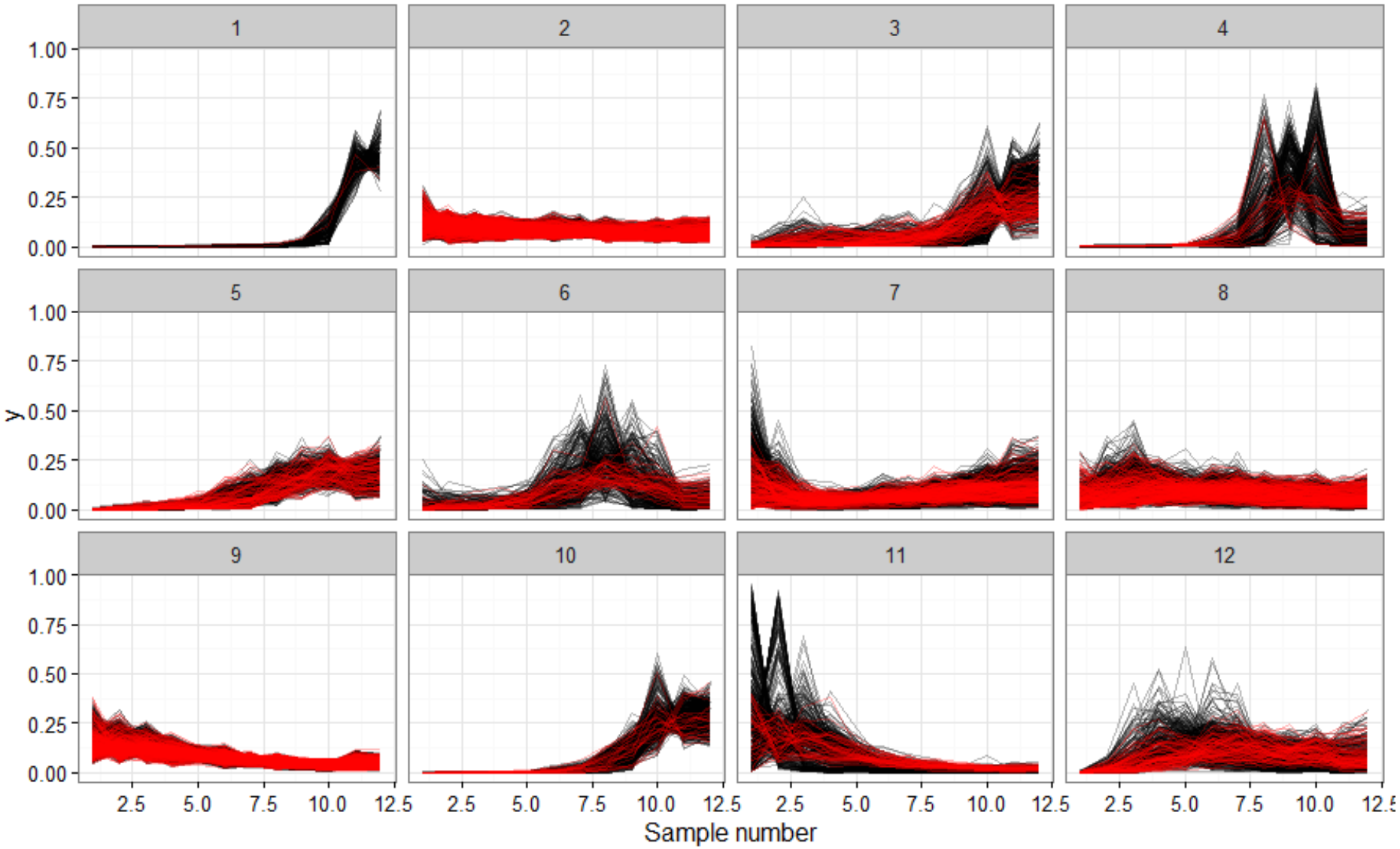
And fit a GMM to the transformed normalized profiles:

$$f(\tilde{\mathbf{p}}|K, \boldsymbol{\Psi}_K) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \phi(\tilde{\mathbf{p}}_i|\boldsymbol{\theta}_k, \Sigma_k)$$

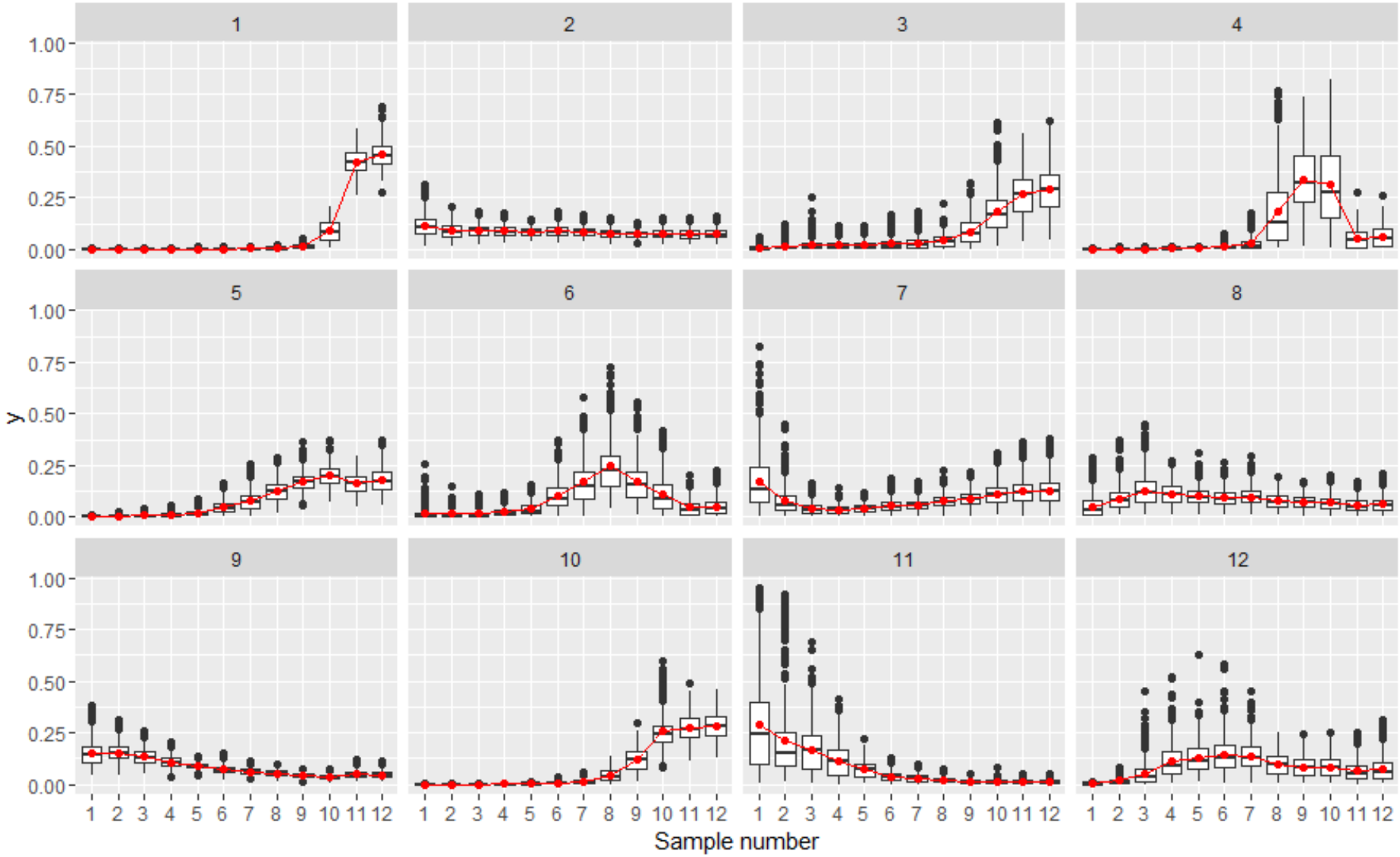
Fitting a GMM for RNA-seq data with coseq

```
> library(coseq)
>
> GMM <- coseq(counts, K=2:10, model="Normal",
>              transformation="arcsin")
> summary(GMM)
> plot(GMM)
```

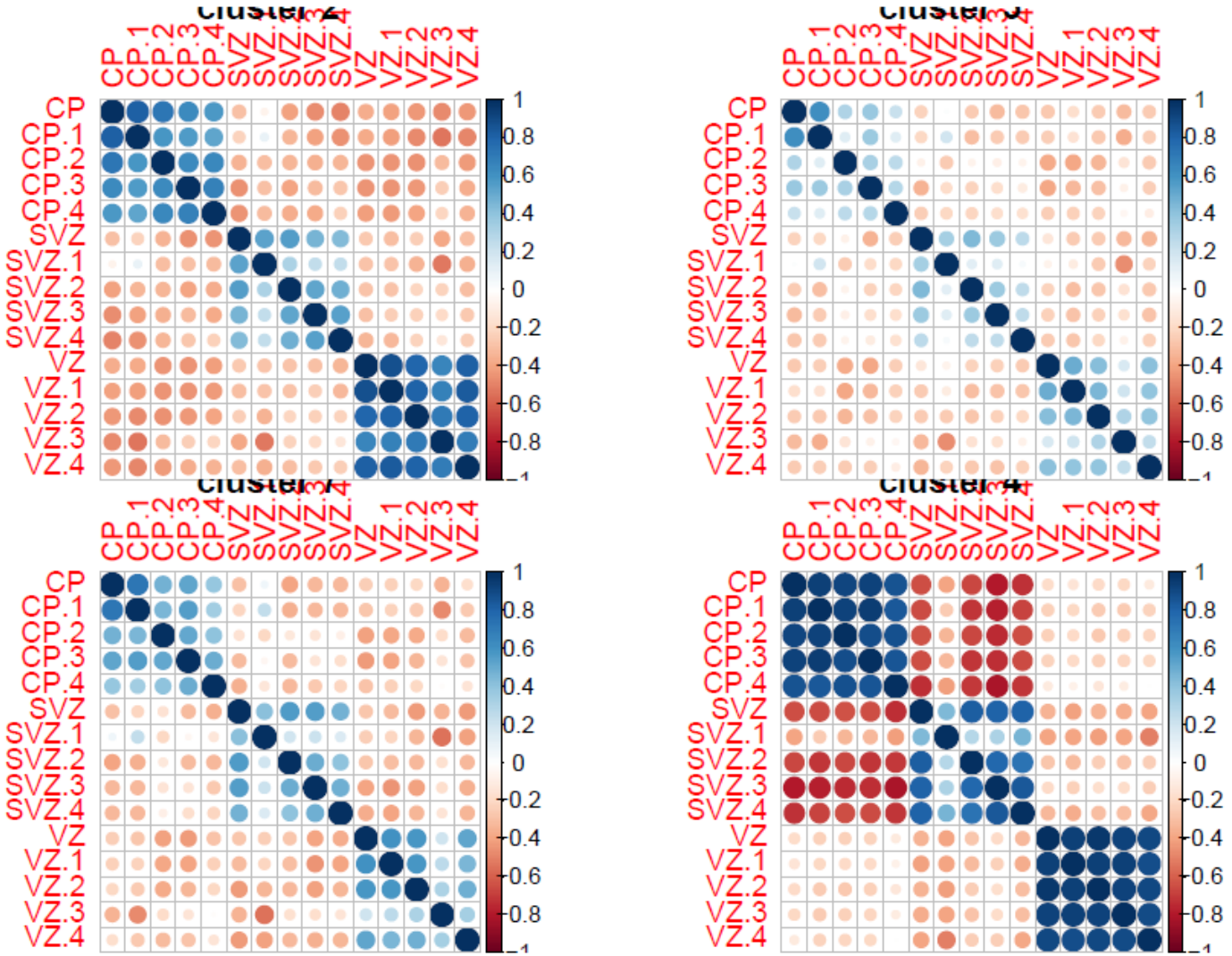
Examining GMM results



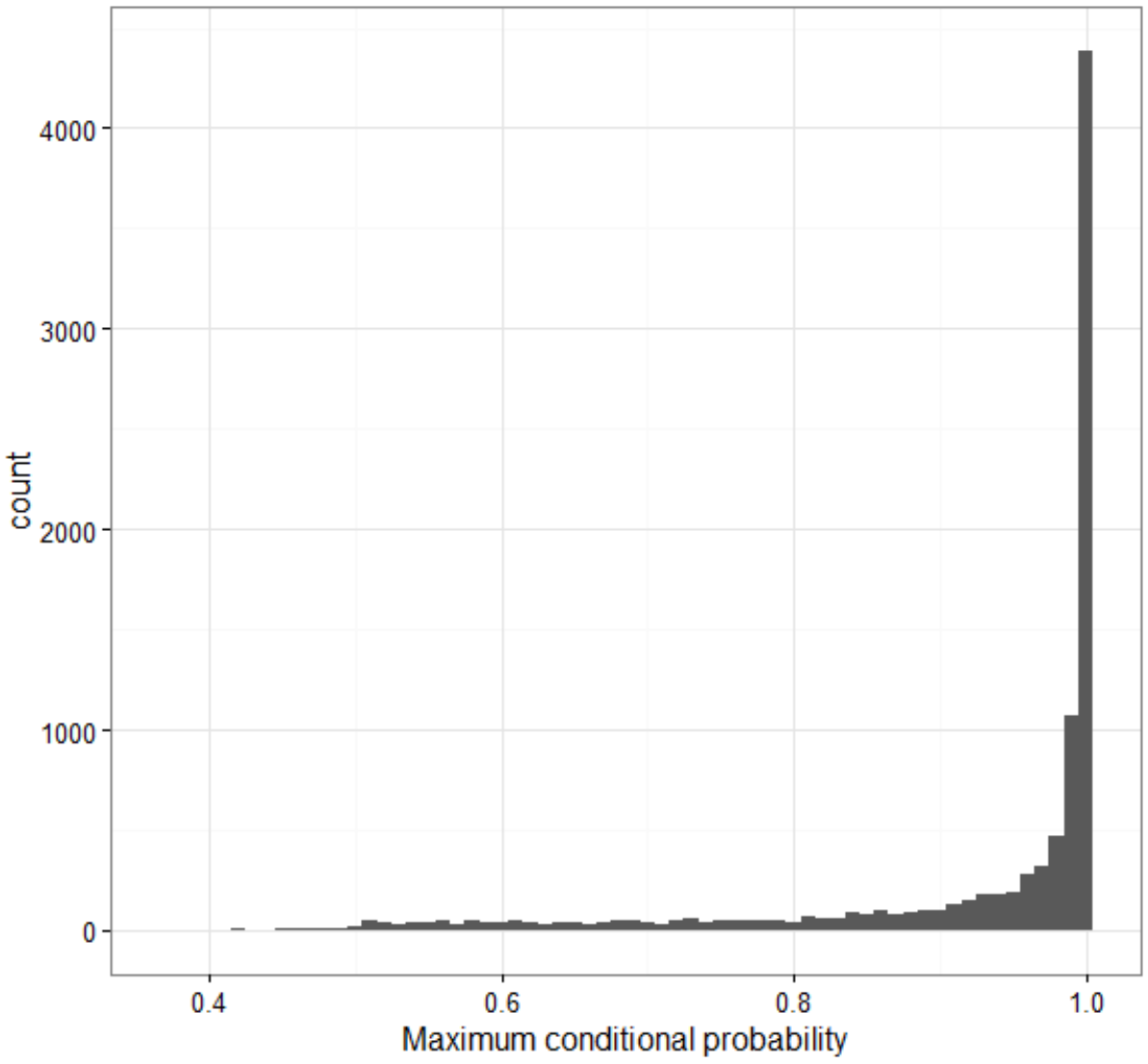
Examining GMM results



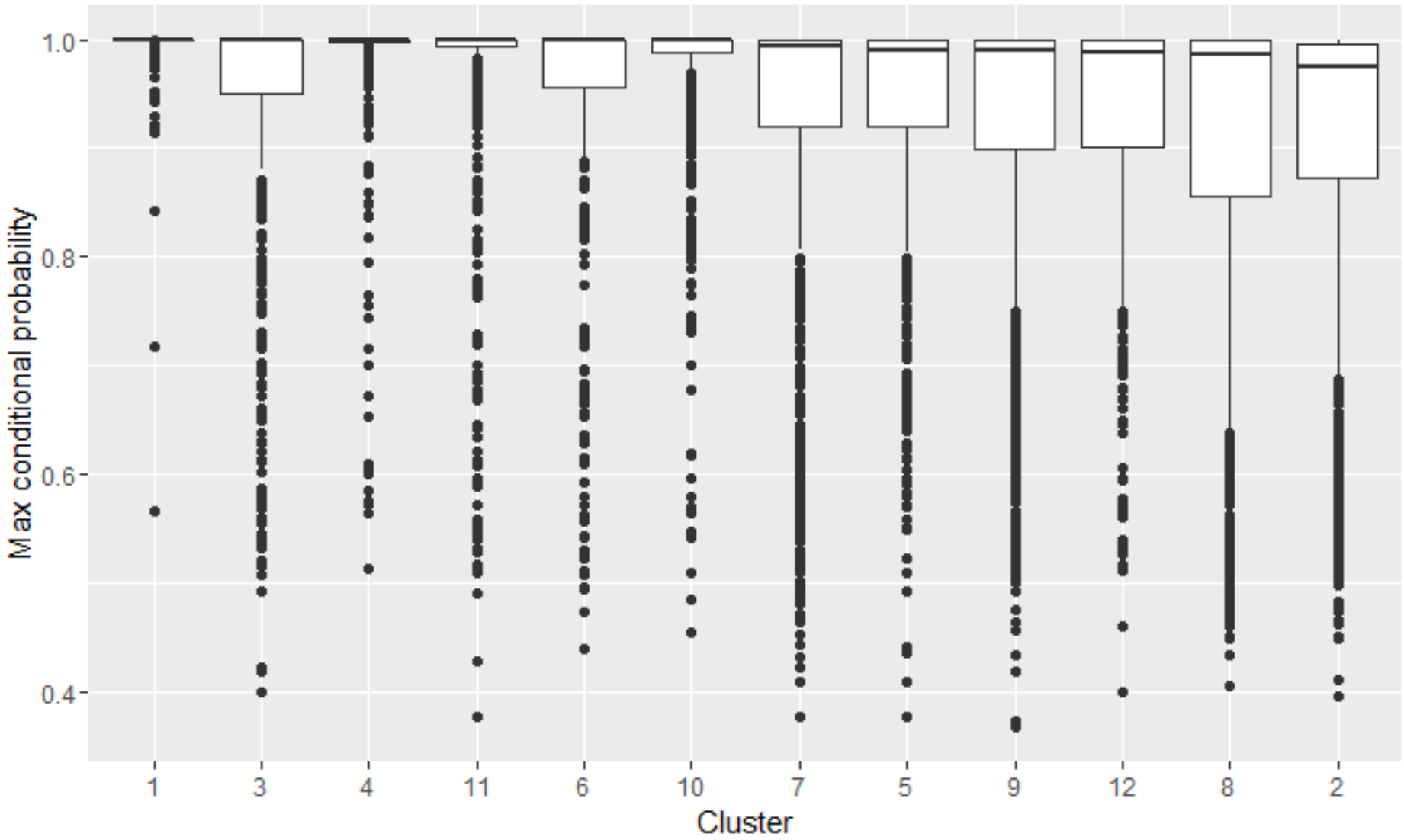
Examining GMM results



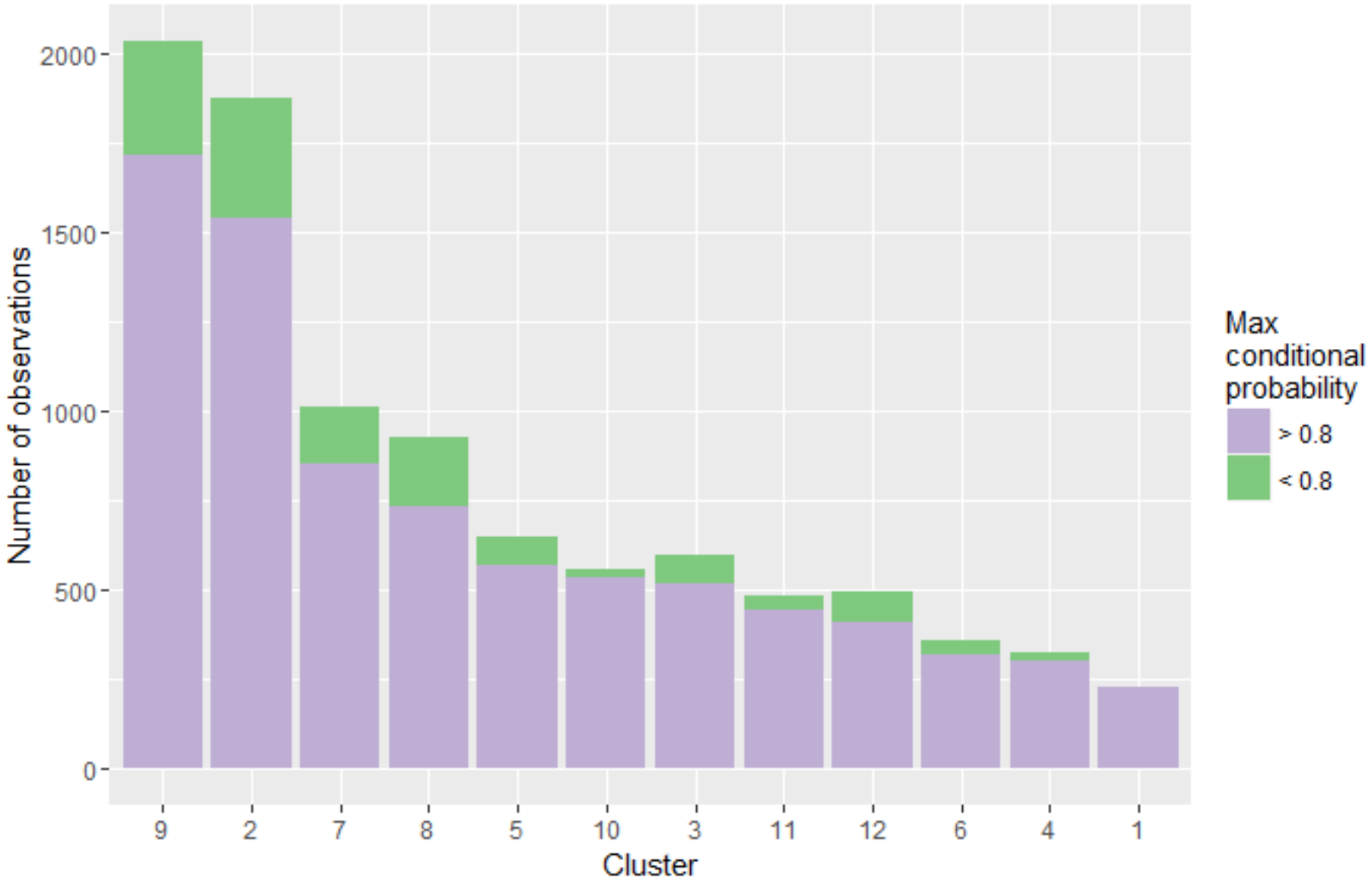
Evaluation of clustering quality



Evaluation of clustering quality



Evaluation of clustering quality



A note about evaluating clustering approaches[4]

- Clustering results can be evaluated based on internal criteria (e.g., statistical properties of clusters) or external criteria (e.g., functional enrichment of GO terms)
- Preprocessing details (normalization, filtering, dealing with missing values) can affect clustering outcome
- Methods that give different results depending on the initialization should be rerun multiple times to check for stability
- Most clustering methods will find clusters even when no actual structure is present ⇒ good idea to compare to results with randomized data!

[4] D'haeseller, 2005

Outline

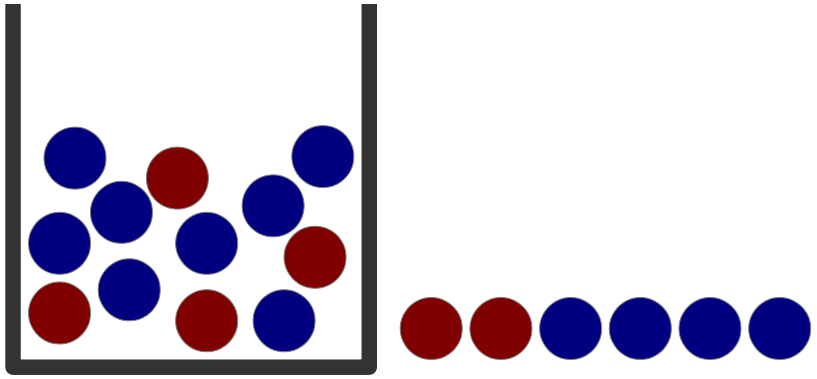
Enriched functional categories

What biological processes are over-represented among the genes identified to be differentially expressed?

- Systems biology technique known as gene category enrichment analysis
- $\Rightarrow$ Genes are grouped into categories by a common biological property and tested to find categories over-represented among DE genes
- Commonly Gene Ontology (GO) categories are used for such an analysis

GO enrichment analysis

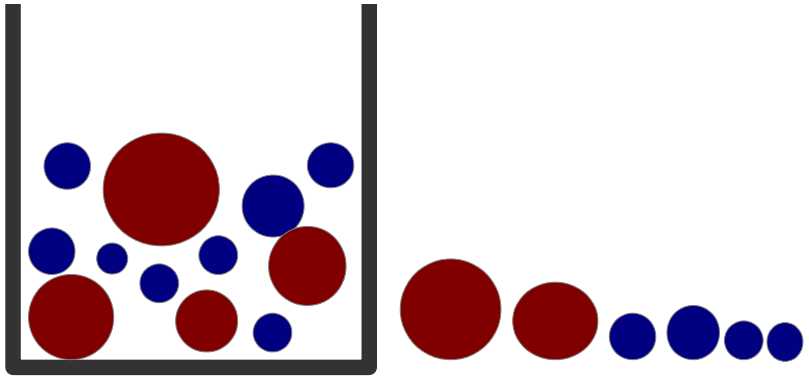
- Assumptions: genes are independent and equally likely to be selected as DE, under the null hypothesis
- If assumptions met, we can test for over-representation using a hypergeometric distribution (Fisher's test):



	Red	Blue	Total
Chosen	2	4	6
Remaining	4	8	12
Total	6	12	18

GO enrichment analysis: Bias due to length

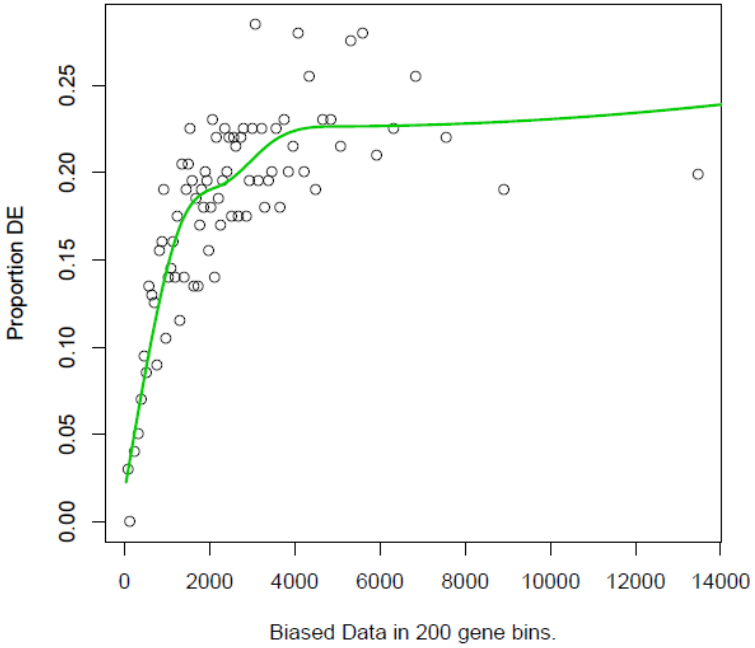
- We have greater statistical power to detect differential expression from longer genes, so under H_0 long and short genes do not have the same probability of being detected as DE
- Without correcting for this bias, categories with many long genes are more likely to show up as over-represented than categories with genes of average length $\Rightarrow$ A biased urn!



goseq package for GO enrichment of RNA-seq data[5]

Once differential analysis (and correction for multiple testing) has been performed (Young et al. (2010)):

Step 1 Calculate the likelihood of DE as a function of transcript length by fitting a monotonic function to DE vs. transcript length



goseq package for GO enrichment of RNA-seq data

Step 2 Incorporate the DE vs. length function into the statistical test of each category's significance using as an approximation the Wallenius non-central hypergeometric distribution

- Extension of classic hypergeometric distribution where probabilities of success and failure differ
- Uses the mean of probability weightings for genes within/outside a given category as the common probability of choosing a gene from within/outside that categroy

Step 3 Correction of p-values for multiple testing!

References I

Anders, S. and W. Huber (2010). Differential expression analysis for sequence count data. *Genome Biology 11*(R106), 1–28.

Bullard, J. H., E. A. Purdom, K. D. Hansen, and S. Dudoit (2010). Evaluation of statistical methods for normalization and differential expression in mRNA-seq experiments. *BMC B 11*(94).

Dillies, M.-A., A. Rau, J. Aubert, C. Hennequet-Antier, M. Jeanmougin, N. Servant, C. Keime, G. Marot, D. Castel, J. Estelle, G. Guernec, B. Jagla, L. Jouneau, D. Laloë, C. Le Gall, B. Schaëffer, S. Le Crom, and F. Jaffrézic (2013). A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in Bioinformatics 14*(6), 671–683.

Law, C., Y. Chen, W. Shi, and G. Smyth (2014). voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology 15*(R29).

Love, M. I., W. Huber, and S. Anders (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology 15*(550).

Marioni, J. C., C. E. Mason, S. M. Mane, M. Stephens, and Y. Gilad (2008). RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research 18*(9), 1509–1517.

References II

Mortazavi, A., B. Williams, K. McCue, L. Schaeffer, and B. Wold (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods 5*(7), 621–628.

Nagalakshmi, U., Z. Wang, K. Waern, C. Shou, D. Raha, M. Gerstein, and M. Snyder (2008). The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science 320*, 1344–1349.

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

Phipson, B. and A. Oshlack (2014). DiffVar: a new method for detecting differential variability with application to methylation in cancer and aging. *Genome Biology 15*(465).

Rau, A. et al. (2013). Data-based filtering for replicated high-throughput transcriptome sequencing experiments. *Bioinf. 29*(17), 2146–2152.

Rau, A. and C. Maugis-Rabusseau (2017). Transformation and model choice for rna-seq co-expression analysis. *Briefings in Bioinformatics*.

Rau, A., C. Maugis-Rabusseau, M.-L. Martin-Magniette, and G. Celeux (2015). Co-expression analysis of high-throughput transcriptome sequencing data with poisson mixture models. *Bioinformatics 31*(9), 1420–1427.

References III

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

Smyth, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology 1*(3), 1–26.

Wang, Z., M. Gerstein, and M. Snyder (2009). RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics 10*(January), 57–63.

Young, M. D., M. J. Wakefield, G. K. Smyth, and A. Oshlack (2010). Gene ontology analysis for RNA-seq: accounting for selection bias. *Genome Biology 11*(R14).