Bulk RNA-sequencing workflows

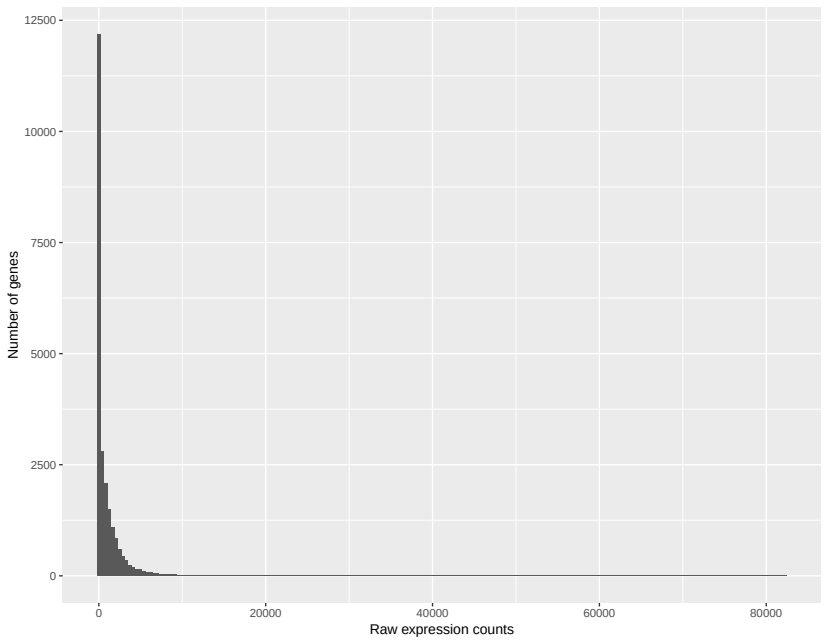
Linda Mhalla[1]

[1]Institute of Mathematics, EPFL, Switzerland

Summer School on Bioinformatic Analysis of RNA-seq

Features of RNA-seq data

- integer counts instead of continuous measurements and non-normally distributed data
- long right tail due to the lack of any upper limit for expression
- low number of counts associated with a large proportion of genes
- genes with larger average expression levels will tend to have larger observed variances across replicates
- low number of replicates and large expression range $\Rightarrow$ large variability within groups (samples)



Challenges in differential expression analysis

- RNA-seq data are discrete counts
  - counts usually modelled by a binomial distribution (number of successful events with a given occurrence probability)
  - if the probability of occurrence is small and the number of events is large, the Poisson distribution is more suitable
  - mean $\neq$ variance $\Rightarrow$ **Negative binomial**
- Each sample will have a different number of reads assigned to it, due to the fact that one sample might have more low quality reads, or another sample might have a higher concentration on the flow cell
  $\rightarrow$ normalize counts to ensure accurate comparisons
- RNA-seq data exhibit variability due to biological factors
  $\rightarrow$ borrow information across genes to lower variance of estimation

Outline of the course

Several software tools (DESeq2 and edgeR for the R statistical software) have been developed for differential expression analysis and provide comprehensive workflows for

- data processing
- model fitting
- hypothesis testing

We will go through the technical statistical details of some of the tools provided by DESeq2 (Love et al., 2014), namely

1. Normalization of raw gene counts
2. Principal component analysis
3. GLM for overdispersed counts
4. Modelling of RNA-seq data

Normalization method in DESeq2

Let K_{ij} denote the number of sequencing reads mapped to gene i in sample j

DESeq2 uses the **method of median of ratios** (Anders and Huber, 2010) where sample-wise normalisation constants are defined as

$$s_j = \underset{i:K_i^R \neq 0}{\text{median}} \frac{K_{ij}}{K_i^R}, \quad K_i^R = (\prod_{j=1}^{n} K_{ij})^{1/n}$$

with **size factor** K_i^R being the geometric mean of counts mapped to gene i

$\rightarrow$ accounts for sequencing depth and RNA composition of the sample

Normalization method in DESeq2

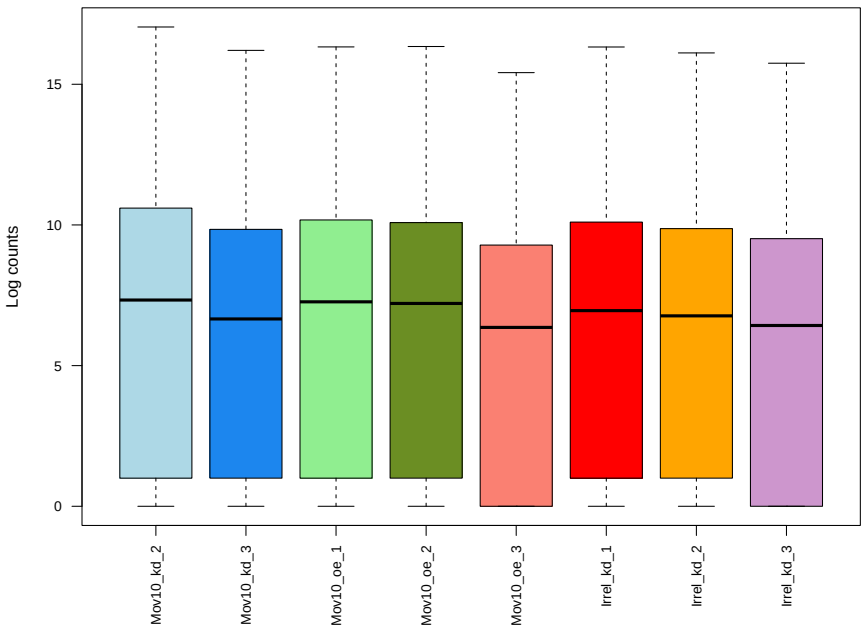
- recommended for gene count comparisons between samples and for DE analysis
  $\rightarrow$ since tools for DEA compare counts between sample groups for the same gene, gene length does not need to be accounted for
- not recommended for comparisons between genes within a sample
- robust to imbalance in up-/down-regulation and large numbers of differentially expressed genes as large outlier genes will not impact the median ratio values
- assumes that most genes are NOT differentially expressed

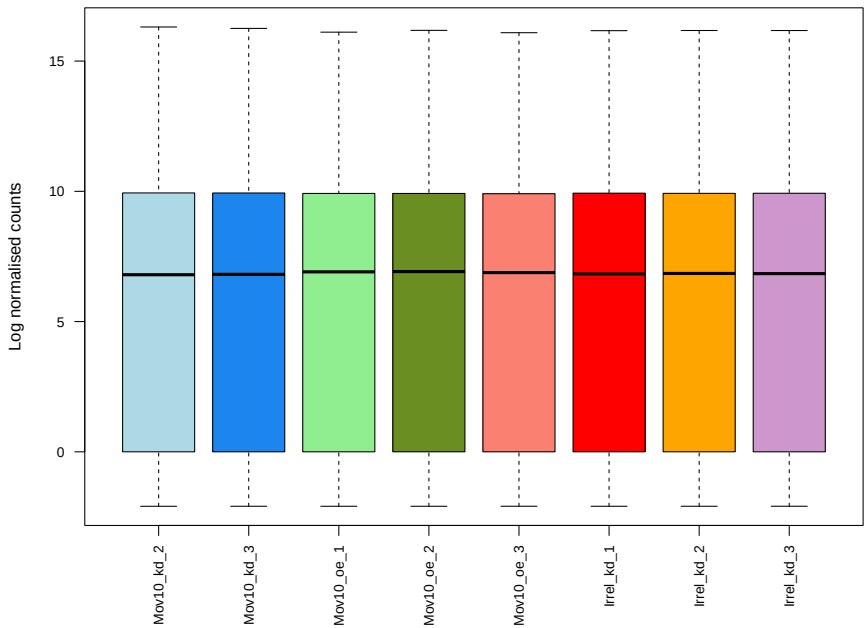
Remark: Usually, the normalization factors (for each sample j) are around 1. If you see large variations between samples it is important to take note since it might indicate the presence of extreme outliers!

Get normalized counts

- Divide each raw count value in a given sample by that sample's normalization factor to generate normalized count values
- DESeq2 doesn't actually use normalized counts, rather it uses the raw counts and models the normalization inside the Generalized Linear Model (GLM): details to follow
- normalized counts will be useful for downstream visualization of results, but cannot be used as input to DESeq2 or any other tools that perform differential expression analysis which use the negative binomial model

Normalized counts





Principal component analysis: Introduction

PCA is a dimensionality reduction technique used to transform a high-dimensional dataset into a lower-dimensional space while preserving its essential features. PCA has various applications:

- **Data Visualization:** PCA can be used to reduce the dimensionality of data for visualization purposes, enabling the identification of patterns and clusters
- **Data Compression:** PCA can compress data by representing it in a lower-dimensional space while retaining most of its variance

By capturing the most significant variations in the data, PCA provides a **lower-dimensional representation** while **preserving the essential information**

$\rightarrow$ explore complex datasets, identify important features

PCA for RNA-seq data

In the context of RNA-seq data, PCA

- is applied before any downstream analysis to check if we should expect to see some differences in the data
- can help identify the most significant patterns of gene expression across samples or conditions
  $\rightarrow$ look for points that cluster with each other, i.e., points that are more similar to each other than they are to other group points
- can help detect outliers (features that should be dropped to avoid skewing results, mislabeling issues)

PCA: Key steps

PCA involves several key steps:

1. **Standardization:** Since RNA-seq data often contains genes with widely varying expression levels, it's essential to standardize the data to ensure that all genes contribute equally to the PCA
2. **Covariance Matrix Computation:** Calculate the covariance matrix of the standardized data
3. **Eigenvalue-Eigenvector Decomposition:** Obtain the eigenvalues and corresponding eigenvectors of the covariance matrix
4. **Feature Vector Construction:** Select the top k eigenvectors based on their corresponding eigenvalues to form the feature vector matrix
5. **Projection:** Transform the original data onto the lower-dimensional space using the feature vector matrix

PCA: definition

PCA is performed on the logarithm of normalized data (or rlog)
$\rightarrow$ moderate and stabilize the variance to improve the distance/clustering
$\rightarrow$ avoid genes with large ranges dominating the variance

Let $\mathbf{X} = [\, X_1, \cdots, X_p] \in \mathbb{R}^{n \times p}$ be the matrix of such transformed counts, where n is the number of samples in the dataset and p the number of genes
We compute the covariance matrix

$$\mathbf{C} = \frac{1}{n-1} \mathbf{X}^T \mathbf{X}$$

and proceed with its eigenvalue-eigenvector decomposition given by

$$\mathbf{C} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T$$

where $\mathbf{V}$ is the matrix of eigenvectors, and $\mathbf{\Lambda}$ the diagonal matrix of eigenvalues

PCA looks for a linear combination with maximal variance that would separate out the multidimensional objects
$\rightarrow$ select the top $k(< p)$ eigenvectors by rearranging the eigenvalues in descending order and choosing the corresponding eigenvectors:
$\mathbf{W} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k] \in \mathbb{R}^{p \times k}$

We project the original data onto the lower-dimensional space spanned by the selected eigenvectors

$$\mathbf{Y} = \mathbf{XW}$$

For example,

- sample j: $x_j = [\ gene_1, gene_2, \cdots, gene_p]$
- l-th eigenvector $\mathbf{v}_l = [\ v_{l1}, v_{l2}, \cdots, v_{lp}]^\top$

Thus, the l-th **PC score** of sample j is $PC_{lj} = x_j \mathbf{v}_l$

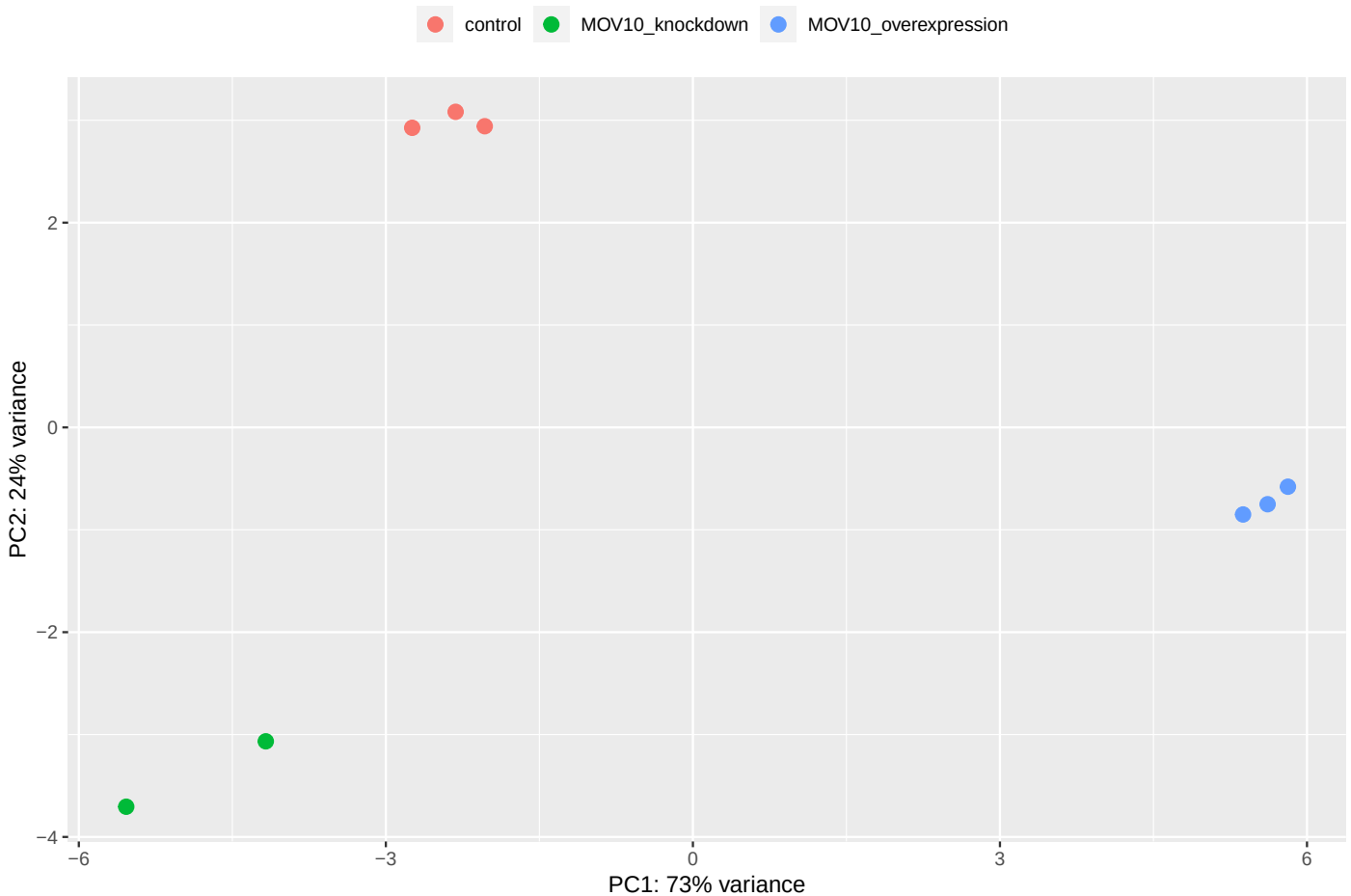
$\rightarrow$ the position of sample j in the new coordinate system of PCs

PCA: interpretation

The principal components represent new orthogonal variables obtained from the original dataset

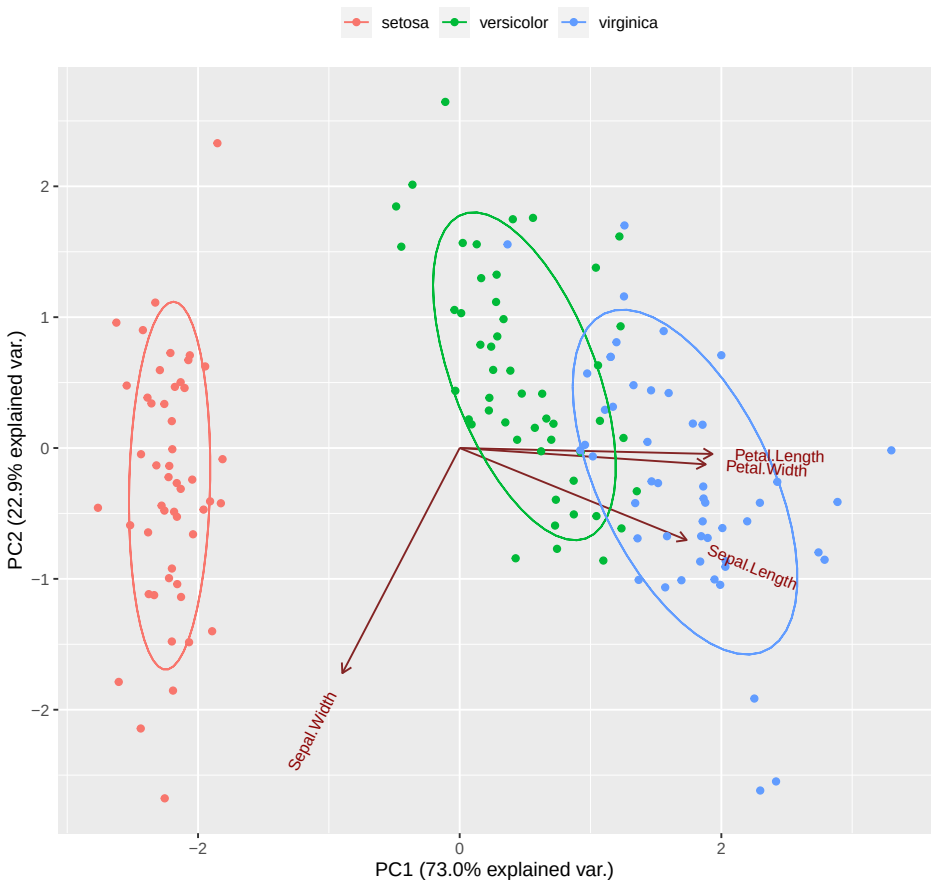
- the first principal component captures the maximum variance in the data
- subsequent principal components capture the remaining variance in decreasing order
- the size of the **loadings** v_{li} indicates how much the i-th gene contributes to the l-th PC
  $\rightarrow$ suppose that v_{1i}s are large for a certain class of genes but small for the others. Then, PC_1 can be interpreted as representing that class

Example


control
MOV10_knockdown
MOV10_overexpression
PC2: 24% variance
2
0
−2
−4
−6
−3
0
3
6
PC1: 73% variance

Example: mislabeling

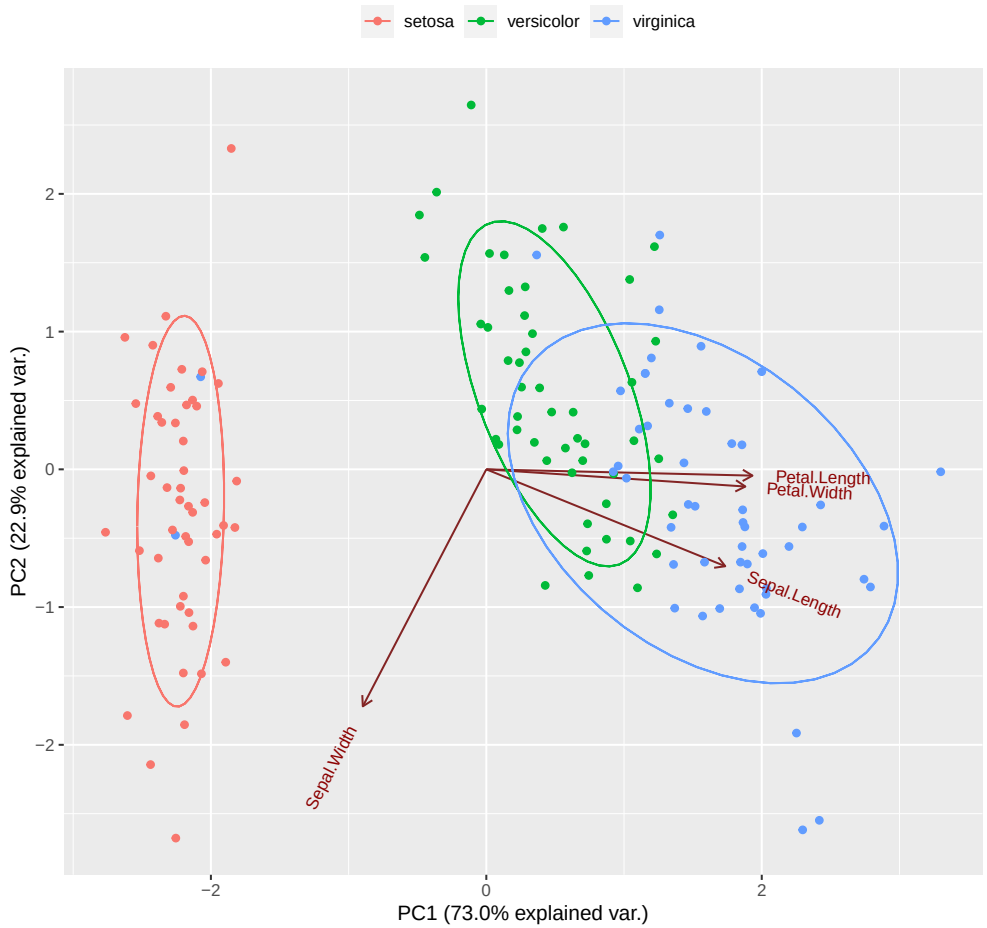
- Iris dataset: sepal length, sepal width, petal length, petal width
- biplot shows how strongly each feature influences a PC
  - positive loadings indicate that a feature and a PC are positively correlated whereas negative loadings indicate a negative correlation
  - large loading indicates that a feature has a strong effect on that PC
- biplot: angles between vectors reflect their correlation



→ PC1 represents having smaller sepal widths and larger sepal lengths
→ PC2 refers to having lower sepal measurements

Example: mislabeling

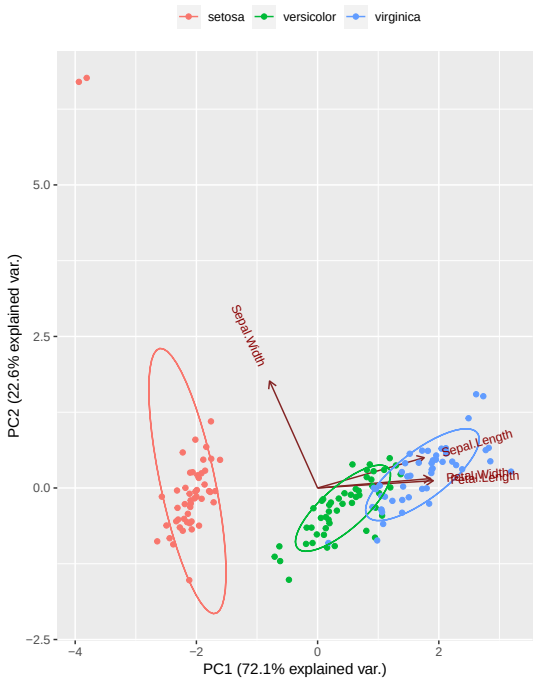
What went wrong?

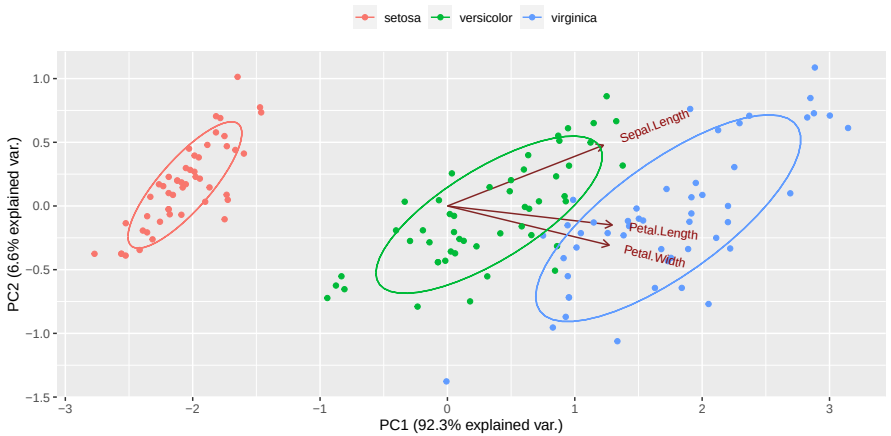


→ check the meta data

Example: outlier?

What went wrong?





$\rightarrow$ throw one feature instead of removing a sample

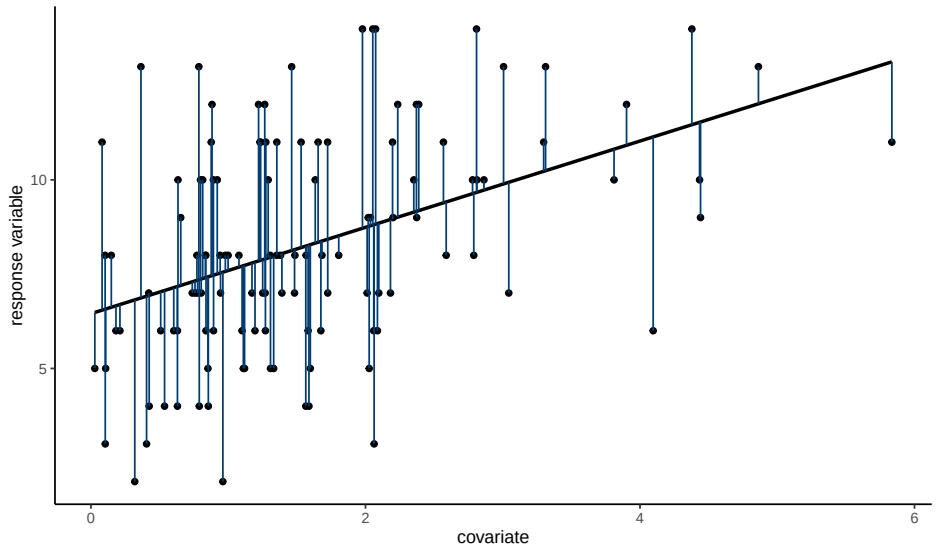
1. Normalization of raw gene counts
2. Principal component analysis
3. GLM for overdispersed counts
4. Modelling of RNA-seq data
5. Likelihood and Bayesian inference

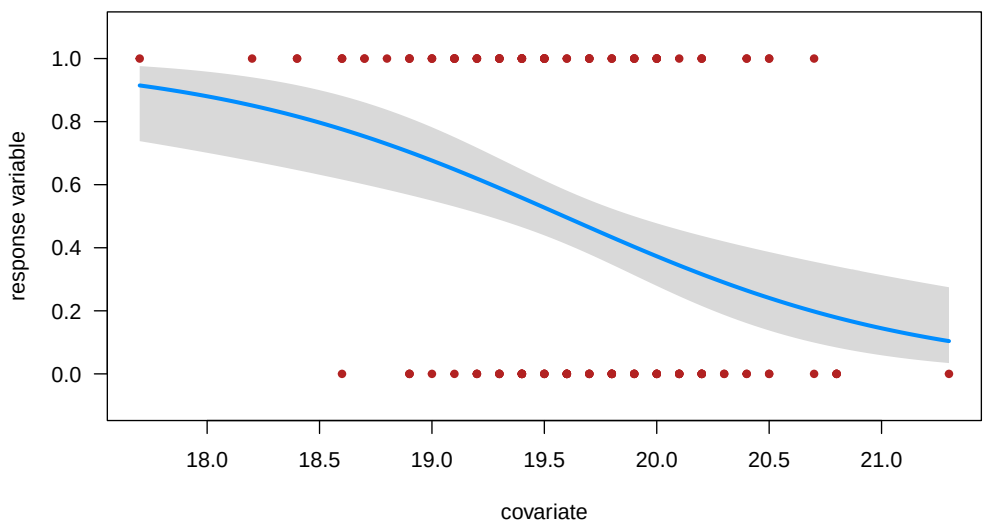
Generalized Linear Model: Introduction

- Linear models are only suitable for data that are (approximately) normally distributed
- However, there are many settings where we may wish to analyse a **response variable** which is not necessarily continuous, including when
  - Y is binary
  - Y is a **count** variable
  - Y is continuous, but non-negative

Generalized linear models (GLMs) extend the linear regression model to handle various types of response variables and non-normal error distributions

Generalized Linear Model: Introduction





Common GLM Families

GLMs can accommodate different types of response variables through specific distributions and link functions. Some common families include:

- **Gaussian Family**: Used for continuous responses
- **Binomial Family**: Used for binary or categorical responses
- **Poisson Family**: Used for count data (discrete and positive)
- **Gamma Family**: Used for positively skewed continuous responses

GLMs combine a model for the **conditional mean** with a **distribution for the response variable** and a **link function** tying predictors and parameters

$\rightarrow$ linear regression (with normal errors) is a special case of a generalized linear model

Families for count data

The Poisson distribution describes the number of events occurring in a given time interval

$\rightarrow$ event: read alignment to gene A: 1 count

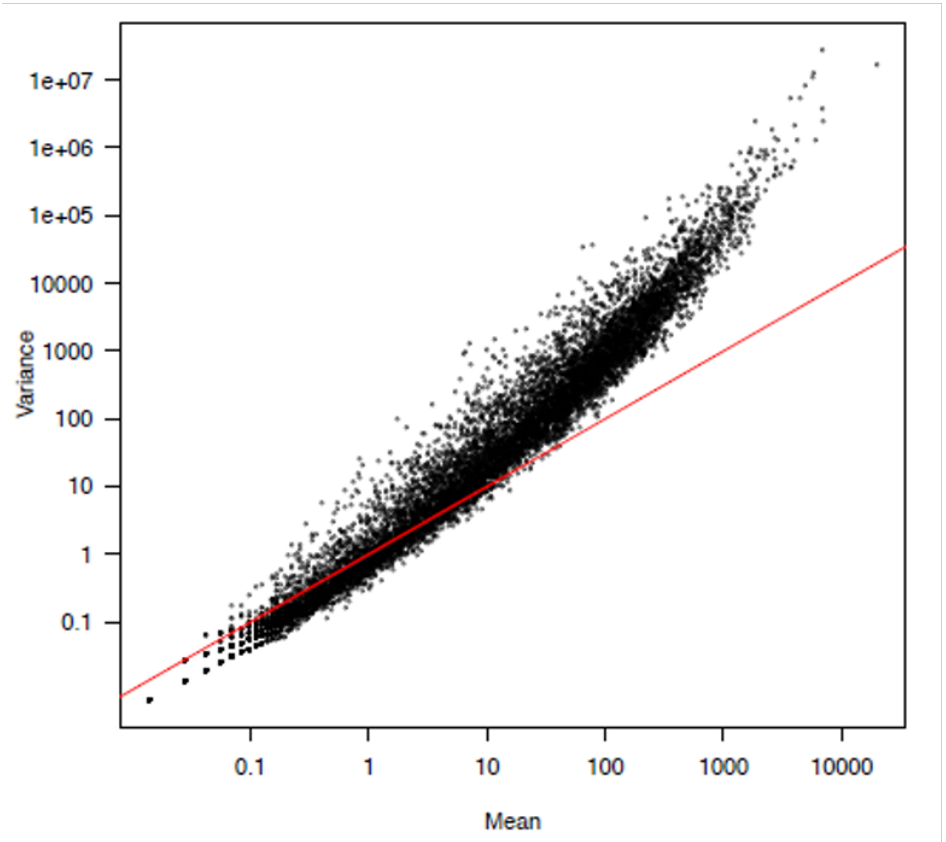
- the probability mass function is

$$\Pr(Y = y) = \frac{\mu^y}{\Gamma(y+1)} e^{-\mu}, \quad y = 0, 1, 2, \ldots$$

- the parameter μ of the Poisson distribution characterizes both its mean and variance, meaning $\mathrm{E(Y)} = \mathrm{Var(Y)} = \mu$

Families for count data

When variability in counts is much larger than the mean, a phenomenon termed **overdispersion**, the Poisson distribution is no longer appropriate



$\rightarrow$ the **negative binomial** model is often used as replacement for overdispersed count data

Negative binomial distribution

- the negative binomial distribution is a probability distribution for integer random variables with two parameters
- we restrict attention to the most common parametrization used in modelling. The probability mass function is

$$\Pr(Y = y) = \frac{\Gamma(y + 1/\alpha)}{\Gamma(y+1)\Gamma(1/\alpha)} \left(\frac{1/\alpha}{1/\alpha + \mu}\right)^{1/\alpha} \left(\frac{\mu}{1/\alpha + \mu}\right)^{y}$$

  for $y = 0, 1, 2, \ldots$, where Γ denotes the gamma function. Both parameters are positive, i.e., $\mu > 0$ and the **dispersion** $\alpha > 0$
- the mean and the variance are

$$\mathrm{E(Y)} = \mu \qquad \mathrm{Var(Y)} = \mu + \alpha\mu^2$$

- the variance of the negative binomial distribution is always **larger** than its mean
- we denote $Y \sim NB(\mu, \alpha)$

Notation for generalized linear models

- the starting point is the same as for linear regression:
  - we have a random sample of independent observations

$$(Y_i, \mathrm{X}_{i1}, \ldots, \mathrm{X}_{ip}), \quad i = 1, \ldots, n$$

    where Y is the response variable and $\mathrm{X}_1, \ldots, \mathrm{X}_p$ are p explanatory variables or covariates which are assumed fixed (non-random)
  - the goal is to model the response variable as a function of the explanatory variables
- let μ_i denote the (conditional) **mean** of Y_i given covariates,

$$\mu_i = \mathrm{E}(\mathrm{Y_i} \mid \mathrm{X_{i1}}, \ldots, \mathrm{X_{ip}})$$

- let η_i denote the linear combination of the covariates that will be used to model the response variable

$$\eta_i = \beta_0 + \beta_1 \mathrm{X}_{i1} + \cdots + \beta_p \mathrm{X}_{ip}$$

Definition of generalized linear model

- there are three building blocks to the generalized linear model:
  - a probability distribution for the outcome Y that is a member of the exponential family (normal, binomial, Poisson, gamma, inverse Gaussian, ...)
  - a linear predictor $\boldsymbol{\eta} = \mathbf{X}\beta$
  - a function g, called **link function**, that links the mean of Y_i to the predictor variables, $g(\mu_i) = \eta_i$
- the link between the mean of Y and the regression "line" is

$$g\left\{\mathrm{E}(\mathrm{Y} \mid \mathrm{X}_1, \ldots, \mathrm{X}_\mathrm{p})\right\} = \beta_0 + \beta_1 \mathrm{X}_1 + \cdots + \beta_p \mathrm{X}_p$$

Negative binomial regression

- negative binomial regression assumes that the response variable Y follows a **negative binomial** distribution and that the **link function** is the logarithmic function

$$g\{\mathrm{E}(\mathrm{Y_i})\} = \log\{\mathrm{E}(\mathrm{Y_i})\} = \beta_0 + \beta_1 \mathrm{X_{i1}} + \ldots + \beta_\mathrm{p} \mathrm{X_{il}}.$$

- the **dispersion** parameter α, is assumed to be **the same for every observation** and therefore doesn't depend on the predictor variables

1. Normalization of raw gene counts
2. Principal component analysis
3. GLM for overdispersed counts
4. Modelling of RNA-seq data
5. Likelihood and Bayesian inference

RNA-seq modelling

Let $\mathbf{K}$ be the matrix of **raw counts** with one row for each gene i and one column for each sample j

$\rightarrow K_{ij}$ is the number of reads aligned to gene i in sample j

For each gene i, we assume

$$K_{ij} \sim NB(s_j q_{ij}, \alpha_i),$$

where

- s_j is the normalization constant in sample j
- q_{ij} is the **normalized count** of gene i in sample j
- α_i is the dispersion of gene i

$\rightarrow \mu_{ij} = s_j q_{ij}$ is the mean number of aligned reads for gene i in sample j

The normalized count q_{ij} is modelled by a GLM with logarithmic link

$$\log_2(q_{ij}) = \sum_r x_{jr} \beta_{\mathbf{ir}}, \tag{1}$$

where $\mathbf{X} = (x_{kl})_{kl}$ is the design matrix

RNA-seq modelling

- Design matrix
  Analyse expression differences between control and treatment, knowing that it should depend on the sex and age

	sex	age	treatment
sample 1	M	11	Ctrl
sample 2	M	10	Ctrl
sample 3	M	13	Treatment
sample 4	M	13	Treatment
sample 5	F	11	Ctrl
sample 6	F	11	Ctrl
sample 7	F	13	Treatment
sample 8	F	13	Treatment

RNA-seq modelling

Coefficients β_{ir} are the **$\log_2$ fold changes (LFC)** for gene *i* in each sample group: the effect of a covariate on gene expression levels

Example: When comparing two groups, β_{i1} is the LFC for gene *i* between treatment and control
$\rightarrow$ A LFC=1.2 for a specific gene in the comparison treatment vs control means that the expression of that gene is increased in the treatment group relative to the control group by a multiplicative factor of $2^{1.2} \approx 2.297$
$\rightarrow$ A positive value indicates an increase in expression, while a negative value indicates a decrease in expression

RNA-seq modelling: Estimation of dispersion

- few biological replicates → high variability/uncertainty of estimates
- but, large number of genes!
  → borrow/pool information across genes
  → genes with similar average expression level have similar dispersion

Thus, we use an **empirical Bayes** inference method to **shrink** individual dispersion estimates towards a global dispersion

How it works:

- low dispersion estimates are shrunken (up) towards the global mean
- high dispersion estimates are shrunken (down) towards the global mean
- genes with extremely high dispersion values are not shrunken. They are considered outliers to the model assumptions

→ reduces estimation uncertainty by pooling information
→ increases power of statistical tests by reducing false positive calls

Empirical Bayes shrinkage of dispersion

Key steps

1. fit the gene-specific GLM
   - get initial estimates of the mean read counts $\hat{\mu}_{ij}^0$
   - get estimates $\hat{\alpha}_i^{gw}$ of gene-wise dispersion
2. obtain a global dispersion trend α_{tr} by regressing $\hat{\alpha}_i^{gw}$ onto the mean counts $\bar{\mu}_i = \sum_j K_{ij}/n$, where $n = \#\text{samples}$

$$\alpha_{tr}(\bar{\mu}) = \frac{a_1}{\bar{\mu}} + \alpha_0$$

$\rightarrow$ similar average expression strengths yield similar dispersions

Empirical Bayes shrinkage of dispersion

3. set a log-normal prior distribution for α_i

$$\log(\alpha_i) \sim \mathcal{N}(\log(\alpha_{tr}(\bar{\mu}_i)), \sigma_d^2)$$

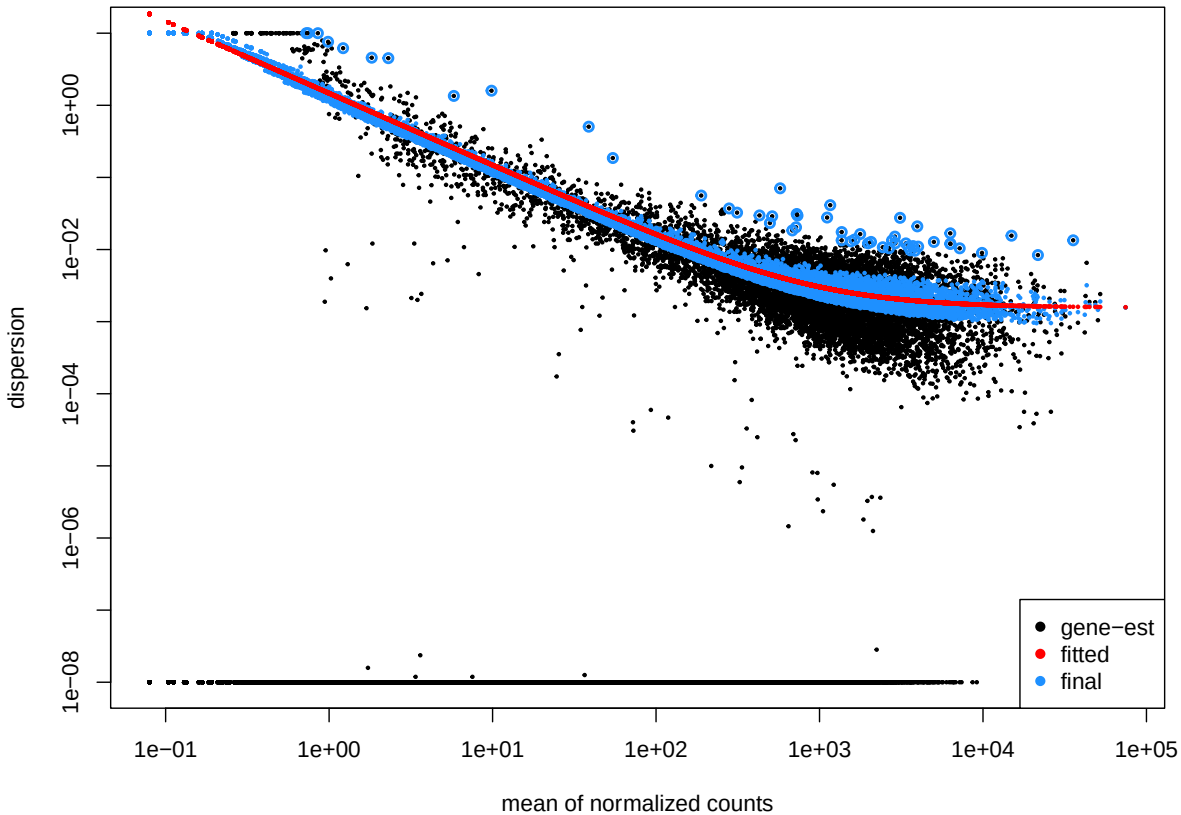
$\rightarrow$ the width σ_d^2 (same for all genes) plays a role in the amount of shrinkage and is estimated from the data (large sample size results in a large width and less shrinkage)

4. get final Maximum A Posteriori estimate $\hat{\alpha}_i^{MAP}$

$$\hat{\alpha}_i^{MAP} = \arg\max_{\alpha}\{\ell_{NB}(\mathbf{K}_{i\cdot}; \boldsymbol{\mu}_{i\cdot}^0, \alpha) + \Lambda_i^{prior}(\alpha)\}, \quad \text{where}$$

$$\Lambda_i(\alpha)^{prior} = \frac{-\{\log(\alpha) - \log(\alpha_{tr}(\bar{\mu}_i)\}^2}{2\sigma_d^2} \text{ is the log density of the prior}$$

Empirical Bayes shrinkage of dispersion



- data scatter around the global dispersion trend
- dispersion decreases with increasing mean expression levels

RNA-seq modelling: Estimation of fold change

Issues

- LFCs for genes with low read counts are highly variable (consequence of taking ratios wrt small values)
- hypothesis testing (and interpretation of results) heavily depends on the variability of the LFC
  $\rightarrow$ the higher the variability the smaller the effect size and the harder is to detect a difference

Solution

When the information for a gene is low (small sample size, low read counts, high dispersion), its LFC is shrunk towards zero

- avoids large absolute values of LFCs for weakly expressed genes
- induces a bias towards zero when the dispersion/variability is large
- with increasing sample size, less shrinkage is applied

$\rightarrow$ quantitative conclusions, e.g., testing and ranking genes, are more reliable

RNA-seq modelling: Shrinkage of fold change

Key steps: estimation and shrinkage of β_{ir}

1. the negative binomial GLM with link (1) is fitted by IRLS $\rightarrow \hat{\beta}_{ir}$
2. for each column of the design matrix (except the first for the intercept), set a prior distribution reflecting the bias towards zero
   - set a normal prior
   $$\beta_{ir} \sim \mathcal{N}(0, \sigma_r^2), \quad r > 0$$
   - estimate the **width** from the empirical variance of $\hat{\beta}_r$ (averaged over all possible contrasts) $\rightarrow$ **amount of shrinkage**

RNA-seq modelling: Shrinkage of fold change

3. get final Maximum A Posteriori estimate $\hat{\beta}_i^{MAP}$

$$\begin{aligned}\hat{\beta}_i^{MAP} &= \arg\max_{\beta}\left\{\sum_j \log f_{NB}(K_{ij};\mu_j(\beta),\hat{\alpha}_i^{MAP}) + \Lambda^{prior}(\beta)\right\} \\ &= \arg\max_{\beta}\left\{\sum_j \log f_{NB}(K_{ij};\mu_j(\beta),\hat{\alpha}_i^{MAP}) - \sum_r \frac{\beta_r^2}{2\sigma_r^2}\right\},\end{aligned}$$

where $\mu_j(\beta) = s_j e^{\sum_r x_{jr}\beta_r}$ is the mean of the negative binomial fitted to the raw counts

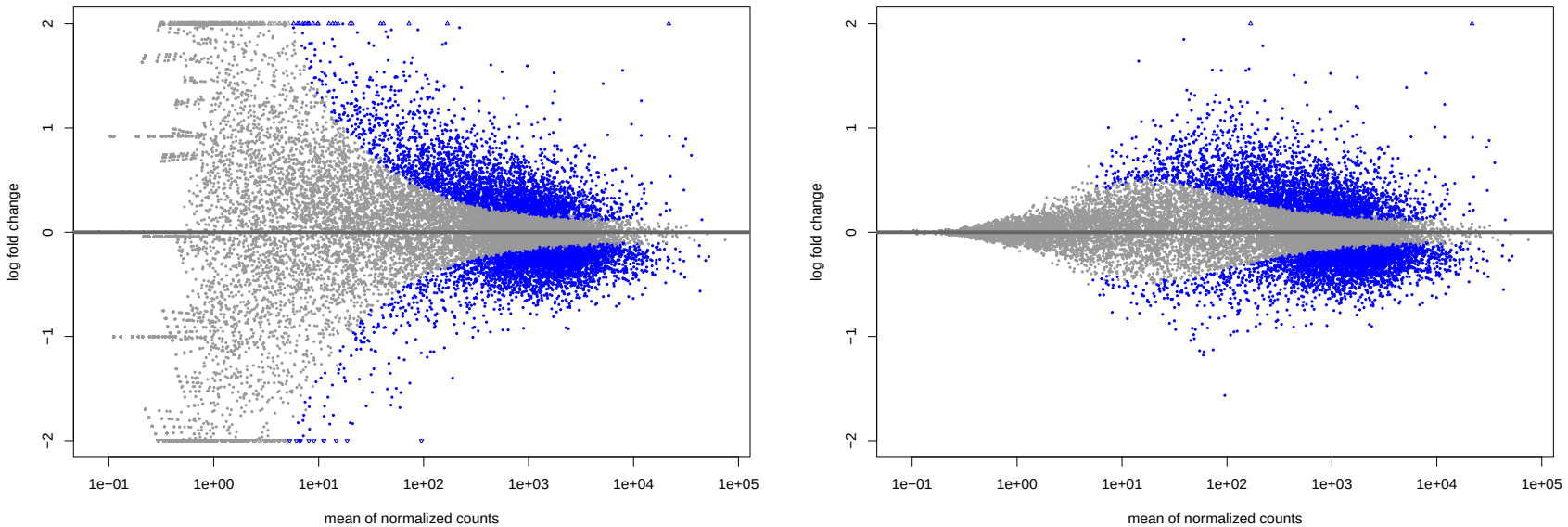
$\rightarrow$ if Step 1 yields a high-uncertainty estimate $\hat{\beta}_{ir}$ due to a flat NB likelihood (small mean, high dispersion, or few samples), then the MAP is pulled closer to zero

Representation of results

The MA plot shows the LFCs between two conditions ("M" values) versus the average of the normalized counts over the two conditions ("A" values) for all tested genes

- illustrates the effect of LFC shrinkage
- identifies genes that exhibit significant differences
  $\rightarrow$ genes with significant upregulation or downregulation correspond to points that deviate significantly from the center line ($M = 0$)
  $\rightarrow$ genes that are not differentially expressed in both conditions tend to cluster around the center line
  $\rightarrow$ genes that are significantly DE are colored to be easily identified

MA plot



$\rightarrow$ genes with similar mean counts might not have the same shrinkage

References

Thank you!

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*, 11(10):R106.

Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for rna-seq data with DESeq2. *Genome Biology*, 15(12):550.

1. Normalization of raw gene counts
2. Principal component analysis
3. GLM for overdispersed counts
4. Modelling of RNA-seq data
5. Likelihood and Bayesian inference

Introduction to likelihood inference

- suppose we want to estimate the probability that an event occurs, which we assume is constant
- suppose that we have a sample of size n with X_i assumed to come from a Bernoulli distribution with probability p, meaning

$$\Pr(X_i = 1) = p, \qquad \Pr(X_i = 0) = 1 - p$$

- by convention, "1" denotes a success and "0" a failure

The probability mass function is

$$\Pr(X_i = x_i \mid p) = p^{x_i}(1-p)^{1-x_i}, \qquad x_i \in \{0, 1\}$$

Since the observations are independent, the joint probability of a given result is the product of the probabilities for each observation,

$$\begin{aligned}\Pr(X_1 = x_1, \ldots, X_n = x_n \mid p) &= \prod_{i=1}^{n} \Pr(X_i = x_i \mid p) \\ &= \prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i}\end{aligned}$$

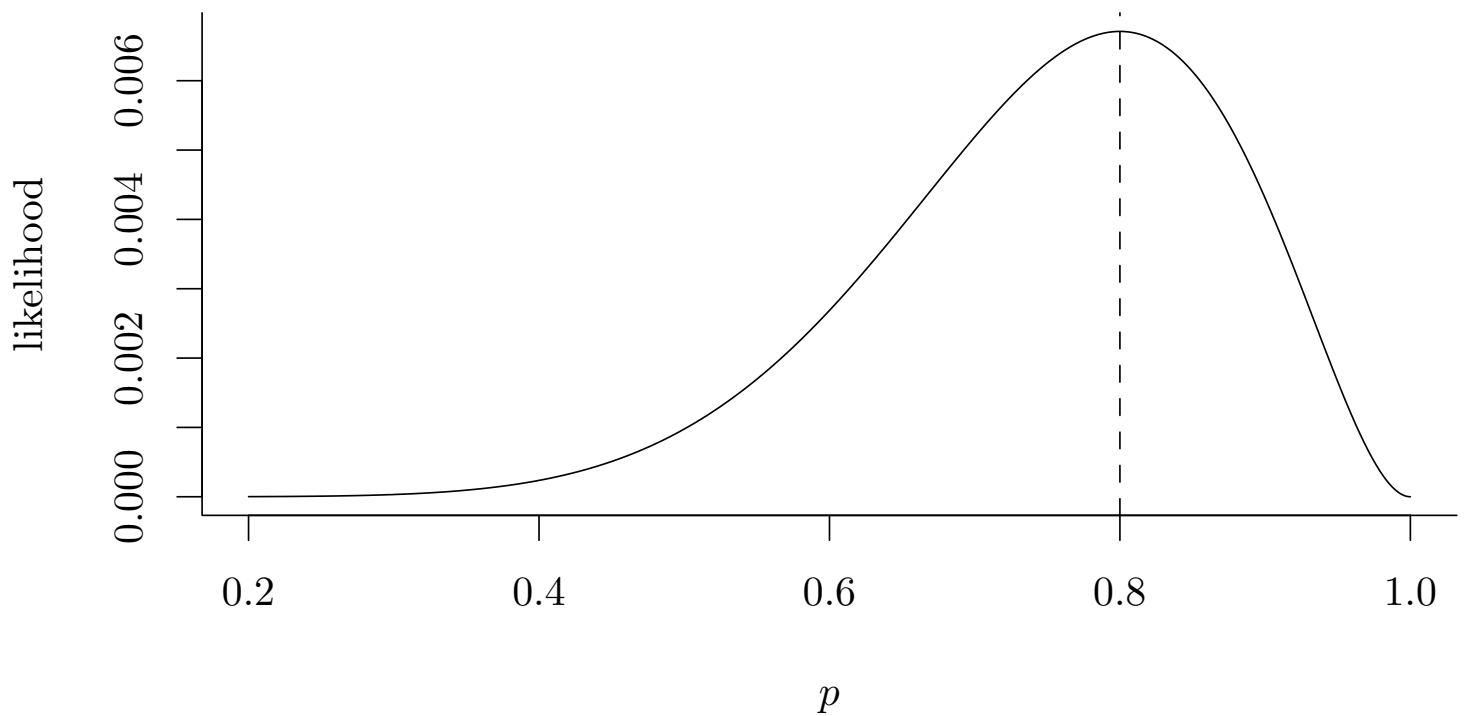
Likelihood

- the **likelihood** $L(\boldsymbol{\theta})$ is a function of the **parameters** of the distribution, say $\boldsymbol{\theta}$
  - the likelihood gives the probability of observing a sample under a postulated distribution whose parameters are $\boldsymbol{\theta}$
  - the likelihood treats the observations as fixed
- the **maximum likelihood** estimator $\hat{\boldsymbol{\theta}}$ is the value of $\boldsymbol{\theta}$ that maximizes the likelihood, i.e., the value that makes the observed sample the most **likely** or **plausible**

Likelihood of the Bernoulli model

The likelihood for a random sample is

$$L(p; \boldsymbol{X}) \equiv \Pr(\boldsymbol{X}|p) = \prod_{i=1}^{n} p^{X_i}(1-p)^{(1-X_i)}$$
$$= p^{\Sigma_{i=1}^{n} X_i}(1-p)^{n-\Sigma_{i=1}^{n} X_i}$$



Bayesian inference

Bayesian methods trace its origin to Thomas Bayes who, along with Pierre-Simon Laplace, discovered the so-called **Bayes' theorem**

- $\Pr(x|\theta)$ likelihood
- $\Pr(\theta)$ prior (the unknown parameter is now a random variable distributed according to our prior knowledge)
- $\Pr(\theta|x)$ posterior (belief after observing some data)
- $\Pr(x)$ marginal distribution

$$\Pr(\theta|x) = \frac{\Pr(\theta, x)}{\Pr(x)} = \frac{\Pr(x|\theta)\Pr(\theta)}{\Pr(x)} \propto \Pr(x|\theta)\Pr(\theta)$$

Back to the Bernoulli case

- Likelihood: $\Pr(\mathbf{x}_{1:n}|p) = p^{\sum_{i=1}^{n} x_i}(1-p)^{n-\sum_{i=1}^{n} x_i}$
- Beta prior $p \sim \text{Beta}(a, b)$:

$$\Pr(p) \propto p^{a-1}(1-p)^{b-1}$$

  on the interval $(0, 1)$
- Posterior of Bernoulli-Beta:

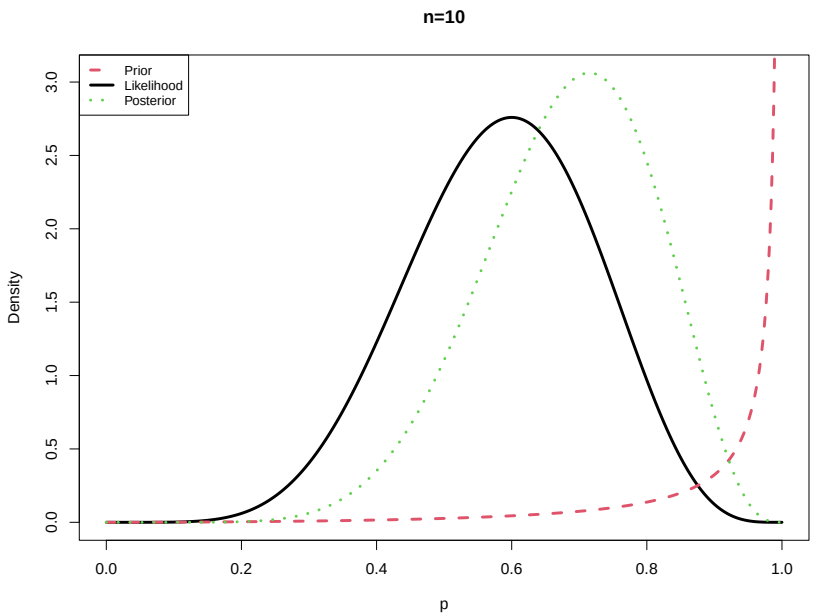
$$\Pr(p|\mathbf{x}_{1:n}) \propto \Pr(\mathbf{x}_{1:n}|p)\Pr(p)$$

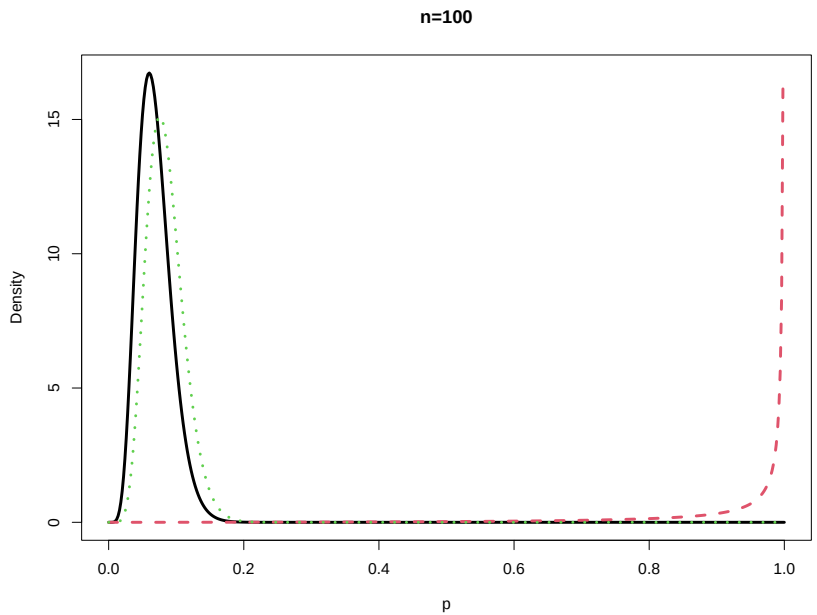
$\Rightarrow$ the maximum a posteriori (MAP) estimator is defined as

$$\hat{p}^{MAP} = \arg\max_p \Pr(p|\mathbf{x}_{1:n}) = \arg\max_p \Pr(\mathbf{x}_{1:n}|p)\Pr(p)$$

Effect of the prior

- as sample size increases, the effect of the prior is washed out





Effect of the prior

- flat priors (uniform, non-informative) have no effect