STATISTICAL METHODS IN BIOINFORMATICS

Analysis of RNA sequencing data

Stefan E Seemann
sse@sund.ku.dk

Center for non-coding RNA in Technology and Heath (http://rth.dk)
Dep. of Public Health, SUND, UCPH

20 November 2026

Programme

Nov 19	08:30 - 11:15	Stefan	Raw sequencing data to count tables
	08:30 - 10:00		*Lecture+Exercise:* Alignment methods
	10:15 - 10:45		*Lecture:* Speed up of local sequence alignments
	10:45 - 11:15		*Lecture:* From Raw Reads to Expression Matrix
	12:15 - 15:00	Nadya	STRING, biological networks, functional enrichment

Nov 20	08:30 - 11:15	Stefan	Unsupervised and supervised data exploration
	08:30 - 09:30		*Lecture:* Unsupervised+supervised data exploration
	09:45 - 11:15		*Exercise:* Expression analysis in R
	12:15 - 15:00	Nadya	Network analysis and visualization with Cytoscape

Unsupervised + supervised data exploration

Explore global and local read count patterns

1. **unsupervised**

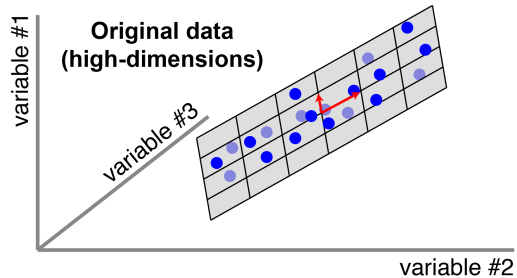
   no *a priori* information is needed
   → to detect technical noise and batch effects
   - Dimensional reduction → Principle Component Analysis (PCA)
   - Clustering → hierarchical, k-means

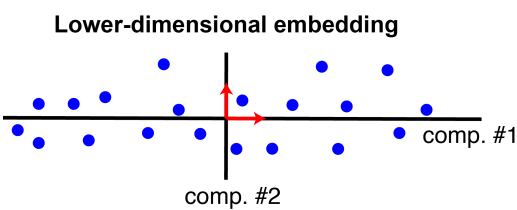
   PCA and clustering should be done on normalized and transformed read counts so that high variability of low read counts does not occlude potentially informative data trends.

2. **supervised**

   usage of known biological labels
   → differential expression

Dimensional reduction





- Maximizing variance in principal component space
- Minimizing least-squares reconstruction error

Williams *et al.* "Everything you did and didn't know about PCA" 2016

Principle component analysis – PCA

Input: $n \times p$ matrix X — n samples (or cells), p genes, normalized counts

What PCA does

Finds orthogonal directions (*principal components*) that explain maximal variance.

PCA: Maximization, SVD, and Reconstruction

1 · Maximization problem

Find c ($\|c\| = 1$) that maximises the variance of the projected data:

$$\max_{c} c^\top X^\top X c$$

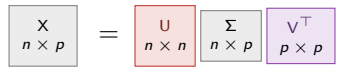
Lagrange multiplier $\rightarrow$ eigenvalue equation:

$$X^\top X c = \lambda c$$

- c = eigenvector of $X^\top X$
- λ = variance explained
- $c^\top X^\top X c = \lambda$
- PCs are orthogonal: each maximises remaining

2 · SVD delivers the solution

$$X = U \Sigma V^\top$$



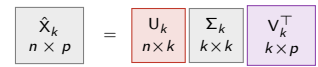
Plug into $X^\top X$:

$$X^\top X = V \underbrace{(\Sigma^\top \Sigma)}_{\text{diag}(\sigma_i^2)} V^\top$$

SVD	PCA
Cols of V	Gene **loadings** (c's)
σ_i^2	Eigenvalues λ_i (var. explained)
$U\Sigma = XV$	Sample **scores**

3 · Rank-k Reconstruction

Truncate to k PCs:



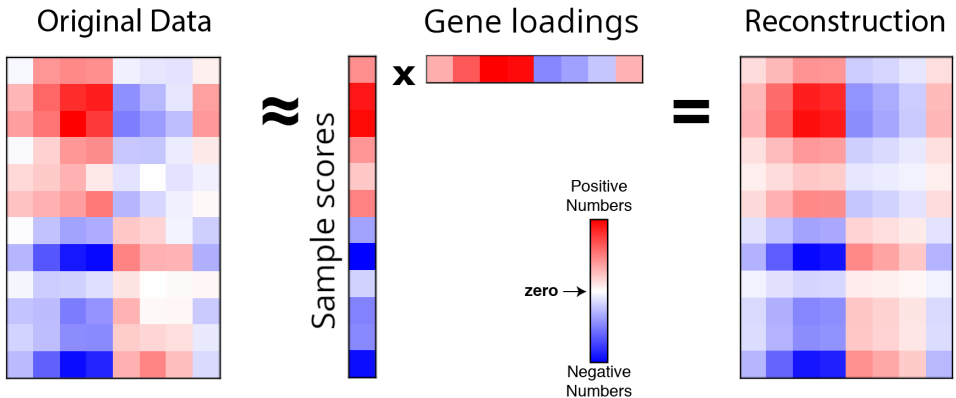
Equivalent forms:

$$\hat{X}_k = \sum_{i=1}^{k} \sigma_i \, u_i \, v_i^\top$$

$$= \underbrace{(XV_k)}_{\text{scores}} \underbrace{V_k^\top}_{\text{loadings}^\top}$$

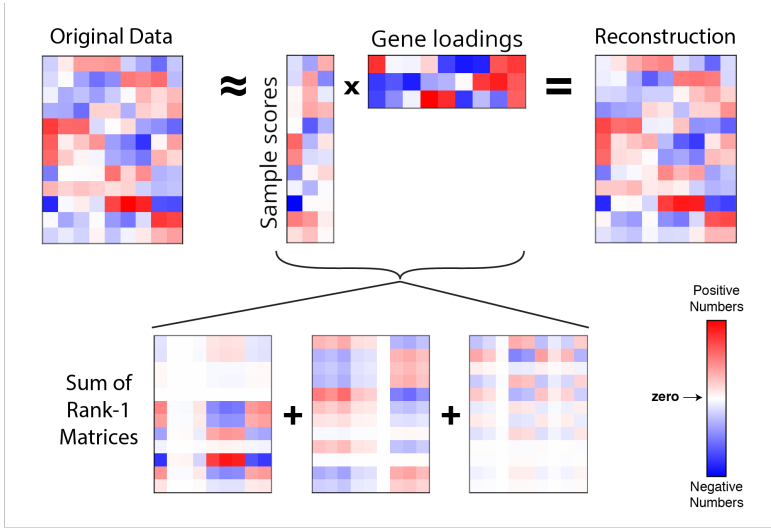
- Each term $\sigma_i u_i v_i^\top$: one rank-1 "gene program"
- Residual $X - \hat{X}_k$: noise not captured

PCA rank-1 reconstruction



Williams *et al.* "Everything you did and didn't know about PCA" 2016

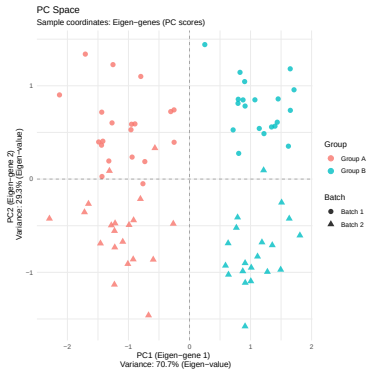
PCA rank-3 reconstruction



Williams *et al.* "Everything you did and didn't know about PCA" 2016

Principle component analysis – PCA

- transforms measurements into new variables that are truly independent
- new variables of most variance are the principal components

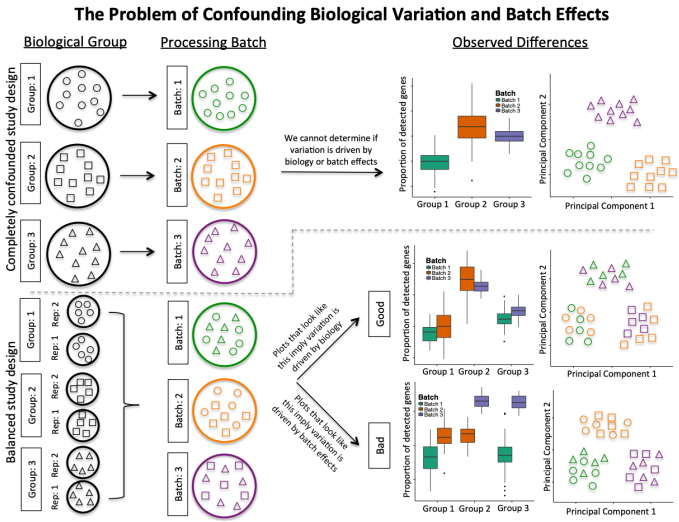


Applications:

- visualization of your data in lower dimensions (2D, 3D)
- find patterns in numeric data
- identify batch effects or other possible covariates (*e.g.* male and female) by labeling them
- **Interpreting loadings:** plot the top genes per PC to identify biological programs

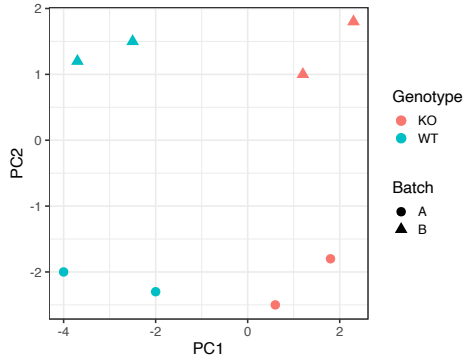
Pradhan *et al.* Behav Brain Funct 2015

PCA to detect batch effects



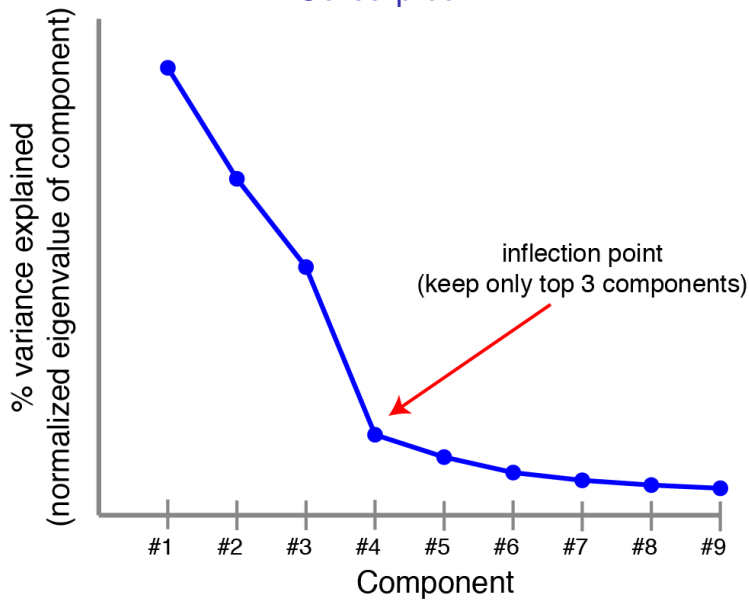
Hicks *et al.* "On the widespread and critical impact of systematic bias and batch effects in single-cell RNA-Seq data" bioRxiv. 2015

PCA to detect batch effects



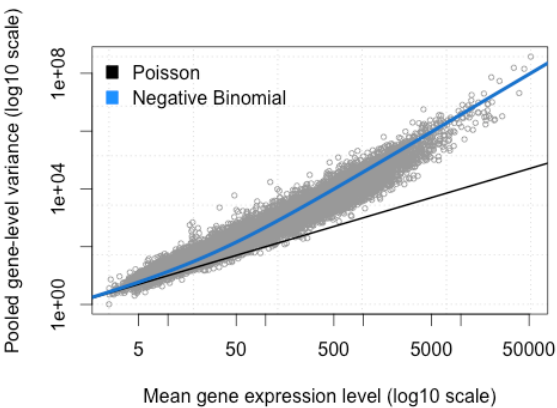
- PC1 separates the *genotype* (group of interest); however
- PC2 separates the *batch* effect (or other covariates)
  - *e.g.* experiment date, sex, experimenter, different RNA isolation kit (you name it!)
- When batch effect is observed in PCA plots, add it as covariate to your GLM
  - $\sim$ Batch + Genotype

Scree plot


% variance explained
(normalized eigenvalue of component)
inflection point
(keep only top 3 components)
#1
#2
#3
#4
#5
#6
#7
#8
#9
Component

The mean–variance problem

RNA-seq counts are *overdispersed*: genes with higher mean expression tend to have higher variance — even if they carry no biological signal.



Consequence for PCA

Highly expressed housekeeping genes dominate the first PC purely because of their count magnitude, not their biological variation.

Solution: first select genes whose variance exceeds what is expected from their mean expression – the **highly variable genes (HVGs)**.

Highly Variable Gene (HVG) Selection

Core idea

Fit the expected variance as a function of the mean. Genes whose *observed* variance exceeds the expectation carry biological signal; the rest are mostly noise.

Bulk RNA-seq (DESeq2 / edgeR)

1. Variance-stabilizing transformation (VST or rlog)
2. Fit loess of $\log(\hat{\sigma}^2)$ vs. $\log(\bar{\mu})$ and select genes above the curve
   OR
   Select genes with largest variance of variance-stabilized counts

scRNA-seq (Seurat / scanpy)

1. Normalize to equal library size
2. Fit loess of $\log(\hat{\sigma}^2)$ vs. $\log(\hat{\mu})$ across genes
3. Rank genes by standardized dispersion; keep top N (commonly $N = 2\,000$–$5\,000$)

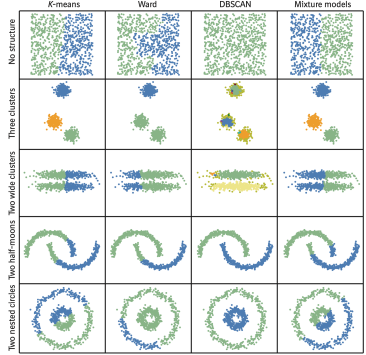
Key point

HVG selection removes genes that are **noisy but uninformative** – not necessarily lowly expressed ones. Housekeeping genes with high but stable expression are correctly excluded.

Clustering

There are several clustering algorithms and more are being developed. Why?

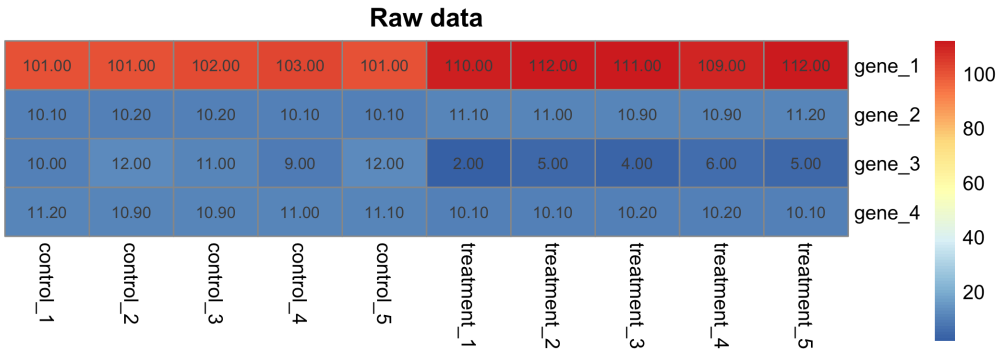
- No clustering algorithm is perfect
- Remember! Always "see" your data and judge if the clusters make sense
- The performance of the clustering algorithm depends on the structure of your data



Ronan *et al.* "Avoiding common pitfalls when clustering biological data" Science Signalling 2016

Finding patterns in your data

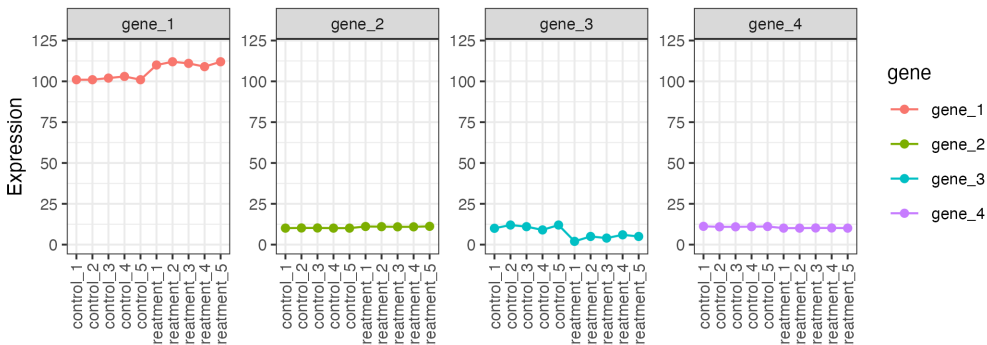
Let's assume that you got these 4 genes differentially expressed in your dataset



Can you identify a pattern?

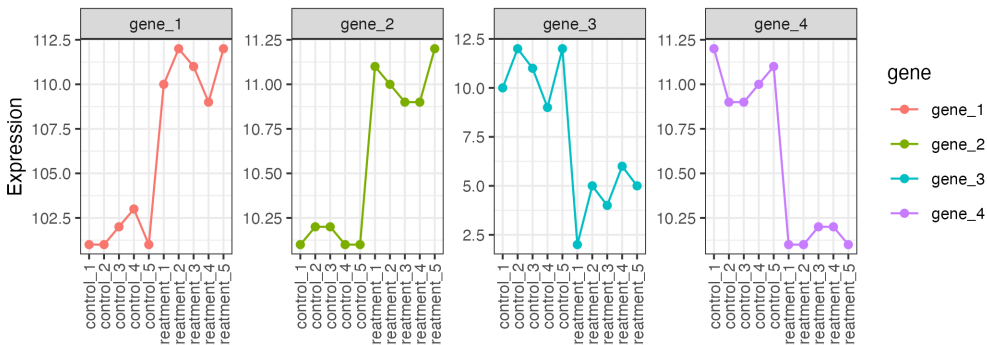
Finding patterns in your data

upload exercise for Statistical Methods in Bioinformatics What about now?



Finding patterns in your data

Perhaps this gives you a better hint

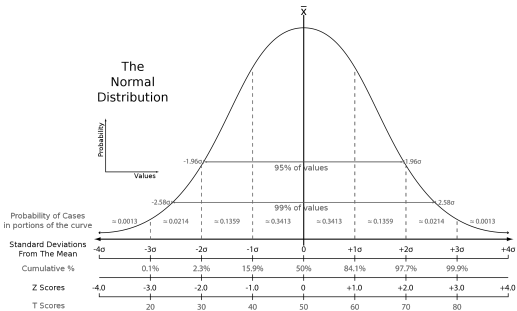


Z-score

$$z = \frac{x-\mu}{\sigma}$$

μ = mean

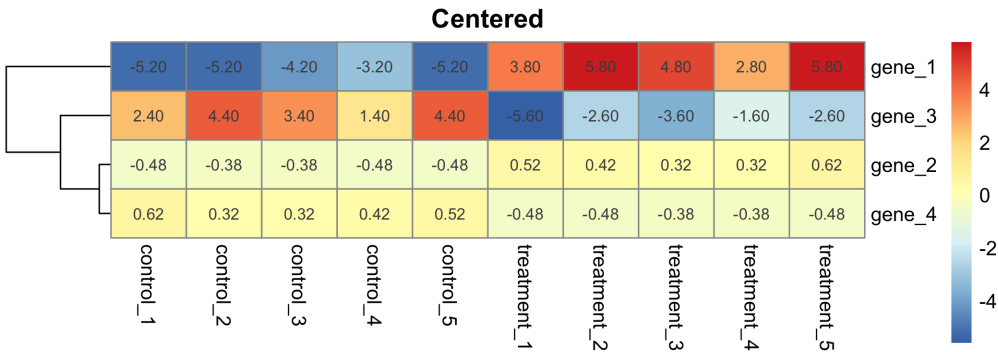
σ = standard deviation



https://en.wikipedia.org/wiki/Standard_score

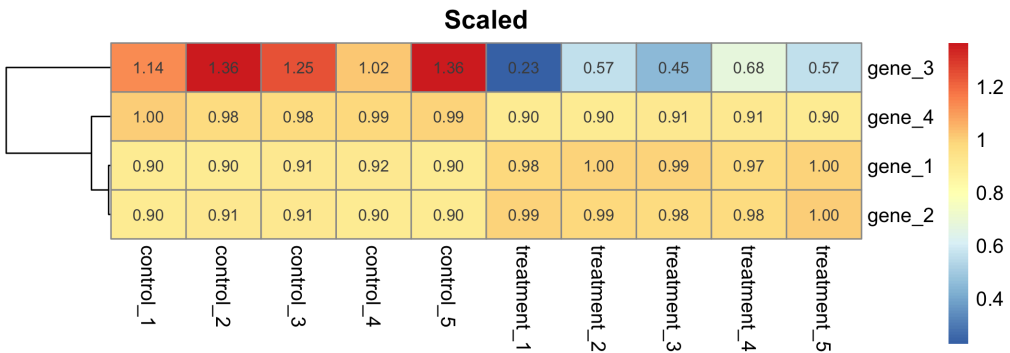
Centering data

$$x - \mu$$



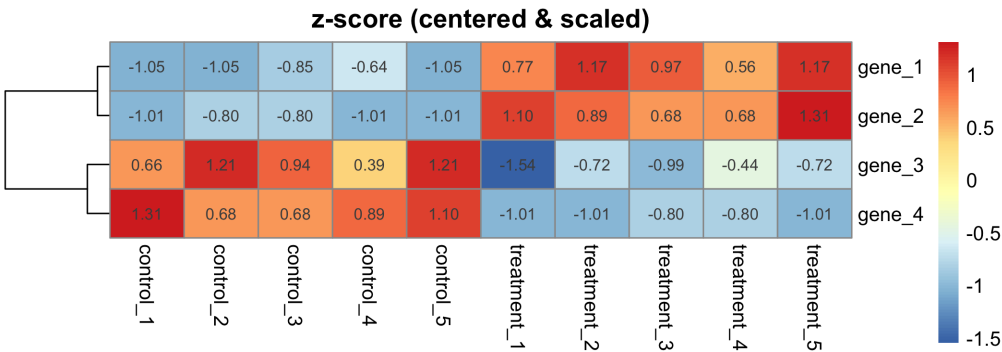
Scaling data

$\frac{x}{\sigma}$



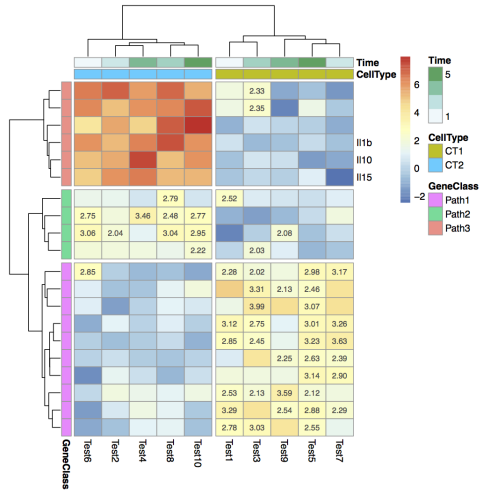
Z-score

$$z = \frac{x-\mu}{\sigma}$$



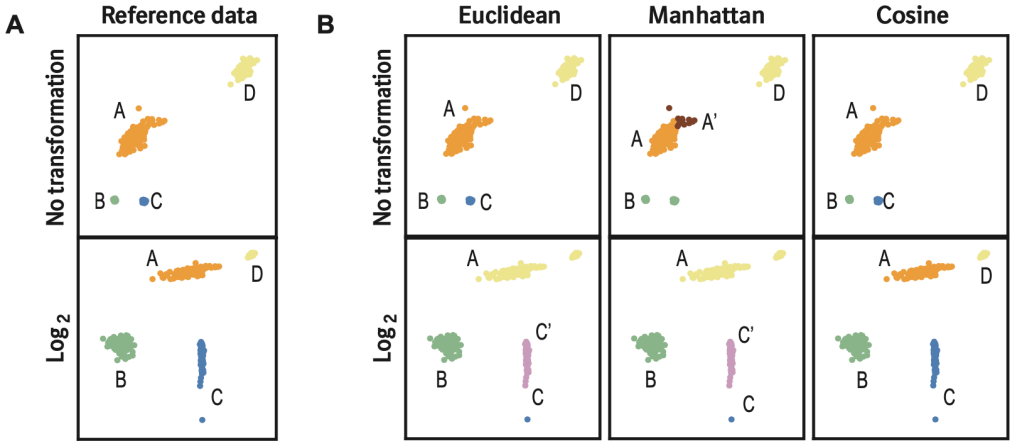
Hierarchical clustering

- No need for pre-defining the number of clusters
- Results dependent on
  1. *distance metric:* euclidean, manhattan, Pearson correlation etc.
  2. *clustering method:* Ward, complete, average etc.
- Used with heatmaps, thus allows better visual inspection of data



Effect of transformations and distance metrics on clustering

- use normalized read counts (rlog, CPM etc.)
- transform your data for better scaling (log2, z-score transformation etc.)



Ronan *et al.* "Avoiding common pitfalls when clustering biological data" Science Signalling 2016

The Compositional Problem in RNA-seq

Raw counts

	Ctrl	Trt	Ratio
G1	10	30	3.0×
G2	30	90	3.0×
G3	5	15	3.0×
G4	1	3	3.0×
G5	**1000**	**240**	0.24×
Total	*1046*	*378*	

G5 = 95.6 % of control library!

Library totals differ 2.8× — *entirely driven by G5.*

Library-size normalization

Divide by sample total ($\times 10^3$):

	Ctrl	Trt
G1	9.6	79.4
G2	28.7	238.1
G3	4.8	39.7
G4	1.0	7.9
G5	956.0	634.9

G1–G4: 8.3× ↑
G5: 0.66× ↓

⇒ G5's dropout frees library capacity, inflating G1–G4 by an artefactual 2.8×.

DESeq2 median-of-ratios

Per-gene ratio to geometric mean:

	Ctrl ratio	Trt ratio
G1	0.577	1.732
G2	0.577	1.732
G3	0.577	1.732
G4	0.577	1.732
G5	*2.04**	*0.49**
Median	**0.577**	**1.732**

*outlier, excluded by median

G1–G4: 1.0× **(stable)**
G5: 12.5× ↓

⇒ G5 is correctly identified as the changing gene; G1–G4 are correctly stable.

Key insight

Library-size normalization is dominated by G5's dropout, inflating G1–G4 by a **spurious** 2.8×. Robust methods (TMM, DESeq2 RLE) use the *median* fold change across genes.

Differential gene expression

Common scientific question:
Quantification and statistical inference of systematic changes between conditions.

Differential gene expression

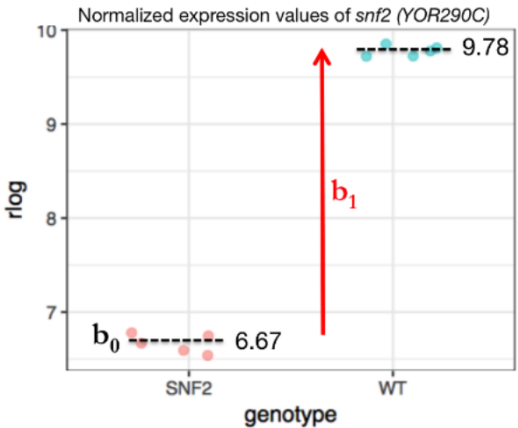
Common scientific question:
Quantification and statistical inference of systematic changes between conditions.

Principles are the same as for all other significance tests:

1. Use the replicates (samples of the same conditions) to estimate the **within-condition variability** (variance) of the expression
2. Use the expression and variance to test whether the difference between conditions is random or not

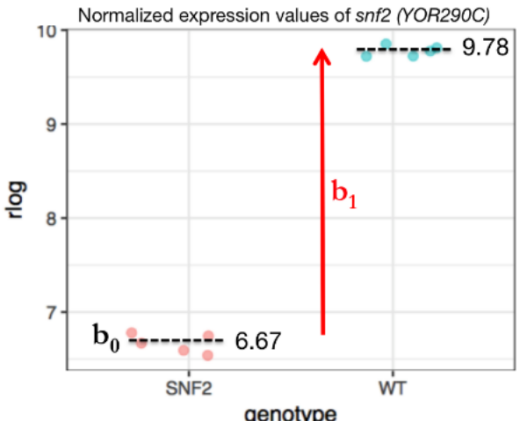
The test's statistical power increases with more biological (and technical) replicates!

Interpretation of the negative binomial GLM



Linear regression model (LM) is evaluated for every gene: $Y = b_0 + b_1 * x_1 + e$

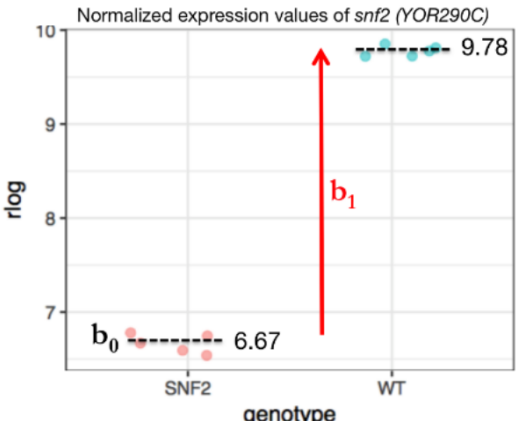
Interpretation of the negative binomial GLM



Linear regression model (LM) is evaluated for every gene: $Y = b_0 + b_1 * x_1 + e$

Y . . . describes all read counts for a gene
b_0 . . . average of baseline group,
e.g., control
x_1 . . . design factor,
e.g., condition (often 0 or 1)
b_1 . . . coefficent that captures the difference between different conditions
e . . . error or uncertainty

Interpretation of the negative binomial GLM



Linear regression model (LM) is evaluated for every gene: $Y = b_0 + b_1 * x_1 + e$

Y . . . describes all read counts for a gene
b_0 . . . average of baseline group, *e.g.*, control
x_1 . . . design factor, *e.g.*, condition (often 0 or 1)
b_1 . . . coefficent that captures the difference between different conditions
e . . . error or uncertainty

$\rightarrow$ the closeness of b_1 to zero will be evaluated during statistical testing steps

$\rightarrow$ DESeq2 and edgeR use a generalized linear model (GLM)

Parametric vs. non-parametric methods

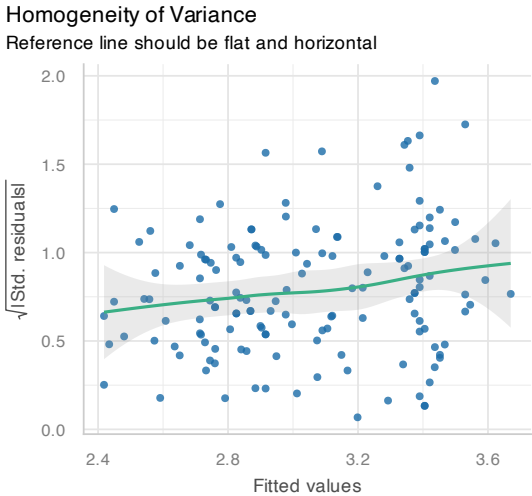
It would be nice to not have to assume anything about the expression value distributions but only use rank-order statistics.

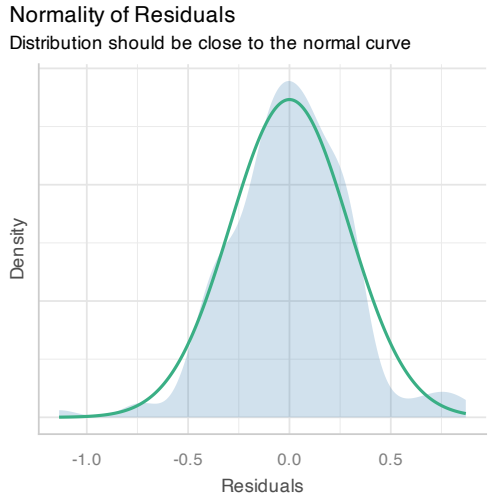
However, it is hard to show statistical significance with non-parametric methods if only few replicates are available (less than 8).

Issues with DGE analysis for RNA-seq

Couldn't we just use a Student's t-test for each gene?

Parametric model assumptions





Issues with DGE analysis for RNA-seq

Couldn't we just use a Student's t-test for each gene?

1. Distribution is not **normal**. Which parametric distribution should I use?
2. **Variance** across groups may not be **homogeneous** e.g. unequal group size
3. The number of replicates is often too small to estimate the variance.
4. If we test each gene for DE, we have to account for **multiple testing**!

Which parametric distribution should I use?

Models for read counts originated from the idea that each read is sampled independently from a pool of reads and hence the number of reads for a given gene follows a ...

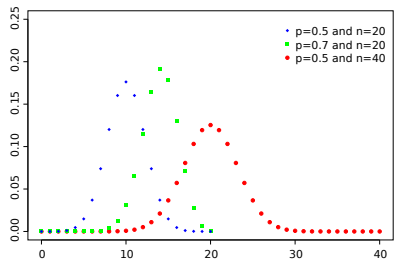
Binomial distribution

The binomial distribution works when we have a fixed number of trials n, each with a constant probability of success p.

The random variable X is the number of k successes:

$$p(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Event: An RNA-seq read "lands" in a given gene (success) or not (failure)



As RNA-seq experiments produce large number of reads (n is large) the Gaussian distribution can replace the binomial.

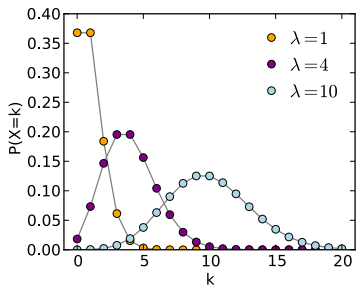
Poisson distribution

RNA-seq experiments produce large number of reads (n is large) and probabilities of success are small (p is small) which can be modelled by the poisson distribution which is an approximation of the binomial.

Instead we know the average number of successes per intervall:

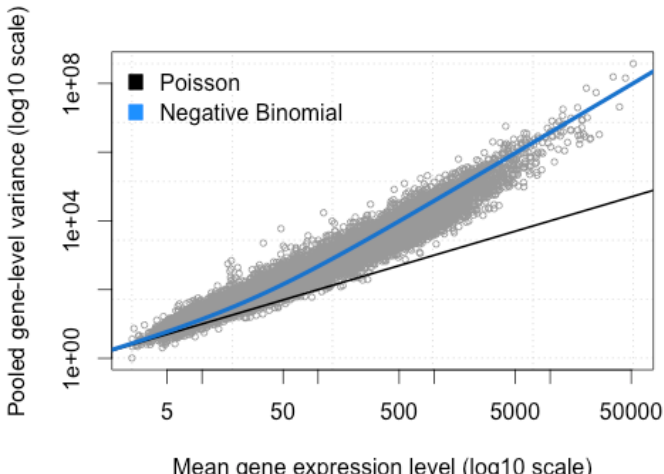
$$\lambda = np$$

For $X \sim \text{Poisson}(\lambda)$: $\lambda = \text{Var}(X) = \mu$, and dispersion $\alpha = 0$.



Poisson versus negative binomial distribution

Many studies have shown that the variance grows faster than the mean in RNAseq data. This is known as **overdispersion**.



Negative binomial distribution

The negative binomial distribution works for discrete data over an unbounded positive range whose sample variance exceeds the sample mean.

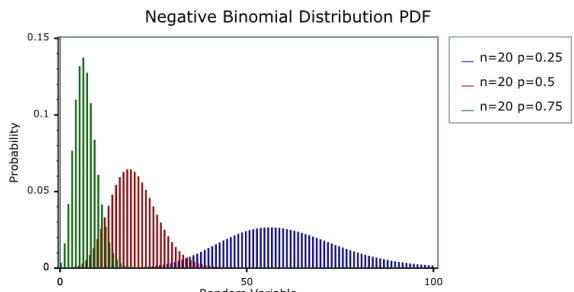
The random variable X is the number of trials needed to make r successes (and k failures) if the probability of a single success is p:

$$NB(X = k) = \binom{k + r - 1}{k} p^r (1 - p)^k$$

both the μ and $\mathrm{Var}(X)$ can be calculated from r and p; dispersion $\alpha > 0$

$$\mathrm{Var}(X) = \mu + \alpha\mu^2$$

$$\hat{\alpha} = \frac{\hat{\sigma}^2 - \hat{\mu}}{\hat{\mu}^2}$$



Fitting a negative binomial GLM

Raw count for gene *i* in sample *j*

Controls the variance

$$K_{ij} \sim \mathrm{NB}\left(\mathrm{mean} = \mu_{ij}, \mathrm{dispersion} = \alpha_i\right)$$

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

Normalization ("size") factor

Normalized count

Design matrix

-- Control or Treatment?

-- Batch (e.g., flow cell, plate, lab)

-- Other co-factors (e.g., gender)

Linear Model:

$$\log_2 q_{ij} = \sum x_{jr} \beta_{ir}$$

Coefficient r

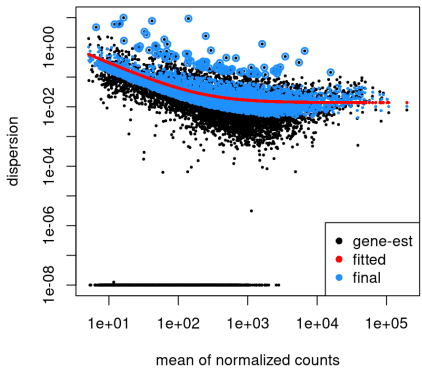
Generalized Linear Model (GLM) coefficients

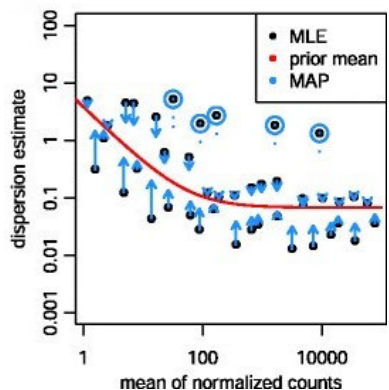
-- One for each Design matrix column (factor)

= strength of effect

= **log2 fold change** for each gene

Dispersion estimate





- Not enough replicates to estimate dispersion for individual genes
- Borrow information from genes of similar expression strength among the replicates
- Genes with very high dispersion left as is
- Do not calculate dispersion from rlog or VST values

Design & contrast matrix

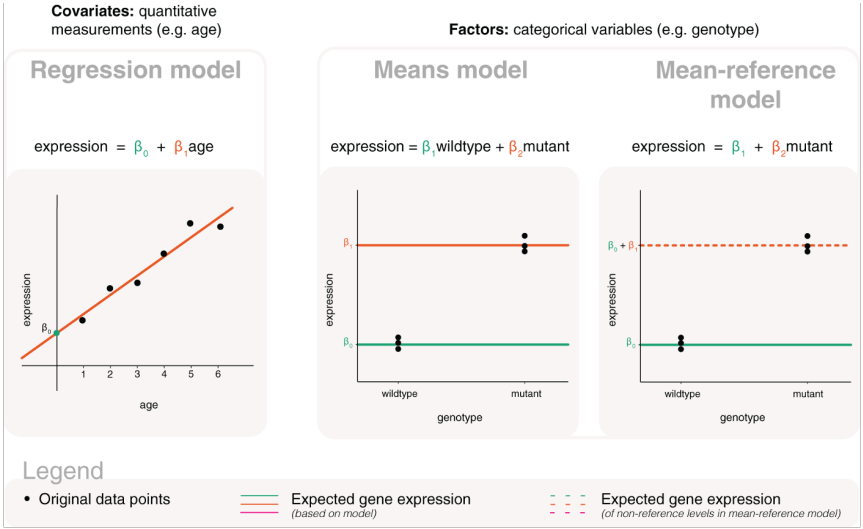
Design matrix (a.k.a. model matrix) has 2 main roles:

1. defines the form of the model, or structure of the relationship between genes and explanatory variables
2. is used to store values of the explanatory variable(s)

Contrast matrix is used for:

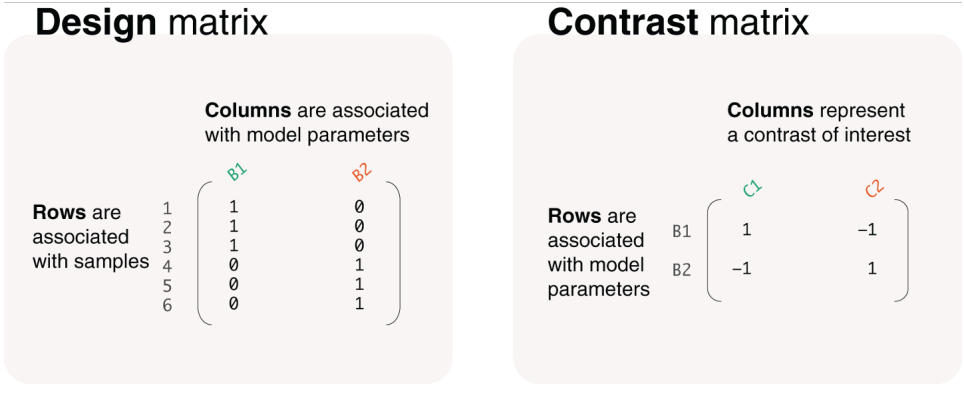
1. identifying the differences (contrast) between explanatory variables
   e.g. $group_1$ vs $group_2$

Basic regression models



Law CW, Zeglinski K, Dong X *et al.*A guide to creating design matrices for gene expression experiments. F1000Research 2020, 9:1444

Design & contrast matrices



Law CW, Zeglinski K, Dong X *et al.*A guide to creating design matrices for gene expression experiments. F1000Research 2020, 9:1444

Design matrix with intercept

x is an indicator variable for **sick** mice:

- x = 1 for sick mice
- x = 0 otherwise

Model

E(y) = 2.95 + 1.62x

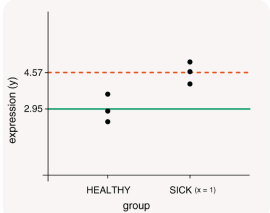
E(y) = 2.95	=	2.95	*(for healthy group)*
E(y) = 2.95 + 1.62	=	4.57	*(for sick group)*

Matrix

```
> model.matrix(~group)
```

	(Intercept)	groupSICK
1	1	0
2	1	0
3	1	0
4	1	1
5	1	1
6	1	1

Plot



contrast = c(0,1)

Law CW, Zeglinski K, Dong X *et al.*A guide to creating design matrices for gene expression experiments. F1000Research 2020, 9:1444

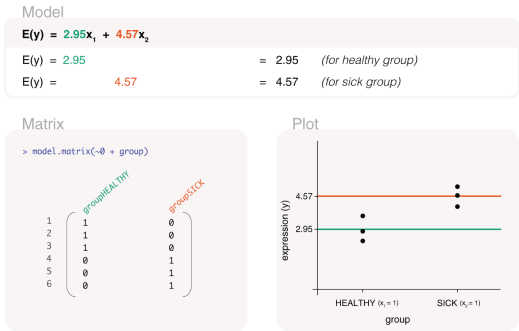
Design matrix without intercept

x1 is an indicator variable for **healthy** mice:

- x1 = 1 for healthy
- x1 = 0 otherwise

x2 is an indicator variable for **sick** mice:

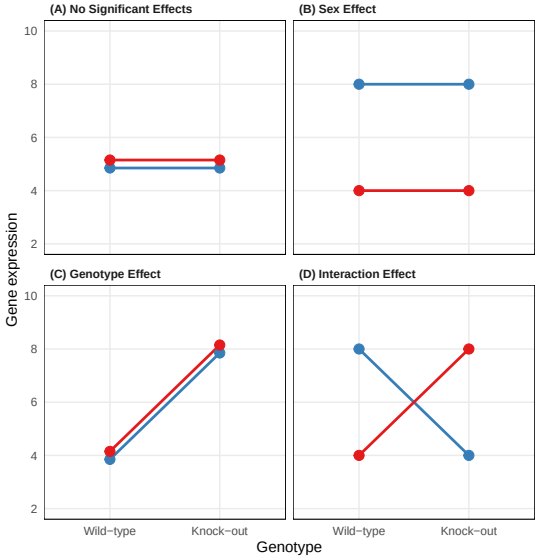
- x2 = 1 for sick
- x2 = 0 otherwise



contrast = c(-1,1)

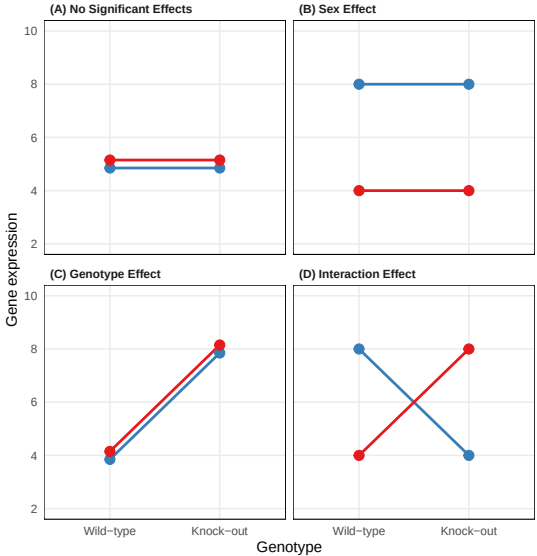
Law CW, Zeglinski K, Dong X *et al.*A guide to creating design matrices for gene expression experiments. F1000Research 2020, 9:1444

2-way designs: main & interaction effects



- Parallel lines → **no** interaction effect
- Crossed lines → **interaction effect**

2-way designs: main & interaction effects



- Parallel lines → **no** interaction effect
- Crossed lines → **interaction effect**

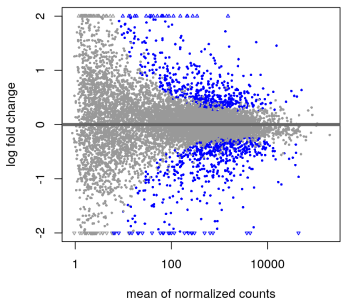
- Interaction effect is **"it depends" effect**
- For example:
  Is gene A is always expressed higher in wild-type compared to knock-out?
- If your answer is **"it depends" on sex**, then there is an interaction effect

Implementation of DGE testing for RNA-seq

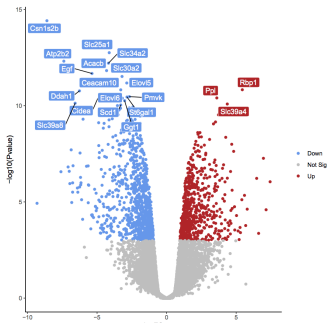
- *Seq. depth normalization:* `DESeq2` uses sample-wise size factor, `edgeR` and `Limma-Voom` use TMM
- *Assumed distribution:* `edgeR` and `DESeq` model the count data using a negative binomial distribution and use their own modified statistical tests based on that. `Limma-Voom` uses log-normal distribution and *t*-test.
- *Dispersion estimate:* `edgeR`, `DESeq2`, `Limma-Voom` (in slightly different ways) "borrow" information across genes to get a better mean-variance estimate.
- *Statistical test to examine if the changes are statistically significant:* `DESeq2` provides the Wald test or the likelihood ratio test; `edgeR` uses quasi-likelihood (QL) F-test or likelihood ratio test
- *Multiple testing issue:* All current packages report false discovery rate FDR (most often Benjamini-Hochberg corrected p values).

Visualization of DE analysis

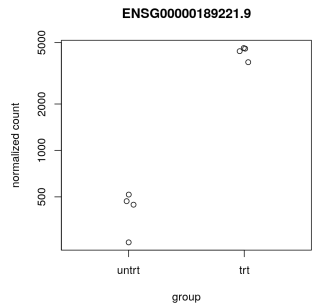
MA-plot (Bland-Altman)



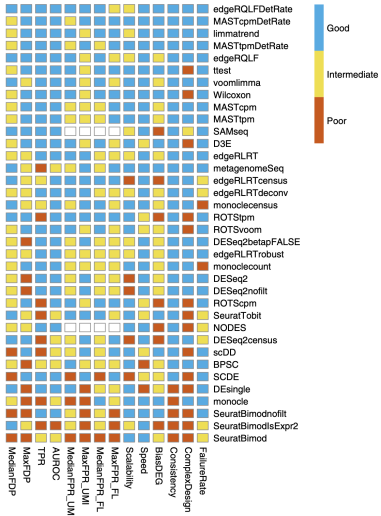
Volcano plot



Normalised counts



An overview of statistical tests for DGE analysis

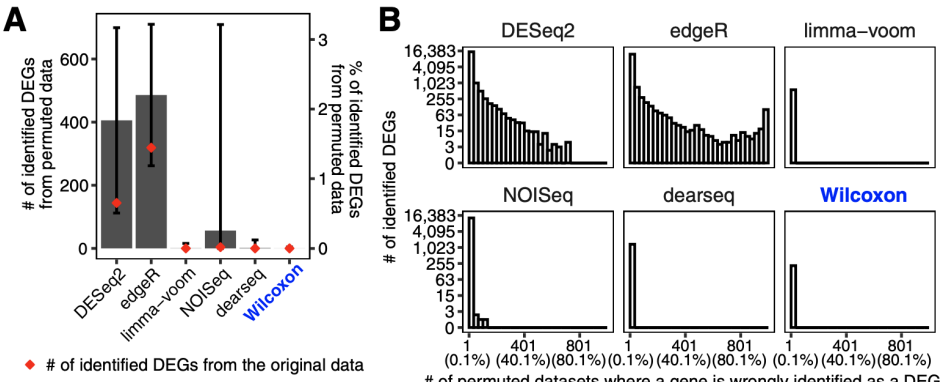


- Classical tests (t-test, Wilcoxon) still perform well
- However, they cannot handle complex designs
- Advanced methods (e.g. DESeq2, edgeR, limma) can handle complex experimental designs
- Choose your method carefully based on your needs
- If you don't know what to do, advanced methods are still the way to go

Soneson & Robinson. Bias, robustness and scalability in single-cell differential expression analysis. Nature Methods (2018)

Very large sample sizes

- population-level RNA-seq studies
- single cell RNA-seq



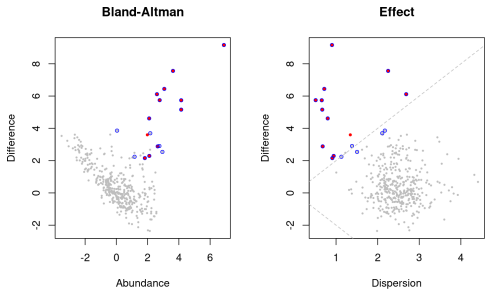
→ Non-parametric approaches, *e.g.*, Wilcoxon rank-sum test, perform best.

Li *et al.* Exaggerated false positives by popular differential expression methods when analyzing human population samples. Genome Biol (2022)

Alternative: Compositional data analysis

ALDex for differential expression analysis:

1. Add a small prior count to the observed counts for taxa j across all samples
2. Draw Monte Carlo samples using the Dirichlet distribution
3. Transform the samples using the Centered log ratio (CLR) transform
4. Hypothesis testing, *e.g.*, Welch's t-test or Wilcoxon rank test
5. Report expected values from statistical tests and effect-size estimate



https://bioconductor.org/packages/release/bioc/vignettes/ALDEx2/inst/doc/ALDEx2_vignette.html

Summary

- Always challenge your data, think of plausible technical explanation first

"I'm a scientist and I know what constitutes proof. But the reason I call myself by my childhood name is to remind myself that a scientist must also be absolutely like a child. If [they] see a thing, [they] must say that [they] see it, whether it was what [they] thought [they] were going to see or not. See first, think later, then test. But always see first. Otherwise you will only see what you were expecting. Most scientists forget that."

– adapted from The Ultimate Hitchhiker's Guide to the Galaxy by Douglas Adams

PhD course "Bioinformatics analysis of gene expression data (BAGED)"

1. Do you have your own bulk or single-cell RNA-seq data in vertebrates or bacteria?
2. Do you want to learn how to analyze your data yourself?
3. **What?** 3 weeks of lectures, tutorials and most importantly student projects
4. **Hardware+Software?** UCloud, (Galaxy), Rstudio, Cytoscape, Bash, R, Snakemake, Python
5. **When?** BAGED-bulk 01.03.2027 – 18.03.2027
6. Sign up early due to limited number of seats