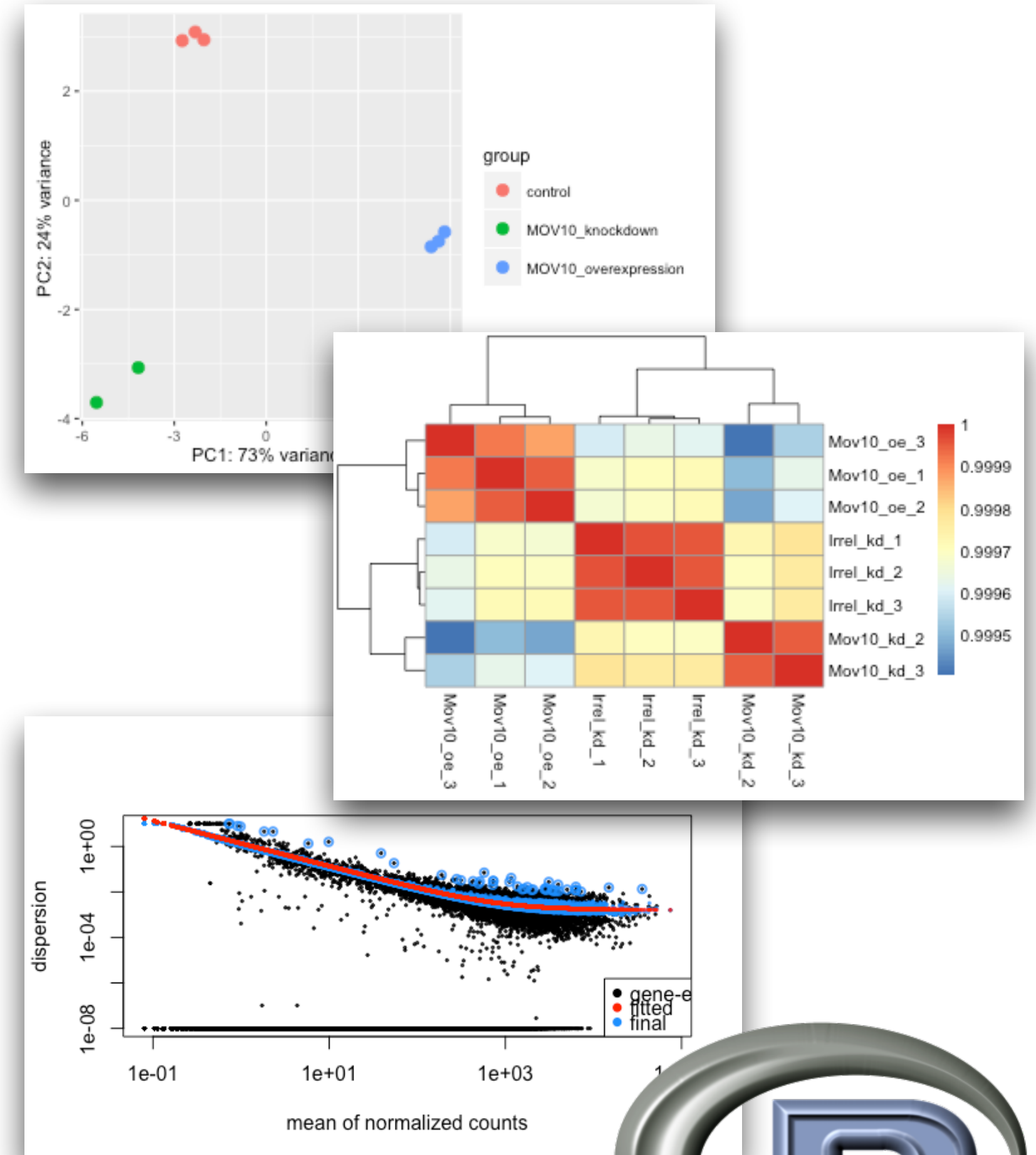
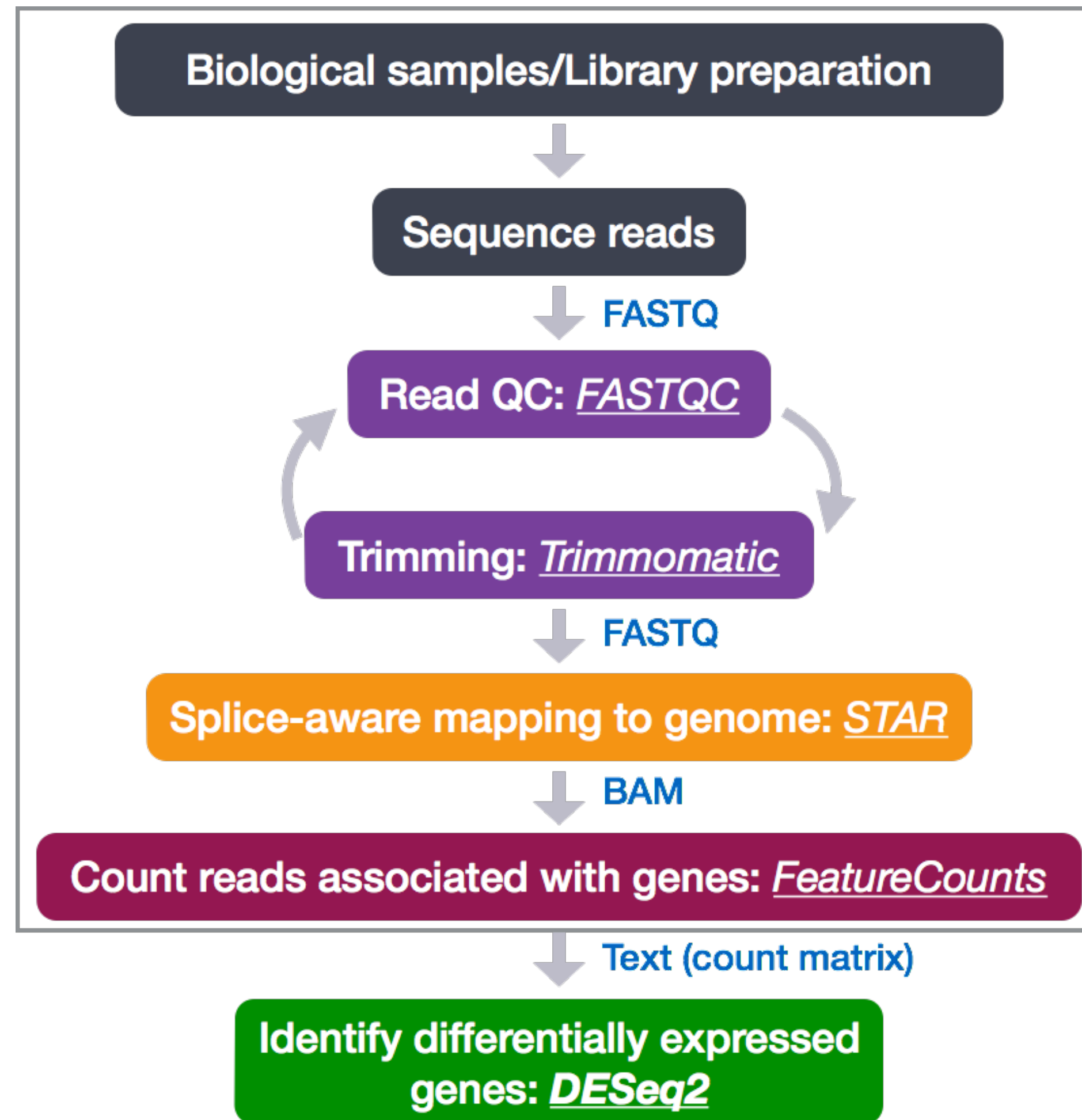


# Statistical analysis for RNA-Seq: *Gene-level differential expression*





From sequence data to count matrix


# High-throughput sequencing data

	SRR1039508	SRR1039509	SRR1039512	SRR1039513	SRR1039516
ENSG000000000003	679	448	873	408	1138
ENSG000000000005	0	0	0	0	0
ENSG000000000419	467	515	621	365	587
ENSG000000000457	260	211	263	164	245
ENSG000000000460	60	55	40	35	78

# High-throughput sequencing data

samples: want to see if differences across  
condition are significant  
(w.r.t. biological and technical variation)

features (e.g. genes)




	SRR1039508	SRR1039509	SRR1039512	SRR1039513	SRR1039516
ENSG000000000003	679	448	873	408	1138
ENSG000000000005	0	0	0	0	0
ENSG000000000419	467	515	621	365	587
ENSG000000000457	260	211	263	164	245
ENSG000000000460	60	55	40	35	78

# High-throughput sequencing data

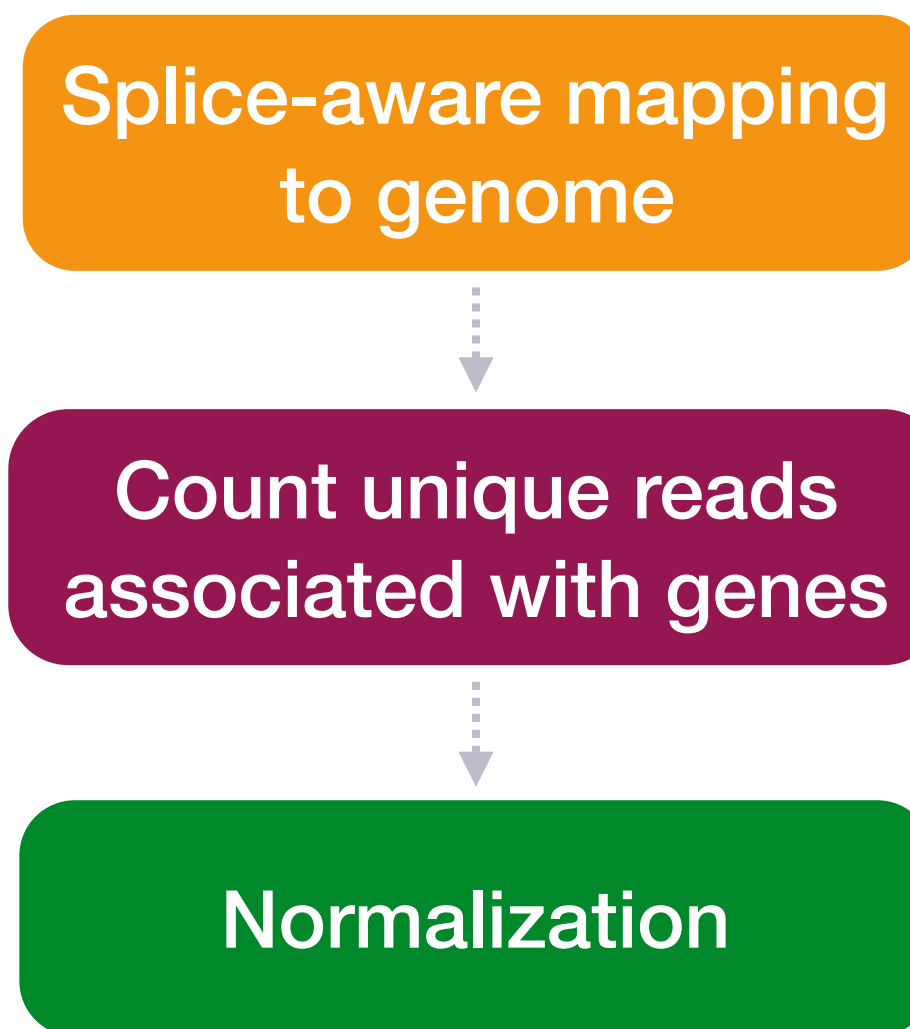
samples: want to see if differences across condition are significant  
(w.r.t. biological and technical variation)

features (e.g. genes)



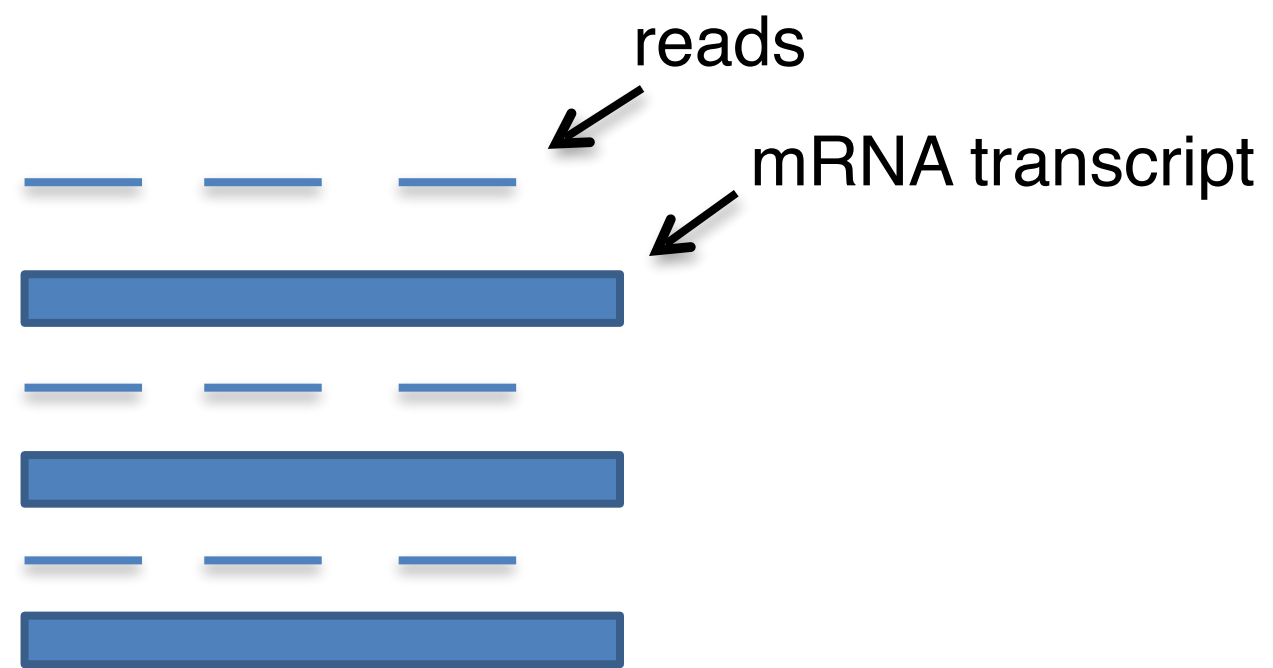
	SRR1039508	SRR1039509	SRR1039512	SRR1039513	SRR1039516
ENSG000000000003	679	448	873	408	1138
ENSG000000000005	0	0	0	0	0
ENSG000000000419	467	515	621	365	587
ENSG000000000457	260	211	263	164	245
ENSG000000000460	60	55	40	35	78

- counts need an appropriate statistical model (normalization and variance modeling)

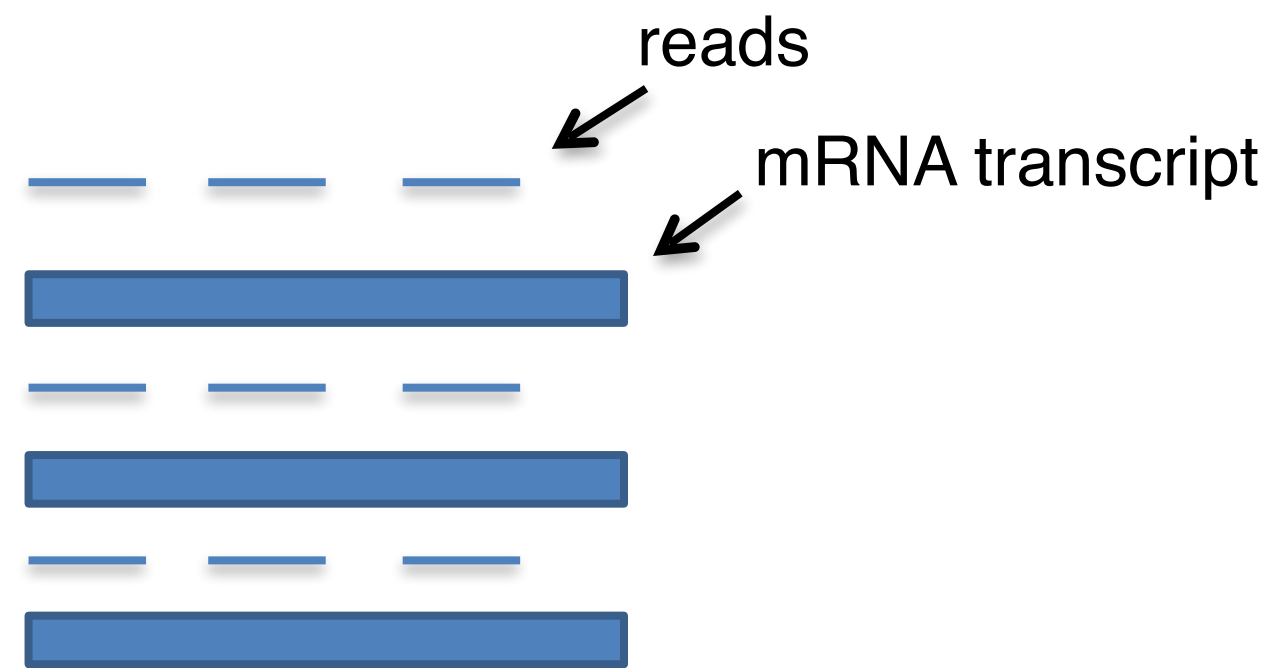


DE workflow :: normalization

# mRNAs to reads



# mRNAs to reads

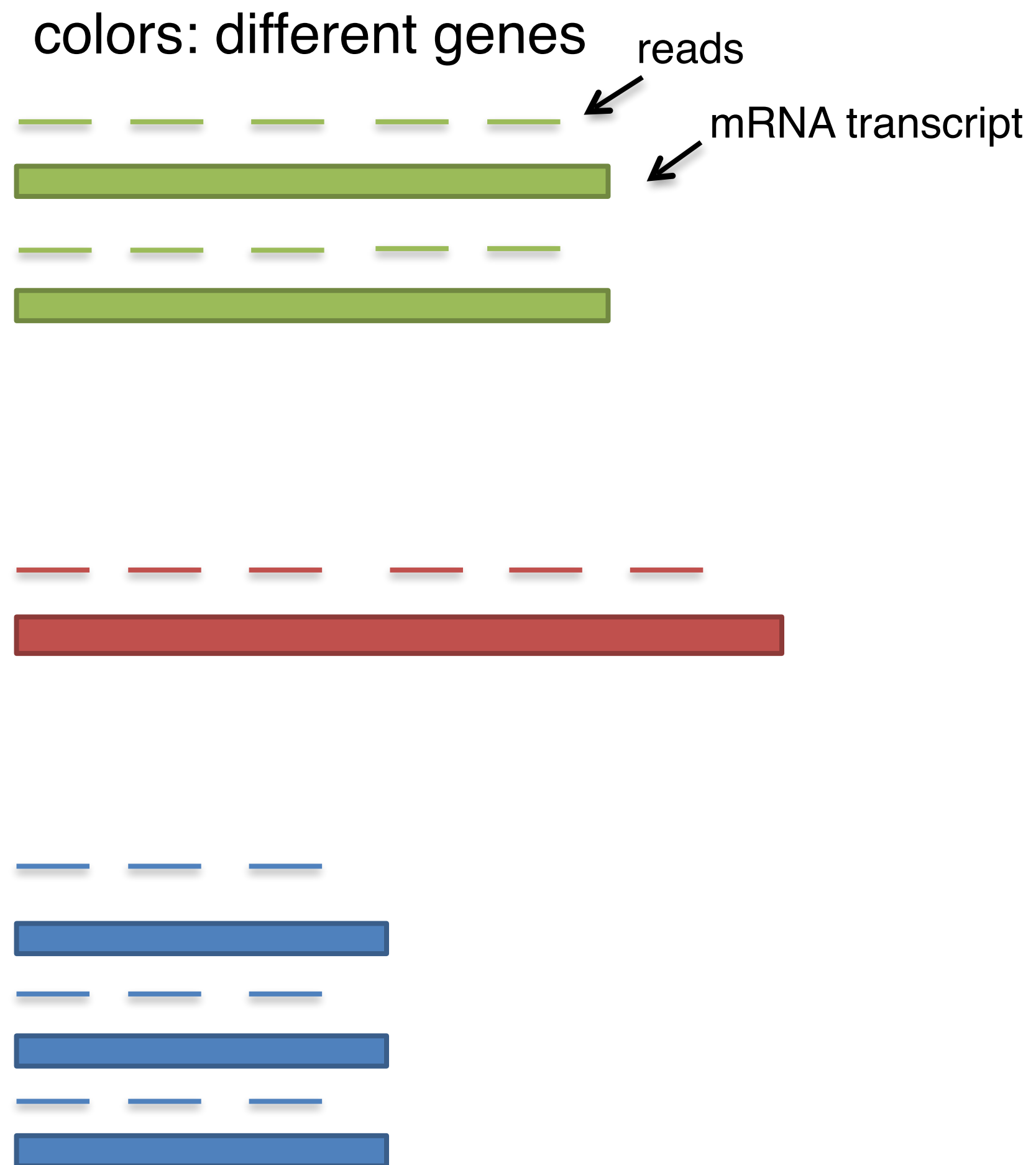


count of mapped reads proportional to:

- expression of RNA
- length of gene
- sequencing depth
- library prep. factors (PCR)
- etc...

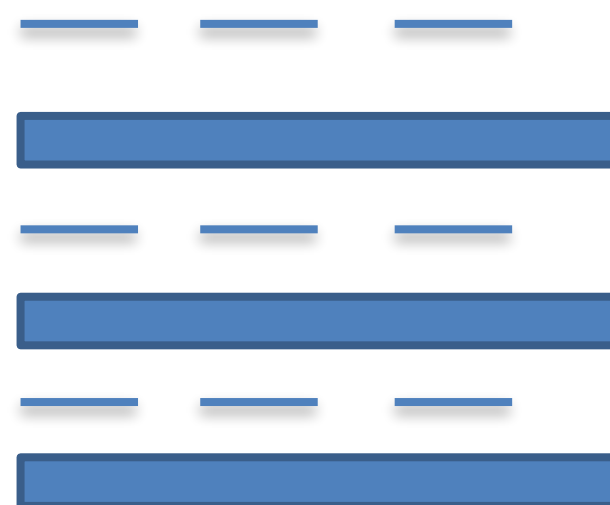
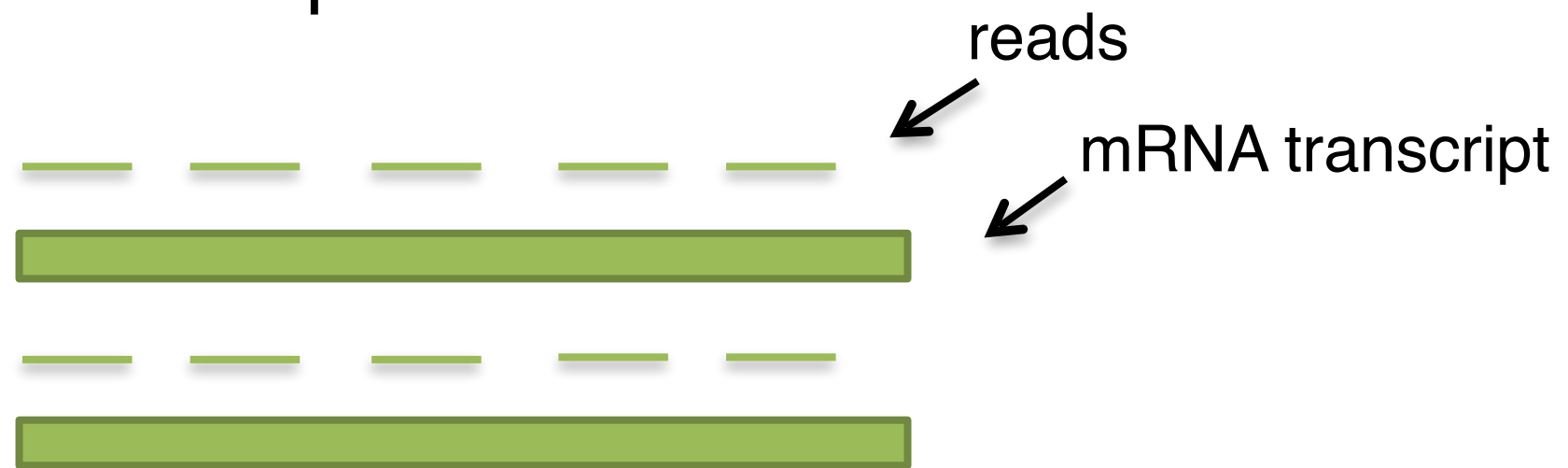


# Length of gene



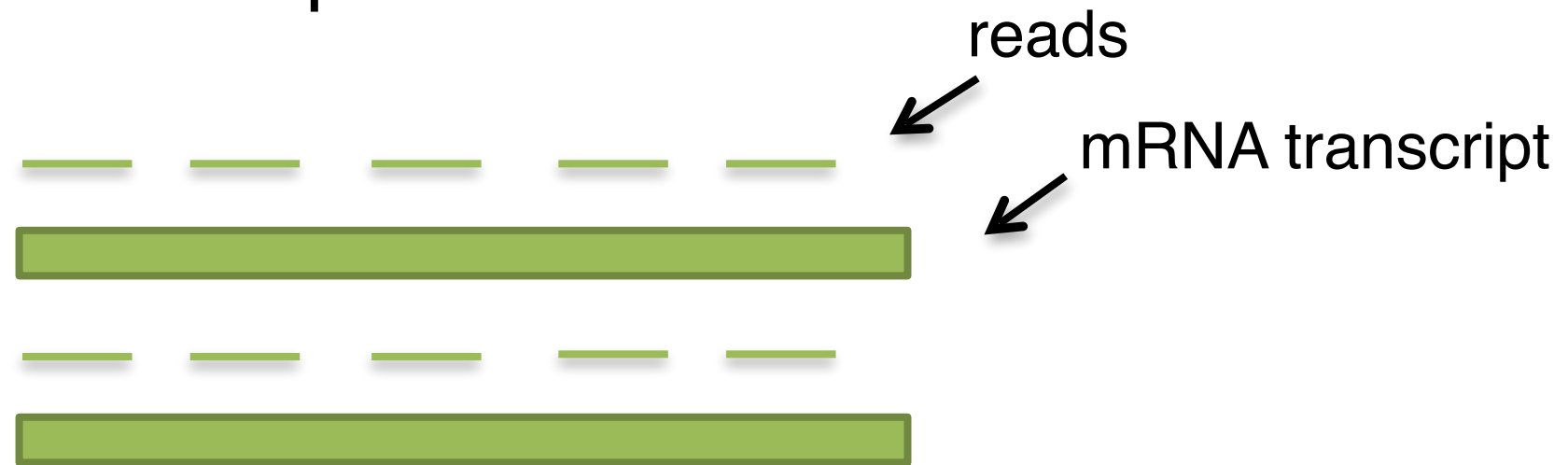
# Sequencing depth

sample 1

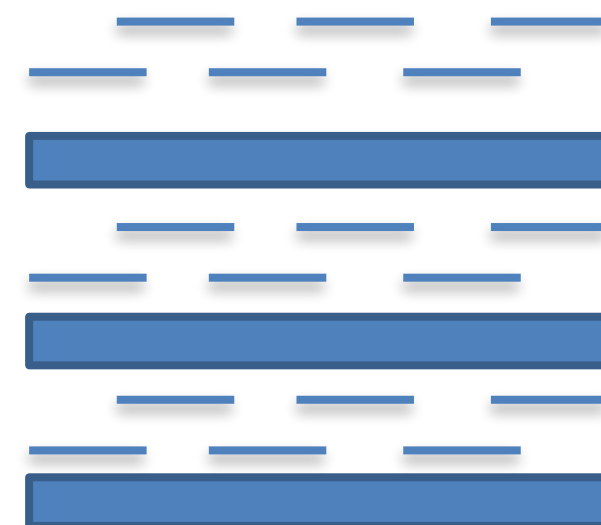
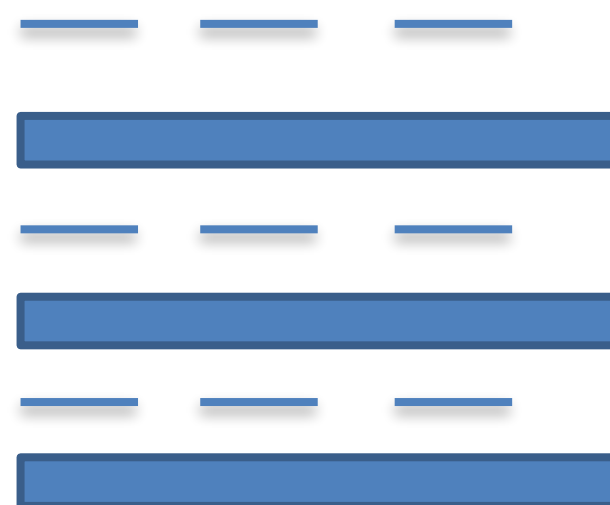


# Sequencing depth

sample 1



sample 2



# (Pre)-scaled measures of expression

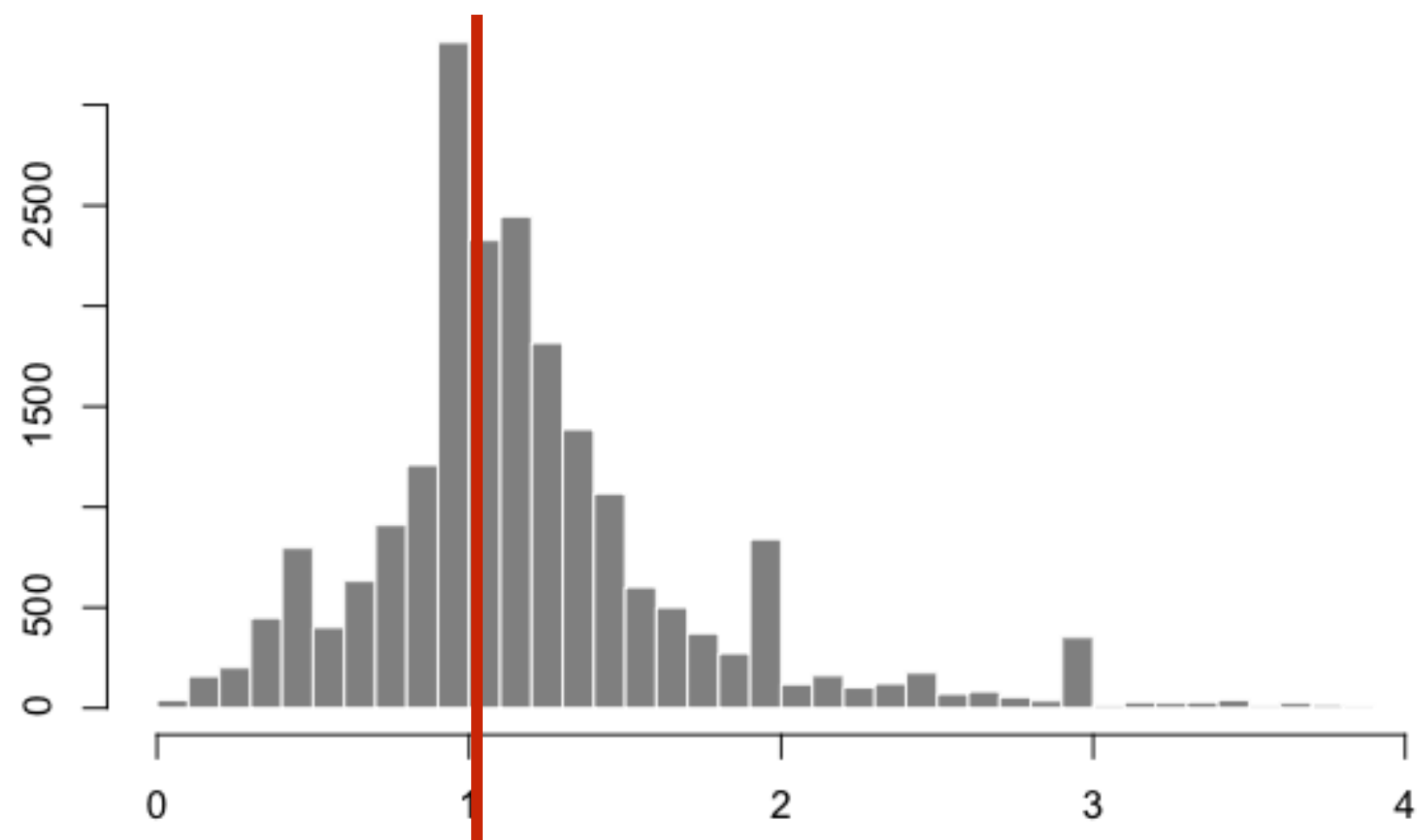
- ▶ CPM (counts per million) counts scaled by the total number of reads
- ▶ TPM (transcripts per million) the proportion of transcripts in your RNA (counts per base)
- ▶ (not recommended) RPKM (reads aligned per kilobase of exon per million reads mapped) – Mortazavi et al 2008
- ▶ (not recommended) FPKM (fragments per kilobase of exon per million fragments mapped). Same idea for paired end sequencing
- ▶ Tool-specific metrics for normalization

Raw counts should be used as input for  
differential expression tools!

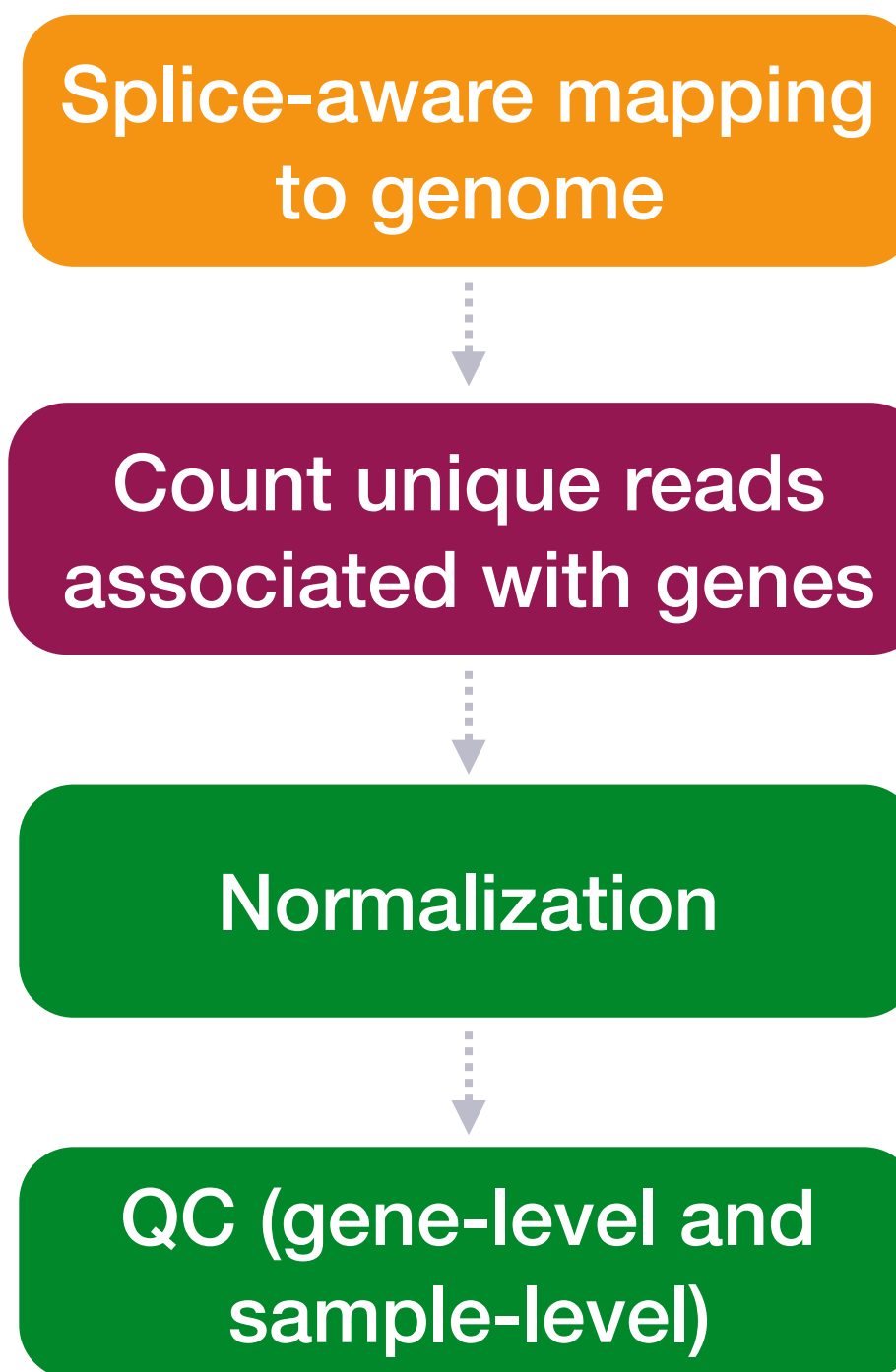
# Median of ratios method

simple approach & works well  
for each gene look at the count ratios:

sample 1 / pseudo-reference sample



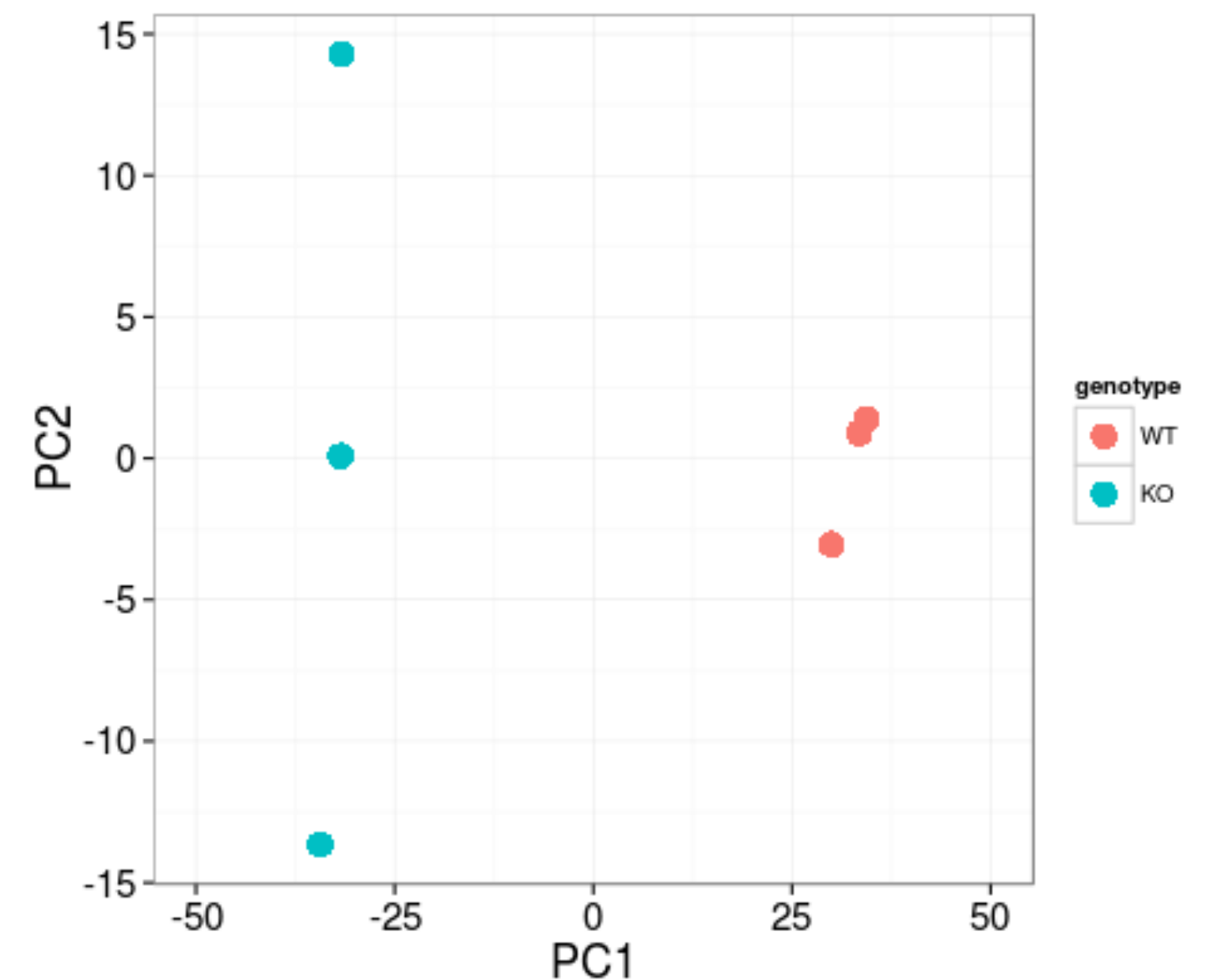
- in general: create a pseudo-reference-sample (row-wise geometric mean)
- calculate ratio of each sample to the reference
- take the median value as the normalization factor
- assumes that not *ALL* genes are DE (differentially expressed)
- **robust** to imbalance in up-/down- regulation and large numbers of DE genes



DE workflow :: quality control

# QC: Sample-level

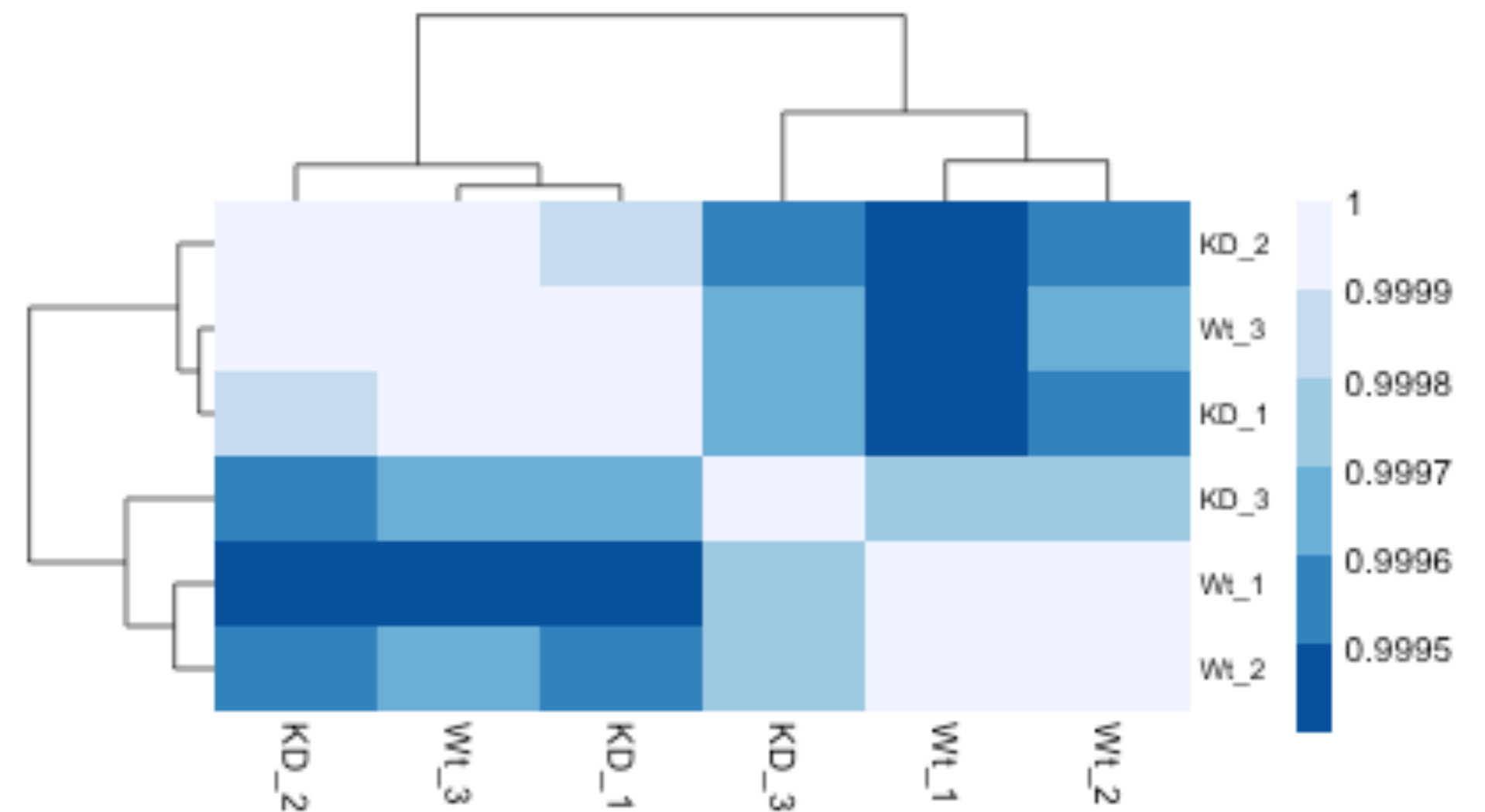
- ▶ **Principal Component Analysis (PCA):** A technique used to emphasize variation and bring out strong patterns in a dataset (dimensionality reduction)
- ▶ Project a line through the data points in  $n$  dimensional space ( $n = \text{genes}$ )
- ▶ Measure how much variance there is from that line (the distance from each point to the line).
- ▶ PC1 explains highest variance, PC2 next highest etc.





# QC: Sample-level

- ▶ Identify strong patterns in a dataset and potential outliers
- ▶ Correlation or distances for all pairwise combinations of samples.
- ▶ Generally high correlations with each other (values higher than 0.80)
- ▶ 'Blocks' indicate substructure in the data



# QC: Gene-level filtering

	SRR1039508	SRR1039509	SRR1039512	SRR1039513	SRR1039516
ENSG000000000003	679	448	873	408	1138
ENSG000000000005	0	0	0	0	0
ENSG000000000419	467	515	621	365	587
ENSG000000000457	260	211	263	164	245
ENSG000000000460	60	55	40	35	78

# QC: Gene-level filtering

	SRR1039508	SRR1039509	SRR1039512	SRR1039513	SRR1039516
ENSG000000000003	679	448	873	408	1138
ENSG000000000005	0	0	0	0	0
ENSG000000000419	467	515	621	365	587
ENSG000000000457	260	211	263	164	245
ENSG000000000460	60	55	40	35	78

Genes with  
zero counts

# QC: Gene-level filtering

	SRR1039508	SRR1039509	SRR1039512	SRR1039513	SRR1039516
ENSG000000000003	679	448	873	408	1138
ENSG000000000005	0	0	0	0	0
ENSG000000000419	467	515	621	365	587
ENSG000000000457	260	211	263	164	245
ENSG000000000460	60	55	40	35	78

Genes with extreme  
count outlier



Genes with  
zero counts



# QC: Gene-level filtering

	SRR1039508	SRR1039509	SRR1039512	SRR1039513	SRR1039516
ENSG000000000003	679	448	873	408	1138
ENSG000000000005	0	0	0	0	0
ENSG000000000419	467	515	621	365	587
ENSG000000000457	260	211	263	164	245
ENSG000000000460	60	55	40	35	78


Genes with extreme  
count outlier

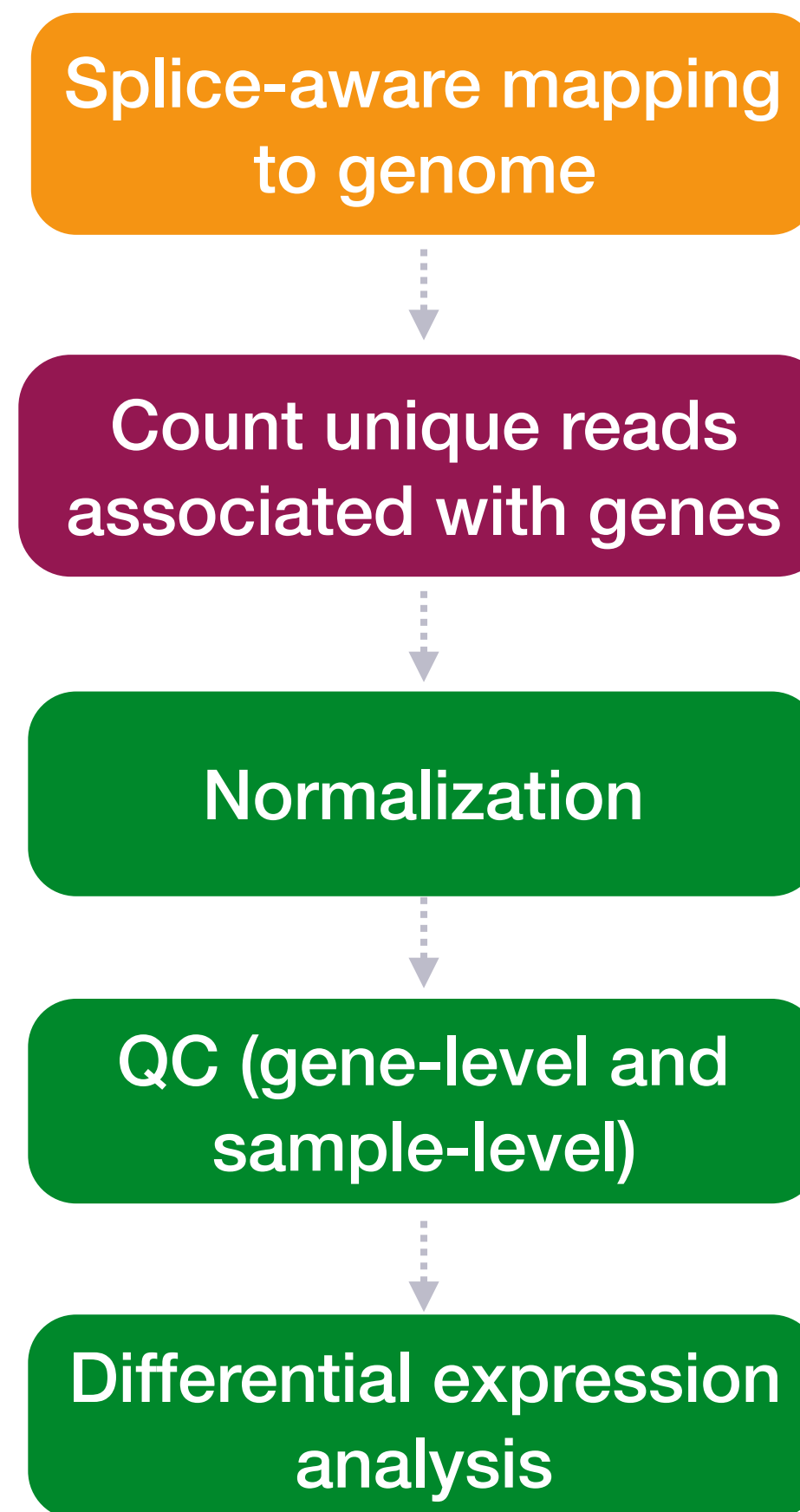


Genes with  
zero counts



Genes with low mean  
normalized counts  
(‘Independent filtering’)



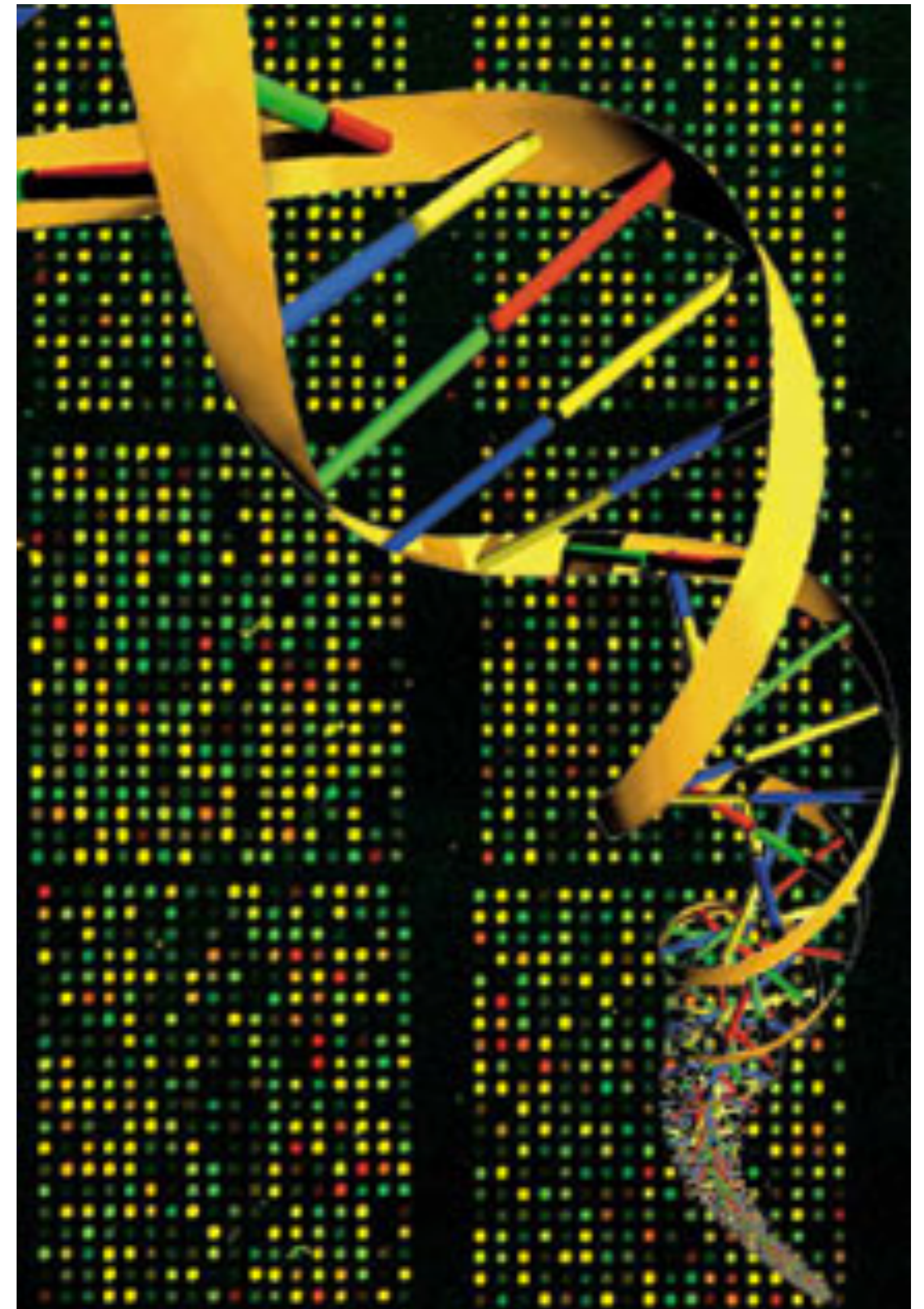


DE workflow :: differential expression

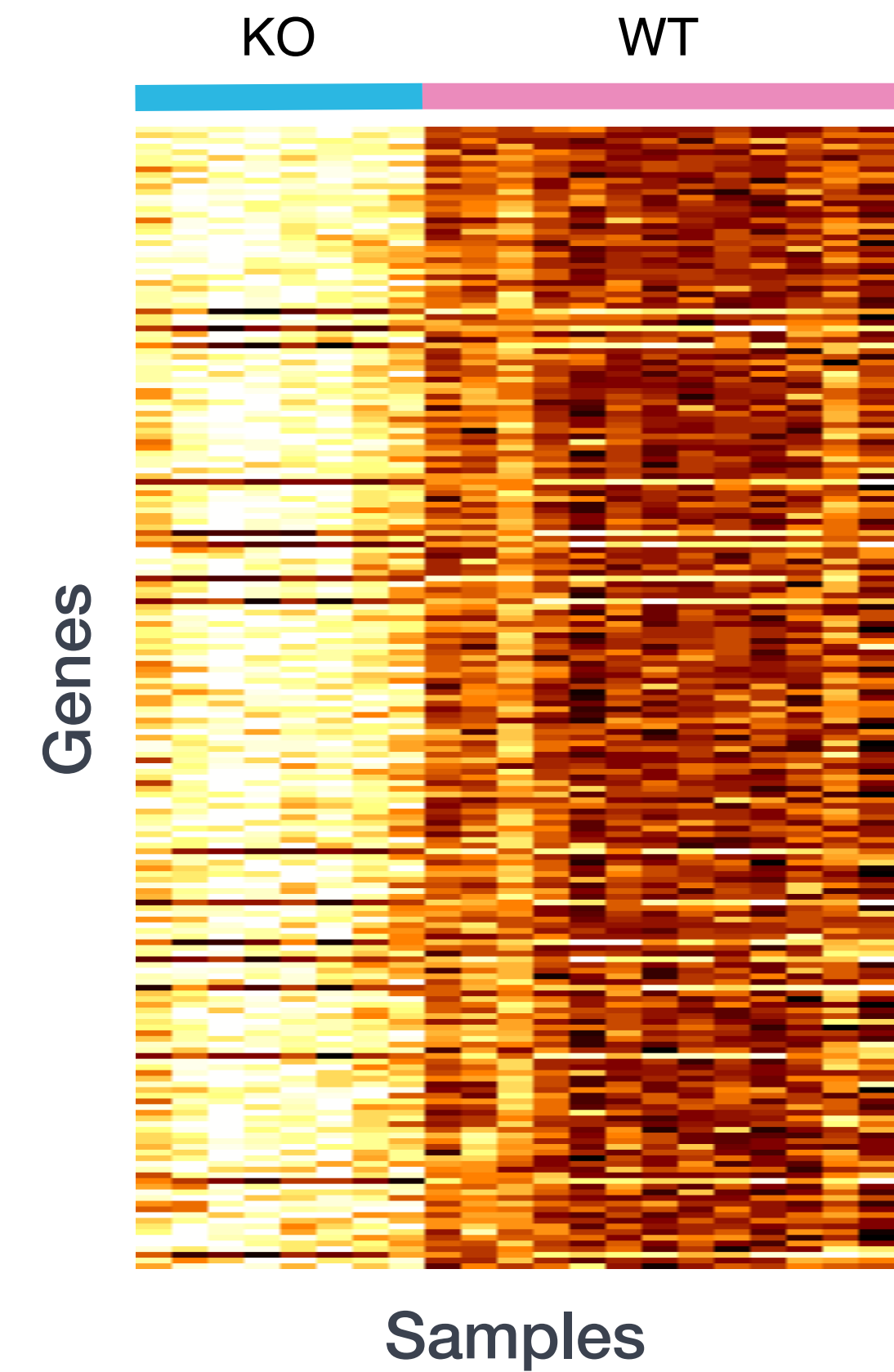


# Transcriptomics

- ▶ to discover functional patterns of biological response to conditions of interest (treatments, environmental influences, mutations etc.)



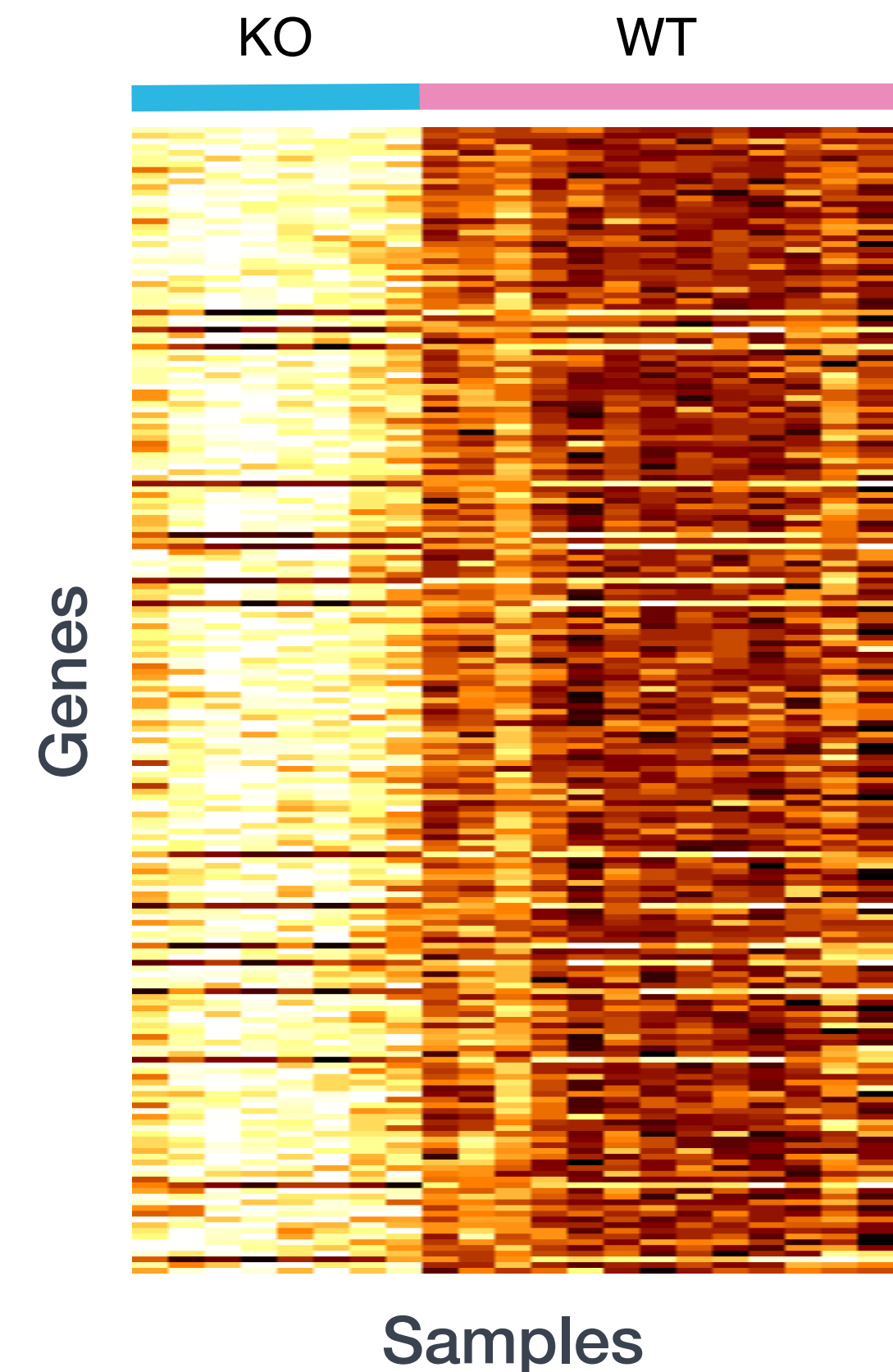
# Identifying differences in gene expression





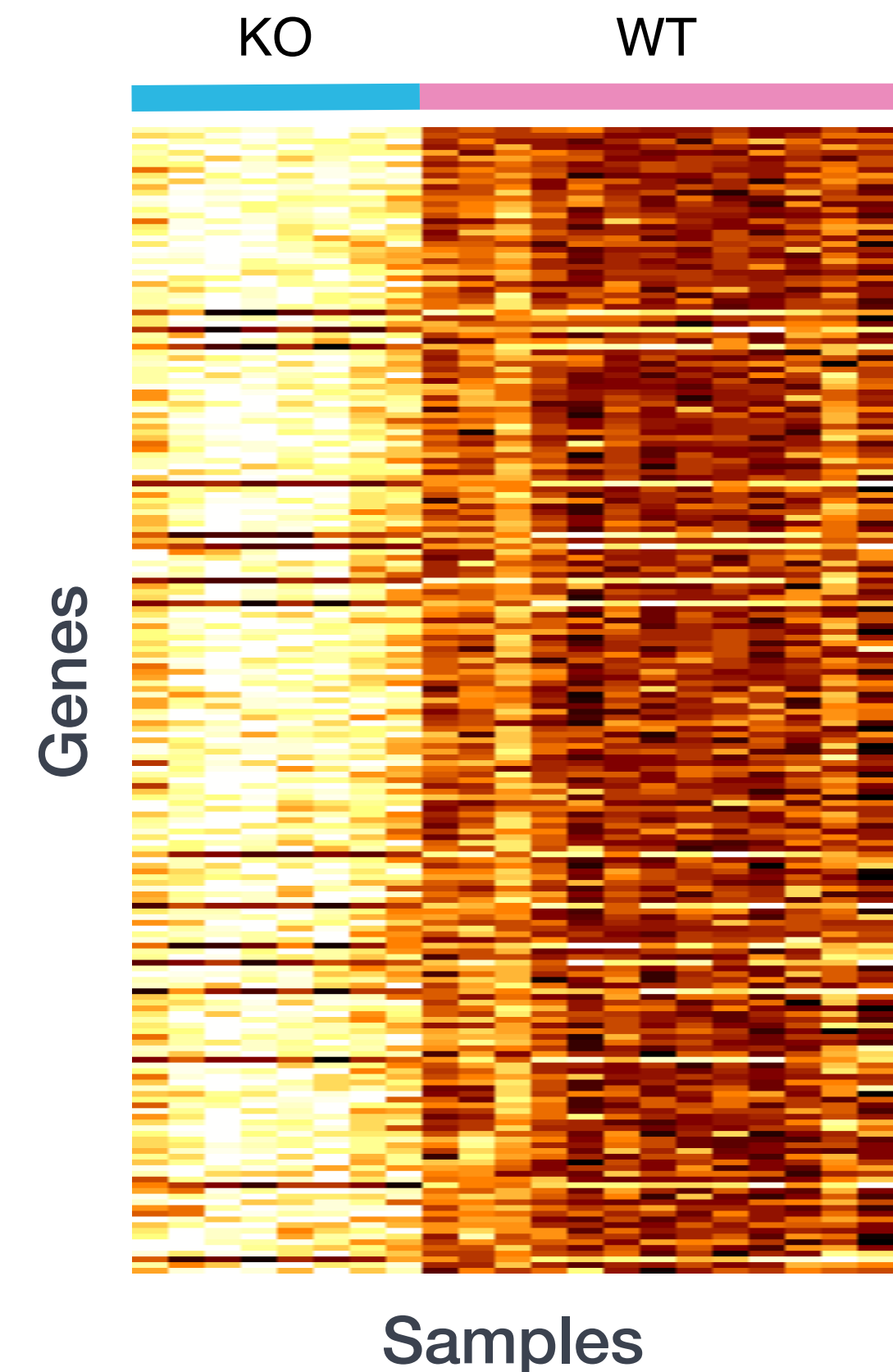
# Identifying differences in gene expression

- ▶ Looking for genes that change in expression between two or more groups
  - ▶ case vs. control
  - ▶ correlation of expression with some variable or clinical outcome

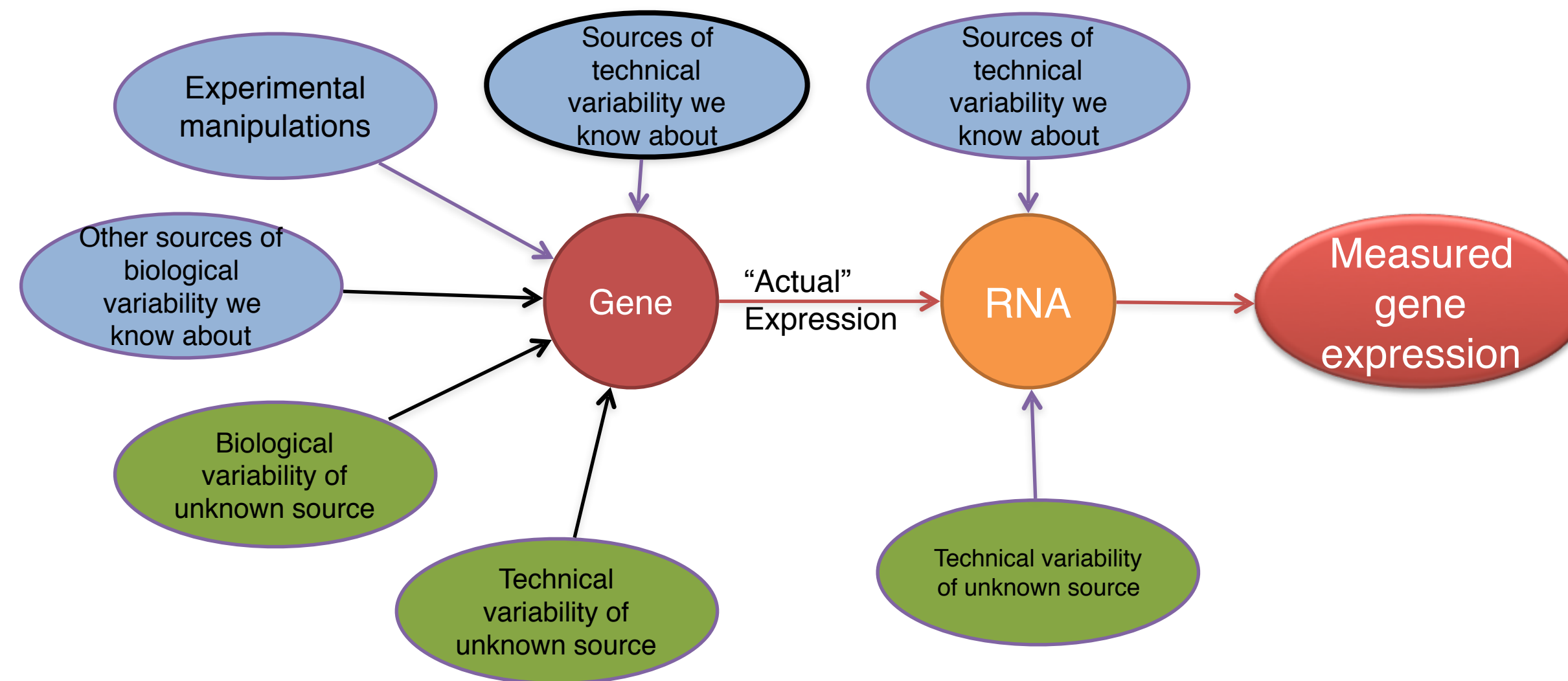


# Identifying differences in gene expression

- ▶ Looking for genes that change in expression between two or more groups
  - ▶ case vs. control
  - ▶ correlation of expression with some variable or clinical outcome
- ▶ Rank the genes by how different they are between the two groups (based on fold change values). **Why does this not work?**



The measurement is the “sum” of many effects

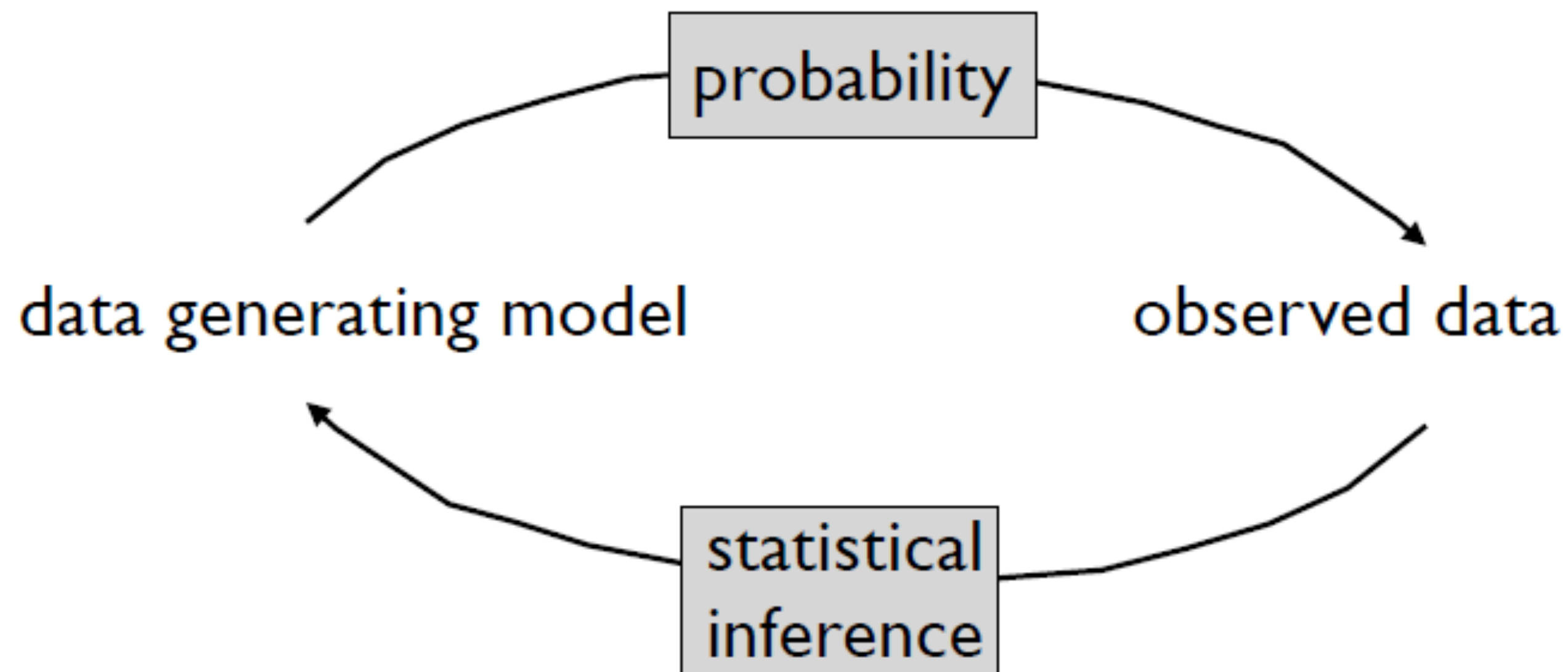


# Modeling gene-level data

Courtesy of Paul Pavlidis, UBC

# Making sense of data

Data is what we observe. We want to infer something about “where it came from”

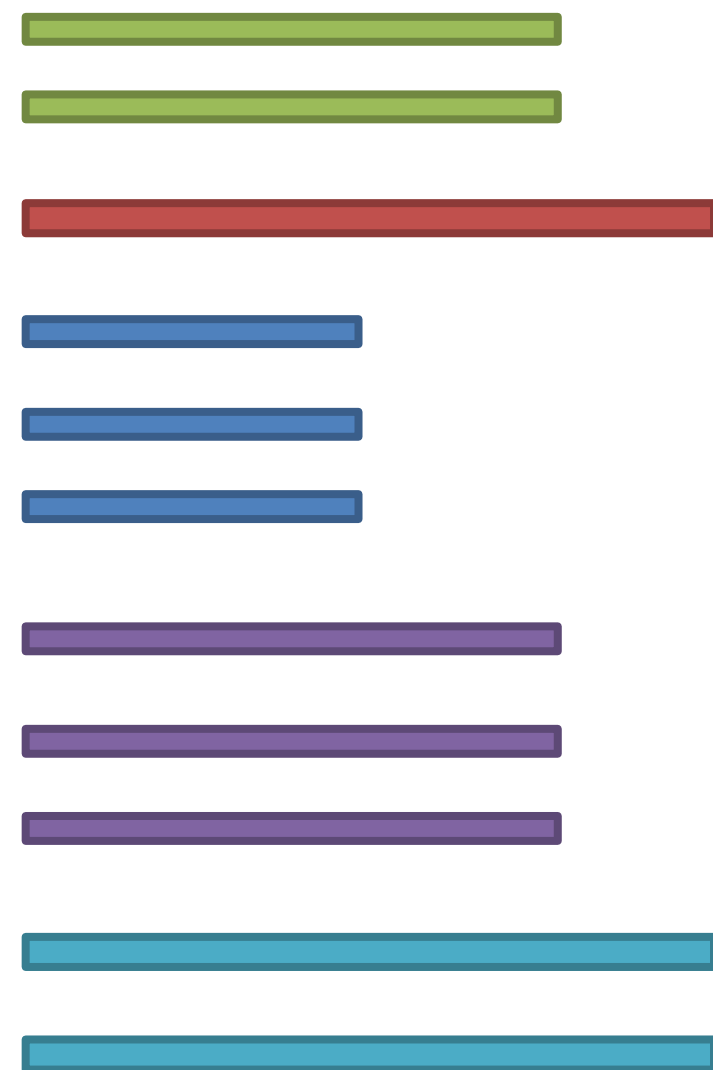


# Characteristics of count data

- Discrete measurement for each gene
- Large dynamic range
- Not normally distributed
- Need regression models specific for count data; generalized linear models

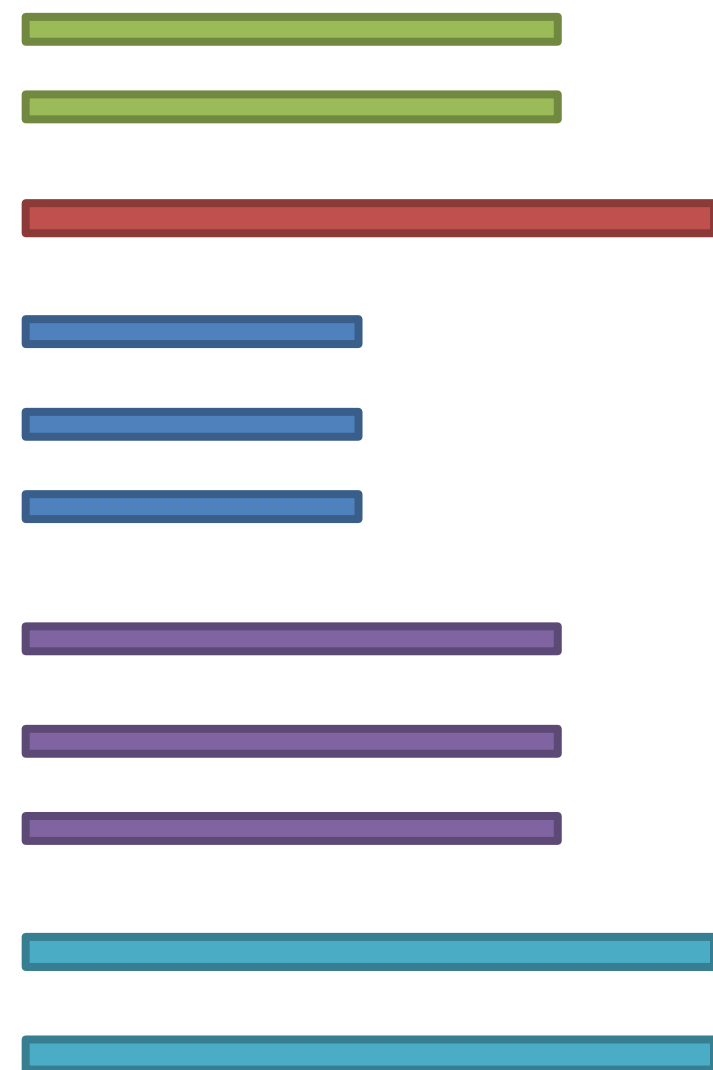
# Distribution of Counts: Poisson versus Negative Binomial

If the proportions of mRNA stays exactly constant between biological replicates we can expect **Poisson distribution**.

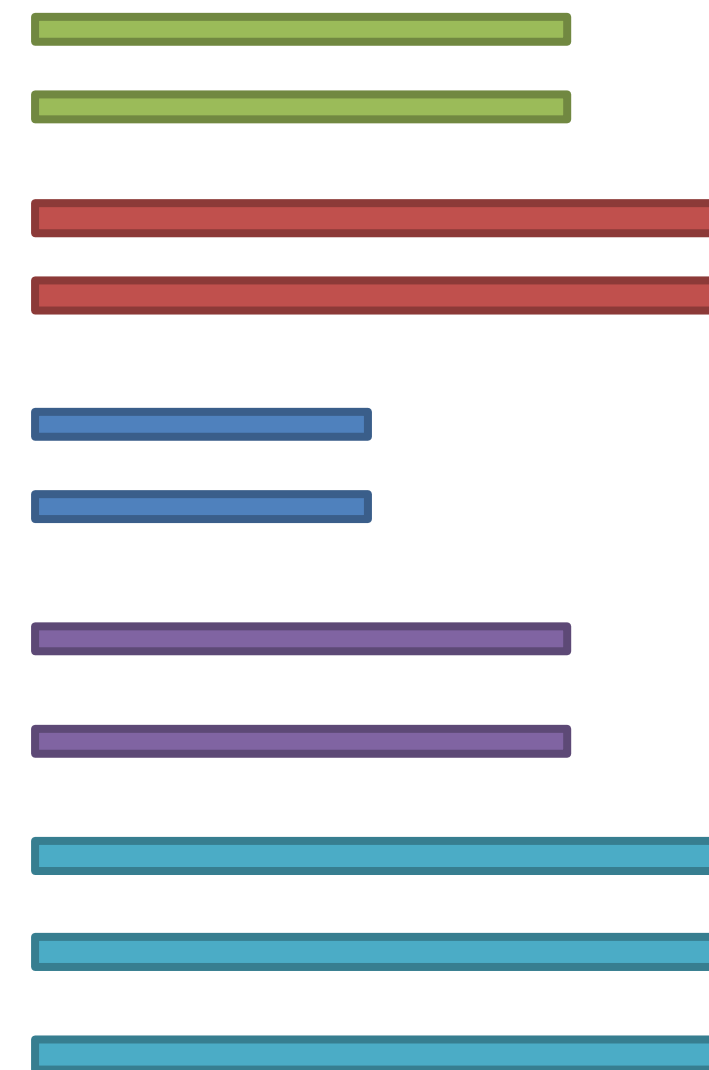


# Distribution of Counts: Poisson versus Negative Binomial

If the proportions of mRNA stays exactly constant between biological replicates we can expect **Poisson distribution**.



But realistically, **biological variation** across sample units is expected and so a **Negative Binomial** distribution is more appropriate.



# Biological vs. Technical Replicates

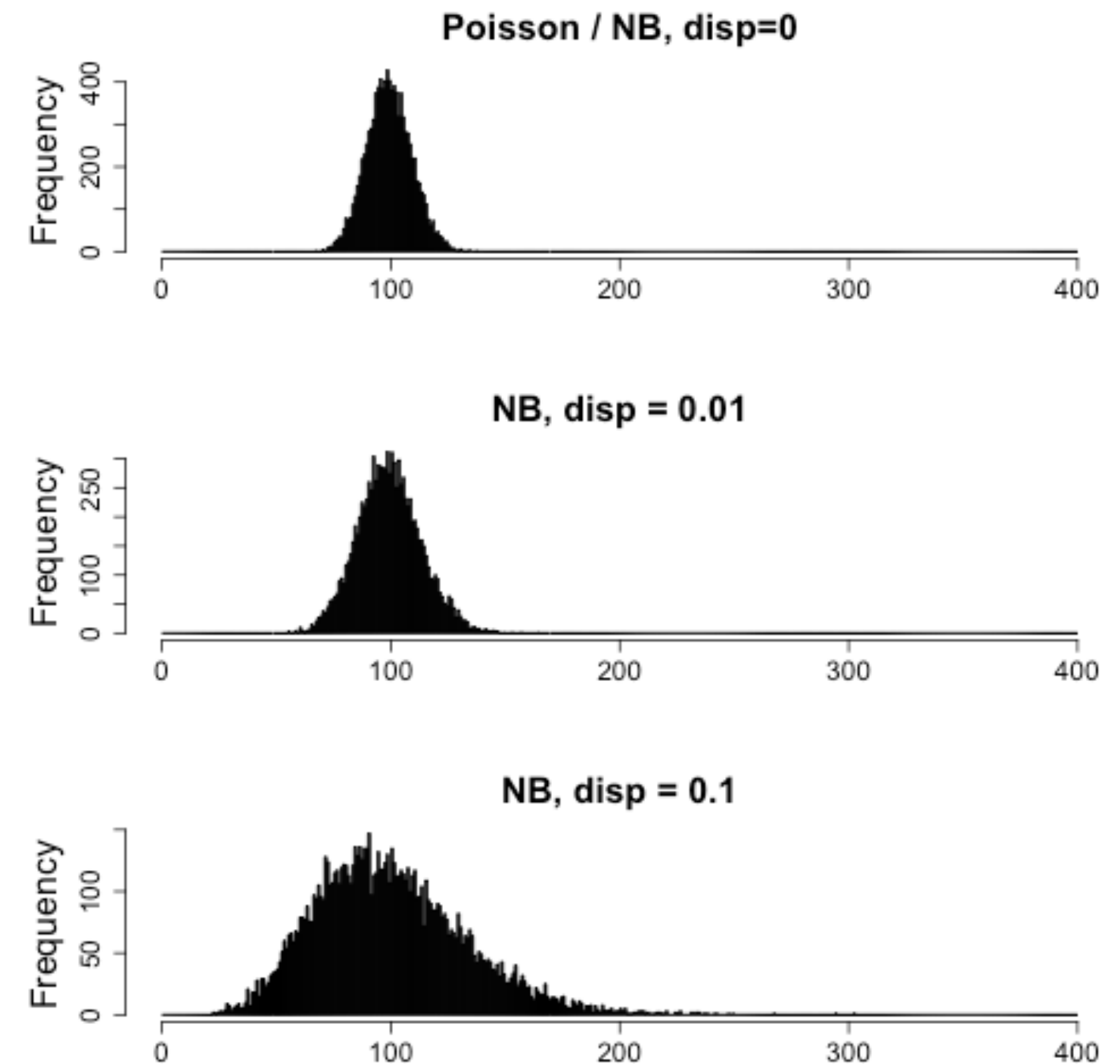
- ▶ **Biological replicates** represent multiple samples representing the same sample class
- ▶ **Technical replicates** represent the same sample but with technical steps replicated
- ▶ Usually **biological variance > technical variance**. They also allow us to make inferences about treatment groups



# Biological replicates produce “over-dispersion” relative to Poisson

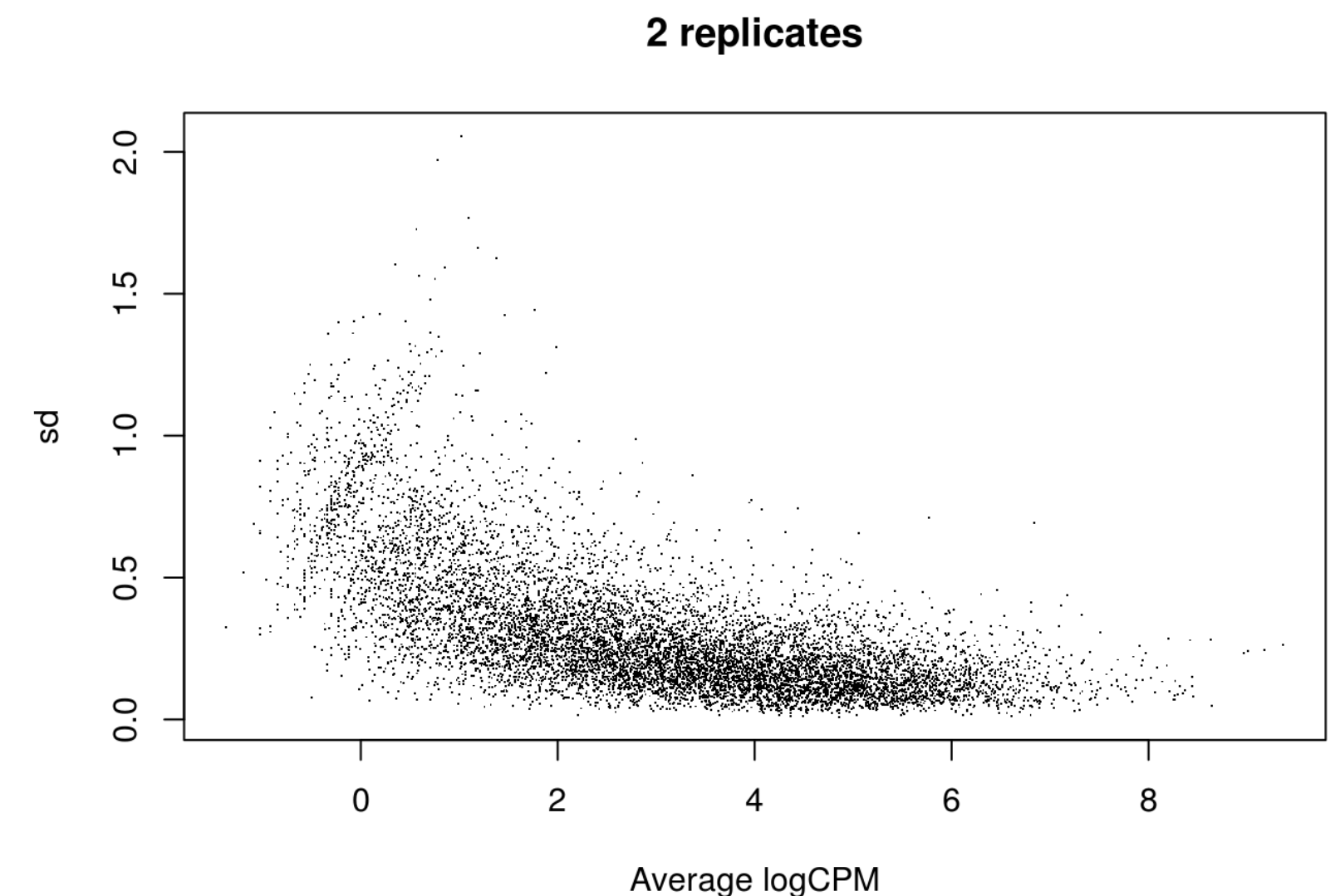
Poisson will underestimate the variability and the effect of the observed differences.

Instead we use the Negative Binomial.



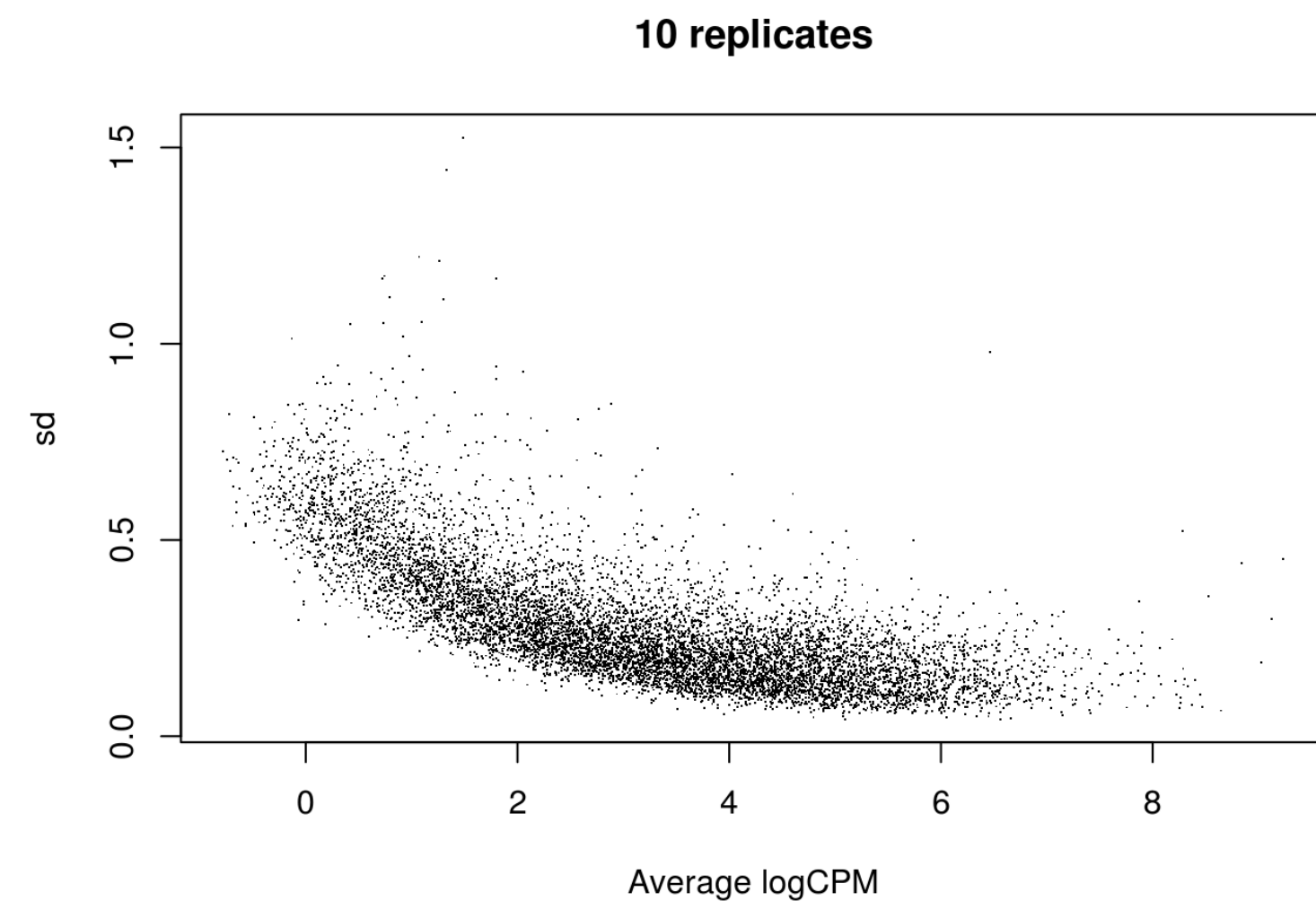
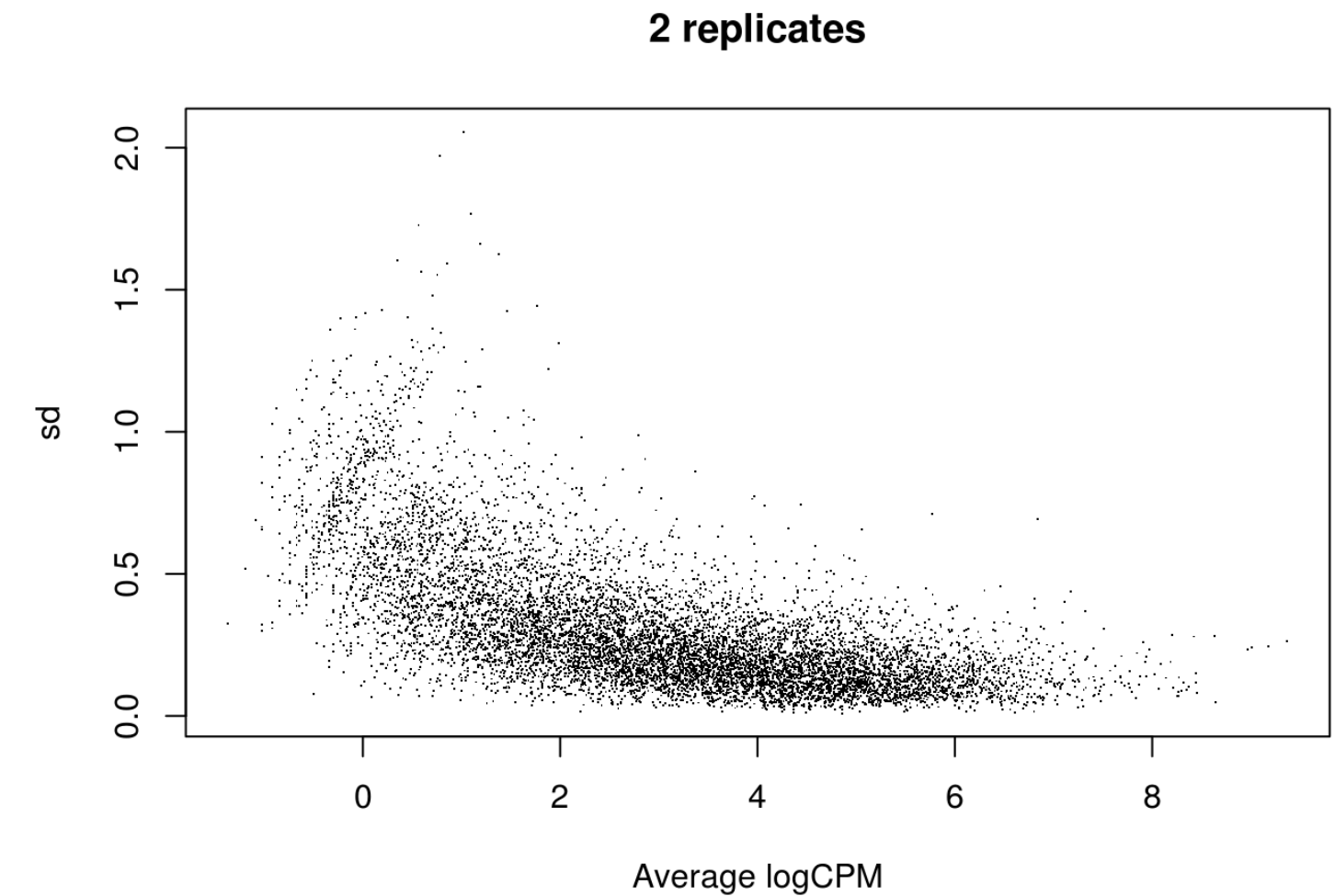
# The need for more replicates

- ▶ Genes with larger average expression have on average larger observed variances across samples, This phenomena of 'having different scatter' is known as data heteroscedasticity.
- ▶ **Mean is not equal to variance**



# Increase the number of replicates...

- ▶ Scatter tends to reduce
- ▶ Standard deviations of averages become smaller
- ▶ With smaller SD, you get more precise estimates of group means, and ultimately greater confidence in the ability to distinguish differences between sample classes



# Counts are modeled using the Negative Binomial (NB) distribution

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

We model read counts  $K_{ij}$ , as following the negative binomial distribution  
with mean  $\mu_i$  and **dispersion**  $\alpha_i$

# Counts are modeled using the Negative Binomial (NB) distribution

raw count for gene  $i$ , sample  $j$



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

We model read counts  $K_{ij}$ , as following the negative binomial distribution  
with mean  $\mu_i$  and **dispersion**  $\alpha_i$

# Counts are modeled using the Negative Binomial (NB) distribution

raw count for gene  $i$ , sample  $j$

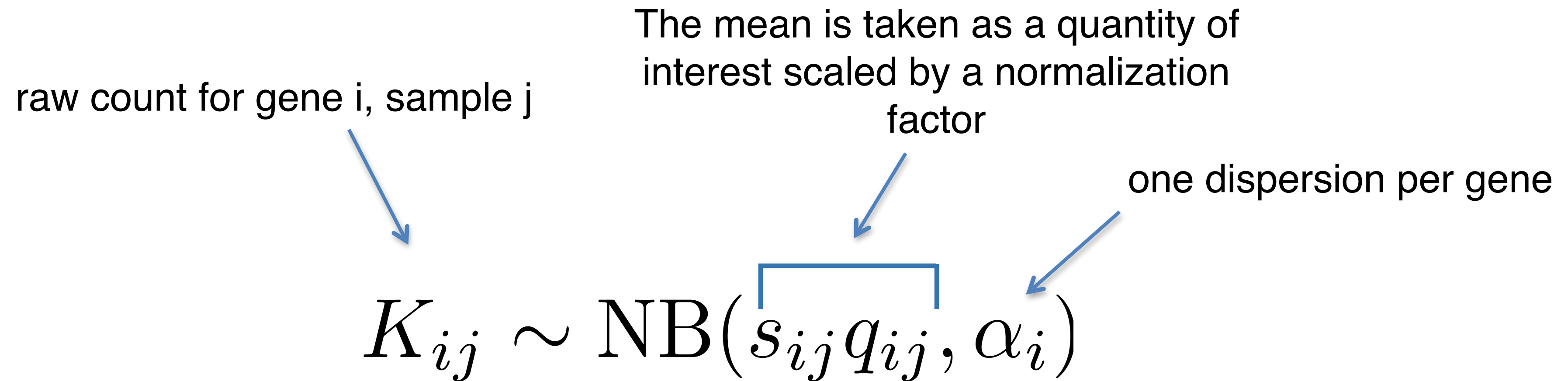
The mean is taken as a quantity of interest scaled by a normalization factor

$$K_{ij} \sim \text{NB}(\overbrace{s_{ij} q_{ij}}^{\text{mean}}, \alpha_i)$$

The diagram illustrates the relationship between the raw count  $K_{ij}$  and the parameters of the Negative Binomial distribution. A blue arrow points from the text 'raw count for gene  $i$ , sample  $j$ ' to the  $K_{ij}$  term in the equation. Another blue arrow points from the text 'The mean is taken as a quantity of interest scaled by a normalization factor' to the term  $s_{ij} q_{ij}$  in the equation, which is enclosed in a blue bracket.

We model read counts  $K_{ij}$ , as following the negative binomial distribution with mean  $\mu_i$  and **dispersion**  $\alpha_i$

# Counts are modeled using the Negative Binomial (NB) distribution



We model read counts  $K_{ij}$ , as following the negative binomial distribution with mean  $\mu_i$  and **dispersion**  $\alpha_i$

# Within group variability is accounted for using the dispersion parameter

The **dispersion parameter**  $\alpha_i$ , describes the variance of counts via:

$$\text{Var}(K_{ij}) = \mu_{ij} + \alpha_i \mu_{ij}^2$$



Poisson part:  
sampling fragments



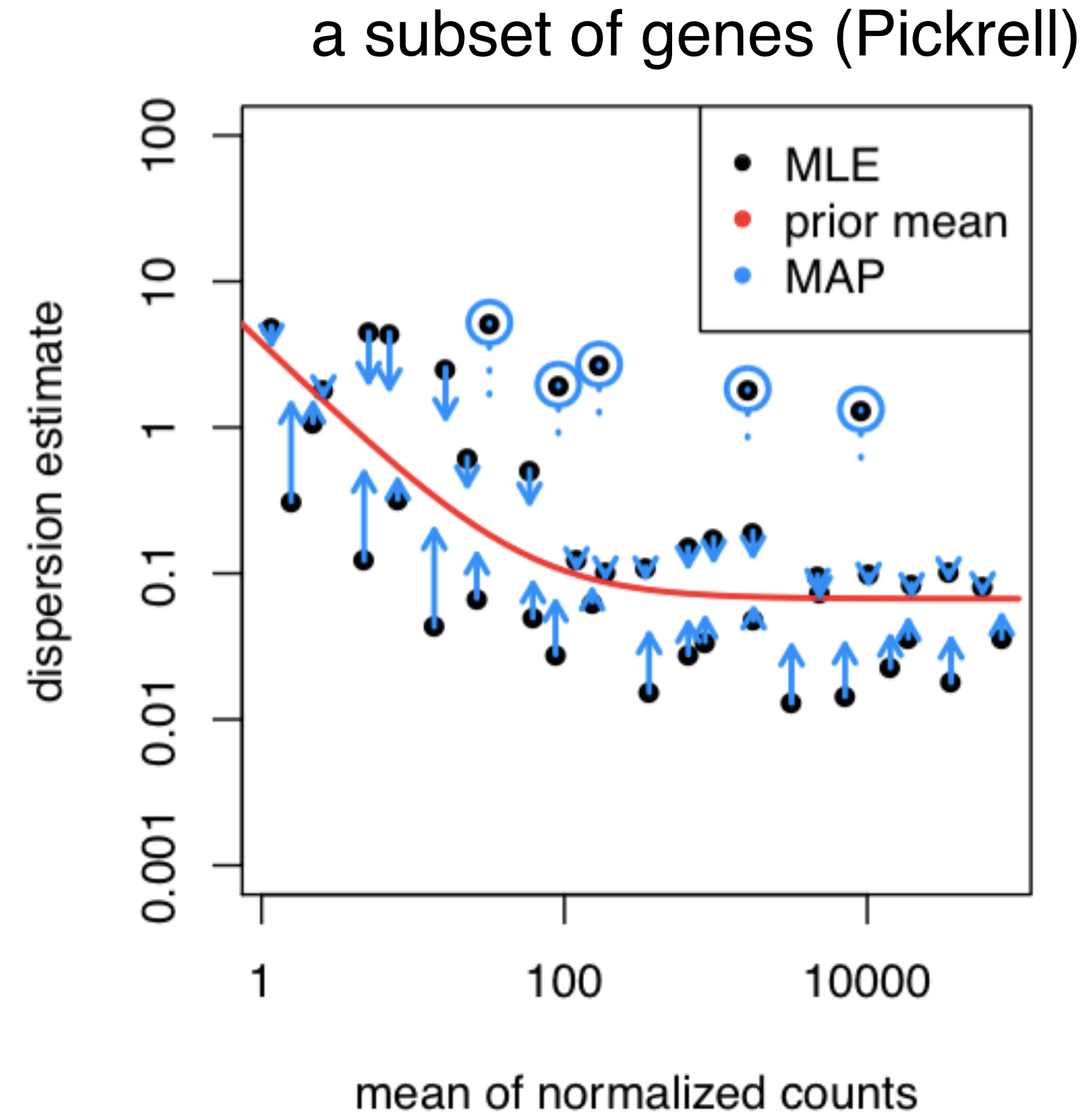
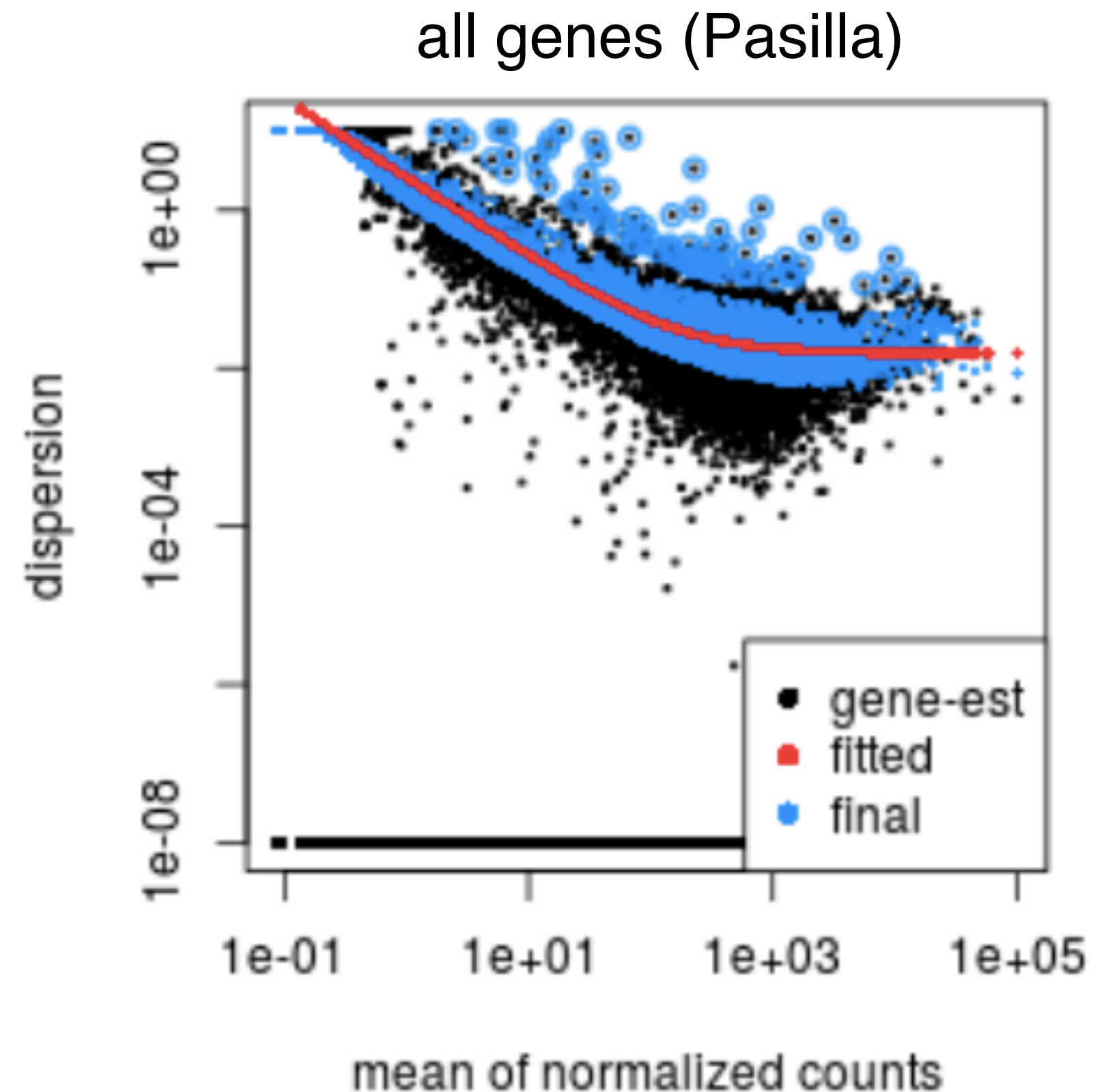
Extra variation  
due to biological variance



# Shrinkage and dispersion

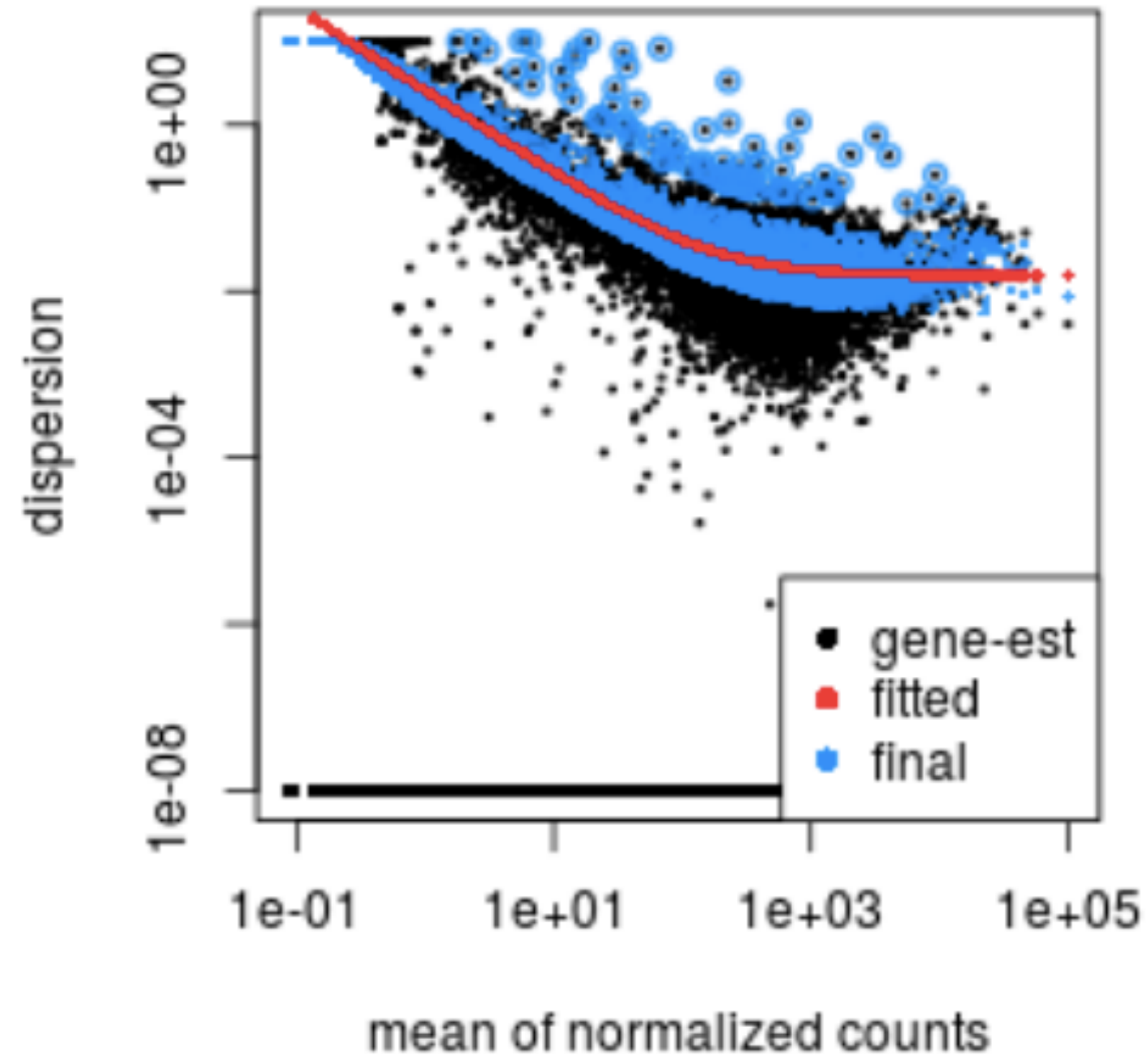
- Different genes naturally have different scales of biological variability
- Over all genes, there will be a distribution of reasonable estimates of dispersion
- With small sample size ( $n=3-5$  replicates per group), we will make *very bad* estimates of gene-wise dispersion unless we **share information across genes**

# Shrinkage of dispersion

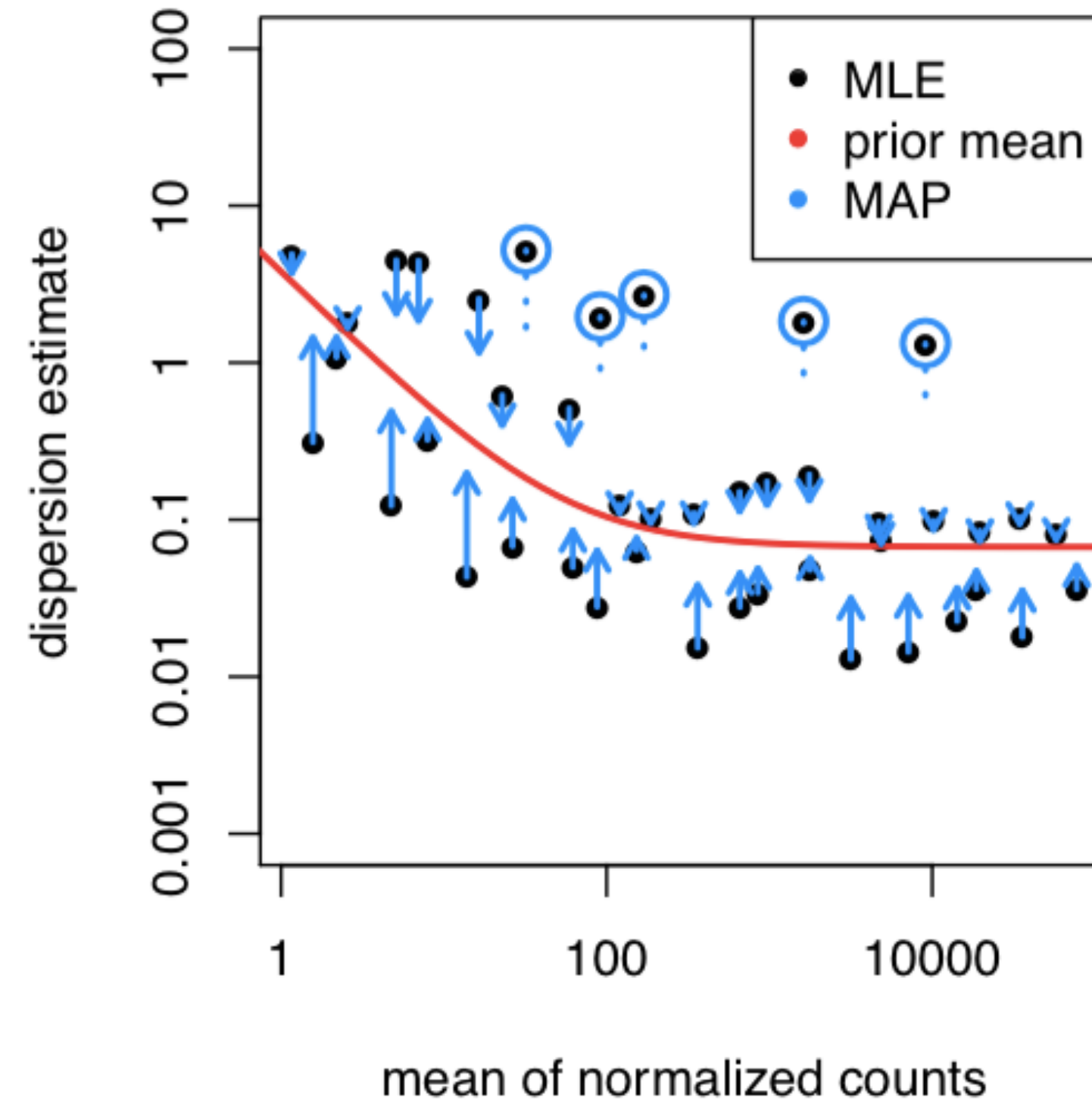


1. Gene-wise estimate = just look at one gene (MLE)
2. Fitted dispersion trend = the middle for the prior
3. Final estimate = posterior, uses shared information (MAP)

all genes (Pasilla)



a subset of genes (Pickrell)



The shrinkage procedure thereby helps avoid potential false positives, which can result from underestimates of dispersion

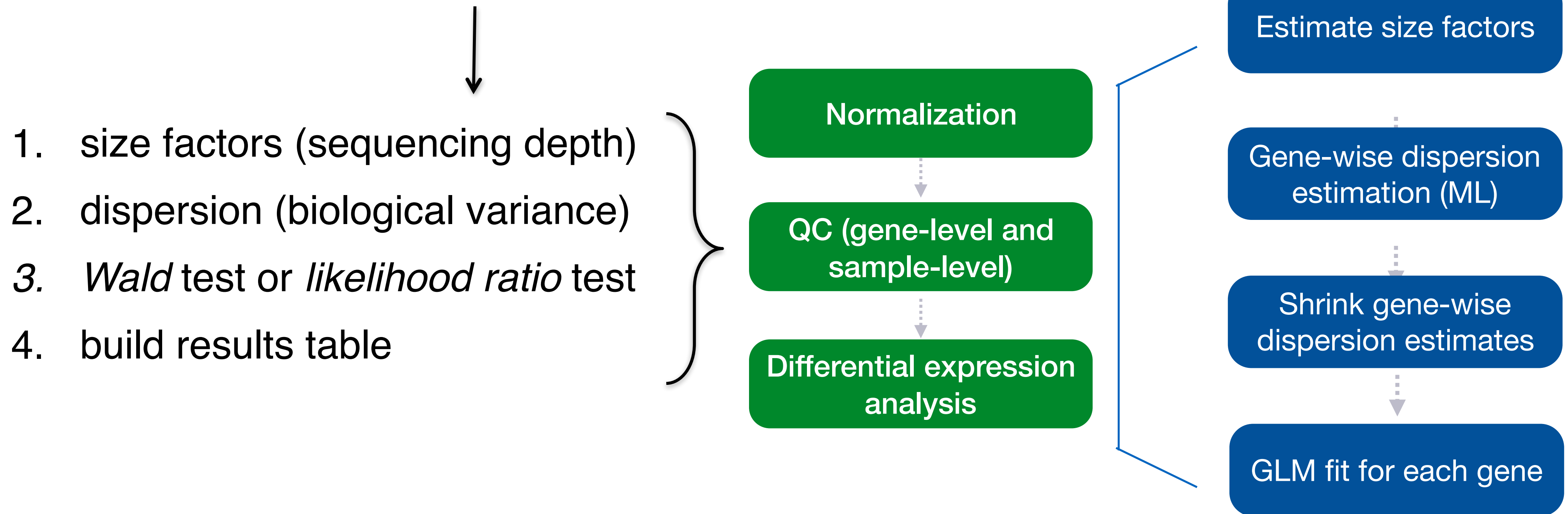
# Differences across conditions

# DESeq2

- Available through Bioconductor since 2013
- Publication: Genome Biology, Dec 2014.  
main text written with non-statisticians in mind
- Builds on good ideas for dispersion estimation and use of GLM from the [DSS](#) and [edgeR](#) methods
- See [bioconductor.org/install](http://bioconductor.org/install) for installation
- Note that the latest Bioconductor packages are only available with **latest R version**. Bioconductor and R versions are *linked*

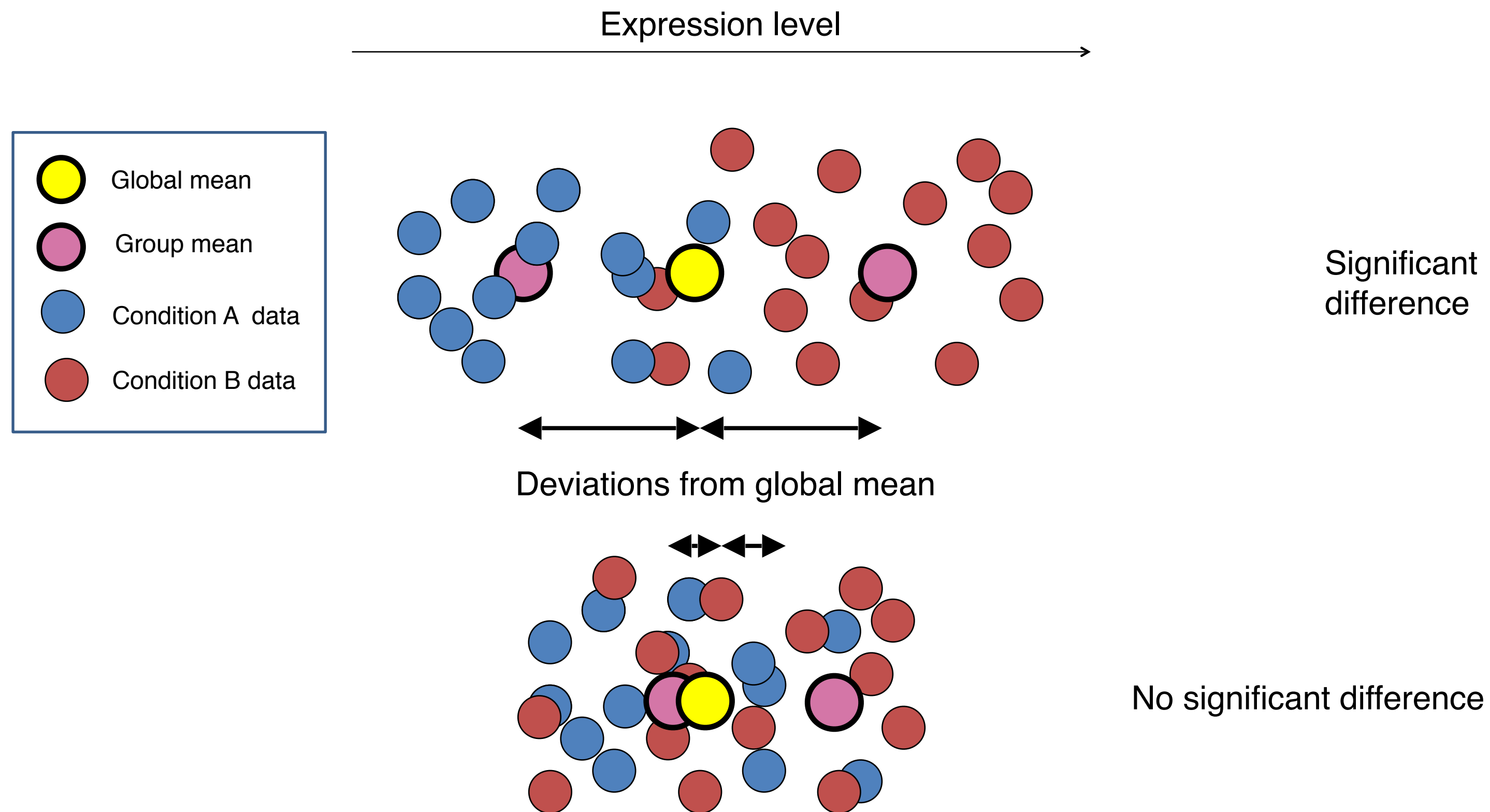
# DESeq2 steps

count matrix (from featureCounts, htseq, tximport, etc.)





# Differences across two conditions



# Differences across two conditions

- Describe experiment with formula, e.g.: ~ condition
- Per gene the design matrix looks like:

$$\begin{array}{lcl} \log_2 q1 & & 1 \quad 0 \\ \log_2 q2 & & 1 \quad 0 \\ \log_2 q3 & = & 1 \quad 1 \\ \log_2 q4 & & 1 \quad 1 \end{array}$$



# Differences across two conditions

- Describe experiment with formula, e.g.:  $\sim$  condition
- Per gene the design matrix looks like:

$$\begin{array}{l} \log_2 q1 \\ \log_2 q2 \\ \log_2 q3 \\ \log_2 q4 \end{array} = \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 1 & 0 \\ \hline 1 & 1 \\ \hline 1 & 1 \\ \hline \end{array} \quad \boxed{\text{Intercept}}$$

All samples get an Intercept term

# Differences across two conditions

- Describe experiment with formula, e.g.: ~ condition
- Per gene the design matrix looks like:

$$\begin{matrix} \log_2 q1 \\ \log_2 q2 \\ \log_2 q3 \\ \log_2 q4 \end{matrix} = \begin{matrix} \begin{matrix} 1 \\ 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} 0 \\ 0 \\ 1 \\ 1 \end{matrix} \end{matrix}$$

Intercept
condition B vs A

All samples get an Intercept term

The B condition samples also get a term that accounts for the difference between B and A

# Differences across multiple conditions

- The design matrix now uses a column for each condition

$\log_2 q1$		1	1	0	0	
$\log_2 q2$		1	1	0	0	Intercept
$\log_2 q3$	=	1	0	1	0	conditionA
$\log_2 q4$		1	0	1	0	conditionB
$\log_2 q5$		1	0	0	1	conditionC
$\log_2 q6$		1	0	0	1	

# Differences across multiple conditions

- The design matrix now uses a column for each condition

$\log_2 q1$	1	1	0	0	
$\log_2 q2$	1	1	0	0	Intercept
$\log_2 q3$	1	0	1	0	conditionA
$\log_2 q4$	1	0	1	0	conditionB
$\log_2 q5$	1	0	0	1	conditionC
$\log_2 q6$	1	0	0	1	

# Differences across multiple conditions

- The design matrix now uses a column for each condition

$\log_2 q1$	1	1	0	0	
$\log_2 q2$	1	1	0	0	Intercept
$\log_2 q3$	1	0	1	0	conditionA
$\log_2 q4$	1	0	1	0	conditionB
$\log_2 q5$	1	0	0	1	conditionC
$\log_2 q6$	1	0	0	1	

# Differences across multiple conditions

- The design matrix now uses a column for each condition

$\log_2 q1$		1	1	0	0	
$\log_2 q2$		1	1	0	0	Intercept
$\log_2 q3$	=	1	0	1	0	conditionA
$\log_2 q4$		1	0	1	0	conditionB
$\log_2 q5$		1	0	0	1	conditionC
$\log_2 q6$		1	0	0	1	

# Differences across multiple conditions

- The design matrix now uses a column for each condition

$\log_2 q1$		1	1	0	0	
$\log_2 q2$		1	1	0	0	Intercept
$\log_2 q3$	=	1	0	1	0	conditionA
$\log_2 q4$		1	0	1	0	conditionB
$\log_2 q5$		1	0	0	1	conditionC
$\log_2 q6$		1	0	0	1	

# Differences across multiple conditions

- The design matrix now uses a column for each condition

$\log_2 q1$		1	1	0	0	
$\log_2 q2$		1	1	0	0	Intercept
$\log_2 q3$	=	1	0	1	0	conditionA
$\log_2 q4$		1	0	1	0	conditionB
$\log_2 q5$		1	0	0	1	conditionC
$\log_2 q6$		1	0	0	1	



# GLM: Generalized linear models

- ▶ Extension of linear models to non-normally distributed response data (in our case, negative binomial)
- ▶ Helps address the different mean-variance relationships
- ▶ GLM fit for a gene will return **coefficients indicating the overall expression** strength of the gene for each design matrix element

# Contrasts

- A contrast is the comparison of coefficients we choose to evaluate

$\log_2 q1$		1	1	0	0	
$\log_2 q2$		1	1	0	0	Intercept
$\log_2 q3$	=	1	0	1	0	conditionA
$\log_2 q4$		1	0	1	0	conditionB
$\log_2 q5$		1	0	0	1	conditionC
$\log_2 q6$		1	0	0	1	

# Contrasts

- A contrast is the comparison of coefficients we choose to evaluate

$\log_2 q1$		1	1	0	0	
$\log_2 q2$		1	1	0	0	Intercept
$\log_2 q3$	=	1	0	1	0	conditionA
$\log_2 q4$		1	0	1	0	conditionB
$\log_2 q5$		1	0	0	1	conditionC
$\log_2 q6$		1	0	0	1	

# Contrasts

- A contrast is the comparison of coefficients we choose to evaluate

$\log_2 q1$		1	1	0	0	
$\log_2 q2$		1	1	0	0	Intercept
$\log_2 q3$	=	1	0	1	0	conditionA
$\log_2 q4$		1	0	1	0	conditionB
$\log_2 q5$		1	0	0	1	conditionC
$\log_2 q6$		1	0	0	1	

# Contrasts

- A contrast is the comparison of coefficients we choose to evaluate

$\log_2 q1$		1	1	0	0	
$\log_2 q2$		1	1	0	0	Intercept
$\log_2 q3$	=	1	0	1	0	conditionA
$\log_2 q4$		1	0	1	0	conditionB
$\log_2 q5$		1	0	0	1	conditionC
$\log_2 q6$		1	0	0	1	

```
results(dds, contrast=c("condition", "B", "C"))
```

# Hypothesis testing

## Wald test

- ▶ use the shrunken estimate of the  $\log_2$  fold-change divided by the SE
- ▶ this gives a Z-statistic which is compared to a standard normal distribution
- ▶ allows testing of individual coefficients, or contrasts of coefficients (i.e. two-level comparison)

# Hypothesis testing

## Wald test

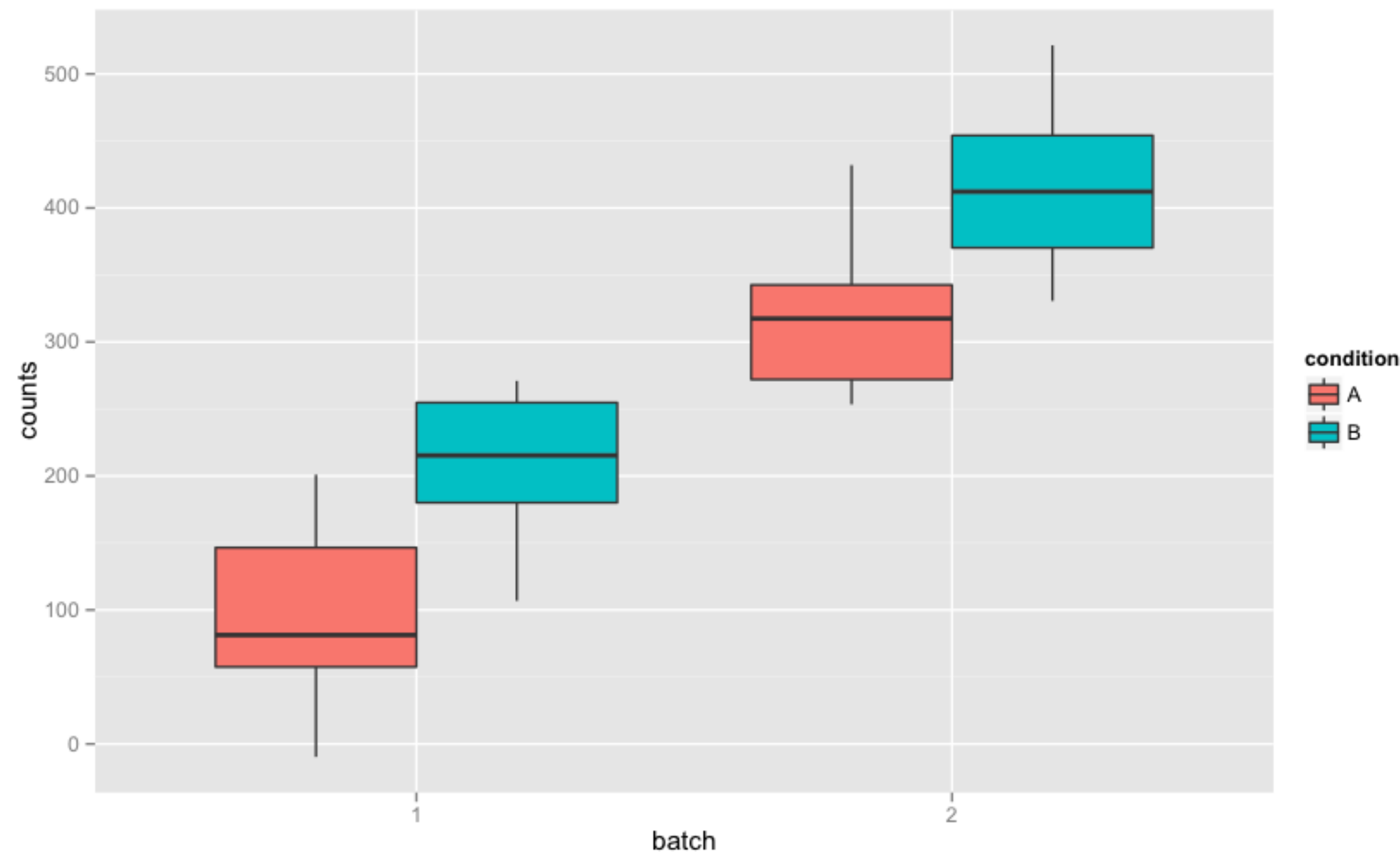
- ▶ use the shrunken estimate of the log2 fold-change divided by the SE
- ▶ this gives a Z-statistic which is compared to a standard normal distribution
- ▶ allows testing of individual coefficients, or contrasts of coefficients (i.e. two-level comparison)

## Likelihood ratio test (LRT)

- ▶ examines two different models: full and reduced model (with some terms removed)
- ▶ determines if the increased likelihood of the data using the extra terms in the full model is more than expected if those extra terms are truly zero
- ▶ useful for identifying any gene that is changing in expression with respect to the biological factor of interest (useful for 3 or more levels)

# Controlling for different batches

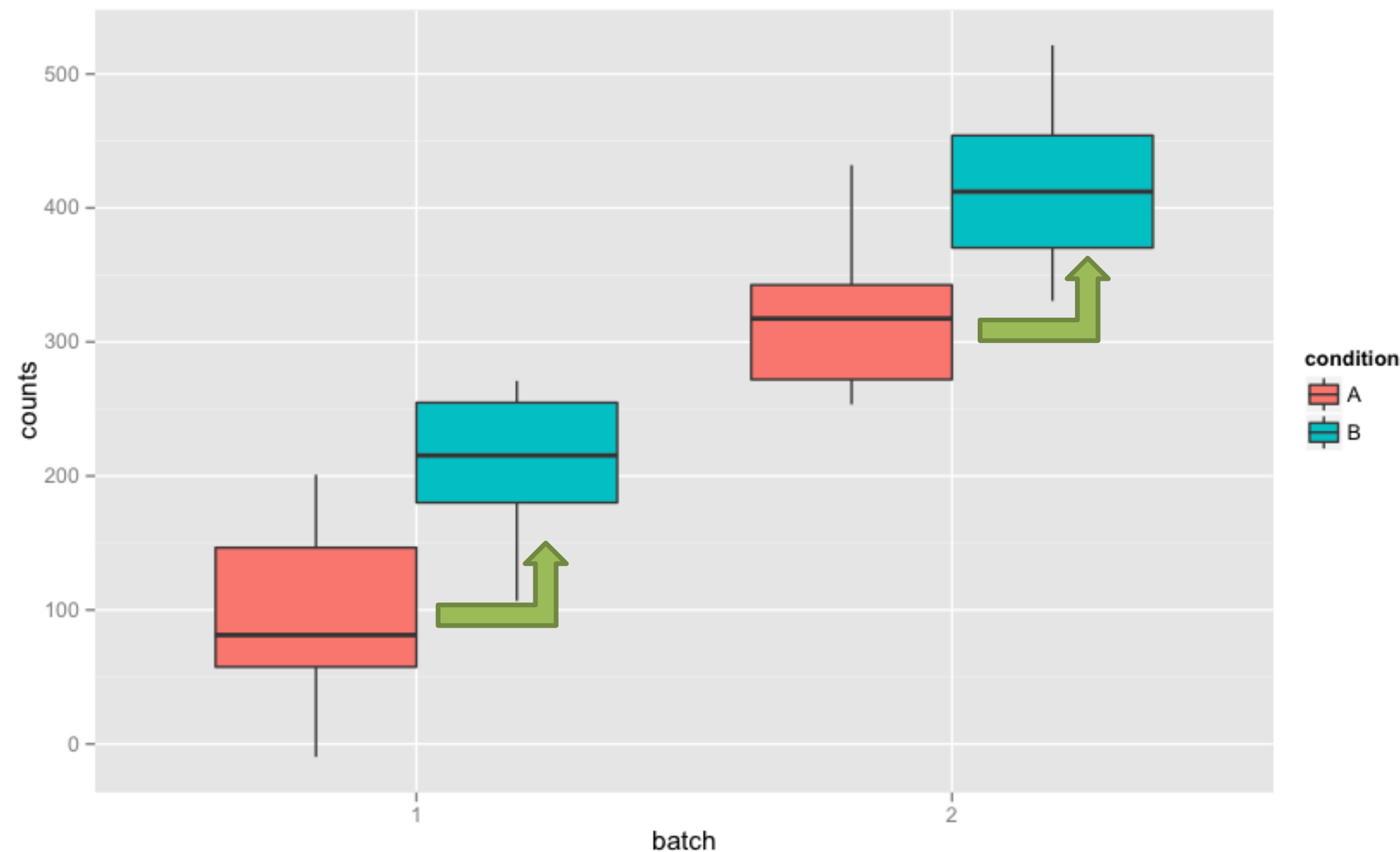
- Using a design formula:  $\sim \text{batch} + \text{condition}$ , adds terms that control for batch differences
- If batches are unknown, possible to detect these with other methods: [svaseq](#), [RUVSeq](#)





# Controlling for different batches

- Using a design formula:  $\sim \text{batch} + \text{condition}$ , adds terms that control for batch differences
- If batches are unknown, possible to detect these with other methods: [svaseq](#), [RUVSeq](#)



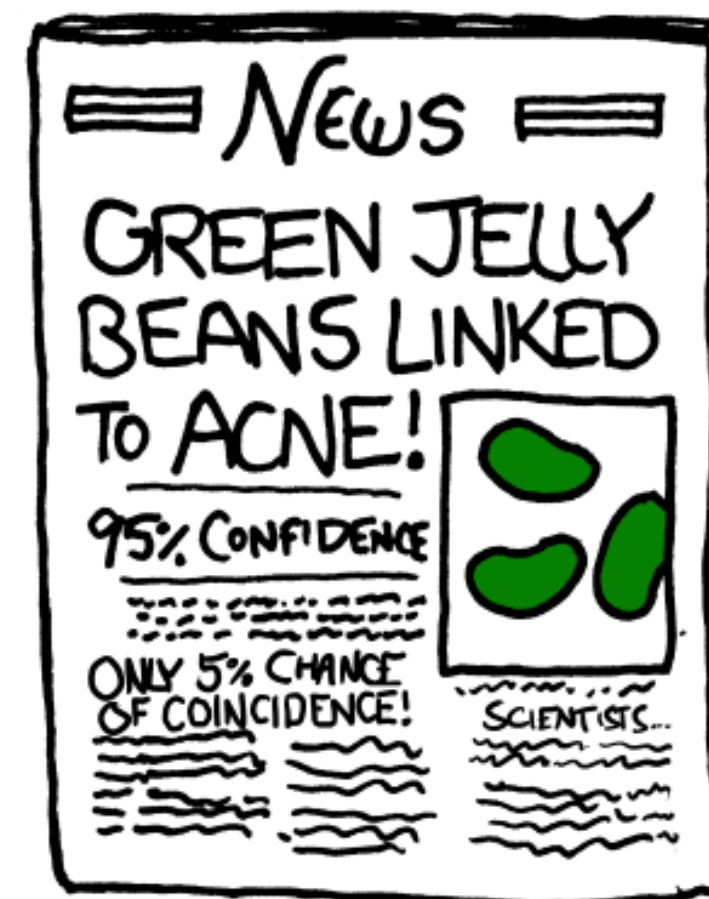
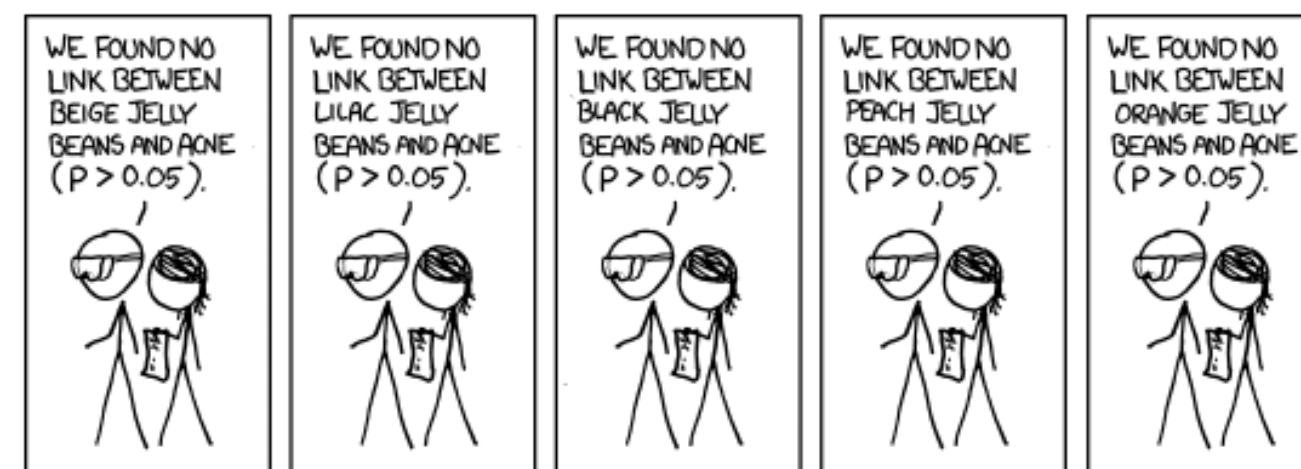
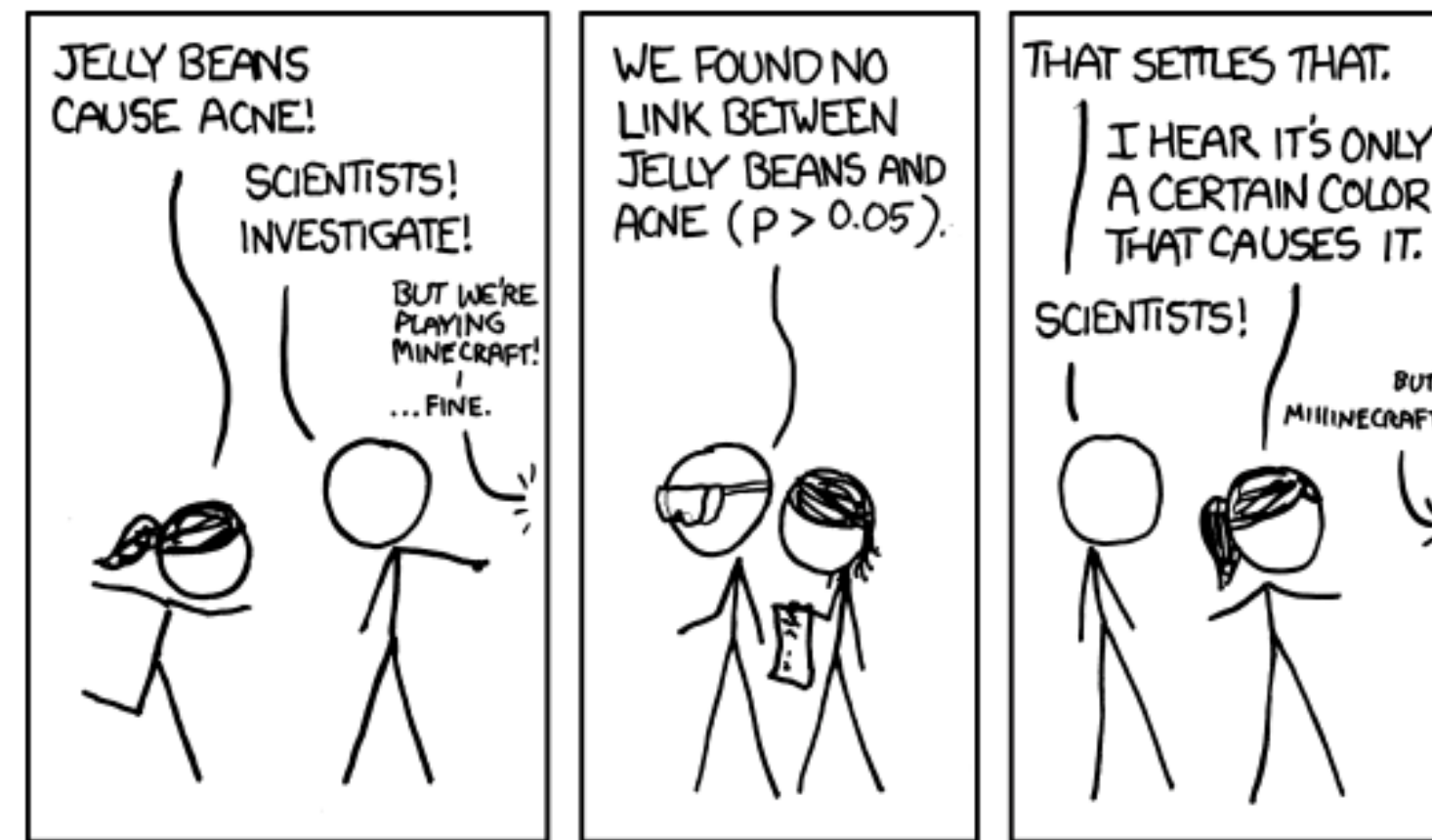
# Complex designs

Want to test: treatment changes for enriched samples over baseline, controlling for individual effects

~individual + enrichment + treatment +  
enrichment:treatment

indiv	.	enrich.	treat.
1	input	control	
1	IP	control	
1	input	treat	
1	IP	treat	
2	input	control	
2	IP	control	
2	input	treat	
2	IP	treat	
...			

# Multiple test correction



# Multiple test correction

# Multiple test correction

- ▶ In our example using condition A vs. condition B we found 2,898 genes to be differentially expressed at  $p < 0.05$

# Multiple test correction

- ▶ In our example using condition A vs. condition B we found 2,898 genes to be differentially expressed at  $p < 0.05$
- ▶ This is 12% of the genes we tested

# Multiple test correction

- ▶ In our example using condition A vs. condition B we found 2,898 genes to be differentially expressed at  $p < 0.05$
- ▶ This is 12% of the genes we tested
- ▶ Expect 5% by chance (this is what  $p < 0.05$ )



# Multiple test correction

- ▶ In our example using condition A vs. condition B we found 2,898 genes to be differentially expressed at  $p < 0.05$
- ▶ This is 12% of the genes we tested
- ▶ Expect 5% by chance (this is what  $p < 0.05$ )
- ▶ Probably  $\sim 1/3$  genes we found are false positives

# Multiple test correction

- ▶ In our example using condition A vs. condition B we found 2,898 genes to be differentially expressed at  $p < 0.05$
- ▶ This is 12% of the genes we tested
- ▶ Expect 5% by chance (this is what  $p < 0.05$ )
- ▶ Probably  $\sim 1/3$  genes we found are false positives

This is the multiple testing problem. The more tests we perform, the more we inflate the number of false positives observed.

# Multiple test correction

## Controlling the FWER

---

- ▶ Control  $\alpha$ , the probability of making an error (false positive)
- ▶ **Bonferroni:** Reject any hypothesis with  $p\text{-value} \leq \alpha/m$ 
  - ▶ Conservative; high probability of false negatives

## Controlling the FDR

---

- ▶ FDR: false discovery rate: the expected percent of false predictions in the set of predictions
- ▶ **Benjamini-Hochberg:** Rank  $j / m$  multiplied by the FDR level
  - ▶ designed to control the FDR
- ▶ **Q-value:** The minimum FDR that can be attained when calling that feature significant

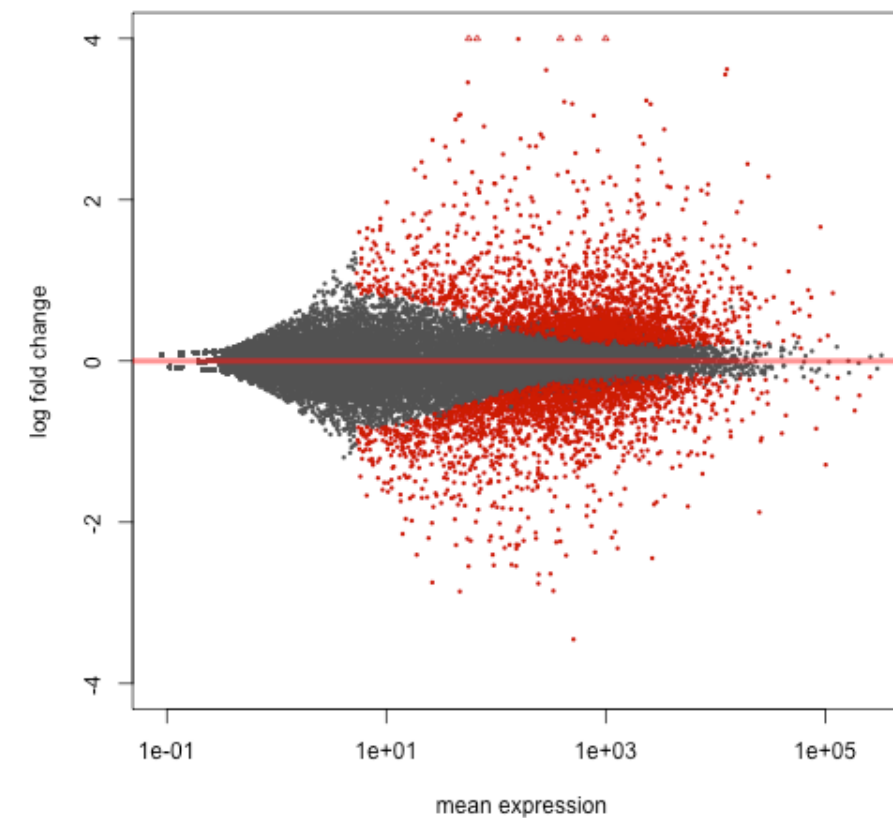
# DE vs EDA

# Two paths in DESeq2

Count matrix

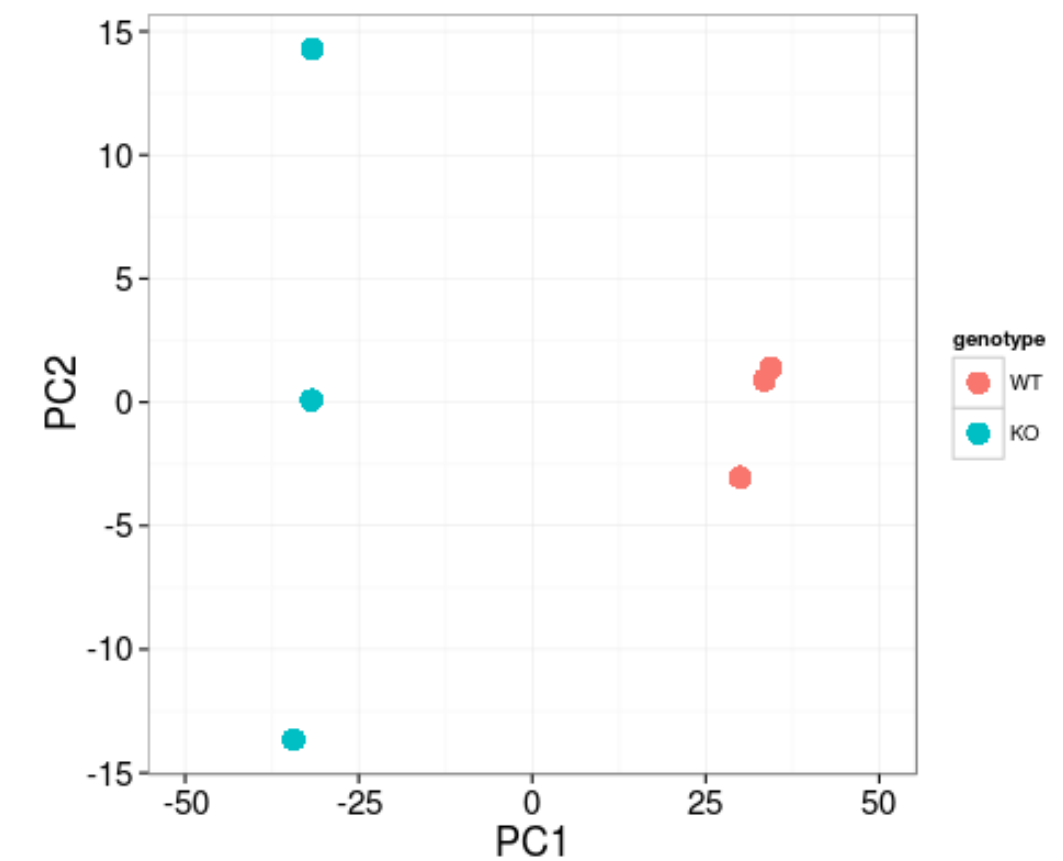
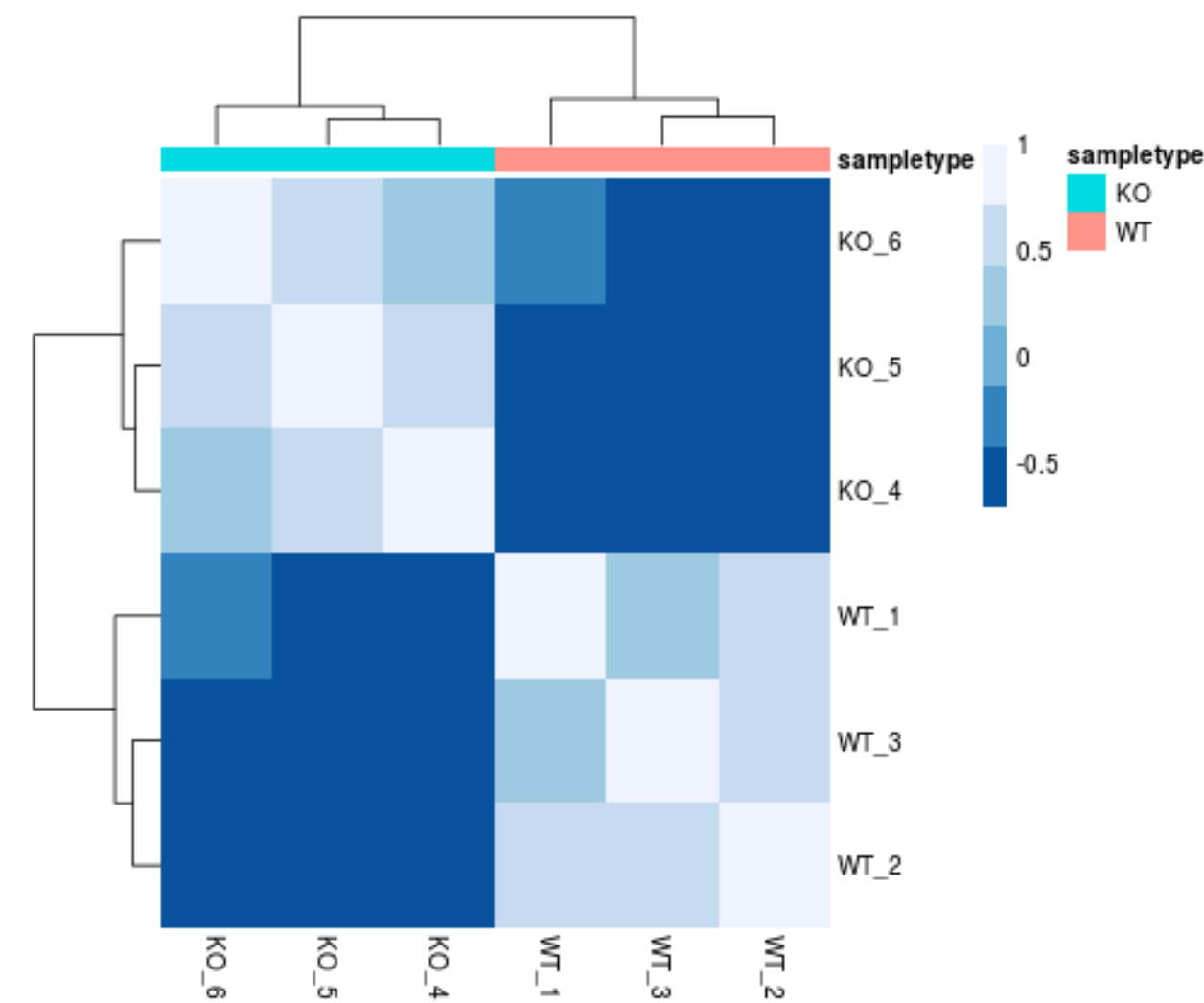
Differential expression

testing, p-values, FDR



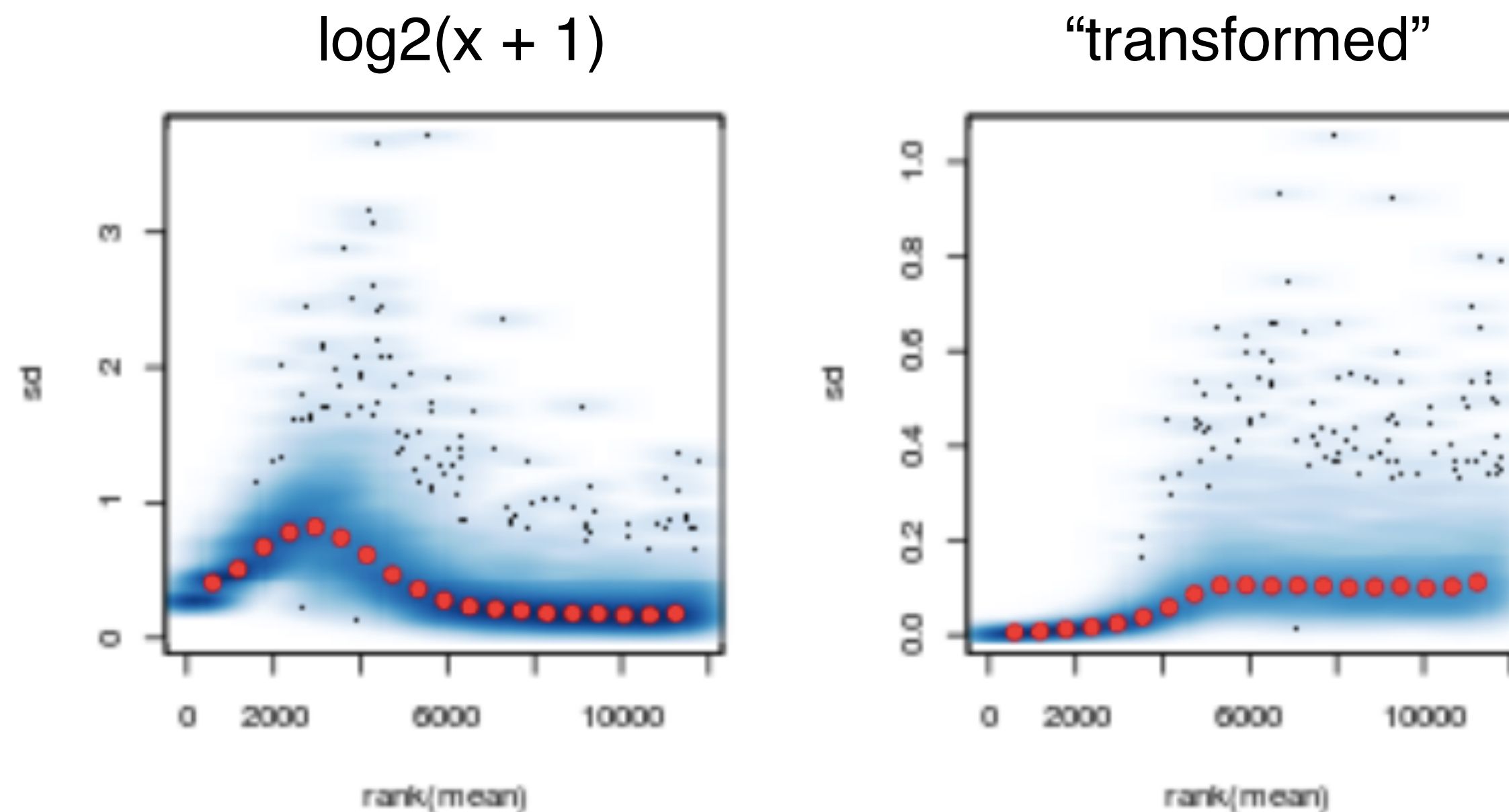
Transformations and  
Exploratory Data Analysis (EDA)

clustering, heatmaps,  
sample-sample distances



# Transformations

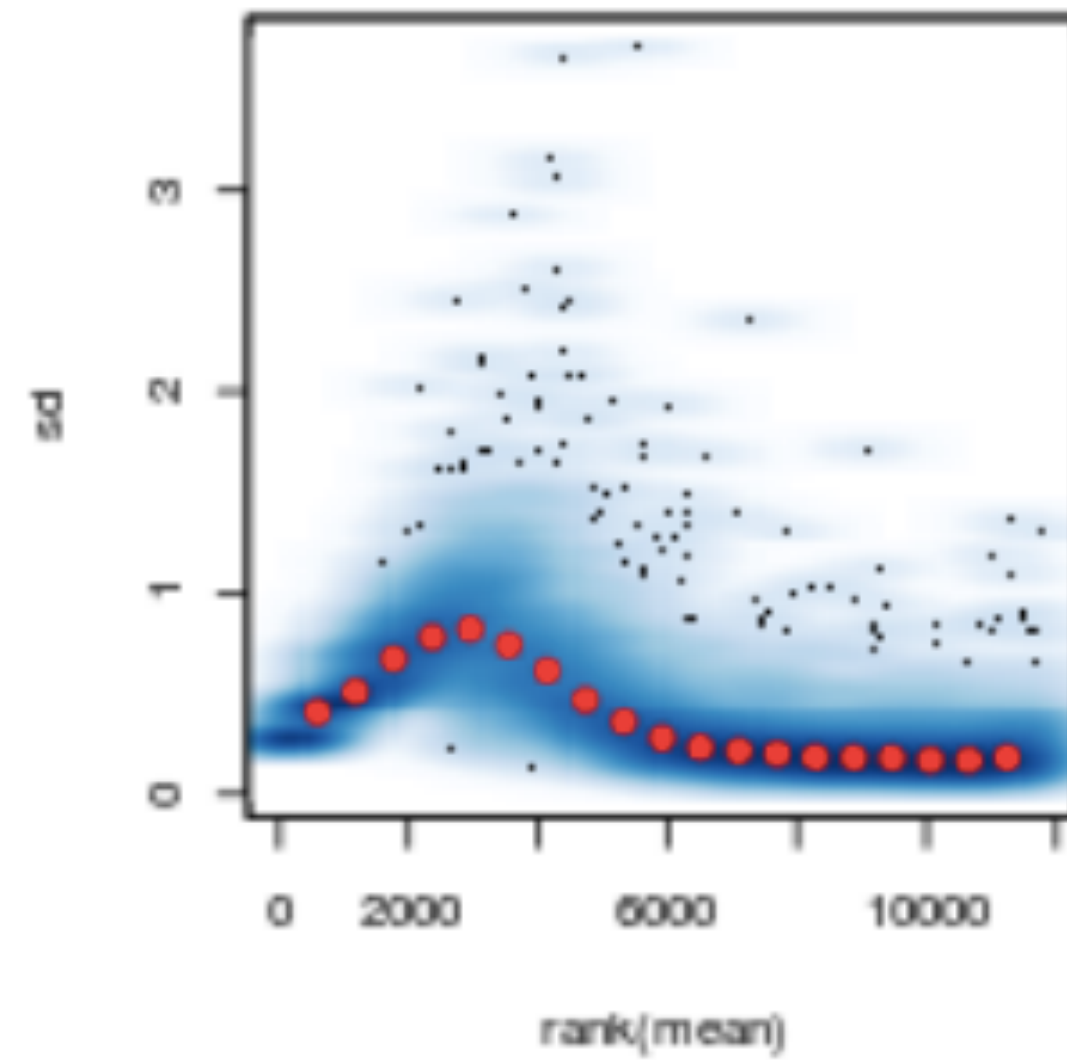
For comparison analyses when using unsupervised techniques, it can be useful to *transform* data. These techniques (VST and rlog) perform better when values have a similar dynamic range



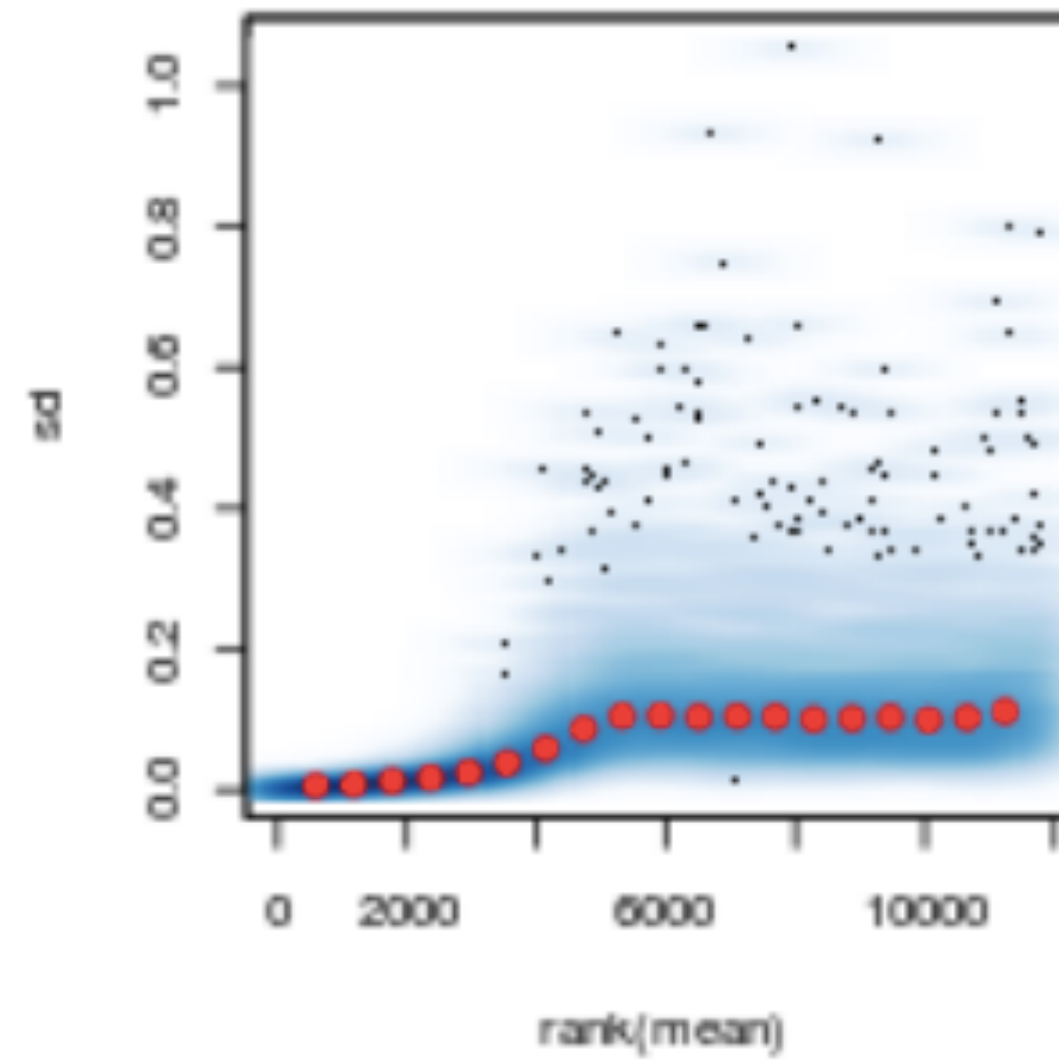
renders data *homoskedastic*  
(variance of the gene is  
stabilized across expression  
levels)

# rlog stabilizes variances along the mean

$\log_2(x + 1)$



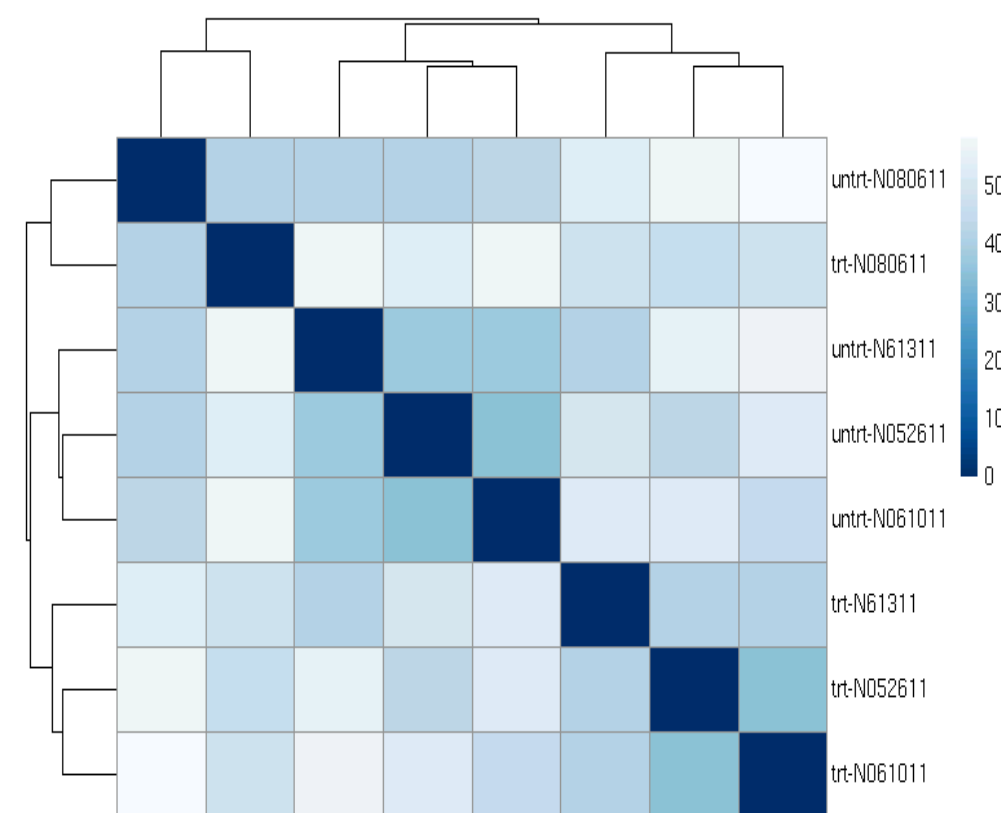
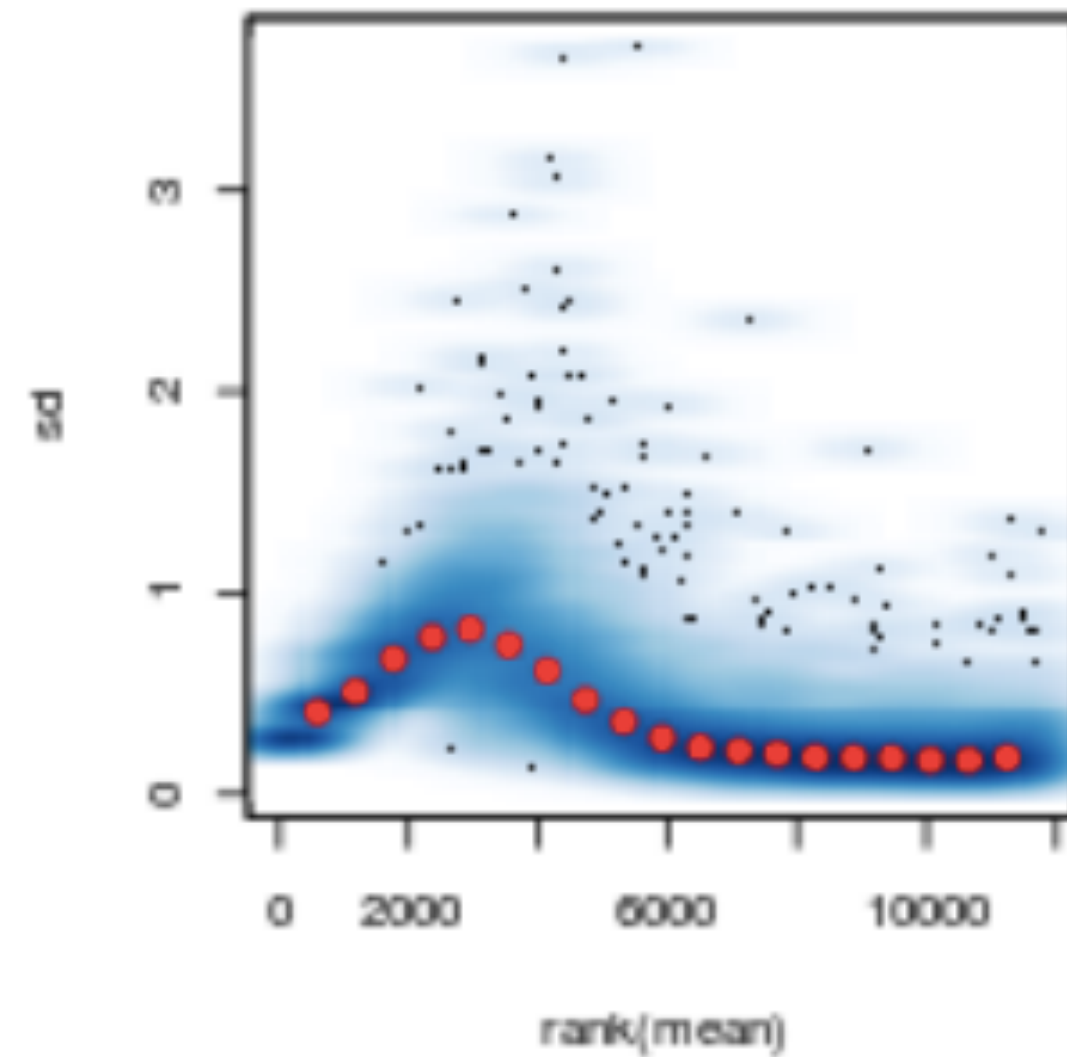
"rlog"



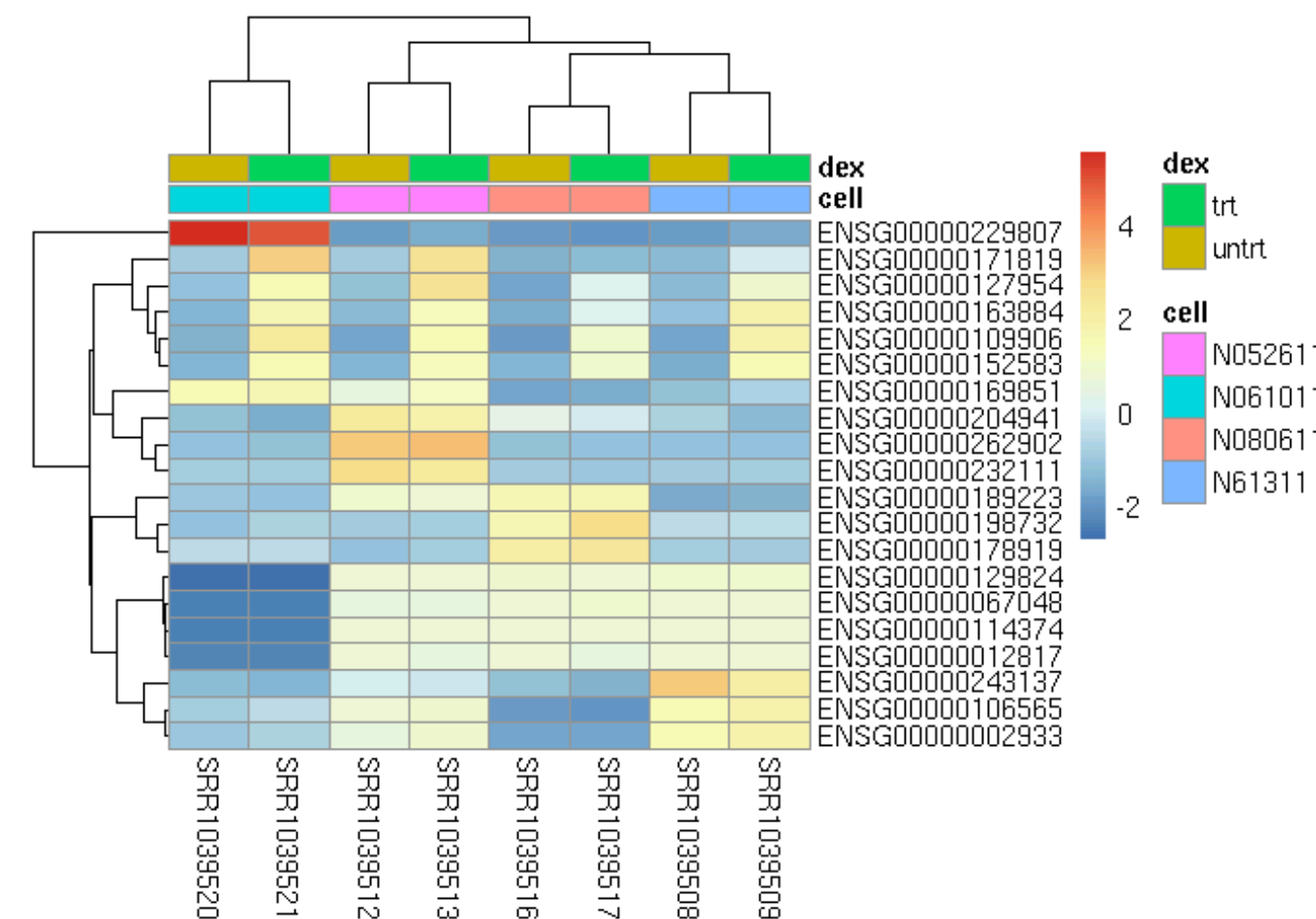
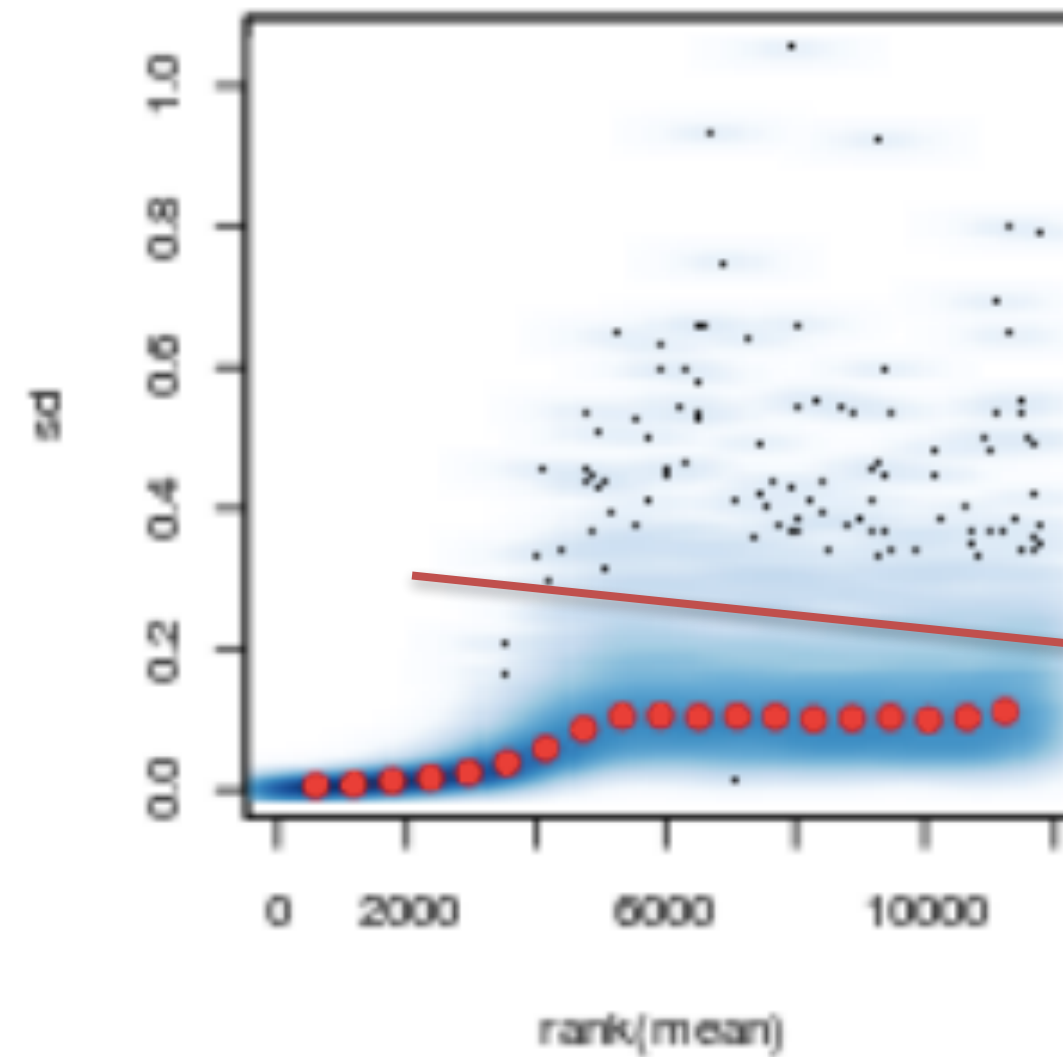


# rlog stabilizes variances along the mean

$\log_2(x + 1)$



"rlog"



Moderating the high variance / low count genes  
Improves distances, clustering, visualizations



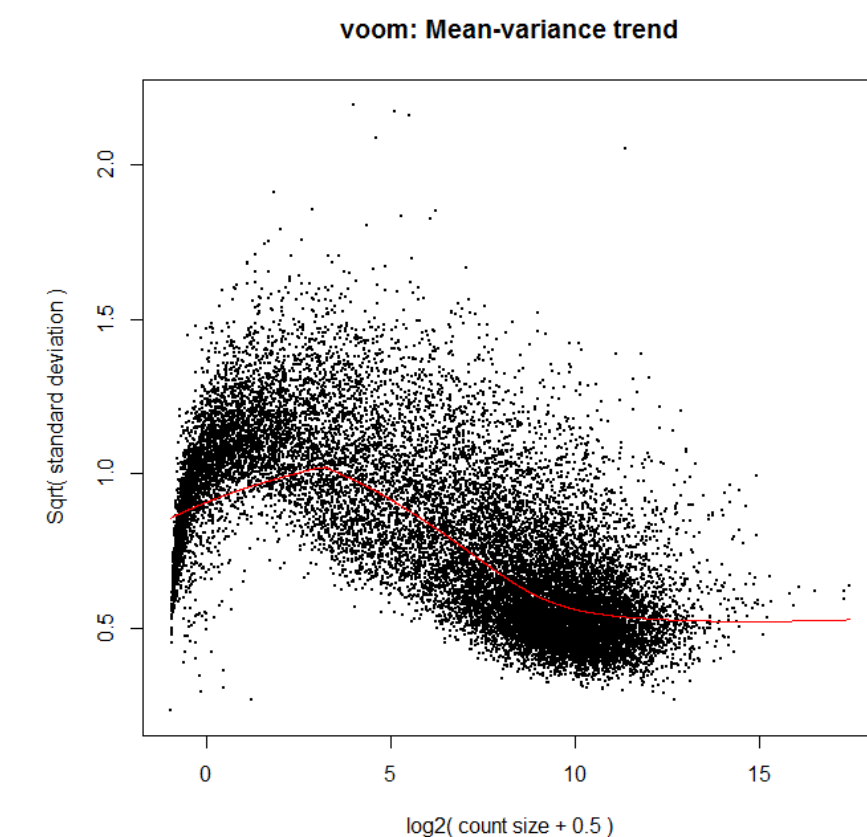
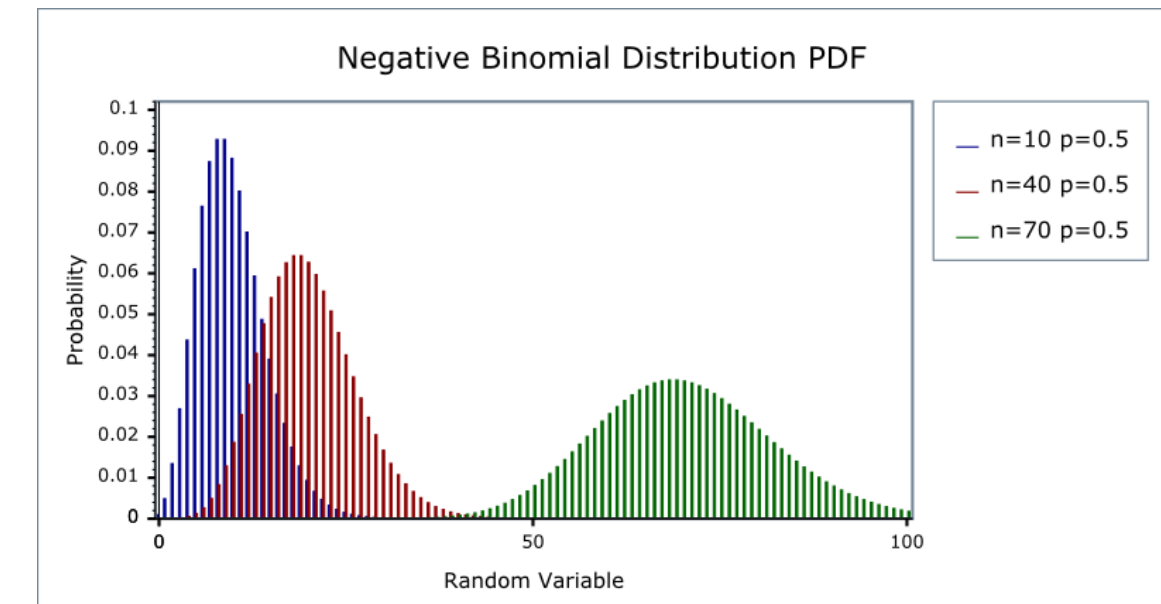
# Variance stabilizing transform (VST)

- The variance stabilizing transformation is an earlier approach (from original DESeq) for transforming counts.
- Uses a function which is log-like but doesn't go to  $-\text{Inf}$  at  $x=0$ .
- VST doesn't use size factors so is **better for data that has consistent sequence depth across samples**
- VST is closed form, so can be **better for large datasets ( $n > 50$ )** due to speed.

# Comparison of methods

# Count model vs linear model

- **DESeq2** and **edgeR** similar approach, similar results
  - very sensitive, may sometimes underestimate FDR
- **limma+voom** uses a linear model, weights determined by variance over mean
  - strong control of FDR, may be less sensitive for small sample size
  - recommended when number of biological replicates per group grows large (e.g. > 20)



# Credits

## *RNA-seq statistical analysis and gene-level differential expression*

Mike Love @mikelove

Dept. of Biostatistics and Computational Biology

Dana-Farber Cancer Institute & Harvard TH Chan School of Public Health

### Acknowledgements:

DESeq2/DEXSeq:

Wolfgang Huber

Simon Anders

Alejandro Reyes

Work supported by:

Rafael Irizarry

Dana-Farber Cancer  
Institute

Harvard TH Chan School  
of Public Health

NIH Training Grant

