Differential expression analysis with DESeq2

Isabelle Dupanloup
BCF - Bioinformatics Core Facility
SIB - Swiss Institute of Bioinformatics
isabelle.dupanloup@sib.swiss

Swiss Institute of Bioinformatics

Teaching material from
Harvard Chan Bioinformatics Core training

QC methods for DE analysis using DESeq2

Learning Objectives

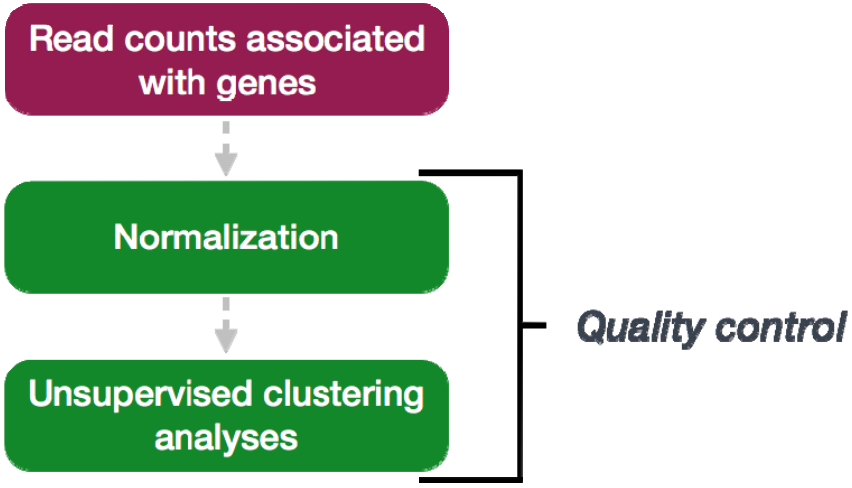
- Transforming counts for unsupervised clustering methods
- Evaluating quality of samples using Principal Components Analysis
- Hierarchical clustering of samples in the dataset

QC methods for DE analysis using DESeq2

Quality Control

- At the sample-level
- At the gene-level

QC checks on the count data to help us ensure that the samples/replicates look good



QC methods for DE analysis using DESeq2

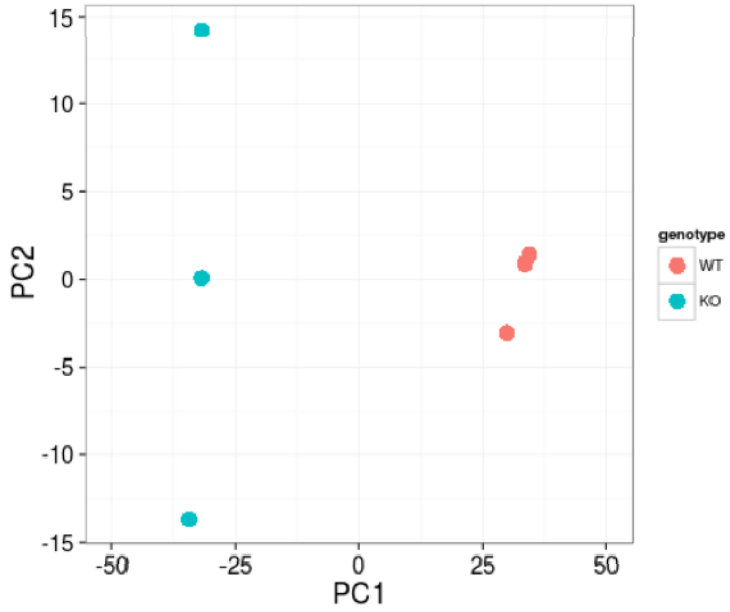
Sample-level QC

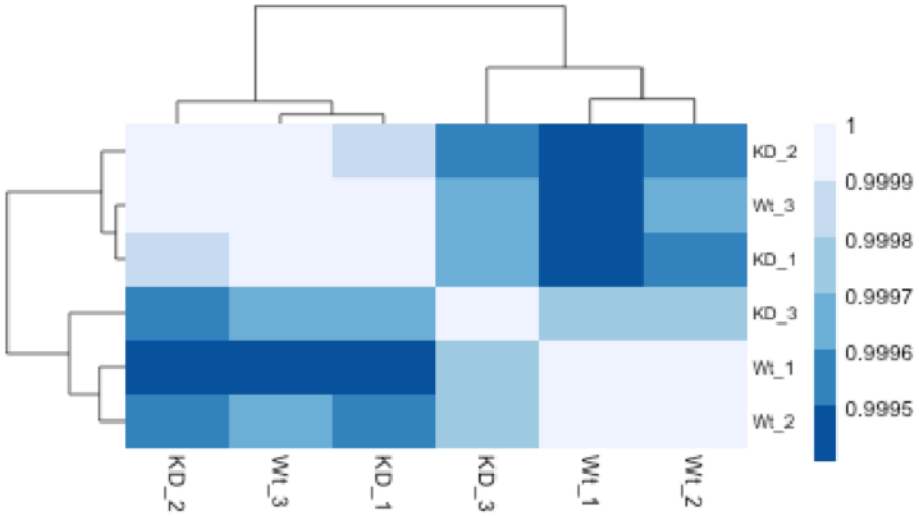
A useful initial step in an RNA-seq analysis is often to assess overall similarity between samples:

- which samples are similar to each other, which are different ?
- does this fit to the expectation from the experiment's design ?
- what are the major sources of variation in the dataset ?

QC methods for DE analysis using DESeq2

Sample-level QC

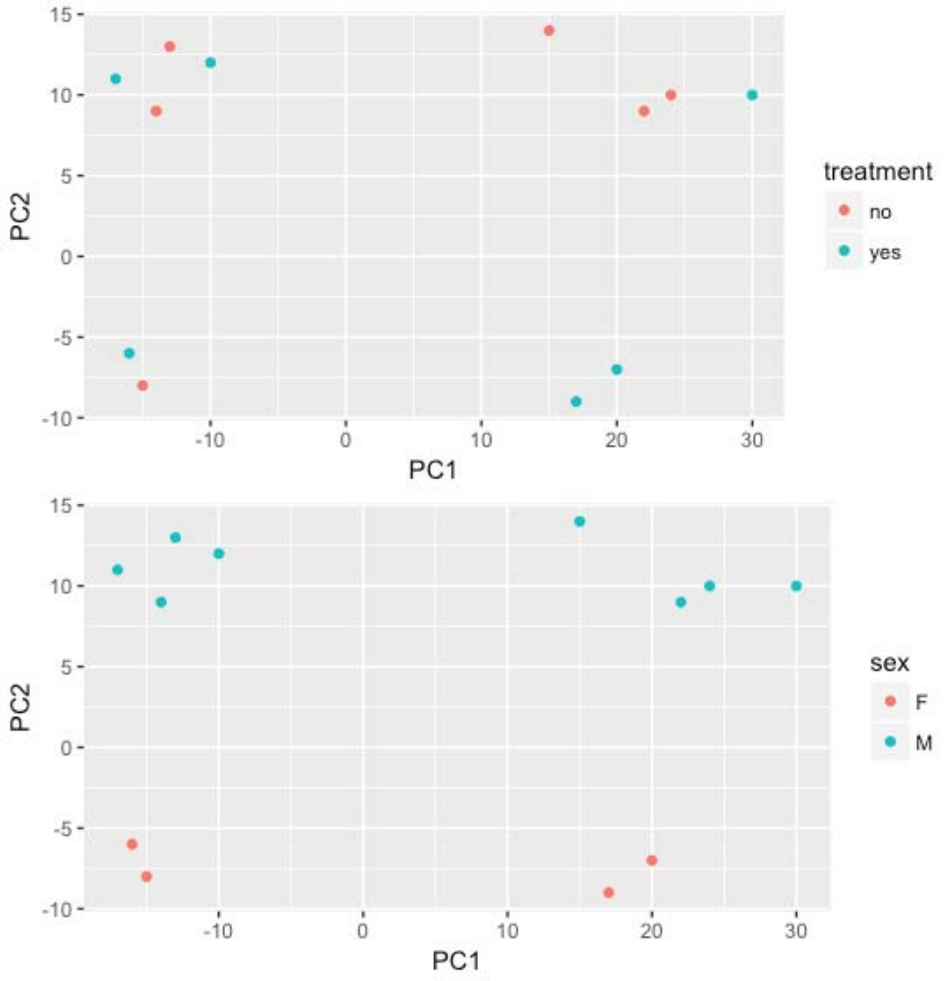




QC methods for DE analysis using DESeq2

Interpreting PCA plots

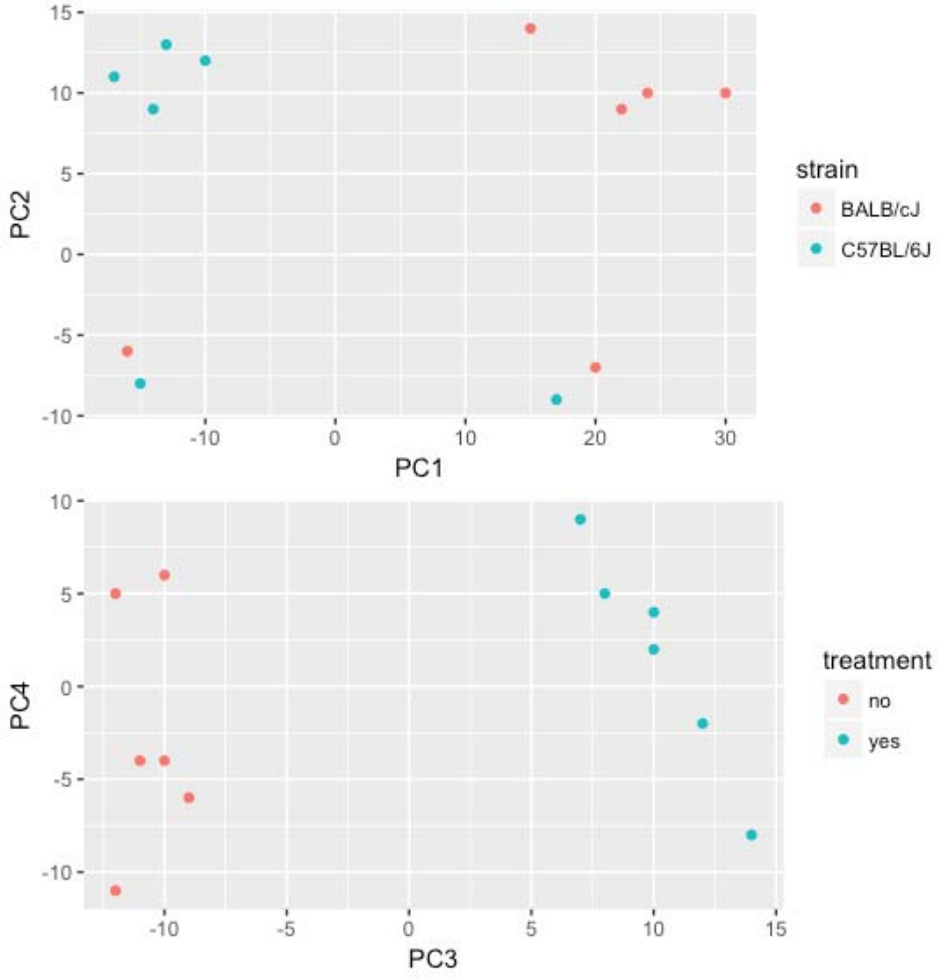
sample	strain	date	cage	treatment	replicate	sex
B1	BALB/cJ	20180515	1	yes	1	M
B2	C57BL/6J	20180515	2	yes	1	M
B3	BALB/cJ	20180515	3	no	1	M
B4	C57BL/6J	20180515	1	no	1	F
B5	BALB/cJ	20180515	2	yes	2	F
B6	C57BL/6J	20180515	3	yes	2	M
B7	BALB/cJ	20180515	1	no	2	M
B8	C57BL/6J	20180515	2	no	2	M
B9	BALB/cJ	20180515	3	yes	3	F
B10	C57BL/6J	20180307	1	yes	3	F
B11	BALB/cJ	20180307	2	no	3	M
B12	C57BL/6J	20180307	3	no	3	M



QC methods for DE analysis using DESeq2

Interpreting PCA plots

sample	strain	date	cage	treatment	replicate	sex
B1	BALB/cJ	20180515	1	yes	1	M
B2	C57BL/6J	20180515	2	yes	1	M
B3	BALB/cJ	20180515	3	no	1	M
B4	C57BL/6J	20180515	1	no	1	F
B5	BALB/cJ	20180515	2	yes	2	F
B6	C57BL/6J	20180515	3	yes	2	M
B7	BALB/cJ	20180515	1	no	2	M
B8	C57BL/6J	20180515	2	no	2	M
B9	BALB/cJ	20180515	3	yes	3	F
B10	C57BL/6J	20180307	1	yes	3	F
B11	BALB/cJ	20180307	2	no	3	M
B12	C57BL/6J	20180307	3	no	3	M



QC methods for DE analysis using DESeq2

Interpreting PCA plots

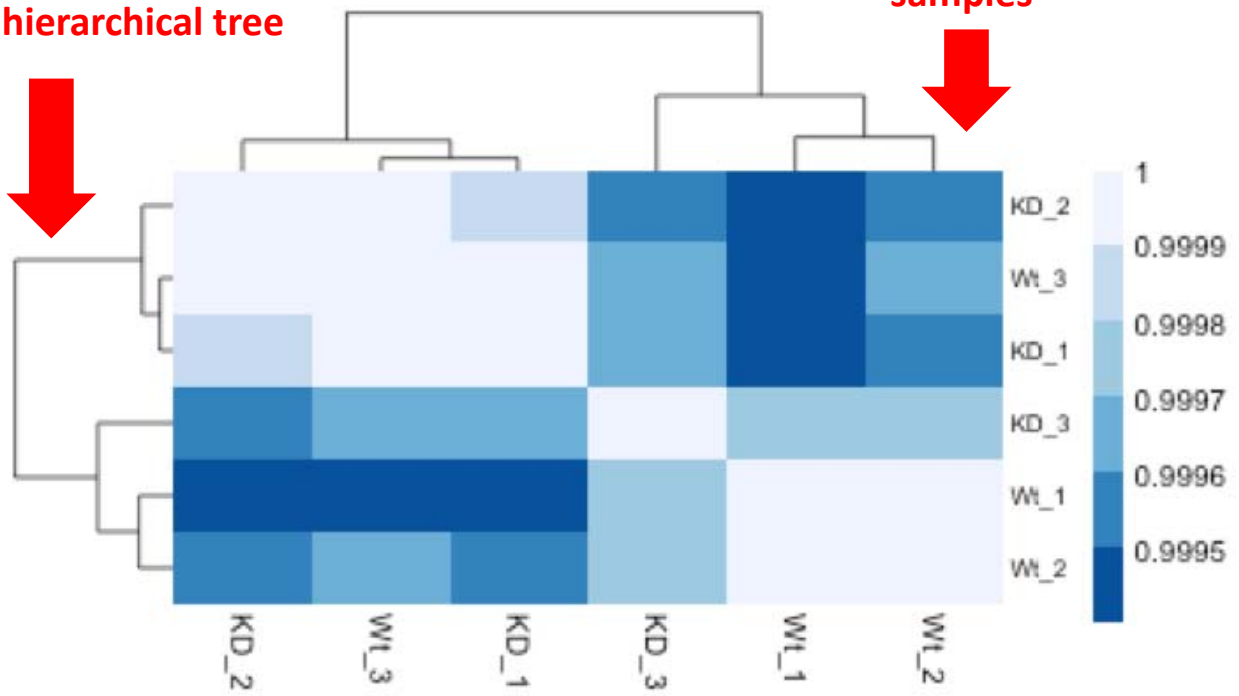
- if low level of variation explained by the first PCs, you **may want to explore more and other factors**
- **where you can identify those sources of variation, it is important to account for these in your model**, as it provides more power to the tool for detecting DE genes

QC methods for DE analysis using DESeq2

Hierarchical Clustering Heatmap

hierarchical tree

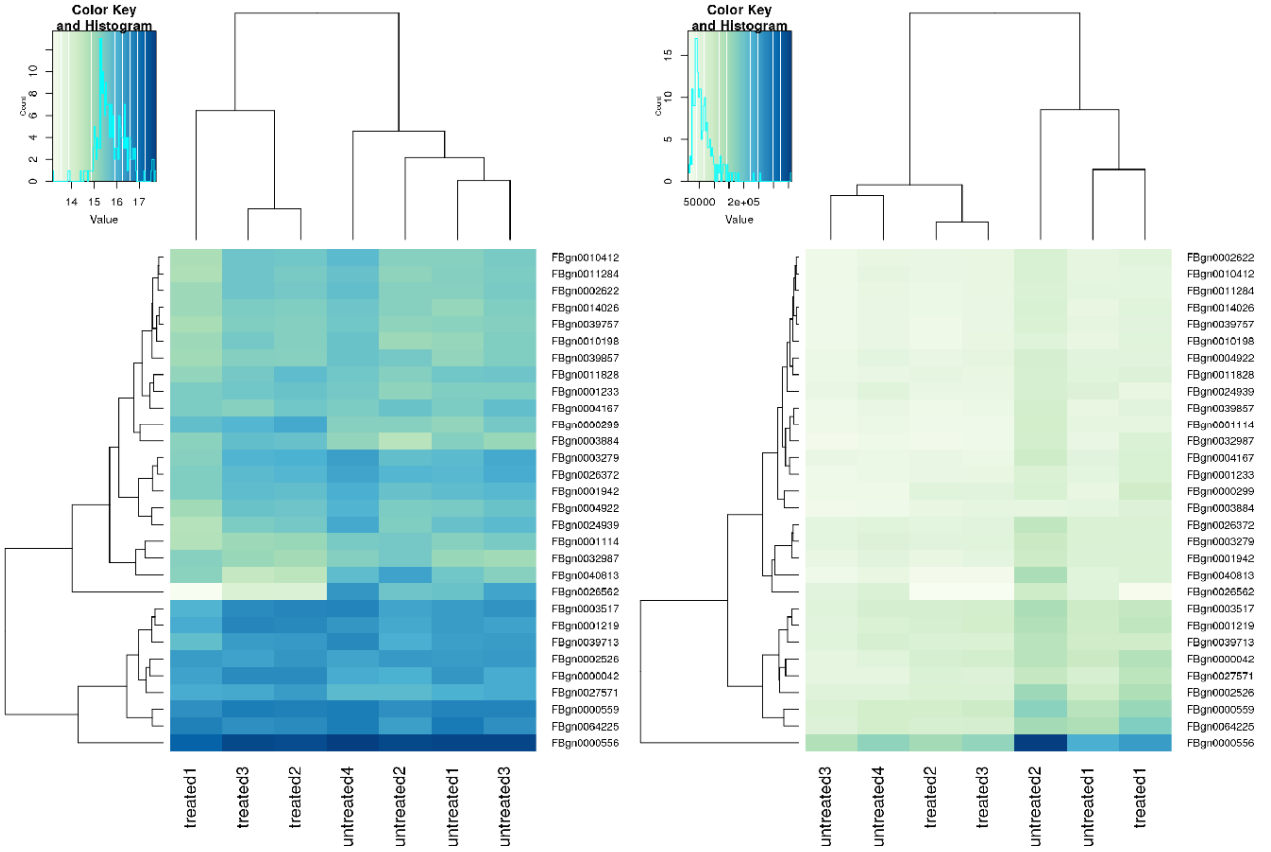
correlation of gene expression for all pairwise combinations of samples



QC methods for DE analysis using DESeq2

Heatmaps showing the expression data of the 30 most highly expressed genes

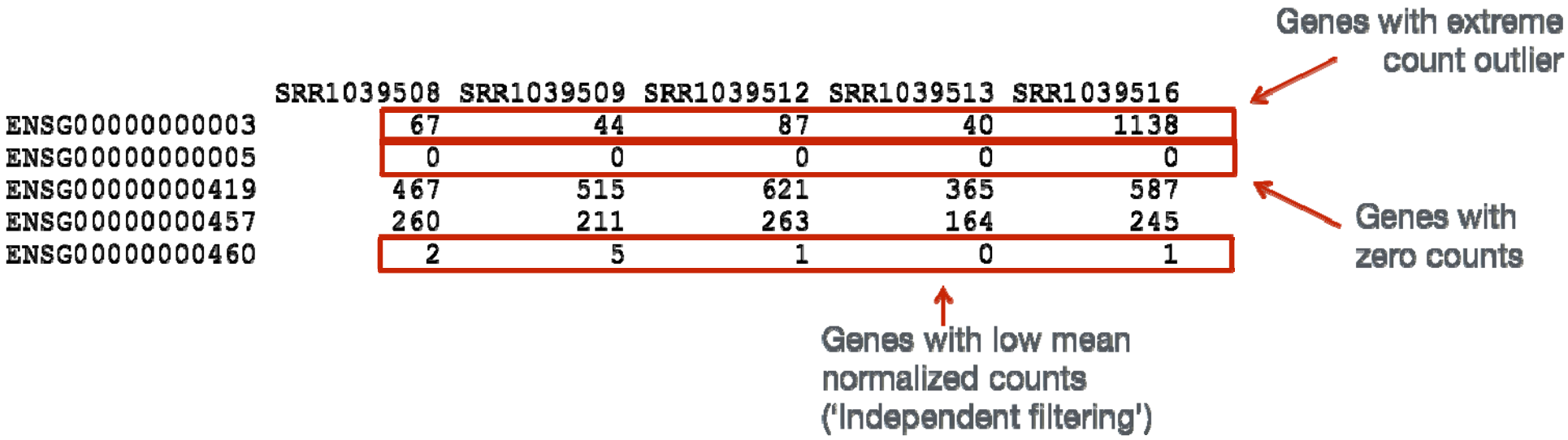
variance stabilisation transformed data: the sample clustering aligns with the experimental factor (treated / untreated)



original count data: the clustering and the colour scale is dominated by a small number of data points with large values

QC methods for DE analysis using DESeq2

Gene-level QC: omit genes that have little or no chance of being detected as differentially expressed



QC methods for DE analysis using DESeq2

Transform normalized counts using the rlog transformation

```
### Transform counts for data visualization
rld <- rlog(dds, blind=TRUE)
```

Principal components analysis (PCA)

```
### Plot PCA
plotPCA(rld, intgroup="sampletype")
```

```
# Input is a matrix of log transformed values
rld <- rlog(dds, blind=T)
rld_mat <- assay(rld)
pca <- prcomp(t(rld_mat))

# Create data frame with metadata and PC3 and PC4 values for input to ggplot
df <- cbind(meta, pca$x)
ggplot(df) + geom_point(aes(x=PC3, y=PC4, color = sampletype))
```

Hierarchical Clustering

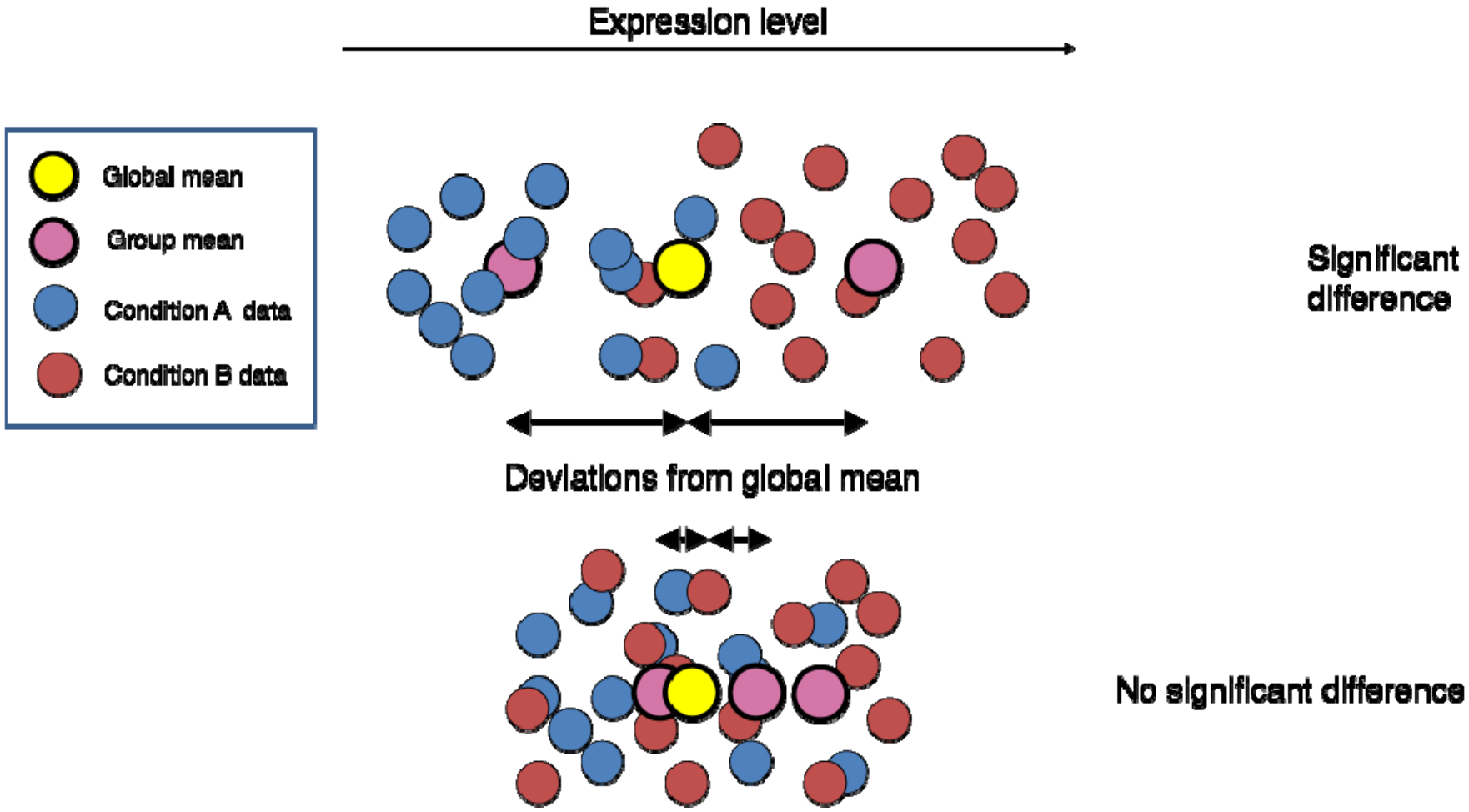
```
### Extract the rlog matrix from the object
rld_mat <- assay(rld)    ## assay() is function from the "SummarizedExperiment" package

### Compute pairwise correlation values
rld_cor <- cor(rld_mat)    ## cor() is a base R function

### Plot heatmap
pheatmap(rld_cor)
```

Differential expression analysis with DESeq2

Learning Objectives

- Understanding the different steps in a differential expression analysis in the context of DESeq2
- Constructing design formulas appropriate for a given experimental design
- Exploring the importance of dispersion during differential expression analysis, and using the plots of the dispersion values to explore assumptions of the NB model

Differential expression analysis with DESeq2

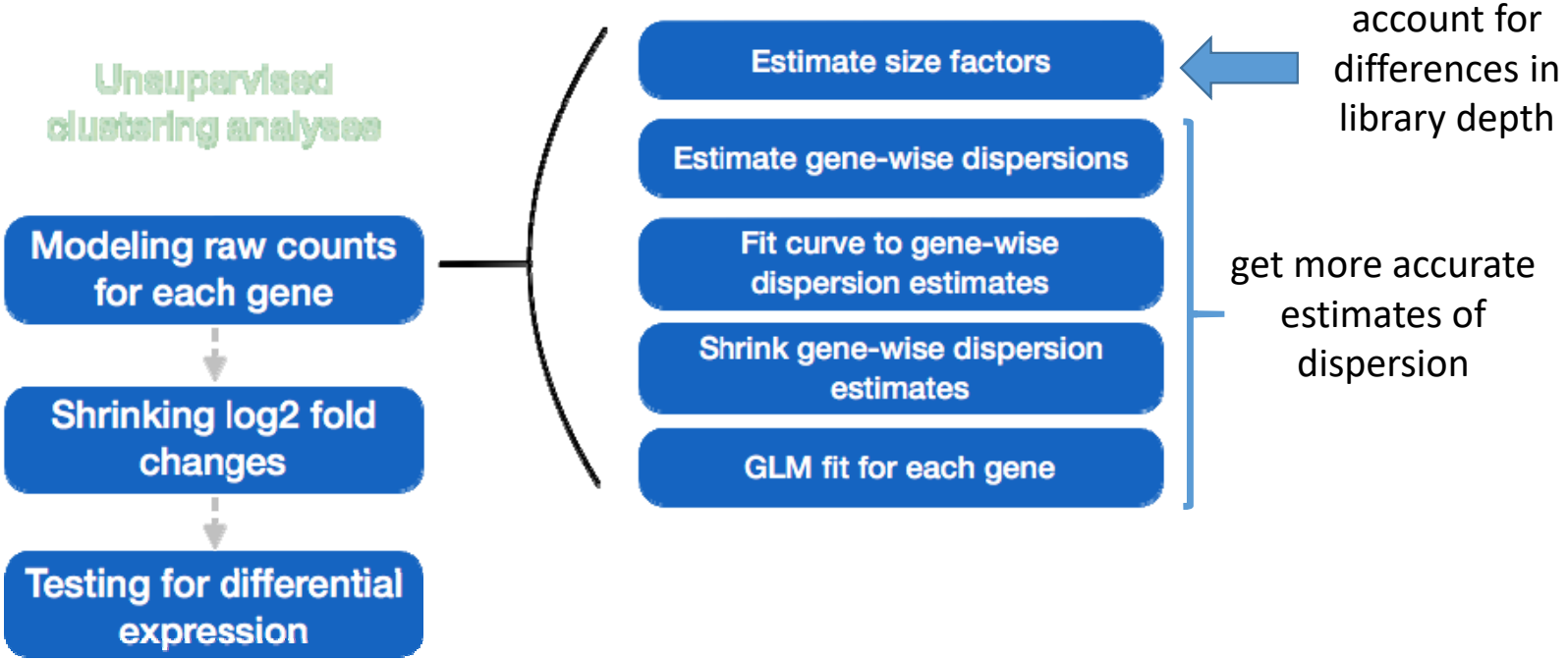


Differential expression analysis with DESeq2

Read counts associated with genes

Normalization

Unsupervised clustering analyses



Differential expression analysis with DESeq2

Running DESeq2

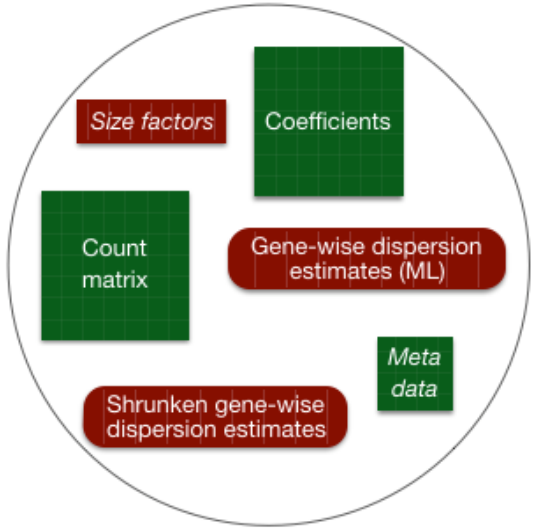
Design formula

The design formula should have all of the factors in your metadata that account for major sources of variation in your data. The last factor entered in the formula should be the condition of interest.

```
design <- ~ sex + age + treatment
```

```
design <- ~ sex + age + treatment + sex:treatment
```

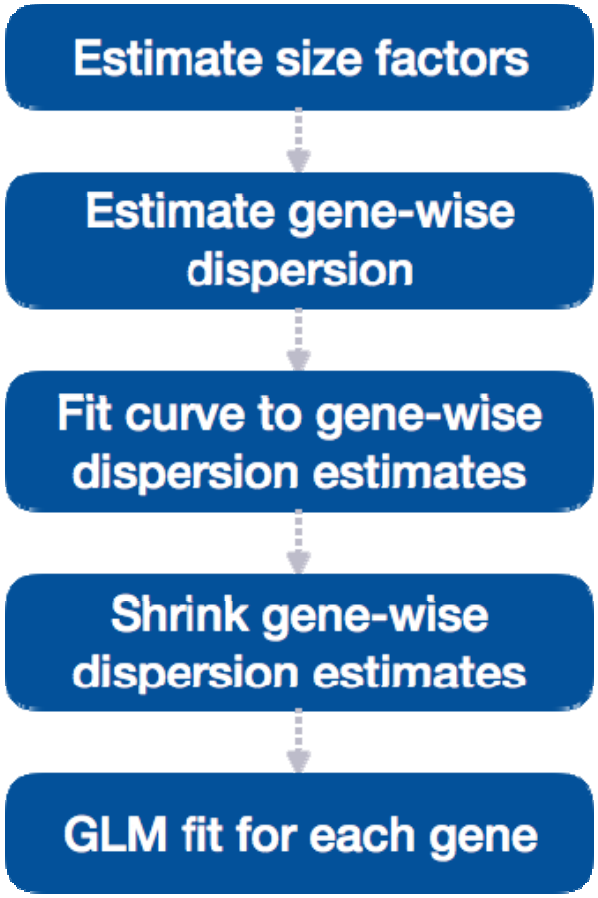
	sex	age	litter	treatment
sample1	M	11	1	Ctrl
sample2	M	13	2	Ctrl
sample3	M	11	1	Treat
sample4	M	13	1	Treat
sample5	F	11	1	Ctrl
sample6	F	13	1	Ctrl
sample7	F	11	1	Treat
sample8	F	13	2	Treat



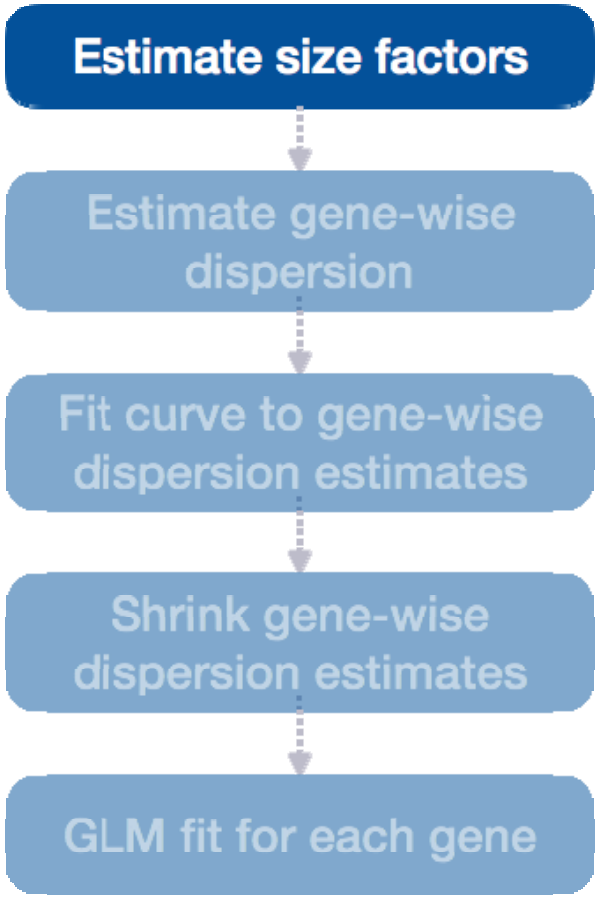
DEA with DESeq2

```
## Create DESeq object
dds <- DESeqDataSetFromMatrix(countData = data, colData = meta, design = ~ sampletype)

## Run analysis
dds <- DESeq(dds)
```

Differential expression analysis with DESeq2



Differential expression analysis with DESeq2



gene	sampleA	sampleB	pseudo-reference sample
EF2A	1489	906	sqrt(1489 * 906) = **1161.5**
ABCD1	22	13	sqrt(22 * 13) = **17.7**

gene	sampleA	sampleB	pseudo-reference sample	ratio of sampleA/ref	ratio of sampleB/ref
EF2A	1489	906	1161.5	1489/1161.5 = **1.28**	906/1161.5 = **0.78**
ABCD1	22	13	16.9	22/16.9 = **1.30**	13/16.9 = **0.77**
MEFV	793	410	570.2	793/570.2 = **1.39**	410/570.2 = **0.72**
BAG1	76	42	56.5	76/56.5 = **1.35**	42/56.5 = **0.74**
MOV10	521	1196	883.7	521/883.7 = **0.590**	1196/883.7 = **1.35**

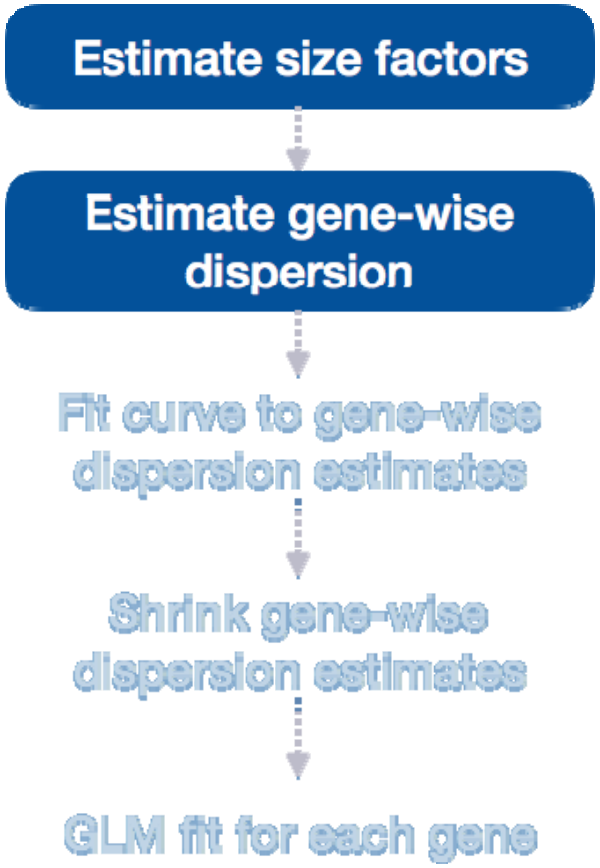
```
normalization_factor_sampleA <- median(c(1.28, 1.3, 1.39, 1.35, 0.59))
```

```
normalization_factor_sampleB <- median(c(0.78, 0.77, 0.72, 0.74, 1.35))
```

Normalized Counts

gene	sampleA	sampleB
EF2A	1489 / 1.3 = **1145.39**	906 / 0.77 = **1176.62**
ABCD1	22 / 1.3 = **16.92**	13 / 0.77 = **16.88**
...	...	...

Differential expression analysis with DESeq2



$$Var = \mu + \alpha * \mu^2$$

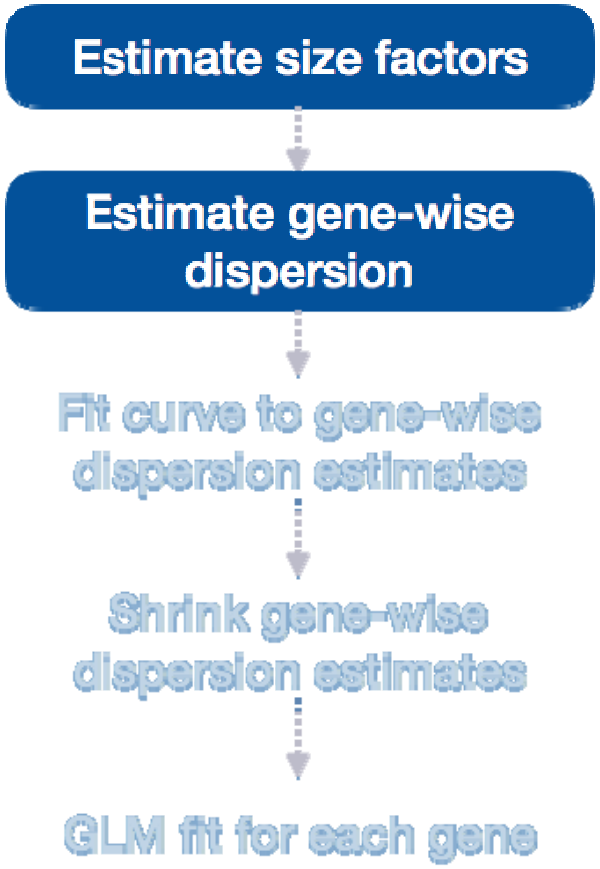
variance mean dispersion

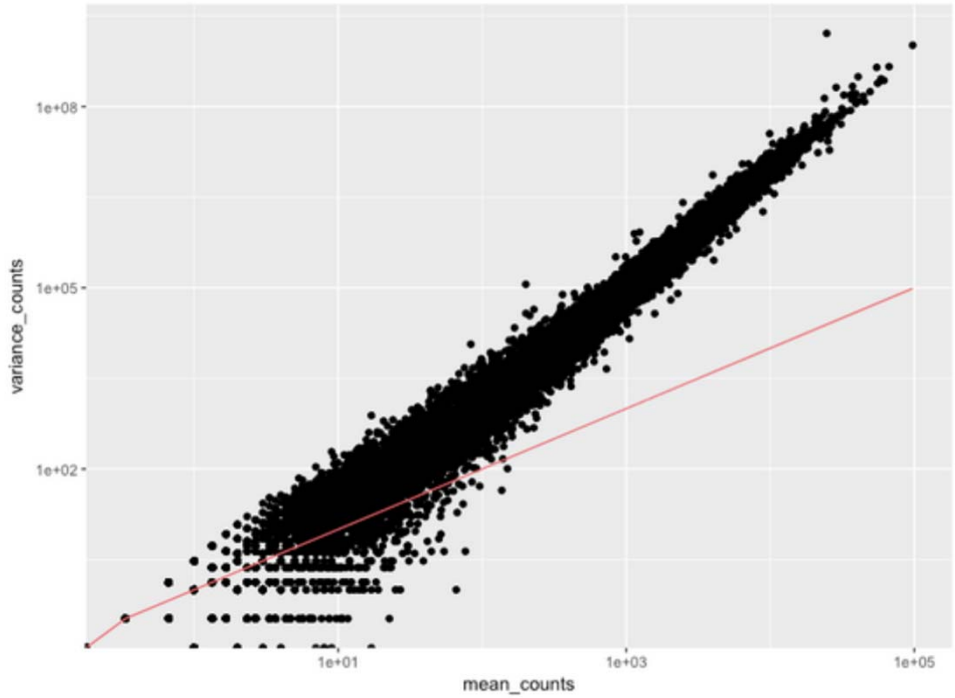
α : **inversely related to the mean** and **directly related to variance**

α : **higher for small mean counts and lower for large mean counts**

α : **reflects the variance in gene expression for a given mean value**

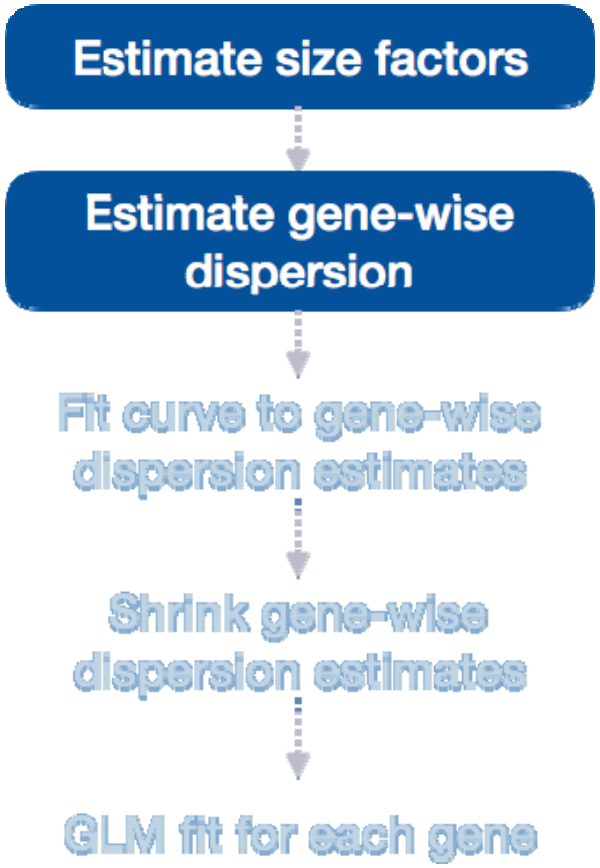
Differential expression analysis with DESeq2





for low mean counts, the variance estimates have a much larger spread
the dispersion estimates will differ much more between genes with small means

Differential expression analysis with DESeq2

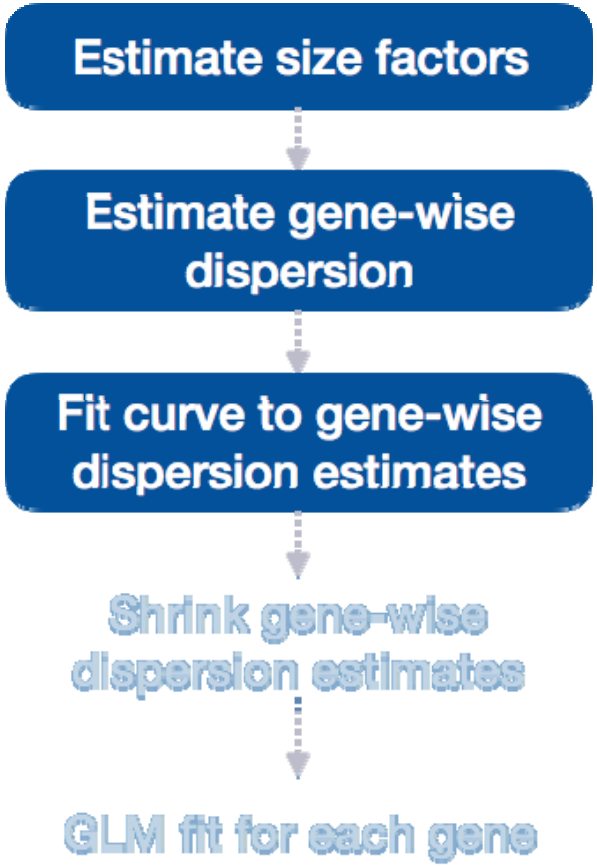


With only a few (3-6) replicates per group, the **estimates of variation for each gene are often unreliable**

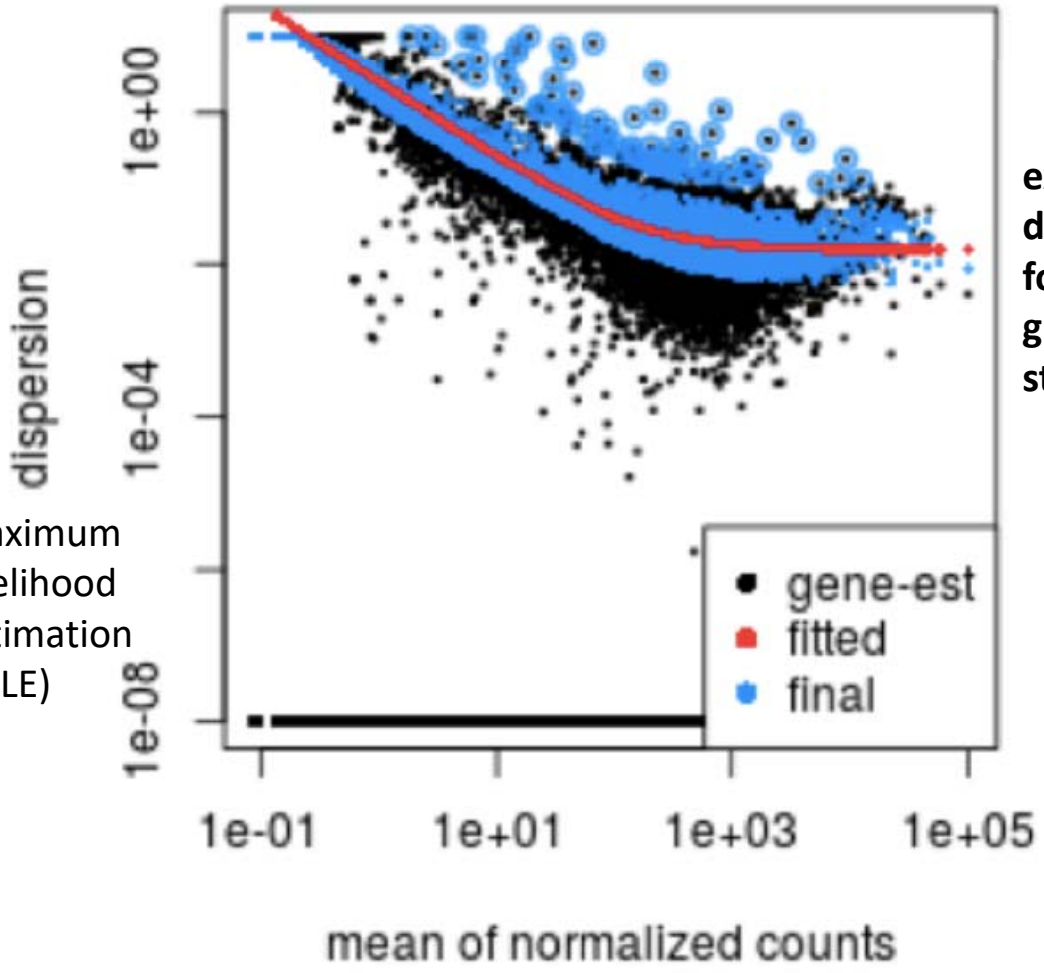
DESeq2 **shares information across genes** : it **assumes that genes with similar expression levels have similar dispersion**

Maximum likelihood approach for estimating the dispersion for each gene

Differential expression analysis with DESeq2



maximum likelihood estimation (MLE)

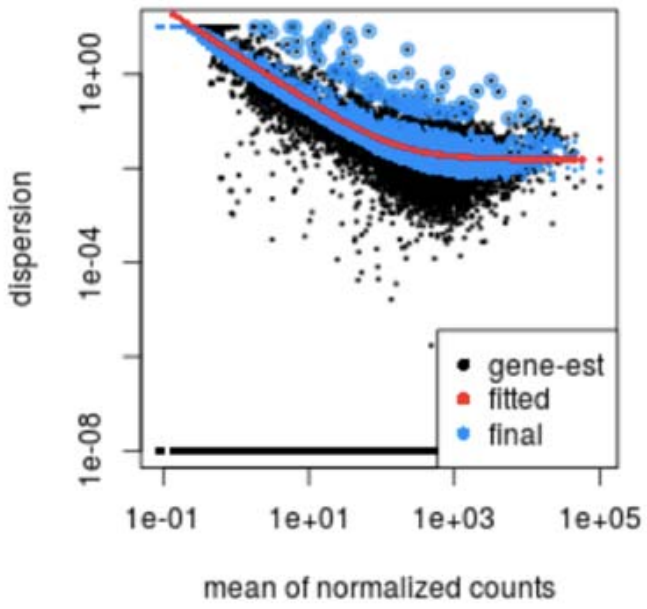


expected dispersion value for genes of a given expression strength

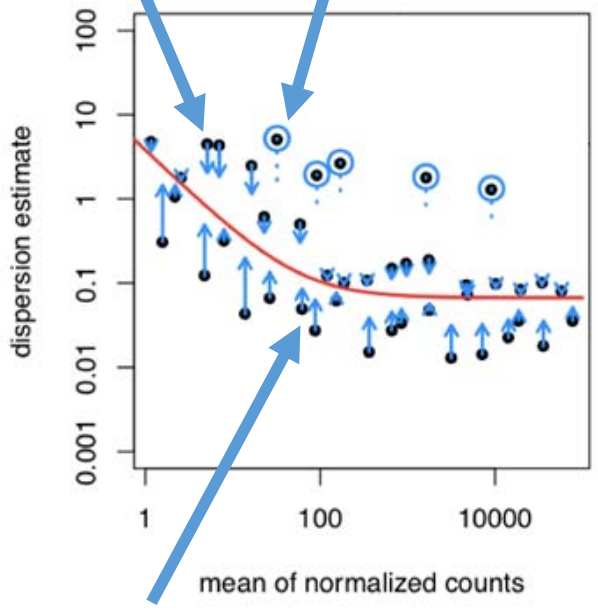
Differential expression analysis with DESeq2

- Estimate size factors
- Estimate gene-wise dispersion
- Fit curve to gene-wise dispersion estimates
- Shrink gene-wise dispersion estimates
- GLM fit for each gene

Dispersion estimates that are slightly above the curve are also shrunk toward the curve

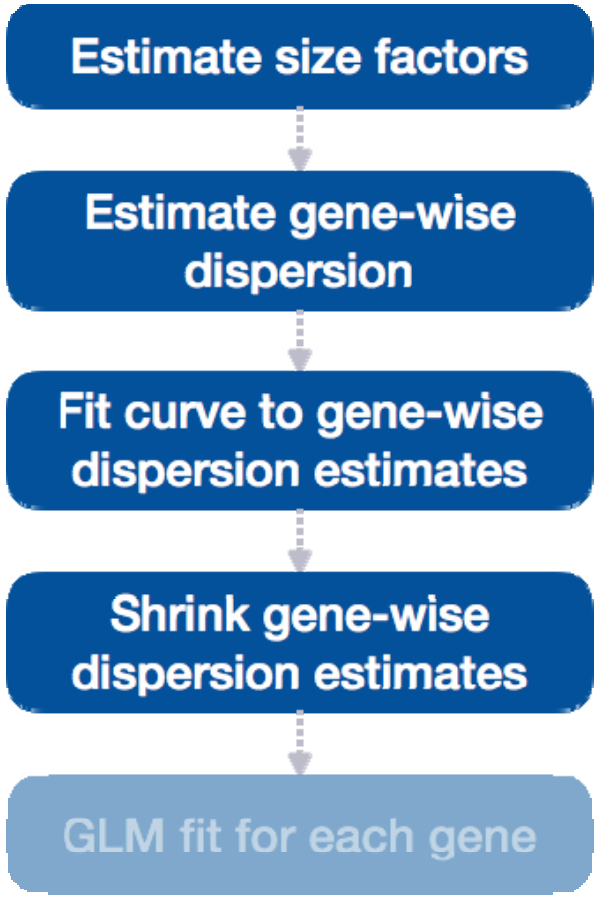


Genes with extremely high dispersion values are not shrunken

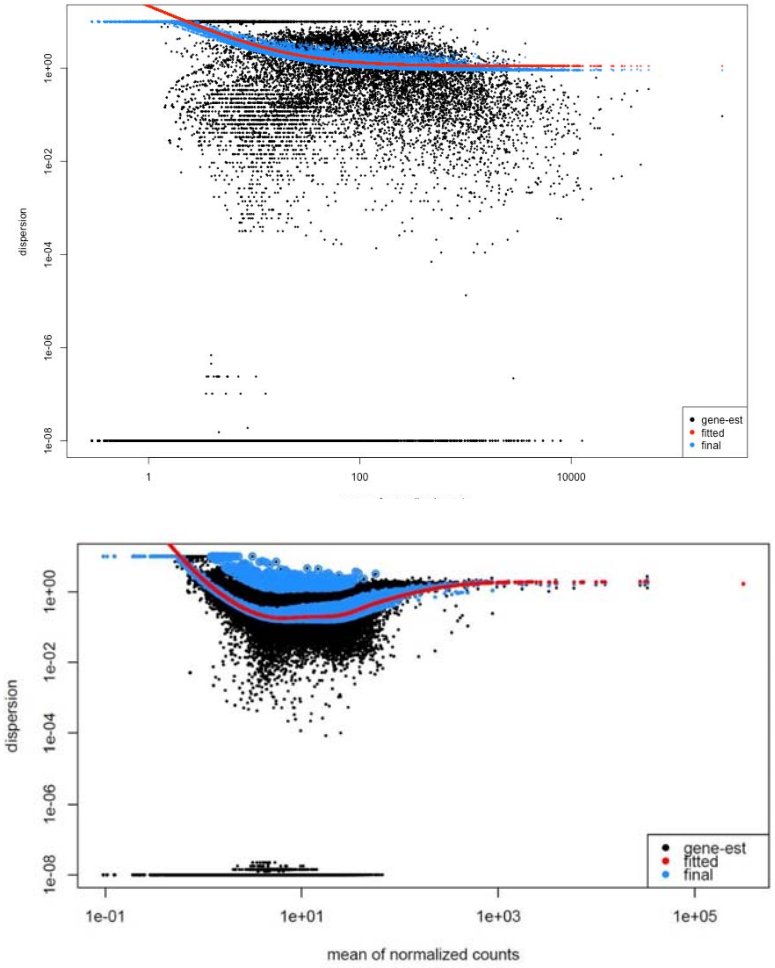


Genes with low dispersion estimates are shrunken towards the curve

Differential expression analysis with DESeq2

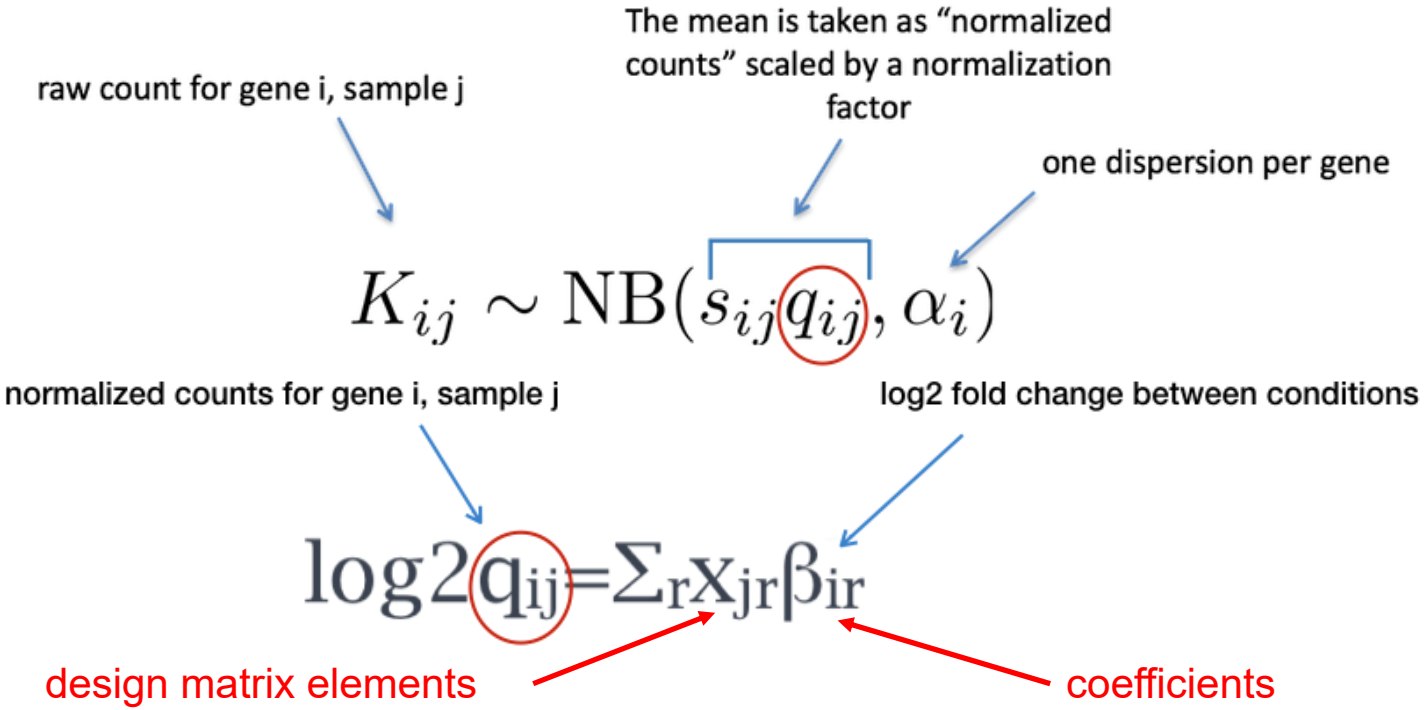


Examples of **worrisome dispersion plots**



Differential expression analysis with DESeq2

Generalized Linear Model fit for each gene



In the simplest case of a comparison between two groups (i.e. treated and control samples), the design matrix elements indicate whether a sample is treated or not, and the GLM fit returns coefficients indicating the overall expression strength of the gene and the log2fold change between treatment and control

Differential expression analysis with DESeq2

Hypothesis testing using the Wald test

- **H_0 : no differential expression across the two sample groups (LFC = 0)**
- Wald test : allows to test if (a set of) explanatory variables have a significant effect on gene expression

```
## Create DESeq object
dds <- DESeqDataSetFromMatrix(countData = data, colData = meta, design = ~ sampletype)
```

```
## Run analysis
dds <- DESeq(dds)
```

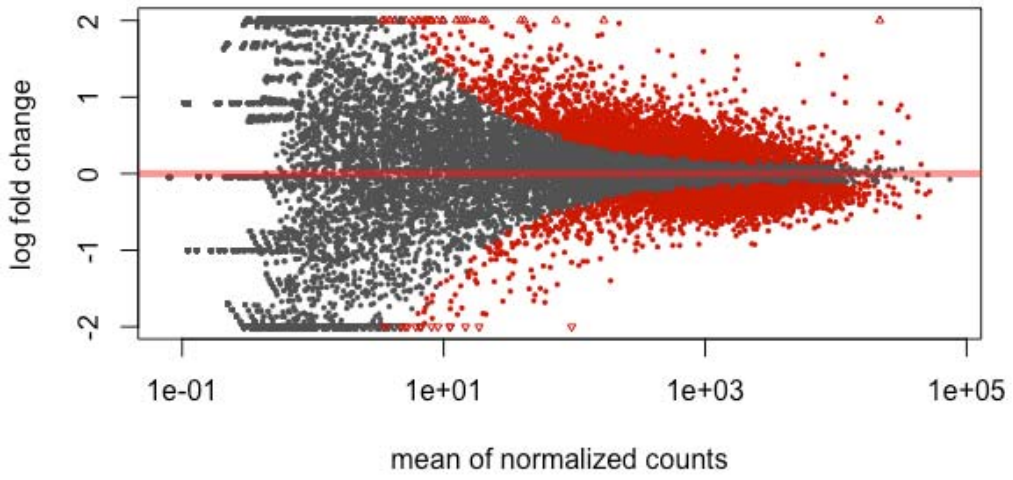
```
estimating size factors
estimating dispersions
gene-wise dispersion estimates
mean-dispersion relationship
final dispersion estimates
fitting model and testing
```

Differential expression analysis with DESeq2

Hypothesis testing using the Wald test

```
## Define contrasts, extract results table, and shrink the log2 fold changes

contrast_oe <- c("sampletype", "MOV10_overexpression", "control")

res_tableOE_unshrunken <- results(dds, contrast=contrast_oe, alpha = 0.05)
```

MA Plot



Differential expression analysis with DESeq2

Hypothesis testing using the Wald test

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>
1/2-SBSRNA4	45.6520399	0.26976764	0.18775752	1.4367874	0.1507784	0.25242910
A1BG	61.0931017	0.20999700	0.17315013	1.2128030	0.2252051	0.34444163
A1BG-AS1	175.6658069	-0.05197768	0.12366259	-0.4203185	0.6742528	0.77216278
A1CF	0.2376919	0.02237286	0.04577046	0.4888056	0.6249793	NA
A2LD1	89.6179845	0.34598540	0.15901426	2.1758136	0.0295692	0.06725157
A2M	5.8600841	-0.27850841	0.18051805	-1.5428286	0.1228724	0.21489067

`baseMean` : mean of normalized counts for all samples

`log2FoldChange` : log2 fold change

`lfcSE` : standard error

`stat` : Wald statistic

`pvalue` : Wald test p-value

`padj` : BH adjusted p-values

Differential expression analysis with DESeq2

Multiple test correction

- **Bonferroni:** The adjusted p-value is calculated by: p-value * m (m = total number of tests). **This is a very conservative approach with a high probability of false negatives**, so is generally not recommended.
- **FDR/Benjamini-Hochberg:** Benjamini and Hochberg (1995) defined the concept of FDR and created an algorithm to control the expected FDR below a specified level given a list of independent p-values. **An interpretation of the BH method for controlling the FDR is implemented in DESeq2 in which we rank the genes by p-value, then multiply each ranked p-value by m/rank.**
- **Q-value / Storey method:** The minimum FDR that can be attained when calling that feature significant. For example, if gene X has a q-value of 0.013 it means that 1.3% of genes that show p-values at least as small as gene X are false positives

Differential expression analysis with DESeq2

Summarizing results

To summarize the results table, a handy function in DESeq2 is `summary()`. Confusingly it has the same name as the function used to inspect data frames. This function when called with a DESeq results table as input, will summarize the results using the alpha threshold: FDR < 0.05 (padj/FDR is used even though the output says `p-value < 0.05`). Let's start with the OE vs control results:

```
## Summarize results
summary(res_tableOE)
```

In addition to the number of genes up- and down-regulated at the default threshold, **the function also reports the number of genes that were tested (genes with non-zero total read count), and the number of genes not included in multiple test correction due to a low mean count.**