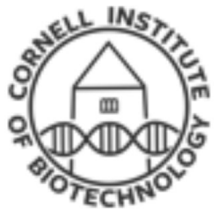
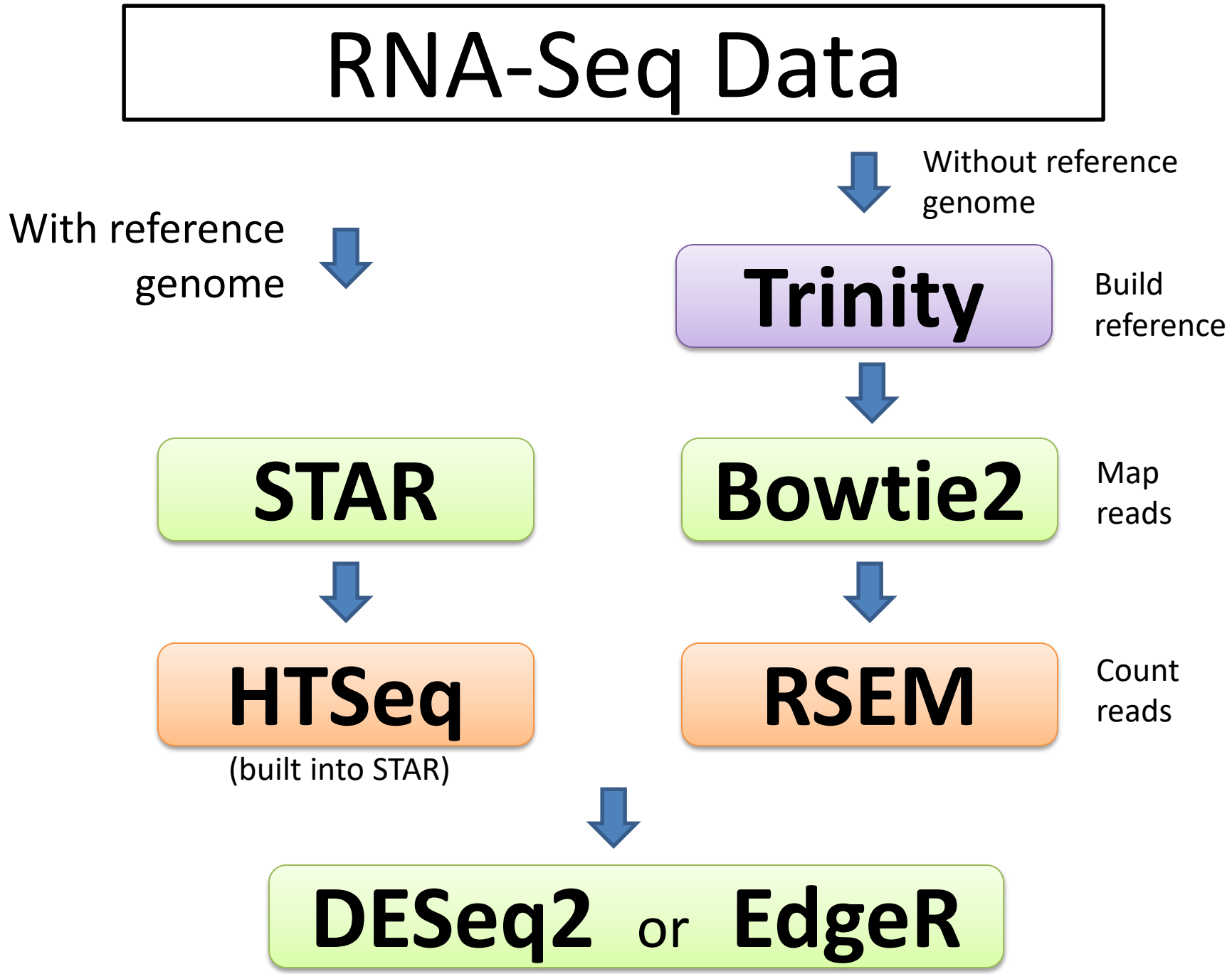
Statistics of RNA-seq data analysis

Jeff Glaubitz and Qi Sun

November 9, 2020
Bioinformatics Facility
Institute of Biotechnology
Cornell University




RNA-Seq Data
Without reference genome
With reference genome
Trinity
Build reference
STAR
Bowtie2
Map reads
HTSeq
(built into STAR)
RSEM
Count reads
DESeq2 or EdgeR

RNA-Seq Statistics:

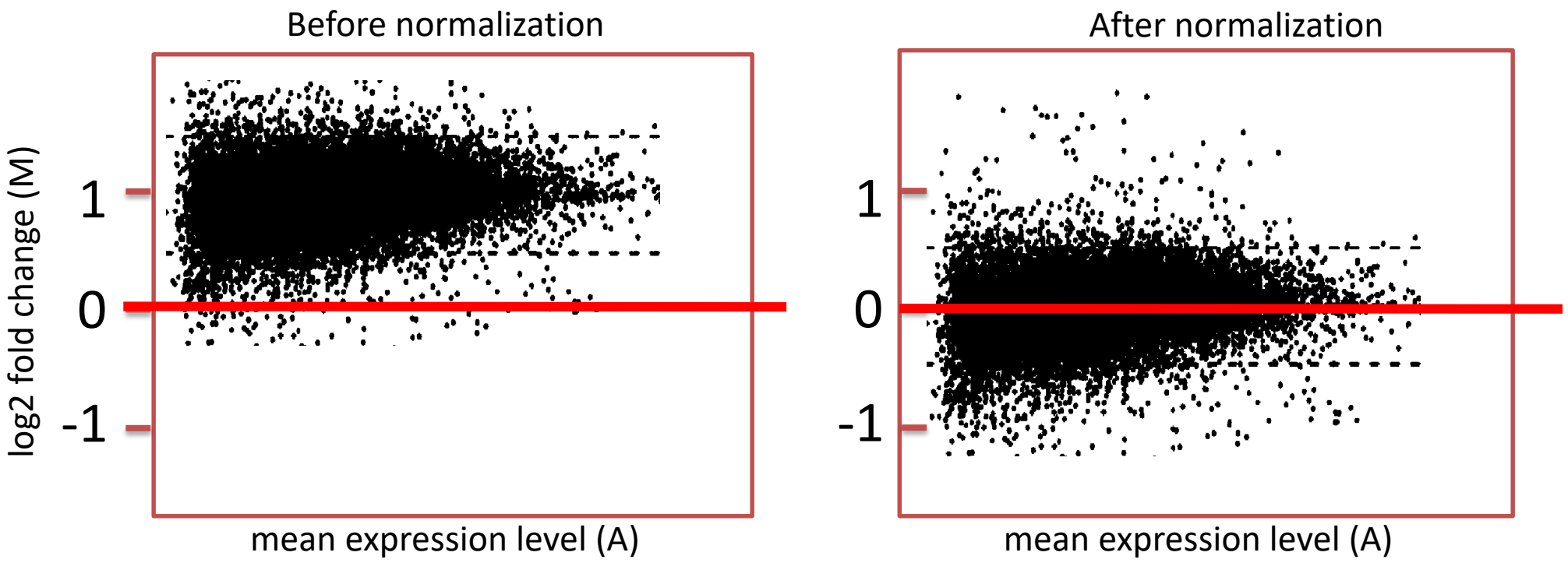
- Normalization between samples
- **D**ifferentially **E**xpressed Genes (**DE**)

Genes	Control	Treated
Gene A	10	30
Gene B	30	90
Gene C	5	15
Gene D	1	3
Gene N	80	240
	126	**378**

Normalization

Genes	Control	Treated
Gene A	10	30
Gene B	30	90
Gene C	5	15
Gene D	1	3
Gene N	80	240
	126	378

MA Plots between samples



- With the assumption that most genes are expressed equally, the log ratio should mostly be close to 0

Simple normalization

CPM (Count Per Million mapped reads)

Normalized by:

- Total number of reads (fragments) aligned to a gene

FPKM (Fragments Per Kilobase Of Exon Per Million Fragments)

Normalized by:

- Total fragment count (number of reads or read pairs)
- Gene length (kb)

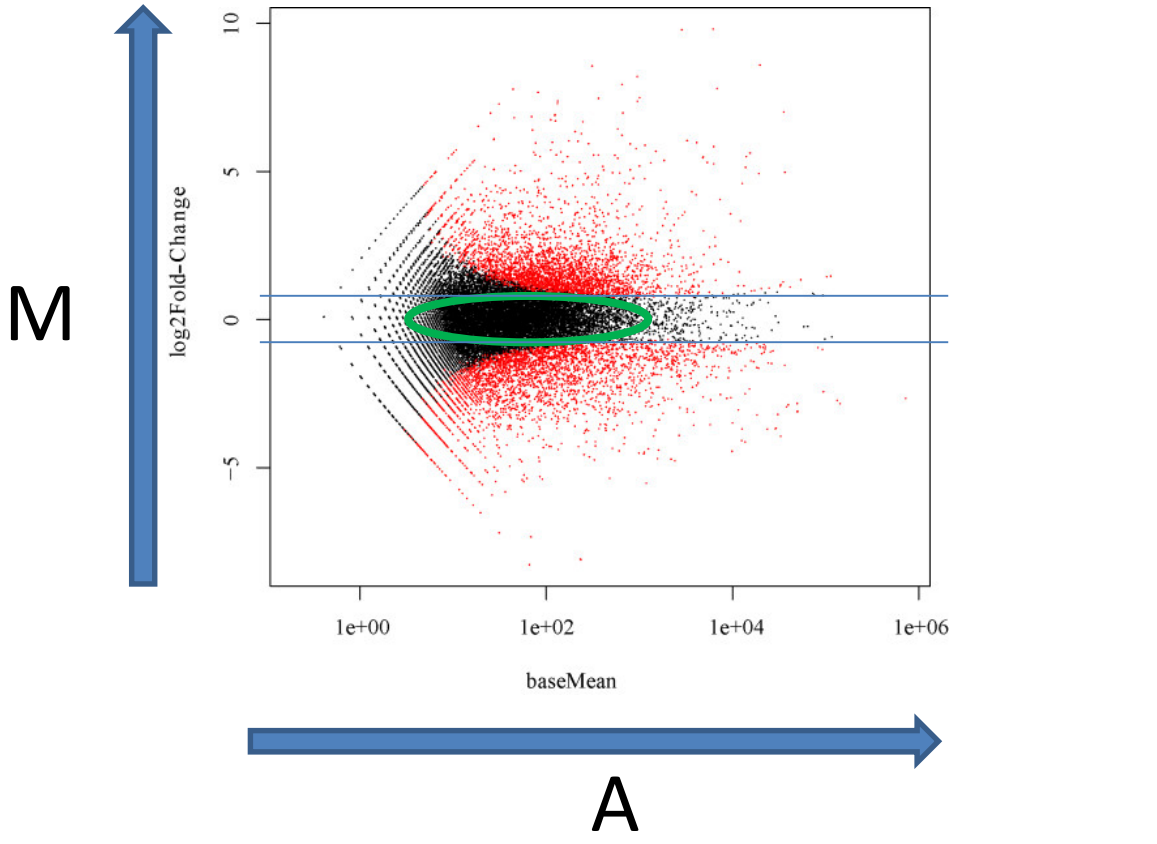
CPM : Not normalized by gene length. Longer genes tend to have higher CPM values than shorter genes. But that is ok, as in RNA-Seq experiments, we do not compare between genes, only compare the same gene between different samples.

Simple normalization could fail

Genes	Control	Treated
Gene A	10	30
Gene B	30	90
Gene C	5	15
Gene D	1	3
Gene N	1000	240
	1046	378

TMM normalization

(Trimmed mean of M-values)



M = log2(Test/Test_total)-log2(Ref/Ref_total)

A =0.5 *log2(Test/Test_total*Ref/Ref_total)

“Effective library size”

DESeq2 normalization

1. For each gene, calculate geometric mean

$$\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}} = \sqrt[n]{x_1 x_2 \cdots x_n}$$

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Geomean
Gene1	34	56	23	12	10	30	**23**
Gene 2	10	6	7	11	12	8	**9**
……							
Gene n	65	78	67	34	56	23	**50**

2. For each gene, calculate ratio to geometric mean

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6
Gene1	1.5	2.4	1.0	0.5	0.4	1.3
Gene 2	1.1	0.7	0.8	1.3	1.4	0.9
……						
Gene n	1.3	1.6	1.4	0.7	1.1	0.5

3. Take median of these ratios as sample normalization factor (“size factor”)

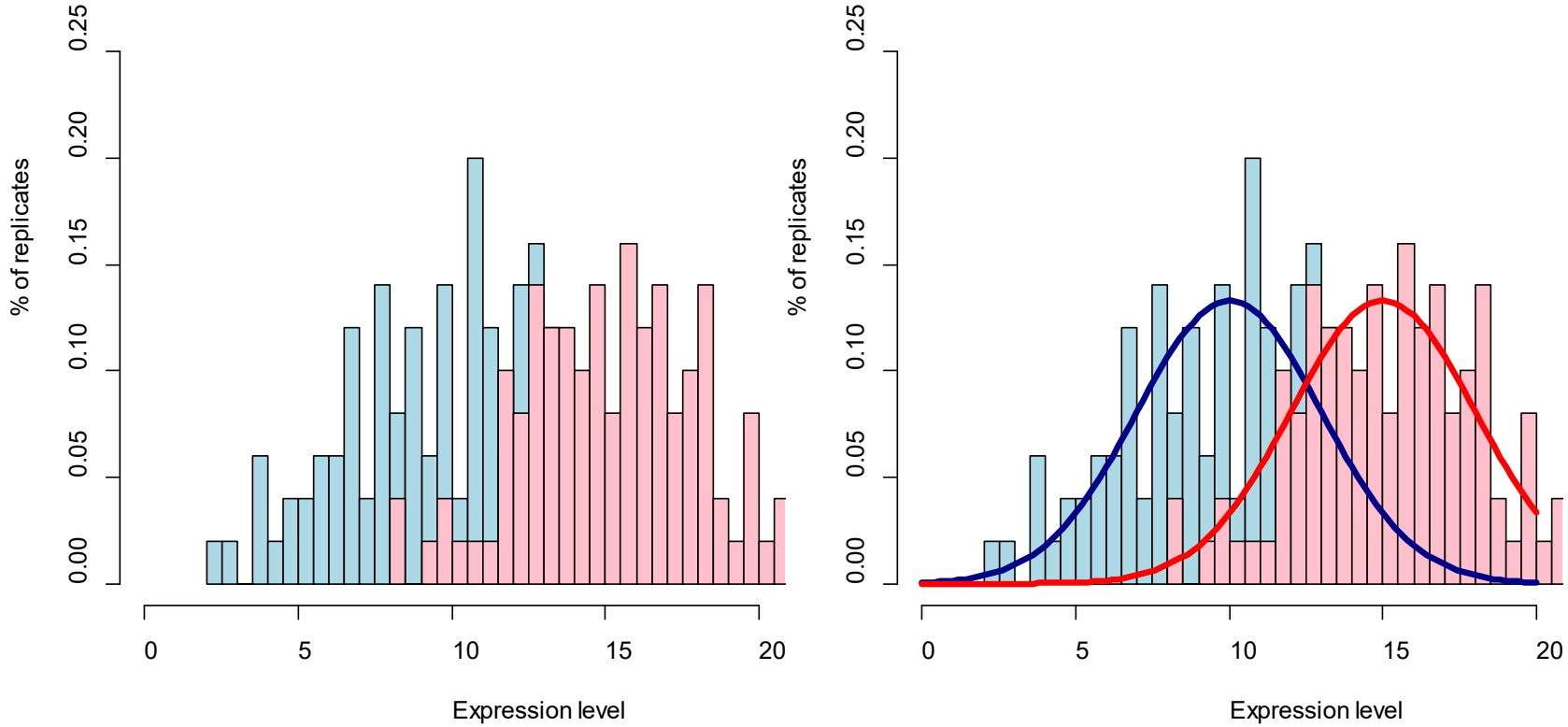
1.3	**1.6**	**1**	**0.7**	**1.1**	**0.9**

Biological vs. technical replicates

Scenario	Replicate Type
Split tissue sample evenly into 2 RNA preps	Technical
Split RNA sample into two library preps	Technical
Split library across two sequencing flow cells	Technical
RNA prep from different leaves on same plant	Technical/Biological
Different clones of the same genotype in same treatment condition	Biological
Different genotypes in same treatment condition	Biological

Differentially expressed genes

If we could do 100 biological replicates:

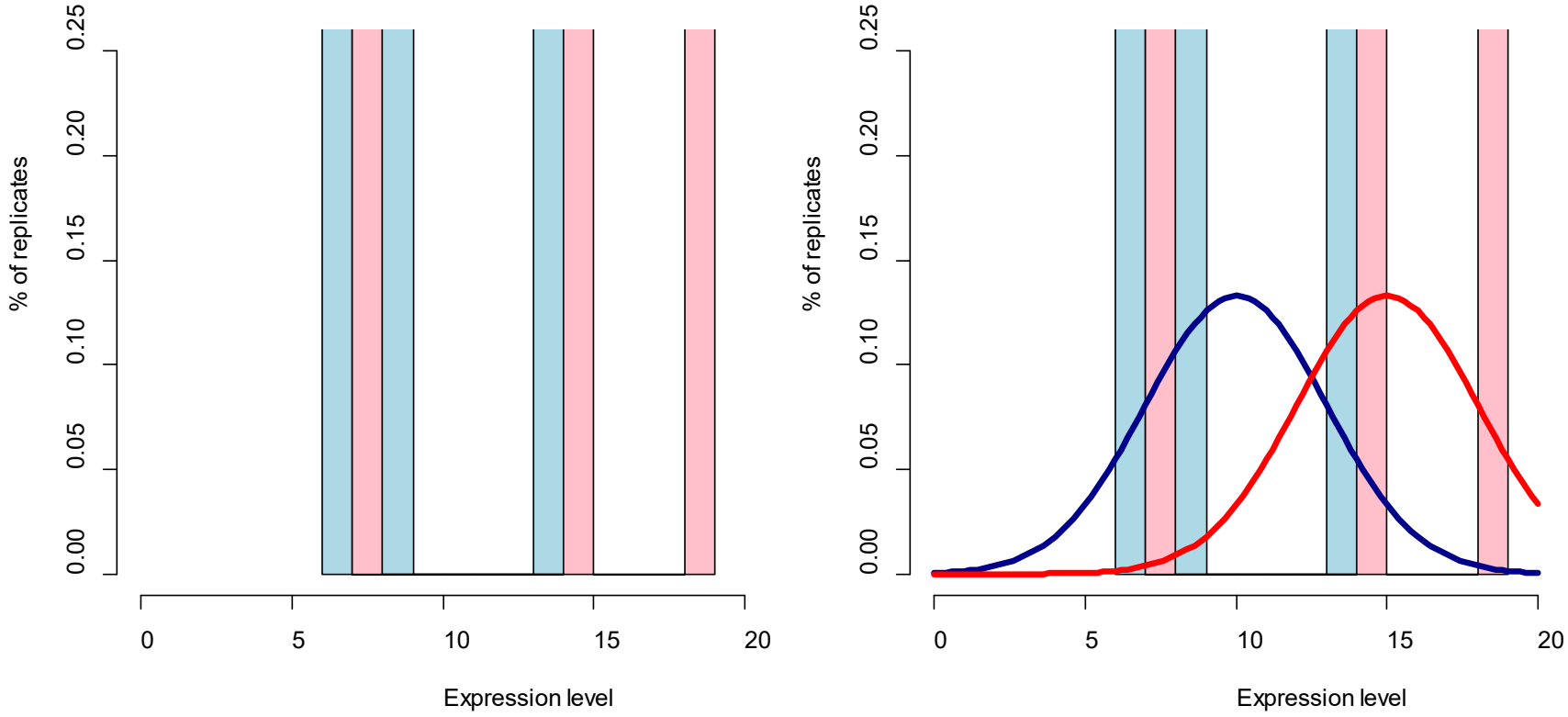


Distribution of Expression Level of A Gene

Control samples

Treated samples

The reality is, often we can only afford 3 or so replicates:



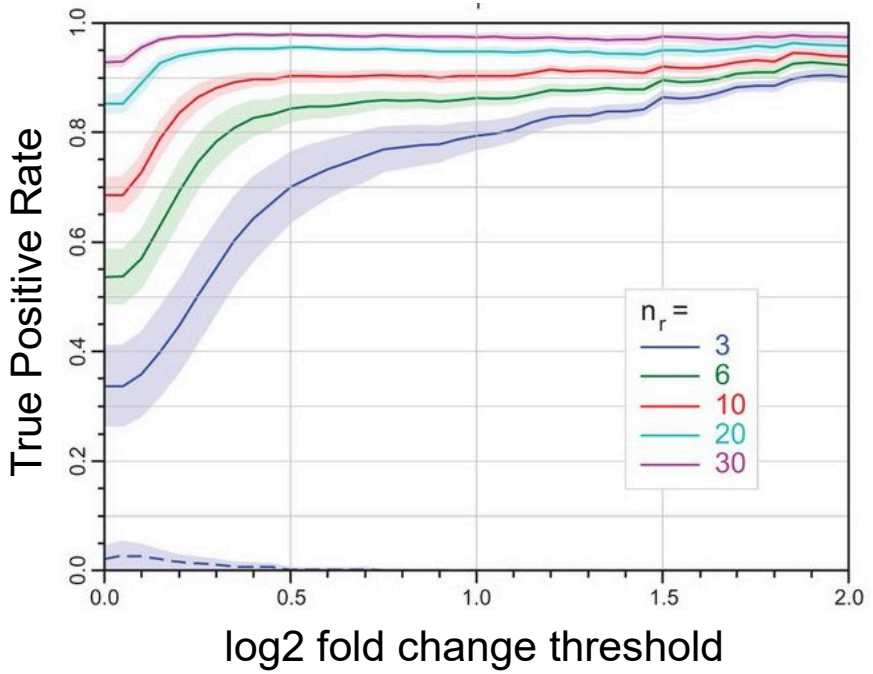
Distribution of Expression Level of A Gene

Control samples

Treated samples

How many biological replicates?

- 3 replicates are the *bare minimum* for publication
- Schurch *et al.* (2016) recommend at least 6 replicates for adequate statistical power to detect DE
- Depends on biology and study objectives
- Trade off with sequencing depth
- Some replicates might have to be removed from the analysis because poor quality (outliers)



From: Schurch *et al.* 2016. *RNA* 22:839-851

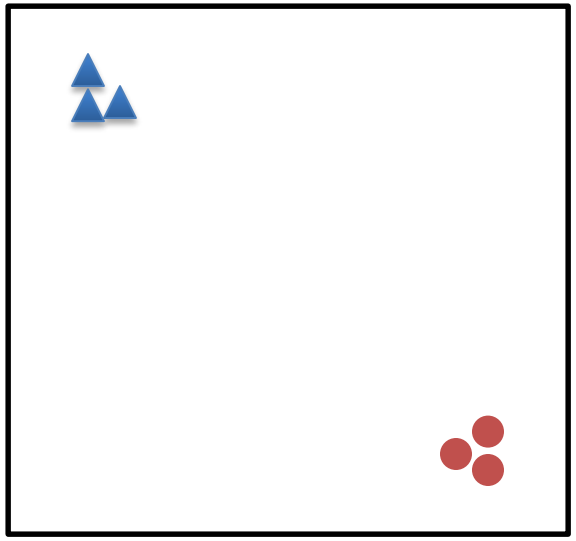
Experimental design

▲: Control samples

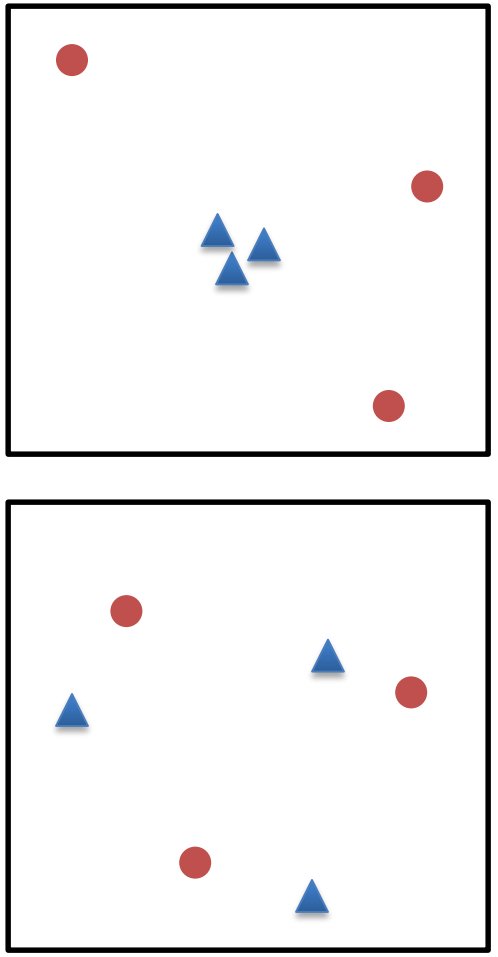
●: Treated samples

PCA Plots

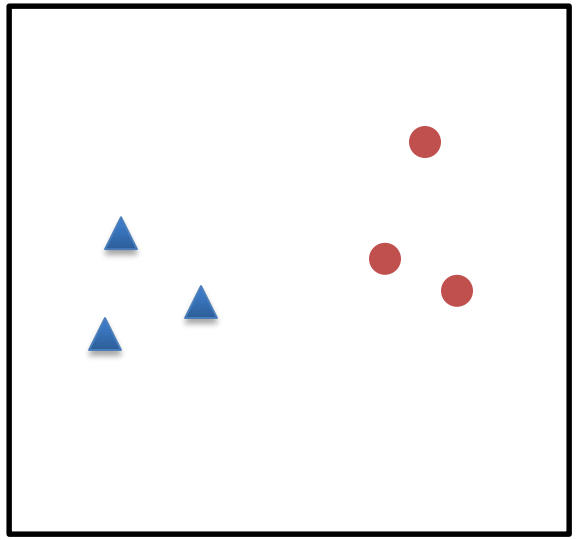
Summarize variation over many genes
(*e.g.*, 500 most variable)



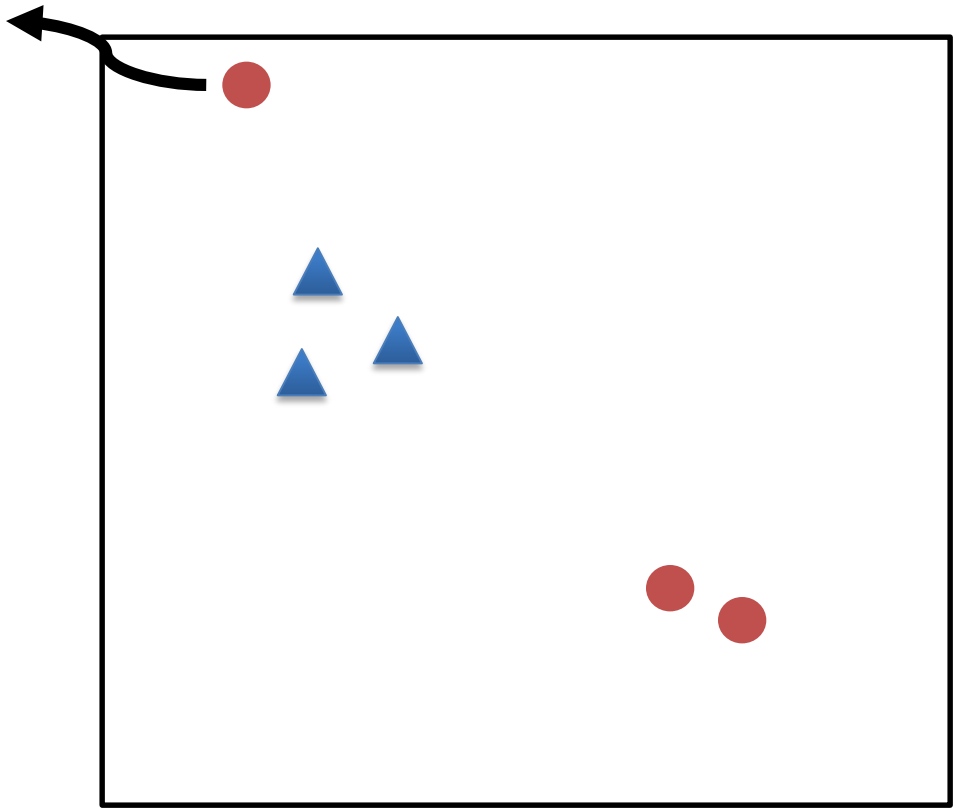
Too many DE genes



Too few DE genes



Remove outlier samples



Differentially Expressed Genes

Expression level of gene 1

Control:		**Treated:**	
Replic 1	**24**	*Replic 1*	**23**
Replic 2	**25**	*Replic 2*	**26**
Replic 3	**27**	*Replic 3*	**102**

Is this a DE gene?

You might get different answers depending of which software you run.

Available RNA-seq analysis packages for DE

TABLE 2. A summary of the recommendations of this paper

	Agreement with other tools[a]	WT vs. WT FPR[b]	Fold-change threshold (T)[c]	Tool recommended for: (# good replicates per condition)[d] ≤3	≤12	>12
DESeq	Consistent	Pass	0	-	-	Yes
			0.5	-	Yes	Yes
			2.0	Yes	Yes	Yes
DESeq2	Consistent	Pass	0	-	-	Yes
			0.5	Yes	Yes	Yes
			2.0	Yes	Yes	Yes
EBSeq	Consistent	Pass	0	-	-	Yes
			0.5	-	Yes	Yes
			2.0	Yes	Yes	Yes
edgeR (exact)	Consistent	Pass	0	-	-	Yes
			0.5	Yes	Yes	Yes
			2.0	Yes	Yes	Yes
Limma	Consistent	Pass	0	-	-	Yes
			0.5	-	Yes	Yes
			2.0	Yes	Yes	Yes
cuffdiff	Consistent	Fail				
BaySeq	Inconsistent	Pass				
edgeR (GLM)	Inconsistent	Pass				
DEGSeq	Inconsistent	Fail				
NOISeq	Inconsistent	Fail				
PoissonSeq	Inconsistent	Fail				
SAMSeq	Inconsistent	Fail				

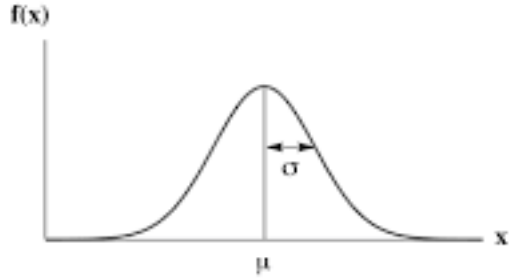
From: Schurch *et al.* 2016. *RNA* 22:839-851

Why DESeq2?

1. Top method recommended by Schurch *et al.* (2016), along with *EdgeR*
2. Cutting-edge tool widely used and accepted: 20,556 citations
   (Google Scholar on Nov 8, 2020)
3. Documentation (and papers) very thorough and well-written
   www.bioconductor.org/packages/devel/bioc/vignettes/DESeq2/inst/doc/DESeq2.html
4. The first author (Mike Love) provides amazing support! Most questions that you Google (e.g., support.bioconductor.org) are clearly and definitively answered by the author himself.
5. R functions in *DESeq2* package are intuitive to R users (and modifiable). Defining the experimental design is easy and intuitive, even for complex, multifactor designs:

```
design= ~ batch + weight + genotype + treatment + genotype:treatment
```

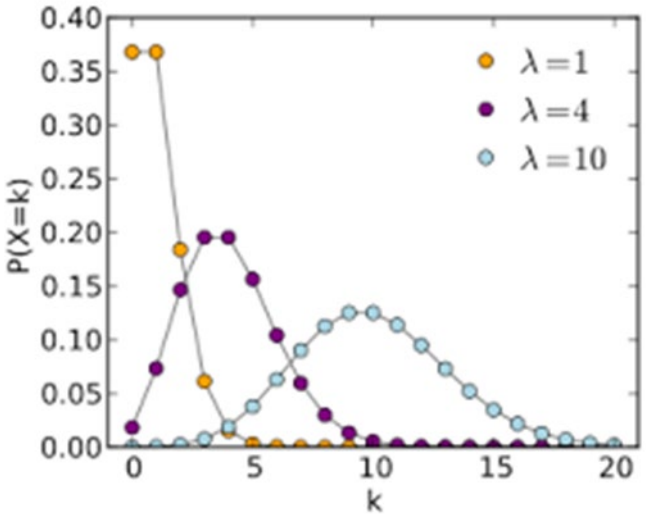
Hypothesis tests require accurate statistical model



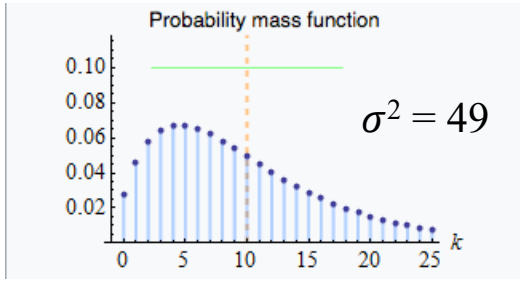
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

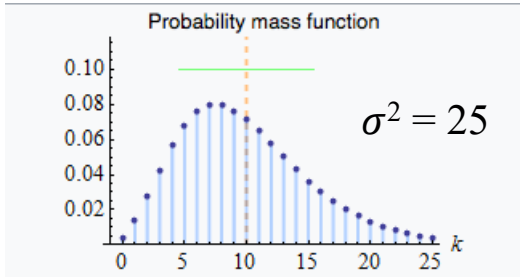
$e = 2.71828$

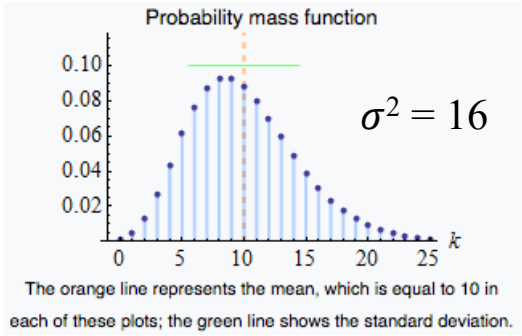
Gaussian (Normal)



Poisson (variance=mean)



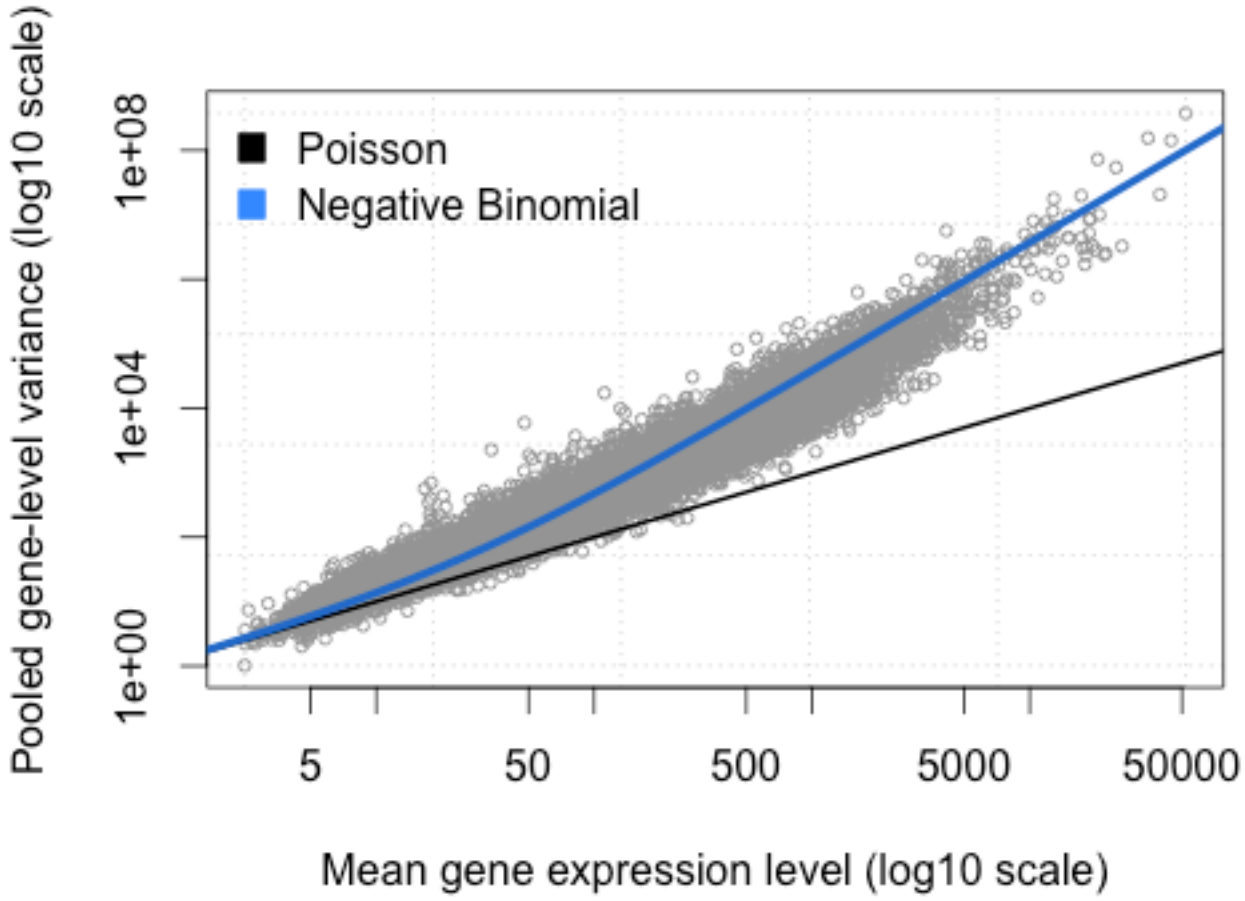




Negative binomial

(variance >= mean)

Negative binomial best fit for RNA-Seq data



DESeq2 fits an 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:

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

Coefficient r

Generalized Linear Model (GLM) coefficients

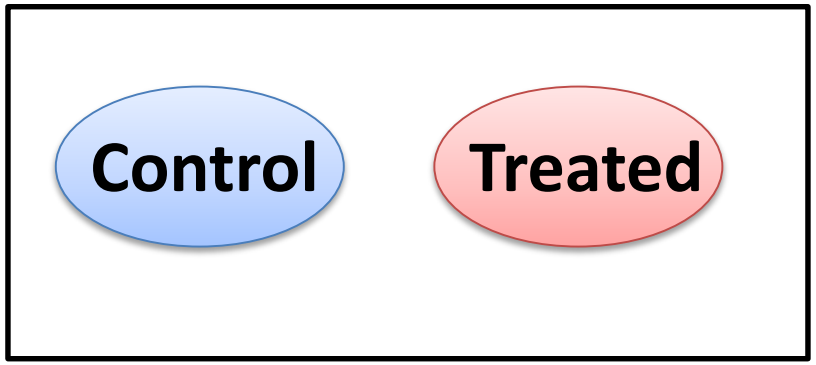
-- One for each Design matrix column (factor)

= strength of effect

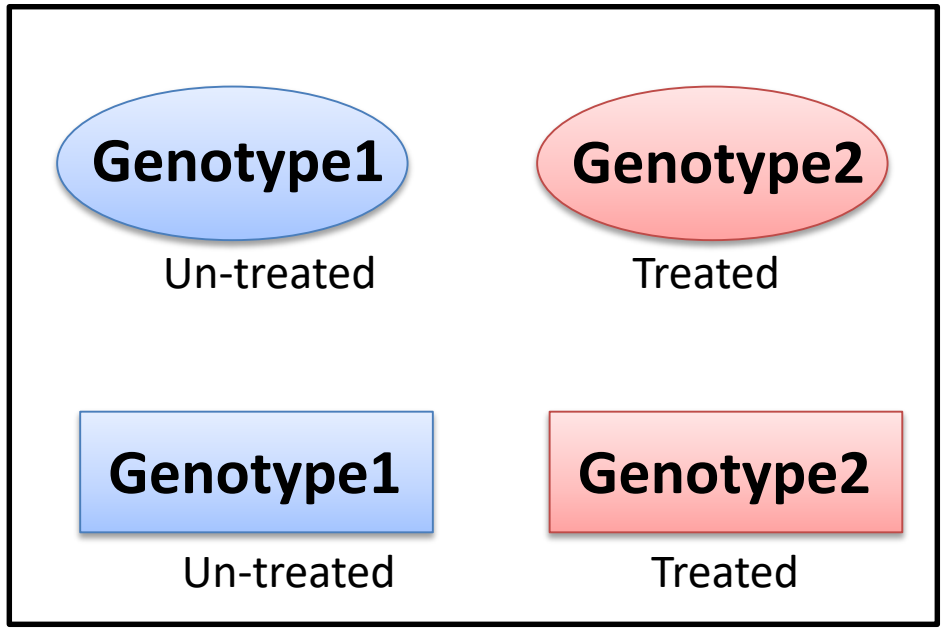
= **log2 fold change** for each gene

Type of analyses

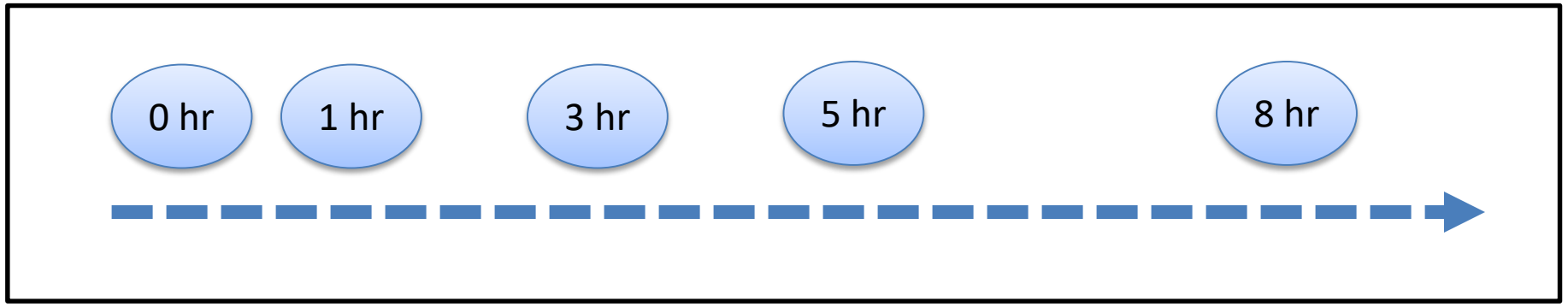
One factor



Two factors



Time series



DESeq2: Design specifications

```
dds <- DESeqDataSetFromMatrix(countData = cts, colData = coldata,
design= ~ treatment)
```

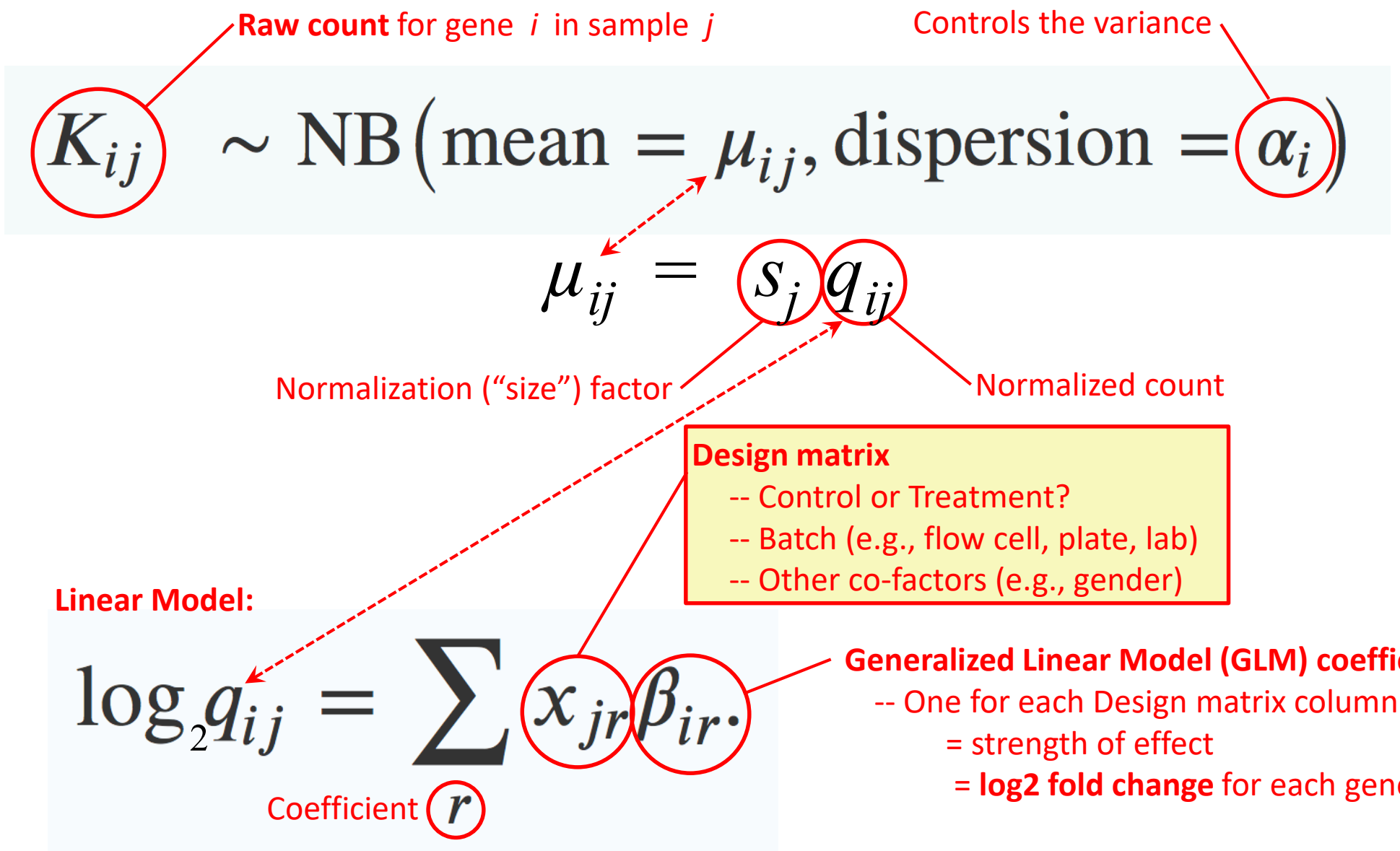
```
dds <- DESeqDataSetFromMatrix(countData = cts, colData = coldata,
design= ~ batch + treatment)
```

```
# Model genotype by treatment interaction:
dds <- DESeqDataSetFromMatrix(countData = cts, colData = coldata,
design= ~ batch + genotype + treatment  + genotype:treatment)
```

```
# Likelihood ratio test for genotype by treatment interaction:
ddsLRT <- DESeq(dds, test="LRT", reduced= ~ batch + genotype + treatment )

resLRT <- results(ddsLRT)
```

DESeq2 fits an negative binomial GLM



DESeq2: One factor design

```
dds <- DESeqDataSetFromMatrix(countData = cts, colData = coldata, design= ~ Genotype)
```

coldata:

Sample	Genotype
wt1	wt
wt2	wt
wt3	wt
mu1	mu
mu2	mu
mu3	mu

Design Matrix:

	(Intercept)	Genotypewt
wt1	1	1
wt2	1	1
wt3	1	1
mu1	1	0
mu2	1	0
mu3	1	0

DESeq2: Two factor design

```
dds <- DESeqDataSetFromMatrix(countData = cts, colData = coldata, design= ~ type + condition)
```

coldata:

file	condition	type
treated1fb	treated	single-read
treated2fb	treated	paired-end
treated3fb	treated	paired-end
untreated1fb	untreated	single-read
untreated2fb	untreated	single-read
untreated3fb	untreated	paired-end
untreated4fb	untreated	paired-end

Design Matrix:

	(Intercept)	typesingle_read	conditiontreated
1)	1	1	1
2)	1	0	1
3)	1	0	1
4)	1	1	0
5)	1	1	0
6)	1	0	0
7)	1	0	0

DESeq2: Two factor design with interaction

```
dds <- DESeqDataSetFromMatrix(countData=cts, colData=coldata, design= ~ strain + minute + strain:minute)
```

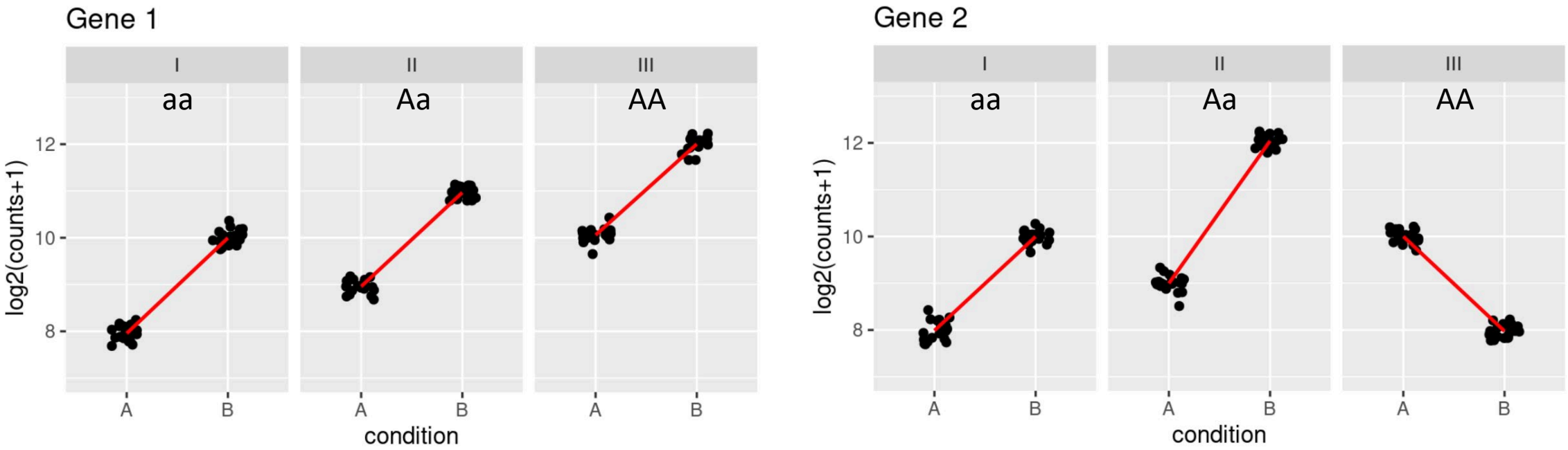
coldata:

	strain	minute
GSM1368273	wt	0
GSM1368274	wt	0
GSM1368275	wt	0
GSM1368285	wt	120
GSM1368286	wt	120
GSM1368287	wt	120
GSM1368291	mut	0
GSM1368292	mut	0
GSM1368293	mut	0
GSM1368303	mut	120
GSM1368304	mut	120
GSM1368305	mut	120

Design Matrix:

	(Intercept)	strainwt	minute120	strainwt:minute120
1)	1	1	0	0
2)	1	1	0	0
3)	1	1	0	0
4)	1	1	1	1
5)	1	1	1	1
6)	1	1	1	1
7)	1	0	0	0
8)	1	0	0	0
9)	1	0	0	0
10)	1	0	1	0
11)	1	0	1	0
12)	1	0	1	0

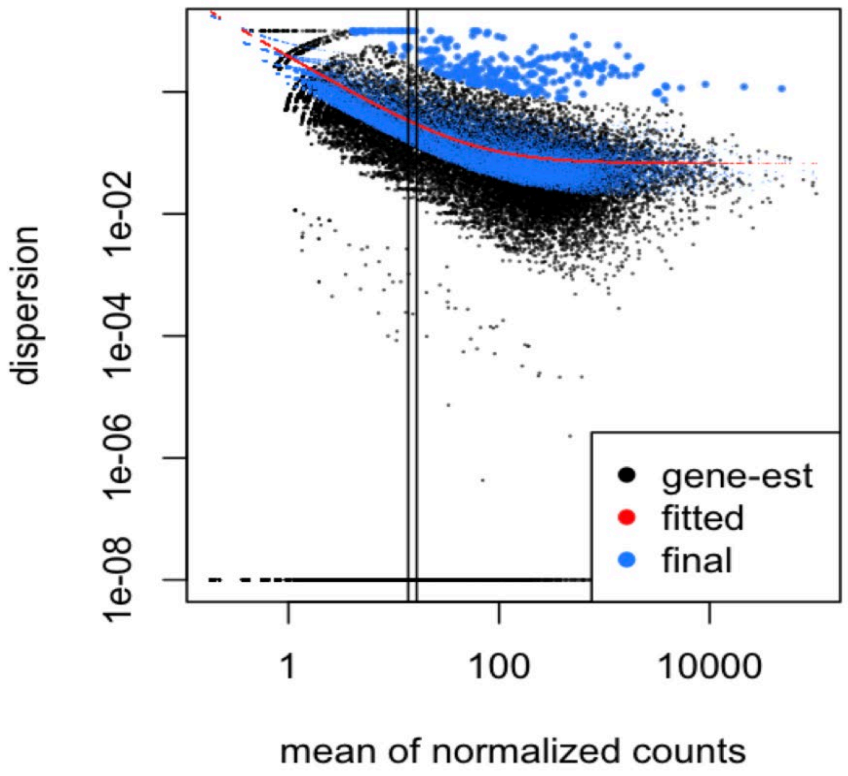
Genotype by Treatment Interaction (3 genotypes)

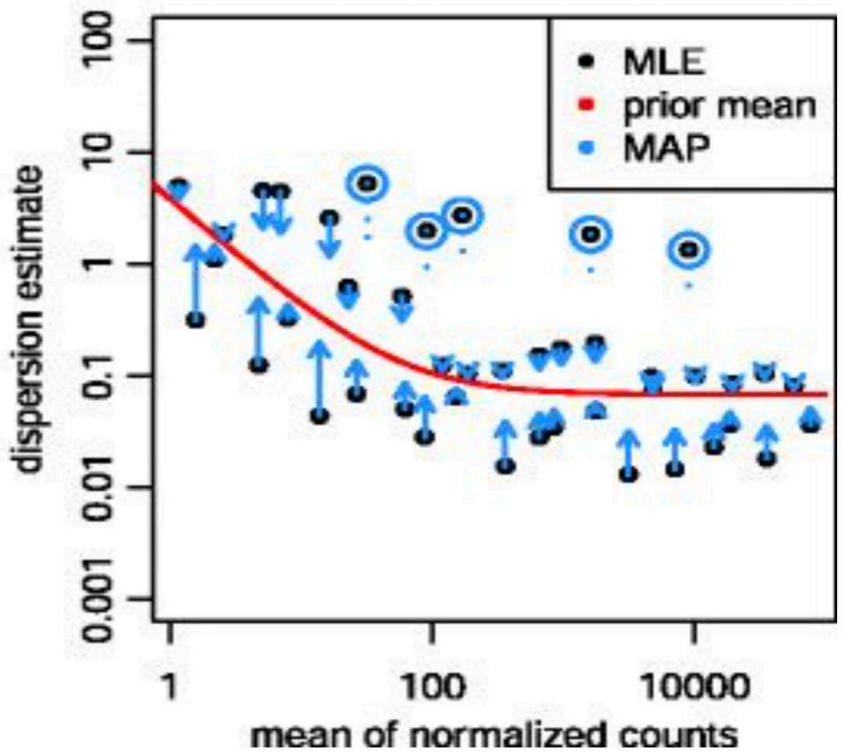



```
dds <- DESeqDataSetFromMatrix(countData = cts, colData = coldata,
design= ~ batch + genotype + treatment  + genotype:treatment)

ddsLRT <- DESeq(dds, test="LRT", reduced= ~ batch + genotype + treatment )
resLRT <- results(ddsLRT)
```

DESeq2: Empirical Bayes shrinkage of dispersion

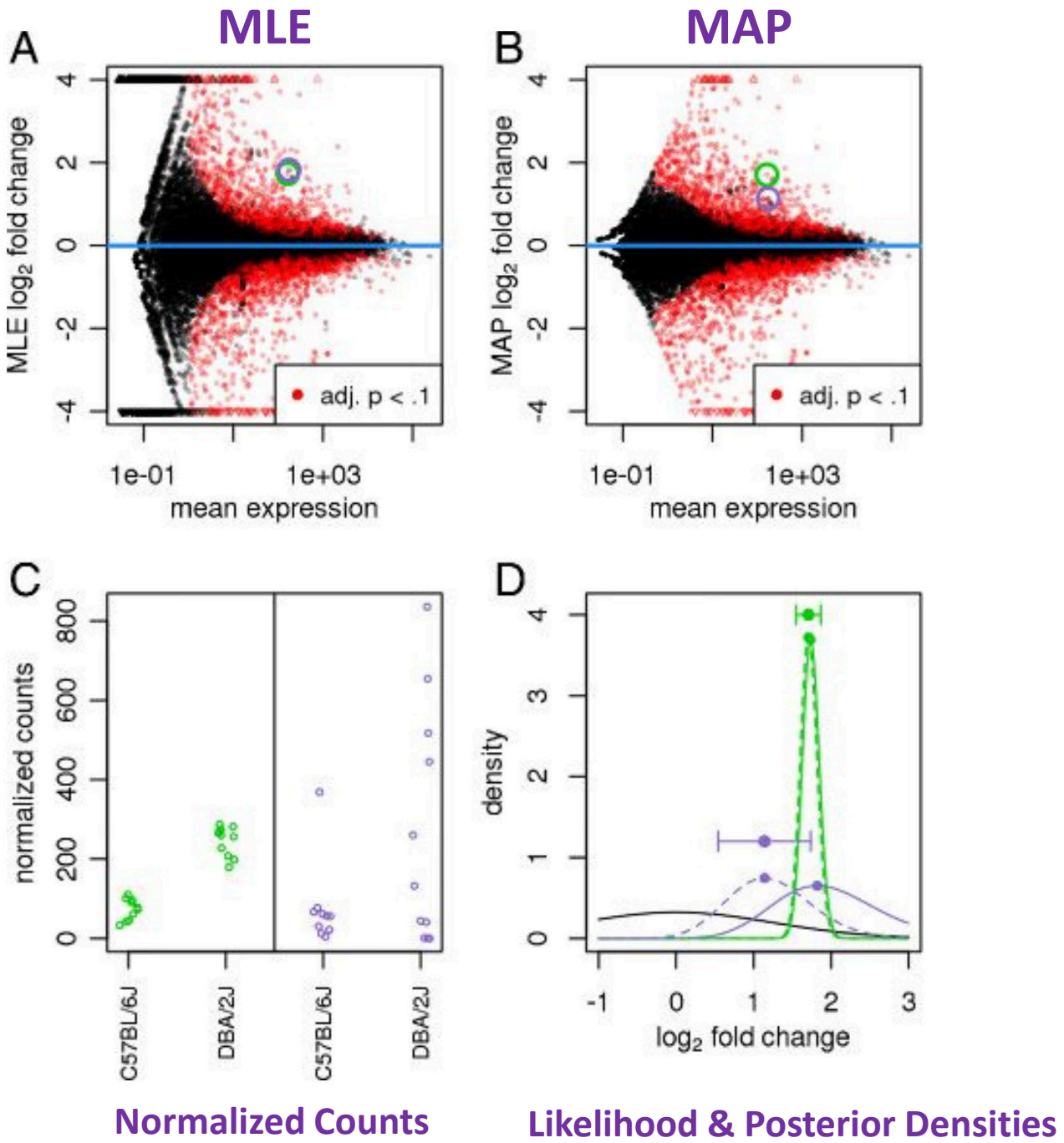




- Not enough replicates to estimate dispersion for individual genes
- Borrow information from genes of similar expression strength among the replicates
- Genes with very high dispersion left as is (violate model assumptions?)

DESeq2: Empirical Bayes shrinkage of fold change

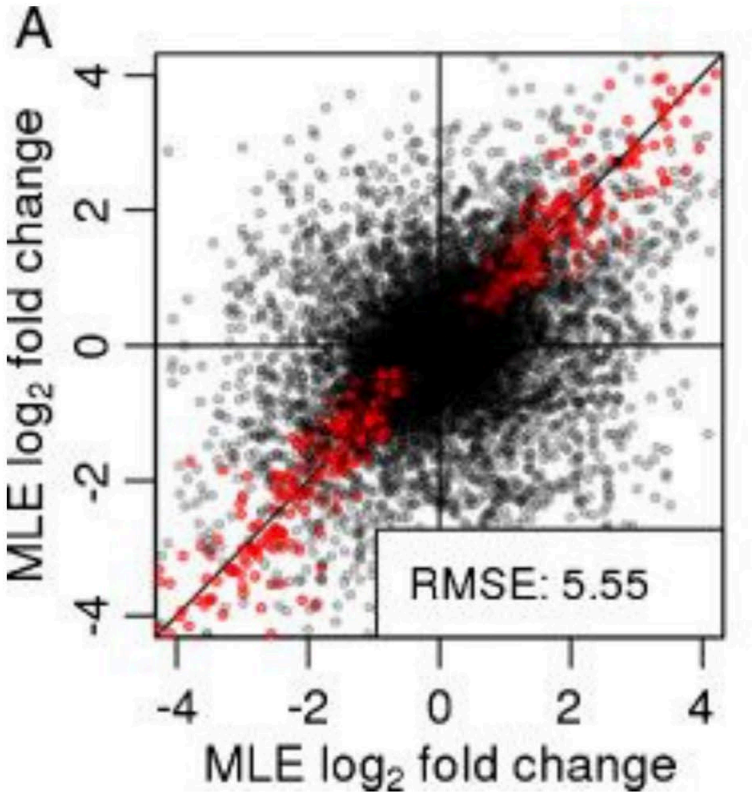
- LFC estimates for weakly expressed genes very noisy and often overestimated



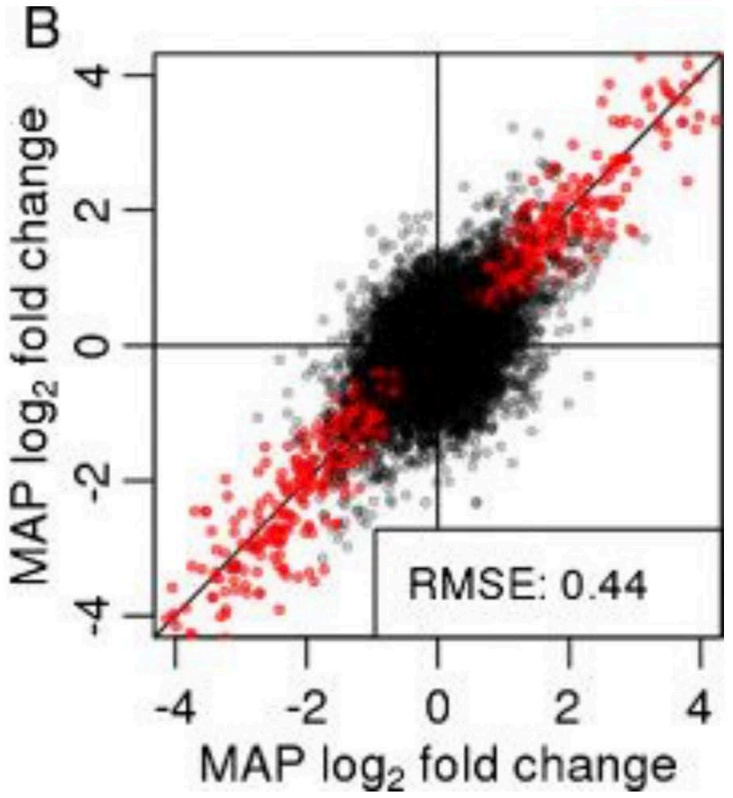
DESeq2: Empirical Bayes shrinkage of log fold change improves reproducibility

- Large data-set split in half → compare log2 fold change estimates for each gene

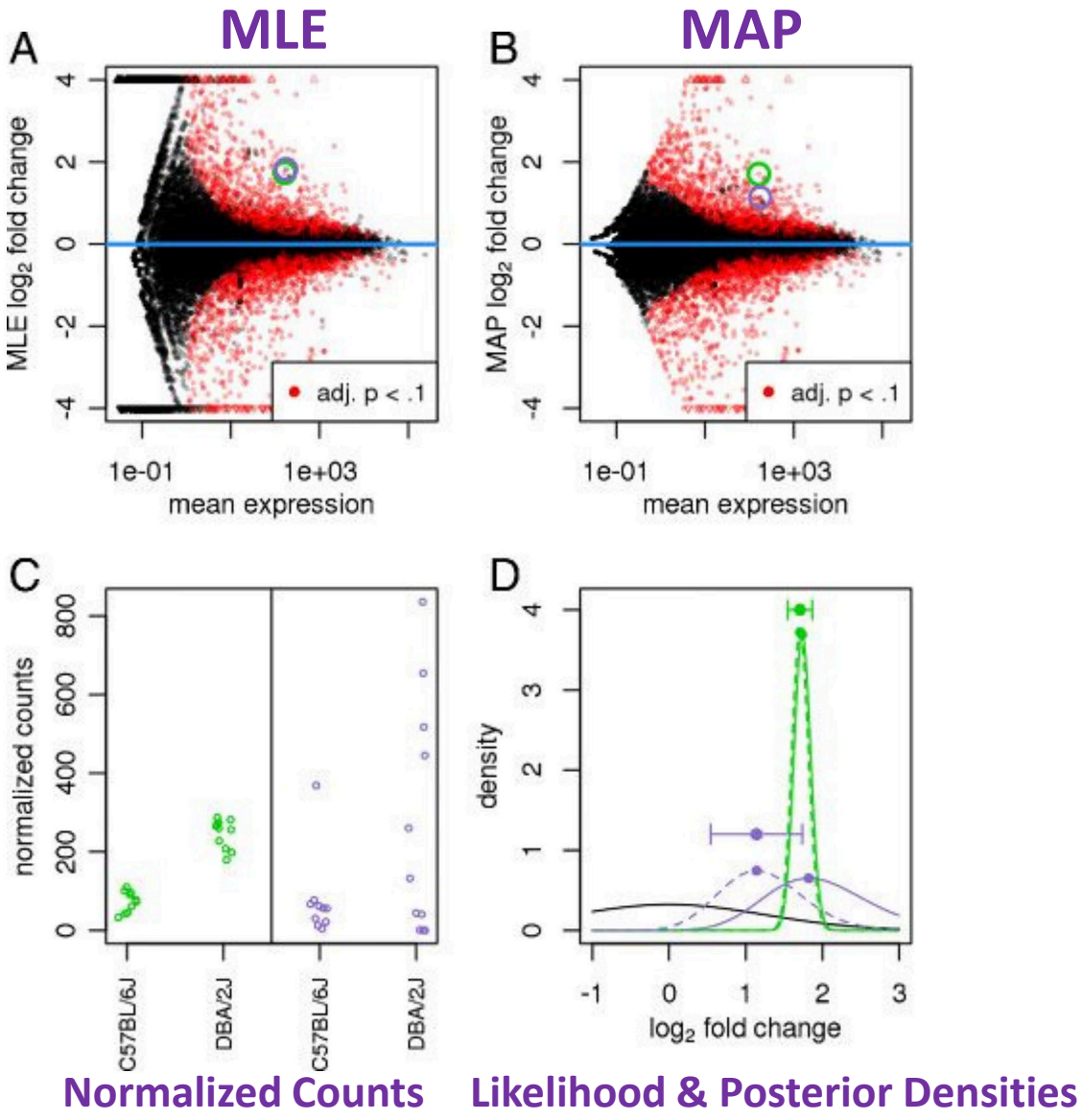
Before shrinkage:



After empirical Bayesian shrinkage:



DESeq2: Statistical test for DE



Test for DE:

- (*shrunkenLFC*) / (*stdErr*) = Wald statistic
- Wald statistic follows std. normal dist.
- *p* value of LFC obtained from standard normal distribution

Automatic Outlier Detection:

- Outlier samples detected using "Cook's distance"
- Few than 7 biological reps: *p* value set to "NA"
- 7 or more biological reps: sample replaced with mean

Multiple Testing:

- *p* values adjusted for multiple testing using Benjamini and Hochberg (1995) procedure
  - Controls *false discovery rate* (FDR)

False Discovery Rate

		Truth		
		Different	**Same**	Total
Experiment	**Different**	TP	FP	R
	Same	FN	TN	m - R
	Total	P	N	m

- **m**: total number of tests (e.g., genes)
- **N**: number of true null hypotheses
- **P**: number of true alternate hypotheses
- **R**: number of rejected null hypotheses (“discoveries”)
- **TP**: number of true positives (“true discoveries”)
- **TN**: number of true negatives
- **FP**: number of false positives (“false discoveries”) (Type I error)
- **FN**: number of false negatives (Type II error)
- **FDR = “false discoveries” / “discoveries” = FP / (FP + TP)**

DESeq2: Automated independent filtering of genes

- DESeq2 automatically omits weakly expressed genes from the multiple testing procedure
  - Fewer tests increase statistical power → more discoveries
- LFC estimates for weakly expressed genes very noisy
  - Very little chance that these will detected as DE (*i.e.*, null hypothesis rejected)
- Threshold overall counts (filter statistic) optimized for target FDR (default FDR = 0.1)

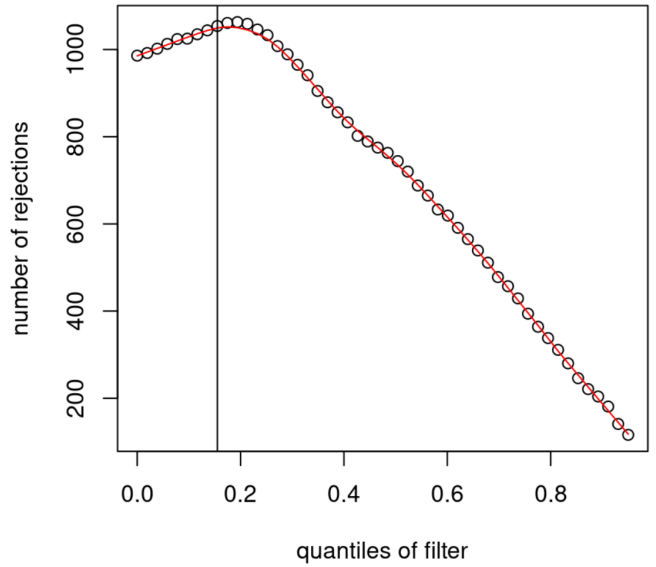


Figure 1: The number of rejected tests for FDR less than 0.1 plotted over theta, the quantiles of the filter statistic.

DESeq2 analysis feedback & summary

```
> dds <- DESeq(dds)
estimating size factors
estimating dispersions
  gene-wise dispersion estimates
  mean-dispersion relationship
  final dispersion estimates
fitting model and testing
```

```
> summary(res)
out of 14177 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 273, 1.9%
LFC < 0 (down)     : 327, 2.3%
outliers [1]       : 25, 0.18%
low counts [2]     : 1650, 12%
(mean count < 4)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

DESeq2: Output of DE analysis

1	gene	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
2	gene27816	416.9148177	-2.511490321	0.428727267	-5.858013979	4.68E-09	5.11E-05
3	gene27620	101.8253191	-2.778767979	0.553389846	-5.021357008	5.13E-07	0.001440413
4	gene31204	365.5088989	-1.143004071	0.227873895	-5.015950038	5.28E-07	0.001440413
5	gene4446	125.745322	-3.205715488	0.637910711	-5.025335725	5.03E-07	0.001440413
6	gene1					20E-06	0.002553055
7	gene1					40E-06	0.002553055
8	gene3					77E-06	0.002756968
9	gene8					23E-06	0.003038386
10	gene2					89E-06	0.005937898
11	gene2					99E-06	0.007929629
12	gene9777	207.8848249	0.90630494	0.202189437	4.482454445	7.38E-06	0.007929629
13	gene21278	357.2070995	-1.375680007	0.309791352	-4.440666269	8.97E-06	0.008159457
14	gene34591	77.02015308	-2.724251177	0.632181028	-4.309289679	1.64E-05	0.013754978

Get list of interesting genes by filtering on:

1. p_{adj} (FDR) < 0.05, and/or
2. log2FoldChange < -1 or >1, and/or
3. baseMean (optional)

…bottom of file = genes excluded from multiple testing:

22902	gene599	0.182683434	1.286409862	4.409114495	0.291761501	0.770468983	NA
22903	gene11602	0.175175275	1.286401869	4.409114509	0.291759687	0.77047037	NA
22904	gene30325	0.168776837	1.606989135	4.409114563	0.364469807	0.715507216	NA
22905	gene35203	0.159702673	1.846921927	4.40661107	0.419125241	0.675124605	NA
22906	gene25371	0.153270142	1.286411877	4.409114495	0.291761958	0.770468634	NA
22907	gene7678	0.141308727	0.93343057	4.407959907	0.211760222	0.832294103	NA
22908	gene13239	0.132221267	1.492522653	2.686412371	0.55558211	0.578496564	NA
22909	gene1935	0.116143364	0.740578083	4.395665987	0.168479153	0.866206343	NA
22910	gene26270	0.107670322	2.315104965	4.402746292	0.525832017	0.599004927	NA
22911	gene30327	0.060455387	0.580738721	4.403240387	0.131888943	0.895072134	NA
22912	gene26805	0.013434773	1.286431679	4.40911449	0.291766449	0.770465199	NA