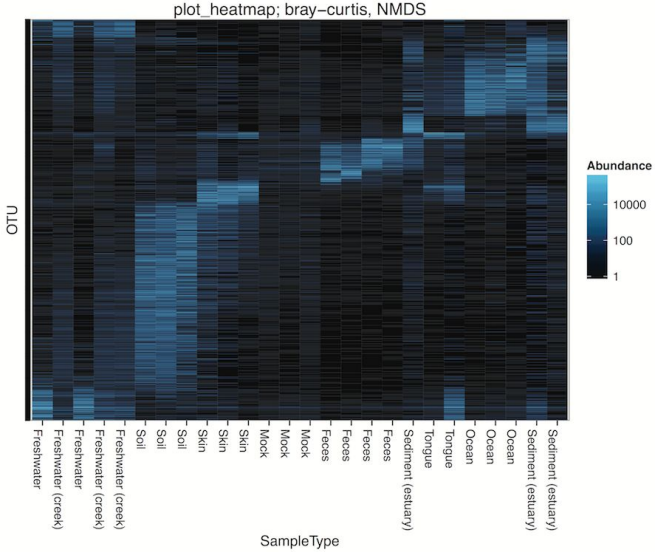
Deconstructing DESeq2

Building Blocks of Differential Abundance Testing

Goal: Characterizing Variation

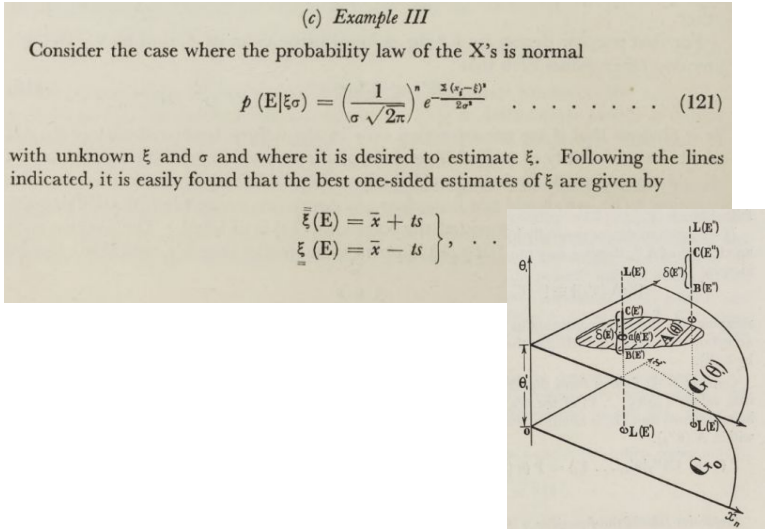
- How does variation in RSV counts reflect sample characteristics?
  - Is it consistent with existing theories?
  - Does it suggest new hypothesis?
- How are characteristics of columns associated with characteristics of rows?



Goal: Characterizing Variation

- Statistical Tools
  - **Inference**: Quantify degree of uncertainty in associations
  - **Visualization**: Compress complexity into interpretable representation

Confidence Intervals (from Neyman's 1937 paper)



(c) *Example III*

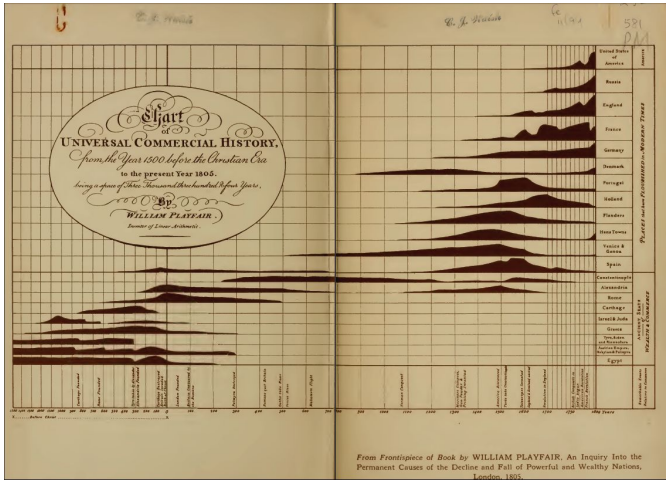
Consider the case where the probability law of the X's is normal

$$p\,(\mathrm{E}|\xi\sigma) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n e^{-\frac{\Sigma(x_i-\xi)^2}{2\sigma^2}} \quad . \; . \; . \; . \; . \; . \; . \; . \quad (121)$$

with unknown ξ and σ and where it is desired to estimate ξ. Following the lines indicated, it is easily found that the best one-sided estimates of ξ are given by

$$\left.\begin{aligned} \bar{\bar{\xi}}(\mathrm{E}) &= \bar{x} + ts \\ \underline{\xi}\,(\mathrm{E}) &= \bar{x} - ts \end{aligned}\right\}, \quad . \; .$$

Time Series (from Playfair, 1805)

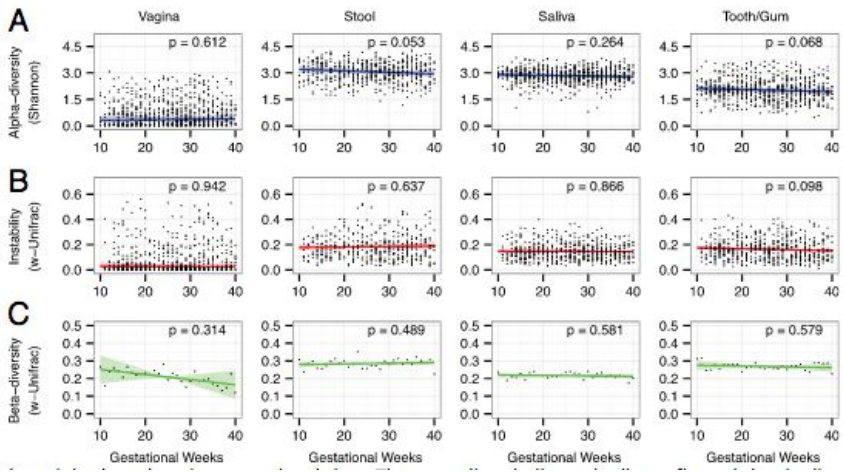


From Frontispiece of Book by WILLIAM PLAYFAIR, An Inquiry Into the Permanent Causes of the Decline and Fall of Powerful and Wealthy Nations, London, 1805.

Goal: Characterizing Variation

- Statistical Tools
  - **Inference**: Quantify degree of uncertainty in associations
  - **Visualization**: Compress complexity into interpretable representation

Confidence intervals **AND** Time Series (Callahan 2015)



Goal: Characterizing Variation

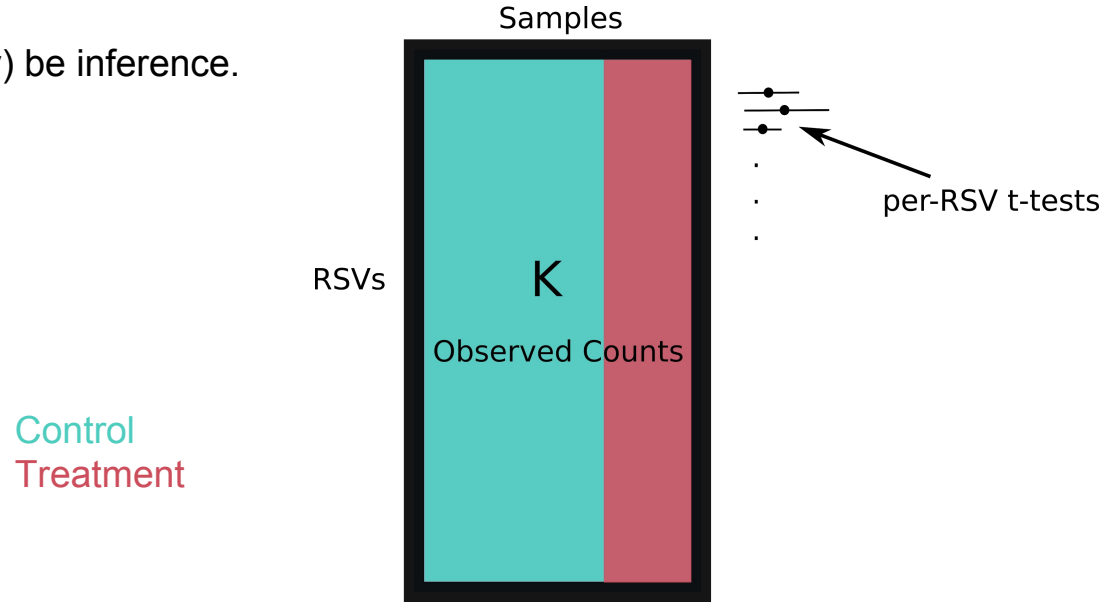
- Statistical Tools
  - **Inference**: Quantify degree of uncertainty in associations
  - **Visualization**: Compress complexity into interpretable representation

Our focus here will (mostly) be inference.

Goal: Characterizing Variation

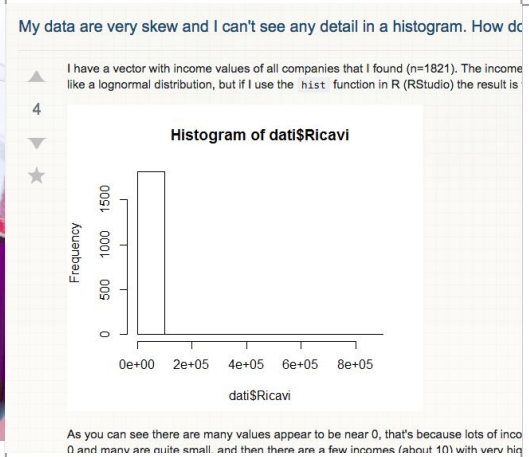
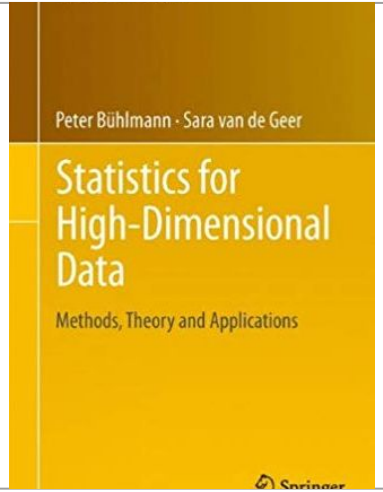
- Statistical Tools
  - **Inference**: Quantify degree of uncertainty in associations
  - **Visualization**: Compress complexity into interpretable representation

Our focus here will (mostly) be inference.
A (naive) starting point:



Challenges

- A few characteristics of microbiome data make it challenging to analyze
- We'll discuss techniques for dealing with these issues
- Especially in relation to DESeq2

Batch Effects ("normalization")	Count structure / Skewness	High-Dimensionality (few samples + multiple testing)
		

DESeq2 Overview

- Method designed for RNA-seq differential expression analysis
- Has been used widely in microbiome studies
  - Microbiome-specific adaptation still open research problem (as far as I am aware)

METHOD | Open Access

Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2

Michael I Love, Wolfgang Huber and Simon Anders

Genome Biology 2014 **15**:550

https://doi.org/10.1186/s13059-014-0550-8

Received: 27 May 2014 | **Accepted:** 19 November 2014 | **Published:** 5 December 2014

DESeq2 Overview

- Method designed for RNA-seq differential expression analysis
- Has been used widely in microbiome studies
  - Microbiome-specific adaptation still open research problem (as far as I am aware)

The underlying math:

$$K_{ij} \sim \mathrm{GP}(\mu_{ij}, \alpha_i)$$

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

$$\log_2(q_{ij}) = \sum_k x_{jk} \beta_{ik}.$$

DESeq2 Overview

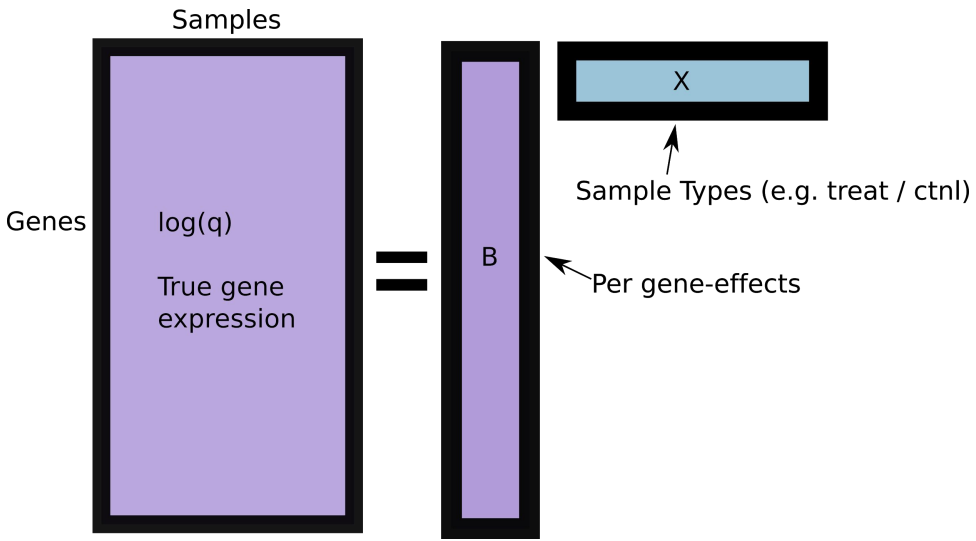
- Method designed for RNA-seq differential expression analysis
- Has been used widely in microbiome studies
  - Microbiome-specific adaptation still open research problem

The underlying math:

$$K_{ij} \sim \mathrm{GP}(\mu_{ij}, \alpha_i)$$

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

$$\boxed{\log_2(q_{ij}) = \sum_k x_{jk} \beta_{ik}.}$$



DESeq2 Overview

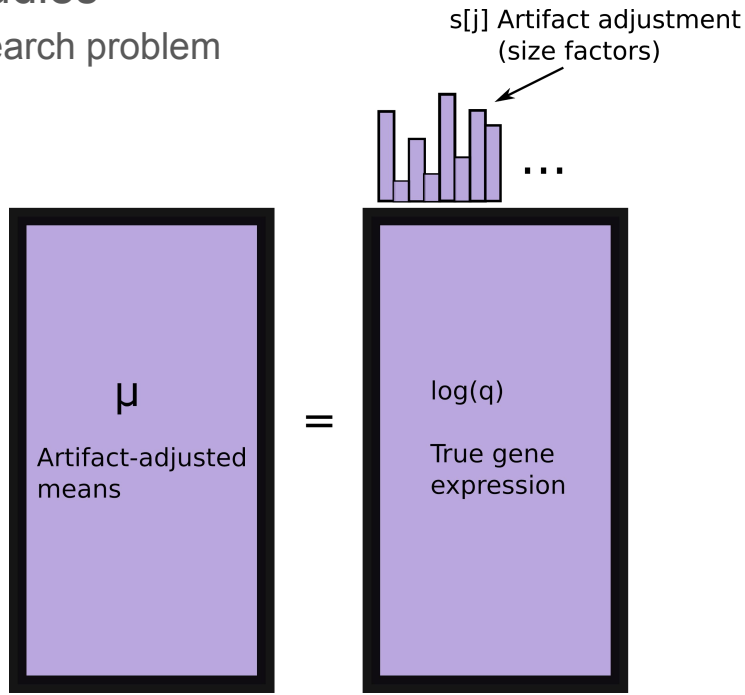
- Method designed for RNA-seq differential expression analysis
- Has been used widely in microbiome studies
  - Microbiome-specific adaptation still open research problem

The underlying math:

$$K_{ij} \sim \mathrm{GP}(\mu_{ij}, \alpha_i)$$

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

$$\log_2(q_{ij}) = \sum_k x_{jk}\beta_{ik}.$$



DESeq2 Overview

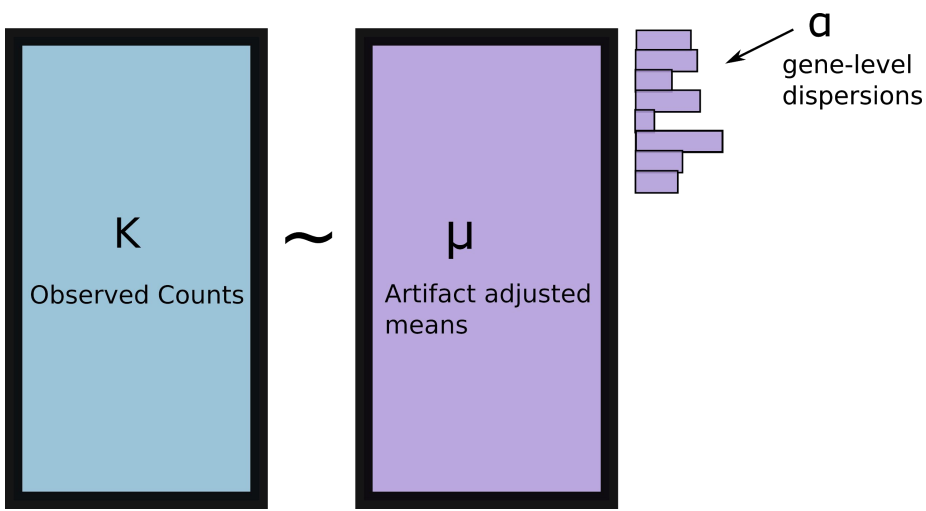
- Method designed for RNA-seq differential expression analysis
- Has been used widely in microbiome studies
  - Microbiome-specific adaptation still open research problem

The underlying math:

$$\boxed{K_{ij} \sim \mathrm{GP}(\mu_{ij}, \alpha_i)}$$

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

$$\log_2(q_{ij}) = \sum_k x_{jk} \beta_{ik}.$$



DESeq2 Overview

- Method designed for RNA-seq differential expression analysis
- Has been used widely in microbiome studies
  - Microbiome-specific adaptation still open research problem

The underlying math:

$$K_{ij} \sim \mathrm{GP}(\mu_{ij}, \alpha_i)$$

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

$$\log_2(q_{ij}) = \sum_k x_{jk} \beta_{ik}.$$

```r
#' @export
DESeq <- function(object, test=c("Wald","LRT"),
                  fitType=c("parametric","local","mean"),
                  sfType=c("ratio","poscounts","iterate"),
                  betaPrior,
                  full=design(object), reduced, quiet=FALSE,
                  minReplicatesForReplace=7, modelMatrixType,
                  useT=FALSE, minmu=0.5,
                  parallel=FALSE, BPPARAM=bpparam()) {
  # check arguments
  stopifnot(is(object, "DESeqDataSet"))
  test <- match.arg(test, choices=c("Wald","LRT"))
```

https://github.com/mikelove/DESeq2/blob/master/R/core.R

DESeq2 Overview

- Method designed for RNA-seq differential expression analysis
- Has been used widely in microbiome studies
  - Microbiome-specific adaptation still open research problem

The underlying math:

$$K_{ij} \sim \mathrm{GP}(\mu_{ij}, \alpha_i)$$

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

$$\log_2(q_{ij}) = \sum_k x_{jk} \beta_{ik}.$$

```
#' dds <- makeExampleDESeqDataSet()
#' dds <- estimateSizeFactors(dds)
#' dds <- estimateDispersions(dds)
#' dds <- nbinomWaldTest(dds)
#' res <- results(dds)
```

https://github.com/mikelove/DESeq2/blob/master/R/core.R

DESeq2 Overview

- Method designed for RNA-seq differential expression analysis
- Has been used widely in microbiome studies
  - Microbiome-specific adaptation still open research problem

The underlying math:

$$K_{ij} \sim \mathrm{GP}(\mu_{ij}, \alpha_i)$$

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

$$\log_2(q_{ij}) = \sum_k x_{jk}\beta_{ik}.$$

```
#' dds <- makeExampleDESeqDataSet()
#' dds <- estimateSizeFactors(dds)
#' dds <- estimateDispersions(dds)
#' dds <- nbinomWaldTest(dds)
#' res <- results(dds)
```

https://github.com/mikelove/DESeq2/blob/master/R/core.R

DESeq2 Overview

- Method designed for RNA-seq differential expression analysis
- Has been used widely in microbiome studies
  - Microbiome-specific adaptation still open research problem

The underlying math:

$$K_{ij} \sim \mathrm{GP}(\mu_{ij}, \boxed{\alpha_i})$$

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

$$\log_2(q_{ij}) = \sum_k x_{jk} \beta_{ik}.$$

```
#' dds <- makeExampleDESeqDataSet()
#' dds <- estimateSizeFactors(dds)
#' dds <- estimateDispersions(dds)
#' dds <- nbinomWaldTest(dds)
#' res <- results(dds)
```

https://github.com/mikelove/DESeq2/blob/master/R/core.R

DESeq2 Overview

- Method designed for RNA-seq differential expression analysis
- Has been used widely in microbiome studies
  - Microbiome-specific adaptation still open research problem

The underlying math:

$$\boxed{K_{ij} \sim \mathrm{GP}(\mu_{ij}, \alpha_i)}$$

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

$$\log_2(q_{ij}) = \sum_k x_{jk} \boxed{\beta_{ik}}$$

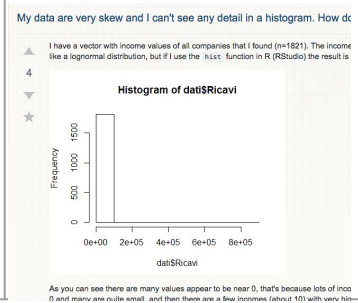
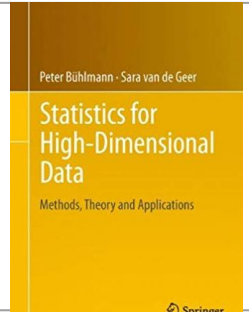
```
#' dds <- makeExampleDESeqDataSet()
#' dds <- estimateSizeFactors(dds)
#' dds <- estimateDispersions(dds)
#' dds <- nbinomWaldTest(dds)
#' res <- results(dds)
```

https://github.com/mikelove/DESeq2/blob/master/R/core.R

DESeq2 Overview

- Method designed for RNA-seq differential expression analysis
- Has been used widely in microbiome studies
  - Microbiome-specific adaptation still open research problem

Let’s try to motivate each component.

Batch Effects (“normalization”)	**Count structure / Skewness**	**High-Dimensionality (few samples + multiple testing)**
		

Normalization

Why do we need normalization?

- Sources of technical variation resulting from experimental setup
- Confounds true biological variation of interest

Normalization

Why do we need normalization?

- Sources of technical variation resulting from experimental setup
- Confounds true biological variation of interest

Examples

- Differences in sample prep or sequencing protocol
    - Sequencing depth

Normalization

Why do we need normalization?

- Sources of technical variation resulting from experimental setup
- Confounds true biological variation of interest

Examples

- Differences in sample prep or sequencing protocol
  - Sequencing depth
- True biological effects, unrelated to what you care about
  - Age of person sample was collected from

Simple (but problematic) Solutions

- Rarefaction
  - Subsample counts across samples down to the minimum observed in any
- Convert to proportions
  - Divide all samples by their total counts
- Quantile normalization
  - Divide samples by their value at a particular quantile (e.g., 90%)

Why are these problematic?

Simple (but problematic) Solutions

- Rarefaction
  - Subsample counts across samples down to the minimum observed in any
- Convert to proportions
  - Divide all samples by their total counts
- Quantile normalization
  - Divide samples by their value at a particular quantile (e.g., 90%)

Why are these problematic?

Overall counts are informative

- (4, 7, 2) ≠ (400, 700, 200)
- Larger Counts → Lower Uncertainty

Simple (but problematic) Solutions

- Rarefaction
  - Subsample counts across samples down to the minimum observed in any
- Convert to proportions
  - Divide all samples by their total counts
- Quantile normalization
  - Divide samples by their value at a particular quantile (e.g., 90%)

Why are these problematic?

Overall counts are informative

- (4, 7, 2) ≠ (400, 700, 200)
- Larger Counts → Lower Uncertainty

RESEARCH ARTICLE

Waste Not, Want Not: Why Rarefying Microbiome Data Is Inadmissible

Paul J. McMurdie, Susan Holmes

Published: April 3, 2014 • https://doi.org/10.1371/journal.pcbi.1003531

Factors of Technical Variation

General theme: Remove latent factors likely due to technical variation.

Normalization of RNA-seq data using factor analysis of control genes or samples

Davide Risso[1], John Ngai[2–4], Terence P Speed[1,5,6] & Sandrine Dudoit[1,7]

Normalization of RNA-sequencing (RNA-seq) data has proven essential to ensure accurate inference of expression levels. Here, we show that usual normalization approaches mostly account for sequencing depth and fail to correct for library preparation and other more complex unwanted technical effects. We evaluate the performance of the External RNA Control Consortium (ERCC) spike-in controls and investigate

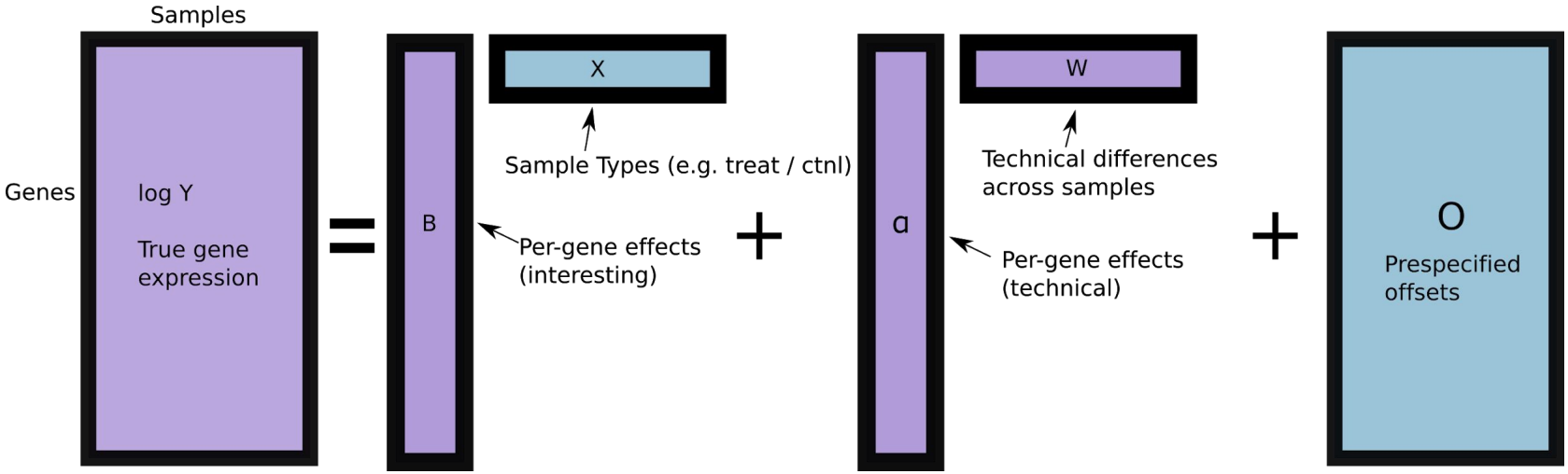
than simply differences in sequencing depths; we refer to such typically unknown nuisance technical effects as unwanted variation.

One largely unexplored direction is the inclusion of spike-in controls in the normalization procedure. Controls have been successfully employed in microarray normalization, for mRNA arrays[7,8] and, more recently, microRNA arrays[9]. One of the advantages of using negative controls in the normalization procedure is the possibility of

Factors of Technical Variation

General theme: Remove latent factors likely due to technical variation.

$$\log E\left[Y \middle| W, X, O\right] = W\alpha + X\beta + O$$



Refinement: Negative Controls

Suppose a gene had two characteristics,

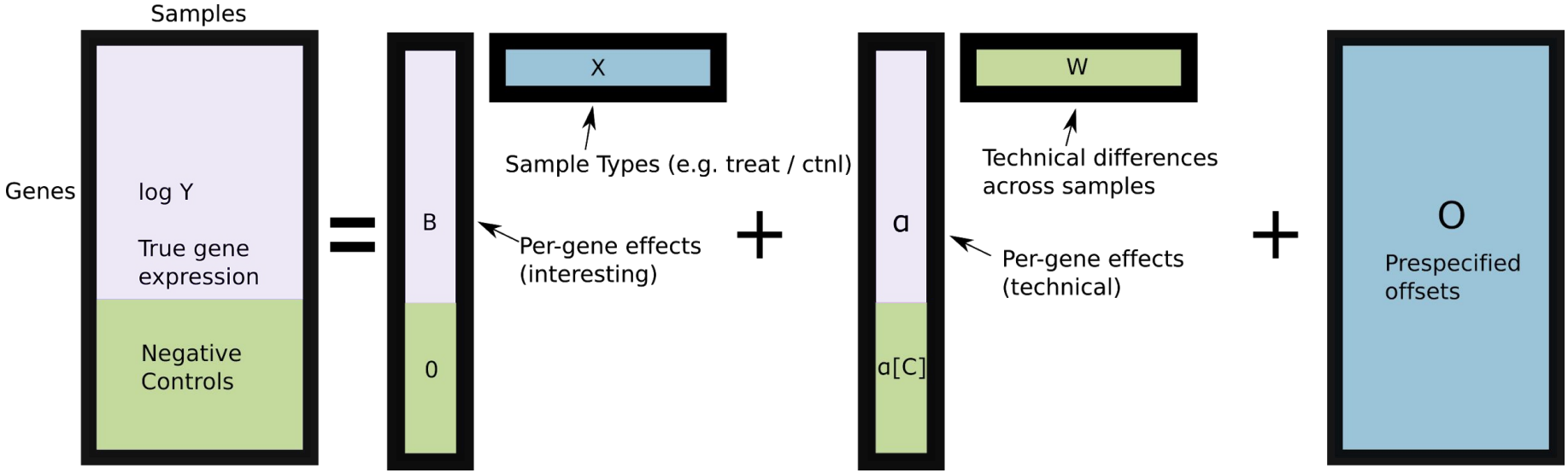
- Gene is unaffected by treatment / control
- Technical variation affects this gene in the same way it affects all others

This gene can be used to “correct” for technical variation in the RUV setup.

Refinement: Negative Controls

Suppose a gene had two characteristics,

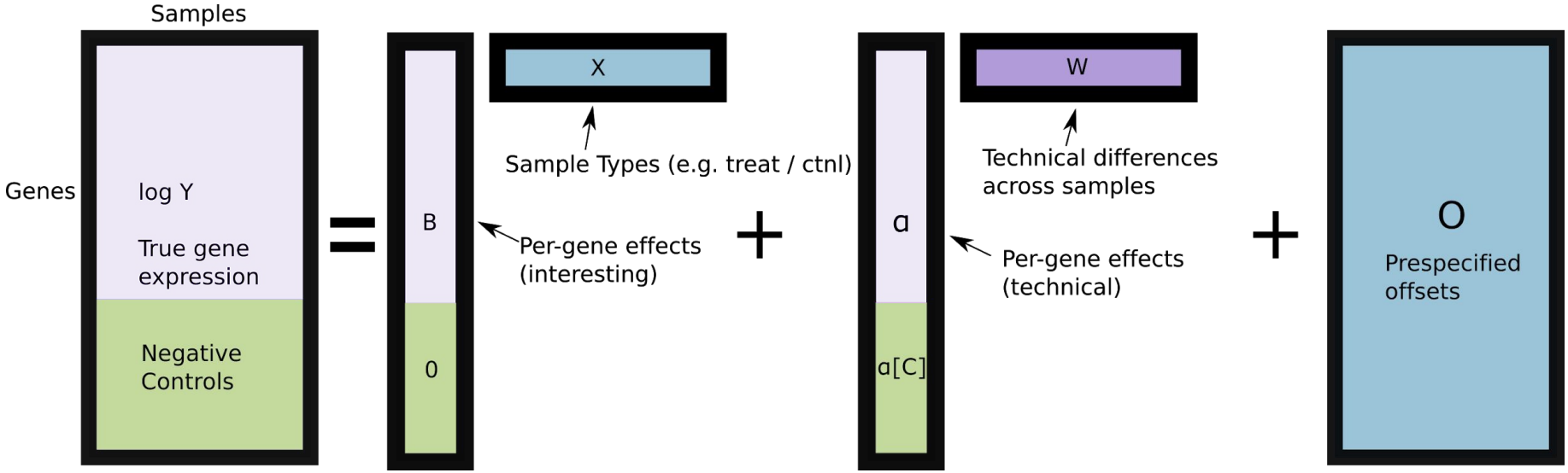
- Gene is unaffected by treatment / control
- Technical variation affects this gene in the same way it affects all others



Refinement: Negative Controls

Suppose a gene had two characteristics,

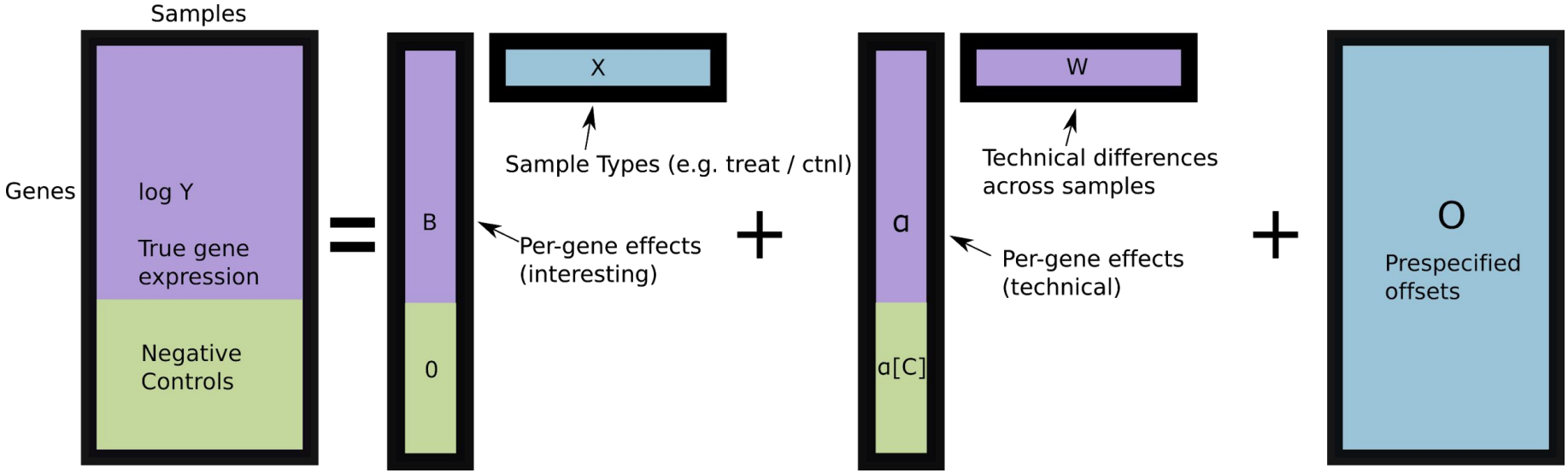
- Gene is unaffected by treatment / control
- Technical variation affects this gene in the same way it affects all others



Refinement: Negative Controls

Suppose a gene had two characteristics,

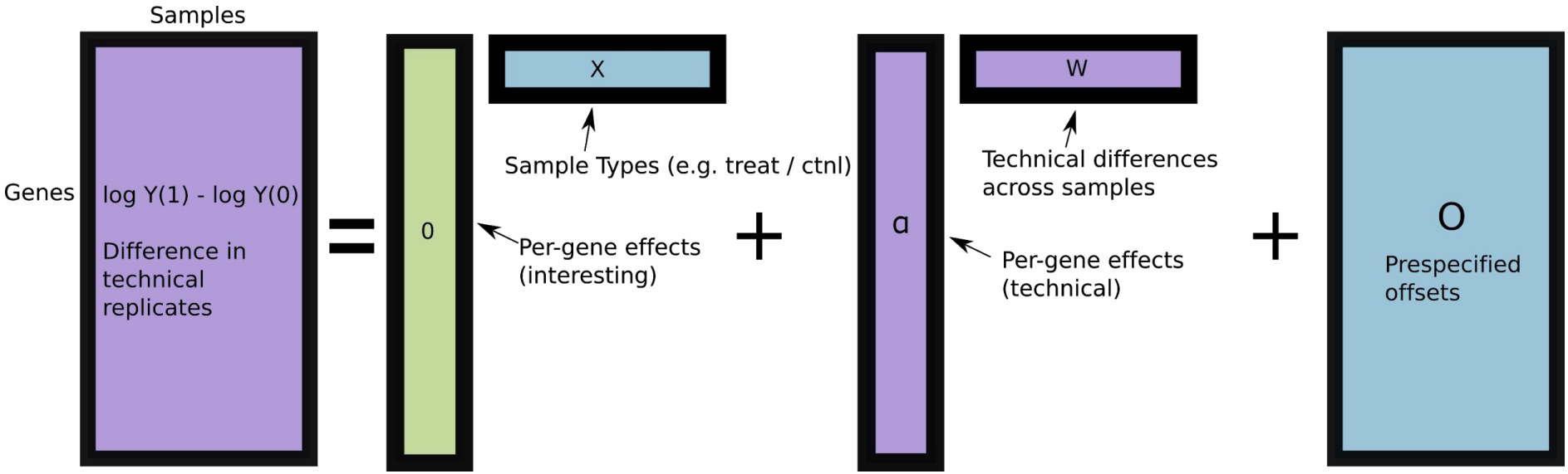
- Gene is unaffected by treatment / control
- Technical variation affects this gene in the same way it affects all others



Refinement: Technical Replicates

Negative control correction is sensitive to the choice of control genes.

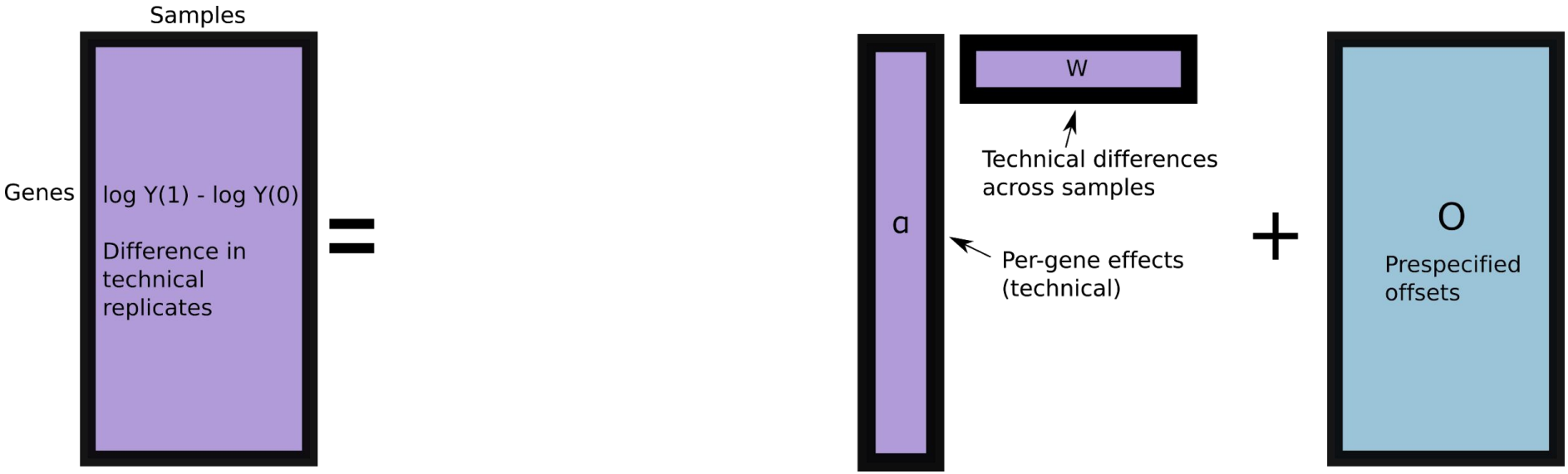
Alternatively, we can use technical replicates → should exhibit no true variation



Refinement: Technical Replicates

Negative control correction is sensitive to the choice of control genes.

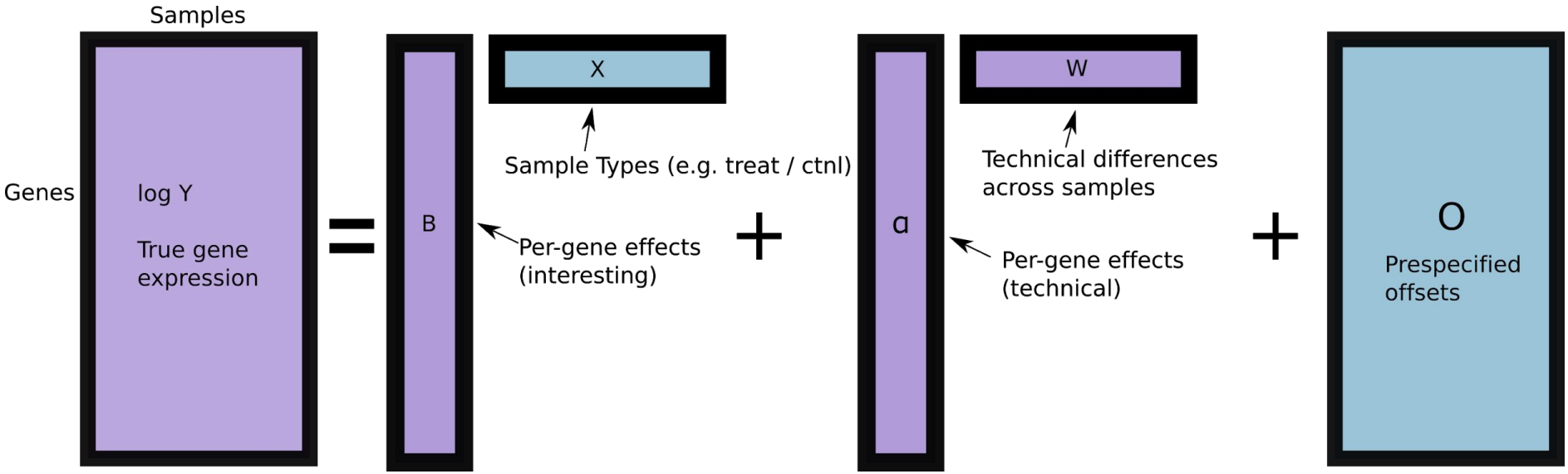
Alternatively, we can use technical replicates → should exhibit no true variation



Refinement: Technical Replicates

Negative control correction is sensitive to the choice of control genes.

Alternatively, we can use technical replicates → should exhibit no true variation



Estimating Factors

- DESeq2 does something closer to upper quantile normalization

$$s_j = \underset{i:\ K_i^{\mathrm{R}} \neq 0}{\mathrm{median}} \frac{K_{ij}}{K_i^{\mathrm{R}}} \quad \text{with} \quad K_i^{\mathrm{R}} = \left(\prod_{j=1}^{m} K_{ij} \right)^{1/m}.$$

Estimating Factors

- DESeq2 does something closer to upper quantile normalization

$$s_j = \underset{i:\ K_i^{\mathrm{R}} \neq 0}{\mathrm{median}} \frac{K_{ij}}{\boxed{K_i^{\mathrm{R}}}} \quad \text{with} \quad K_i^{\mathrm{R}} = \left(\prod_{j=1}^{m} K_{ij} \right)^{1/m}.$$

Typical value for that RSV

Estimating Factors

- DESeq2 does something closer to upper quantile normalization

How much larger is this sample, for that RSV?

$$s_j = \underset{i:\ K_i^{\mathrm{R}} \neq 0}{\mathrm{median}} \boxed{\frac{K_{ij}}{\boxed{K_i^{\mathrm{R}}}}} \quad \text{with} \quad K_i^{\mathrm{R}} = \left(\prod_{j=1}^{m} K_{ij} \right)^{1/m}.$$

Typical value for that RSV

Estimating Factors

- DESeq2 does something closer to upper quantile normalization

How much larger is this sample, across RSVs?

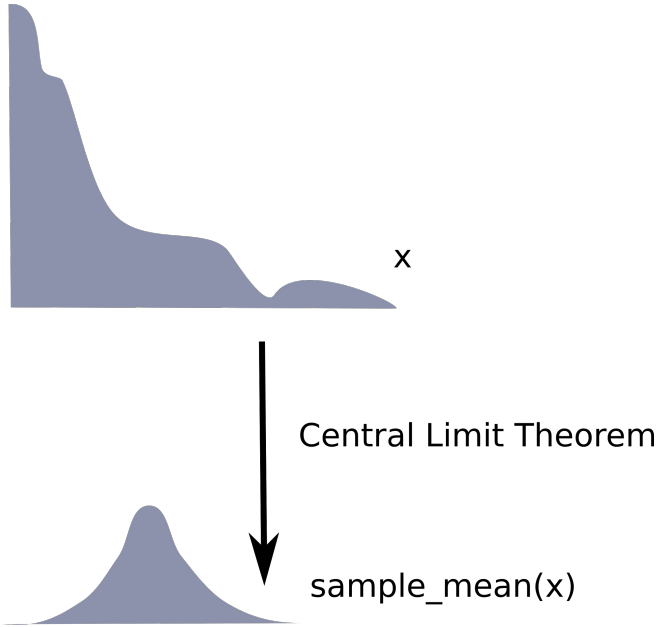
How much larger is this sample, for that RSV?

$$s_j = \boxed{\underset{i:\ K_i^{\mathrm{R}} \neq 0}{\mathrm{median}}} \boxed{\frac{K_{ij}}{\boxed{K_i^{\mathrm{R}}}}} \quad \text{with} \quad K_i^{\mathrm{R}} = \left(\prod_{j=1}^{m} K_{ij} \right)^{1/m}.$$

Typical value for that RSV

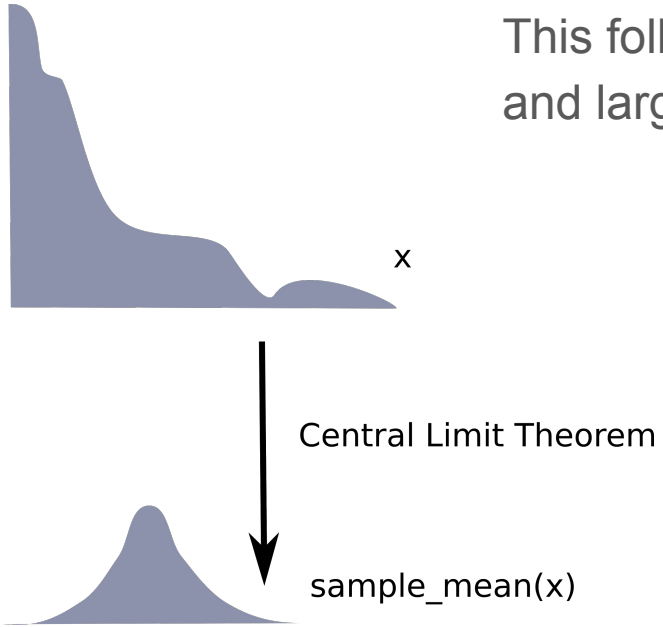
Count Structure (and skewness)

- Misconception: To use a t-test, you need normally distributed data.
- Reality: You only need normality in means



Count Structure (and skewness)

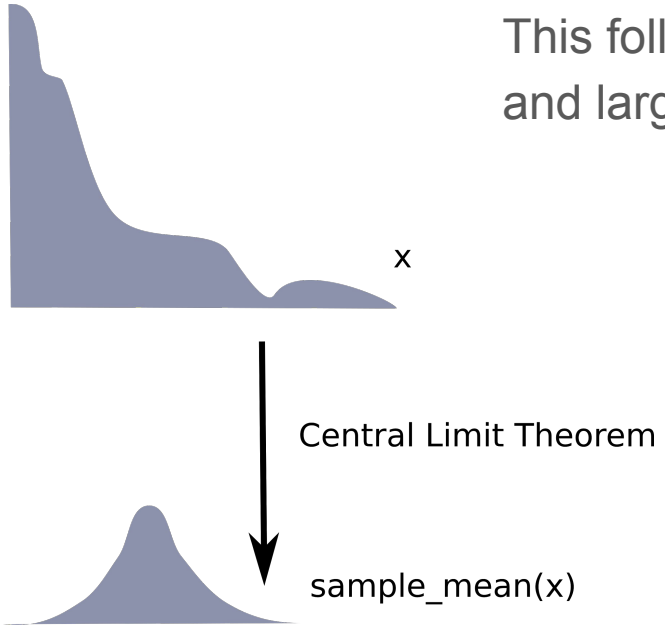
- Misconception: To use a t-test, you need normally distributed data.
- Reality: You only need normality in means



This follows from the central limit theorem and large enough sample sizes.

Count Structure (and skewness)

- Misconception: To use a t-test, you need normally distributed data.
- Reality: You only need normality in means

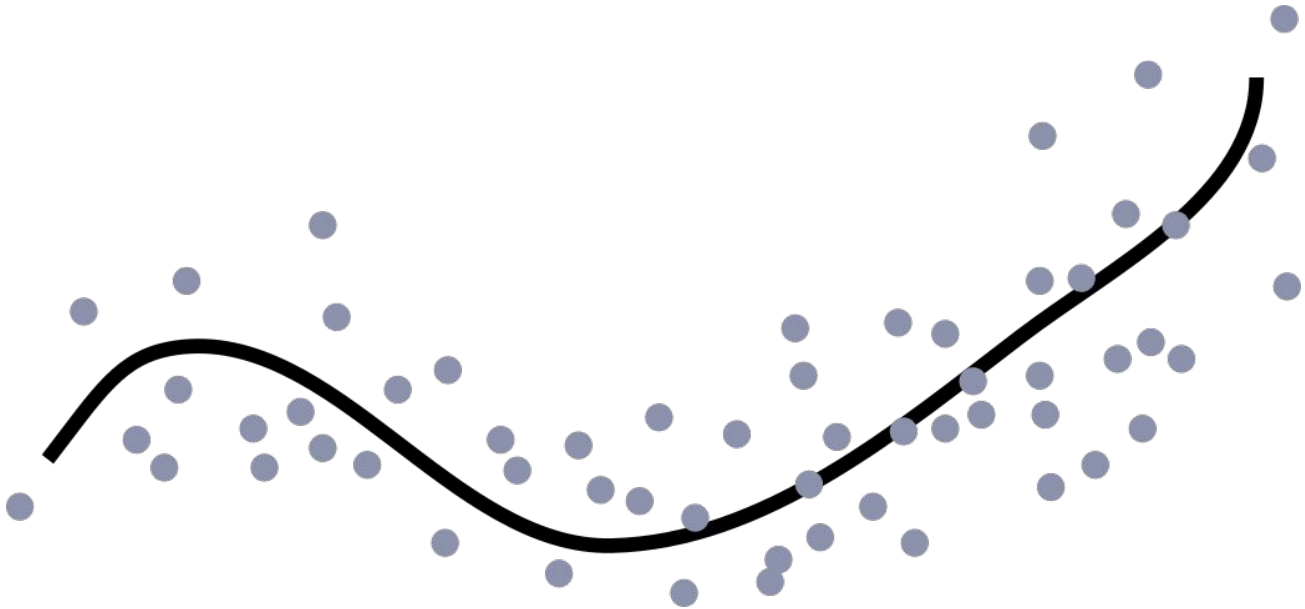


This follows from the central limit theorem and large enough sample sizes.

Fundamental Problem
We usually need covariates (can't just use two-group means), and need to model the original count data.

Usual Linear Regression

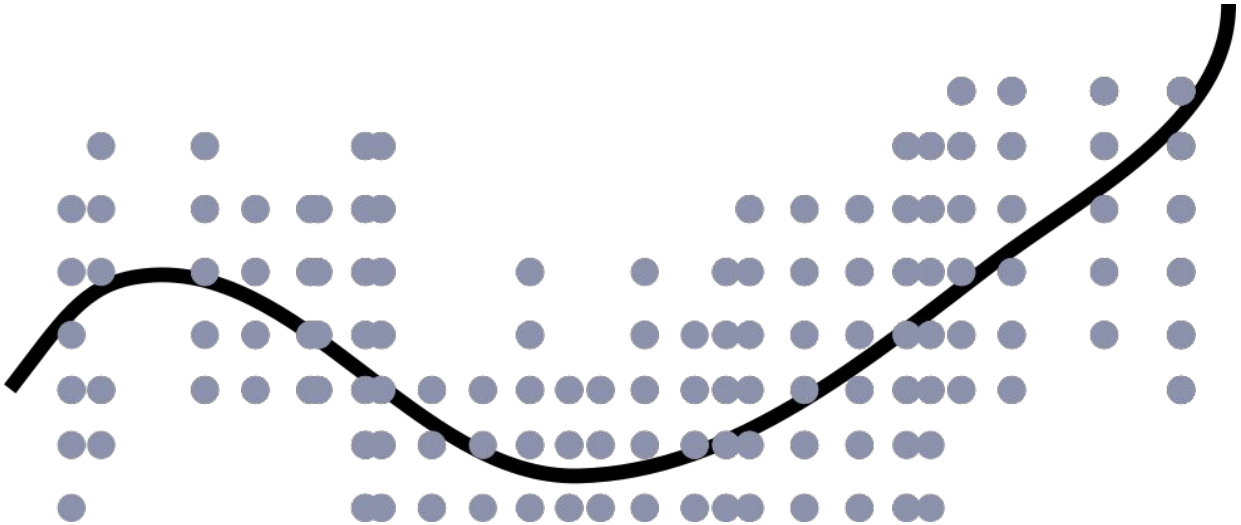
$$y_i \sim N(y_i | \mu(x_i), \sigma^2)$$



Gaussian errors around a regression function.

Usual Linear Regression

$$\cancel{y_i \sim N(y_i | \mu(x_i), \sigma^2)}$$



Gaussian errors around a regression function.

This error structure makes no sense for count data!

Alternative: Generalized Linear Models

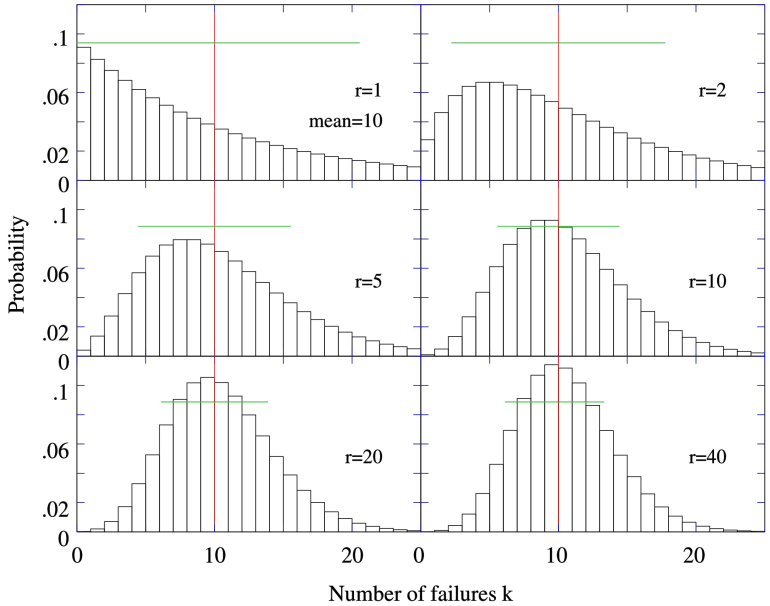
- Generalized linear models extend linear regression to other error structures

$$y_i \sim \mathrm{ExpFam}(y_i | \eta(x_i))$$

- For example, can make the data be Poisson around a regression function
- Inferential theory from linear regression carries over (confidence intervals, prediction intervals, p-values, ...)

Overdispersion → Negative Binomial Distn.

- Poisson models tend to underestimate variance (only one parameter)
- A two-parameter alternative is the Negative Binomial distribution
- (also called “Gamma-Poisson”)



Fixed mean, but different amounts of *dispersion*, according to parameter r [from wikipedia]

Overdispersion → Negative Binomial Distn.

- Poisson models tend to underestimate variance (only one parameter)
- A two-parameter alternative is the Negative Binomial distribution
- (also called “Gamma-Poisson”)

$$\boxed{K_{ij} \sim \mathrm{GP}(\mu_{ij}, \alpha_i)}$$

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

$$\log_2(q_{ij}) = \sum_k x_{jk} \boxed{\beta_{ik}}$$

```
#' dds <- makeExampleDESeqDataSet()
#' dds <- estimateSizeFactors(dds)
#' dds <- estimateDispersions(dds)
#' dds <- nbinomWaldTest(dds)
#' res <- results(dds)
```

https://github.com/mikelove/DESeq2/blob/master/R/core.R

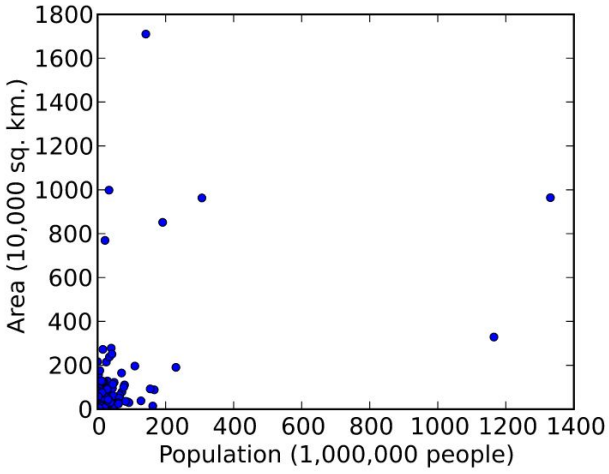
Aside: Unsupervised Versions

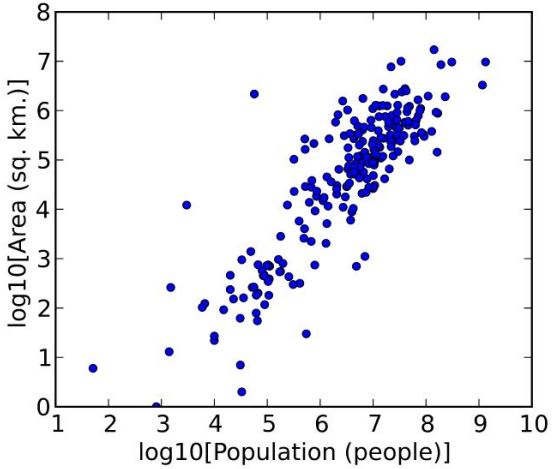
- There are ways to do flexible count modeling in purely unsupervised settings
- (still sort of a research area though)
- References

  - Mohamed, Shakir, Zoubin Ghahramani, and Katherine A. Heller. "Bayesian exponential family PCA." Advances in neural information processing systems. 2009.
  - Lopez, Romain, et al. "A deep generative model for gene expression profiles from single-cell RNA sequencing." arXiv preprint arXiv:1709.02082 (2017).
  - Pierson, Emma, and Christopher Yau. "ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis." Genome biology 16.1 (2015): 241.
  - Sankaran, Kris, and Susan P Holmes; Latent variable modeling for the microbiome, Biostatistics, , kxy018, https://doi.org/10.1093/biostatistics/kxy018.

Aside: Transformations

- An alternative to explicitly modeling count data is to transform it first
  - log(x)
  - log(pseudo + x)
  - asinh(x)
  - Variance Stabilizing Transform, Regularized Log
- Advantage: Can plug transformed data into generic methods
- Disadvantage: Lose probabilistic interpretation





https://en.wikipedia.org/wiki/Data_transformation_(statistics)

High-Dimensionality

- Lots of RSVs, relatively few samples

High-Dimensionality

- Lots of RSVs, relatively few samples
- If we study one gene at a time, this is bad news

(Not enough samples to say anything with certainty)

High-Dimensionality

- Lots of RSVs, relatively few samples (Not enough samples to say anything with certainty)
- If we study one gene at a time, this is bad news
- If we study lots of genes in isolation, this is bad news (Multiple testing problem)

High-Dimensionality

- Lots of RSVs, relatively few samples (Not enough samples to say anything with certainty)
- If we study one gene at a time, this is bad news
- If we study lots of genes in isolation, this is bad news (Multiple testing problem)

Two general solution strategies,

- Share information whenever possible
- Control False Discovery Rates

Sharing Information: Random Effects Models

- When we are trying to estimate across related problem instances, it makes sense to (partially) pool across them
- Cases with few examples will be regularized, cases with many will be unaffected

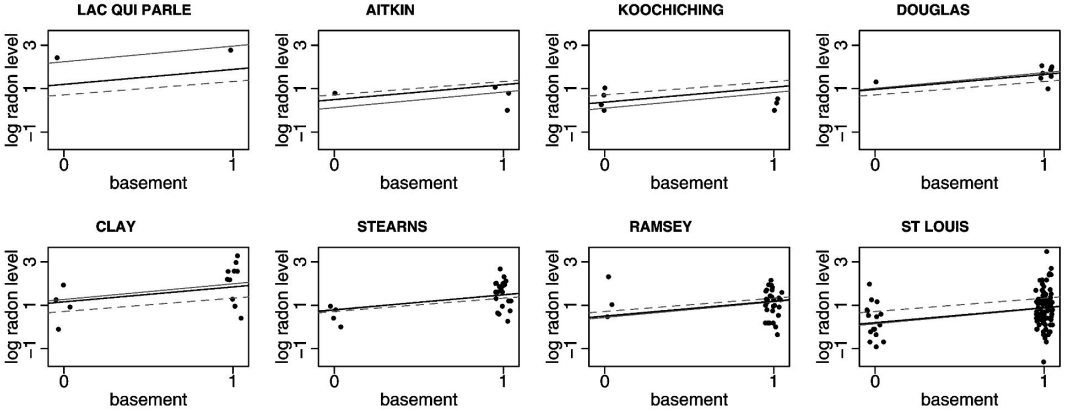


Figure 1. Multilevel (partial pooling) Regression Lines $y = a_j + \beta x$ Fit to Radon Data From Minnesota, Displayed for Eight Counties j With a Range of Sample Sizes. Light-colored dotted and solid lines show the complete-pooling and no-pooling estimates. The x-positions of the points are jittered slightly to improve visibility.

Figure from “Multilevel (hierarchical) modeling: what it can and cannot do”

Sharing Information: Random Effects Models

- When we are trying to estimate across related problem instances, it makes sense to (partially) pool across them
- Cases with few examples will be regularized, cases with many will be unaffected

$$K_{ij} \sim \mathrm{GP}(\mu_{ij}, \alpha_i)$$
$$\mu_{ij} = s_j\, q_{ij}$$
$$\log_2(q_{ij}) = \sum_k x_{jk}\beta_{ik}.$$

$$\beta_{ir} \sim N\left(0, \sigma_r^2\right)$$

We can do this with genes instead of counties!

(eq. 10 in DESeq2 paper)

Sharing Information: Random Effects Models

- When we are trying to estimate across related problem instances, it makes sense to (partially) pool across them
- Cases with few examples will be regularized, cases with many will be unaffected

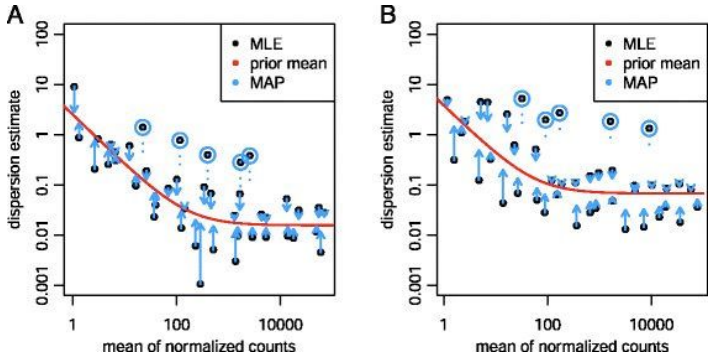
$$K_{ij} \sim \mathrm{GP}(\mu_{ij}, \alpha_i)$$

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

$$\log_2(q_{ij}) = \sum_k x_{jk} \beta_{ik}.$$

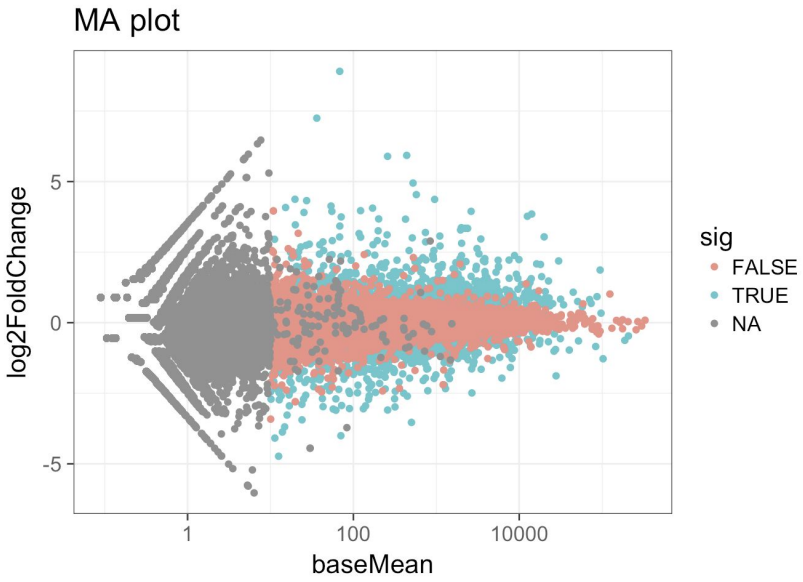
$$\beta_{ir} \sim N\left(0, \sigma_r^2\right)$$

Also for dispersions α

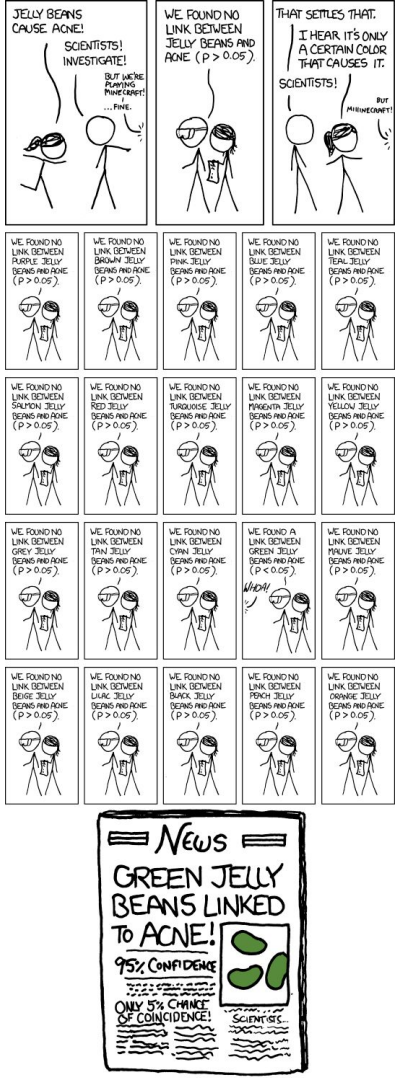


False Discovery Rate control

- Need to protect against the multiple testing problem
- Also, want practical (not just statistical) significance



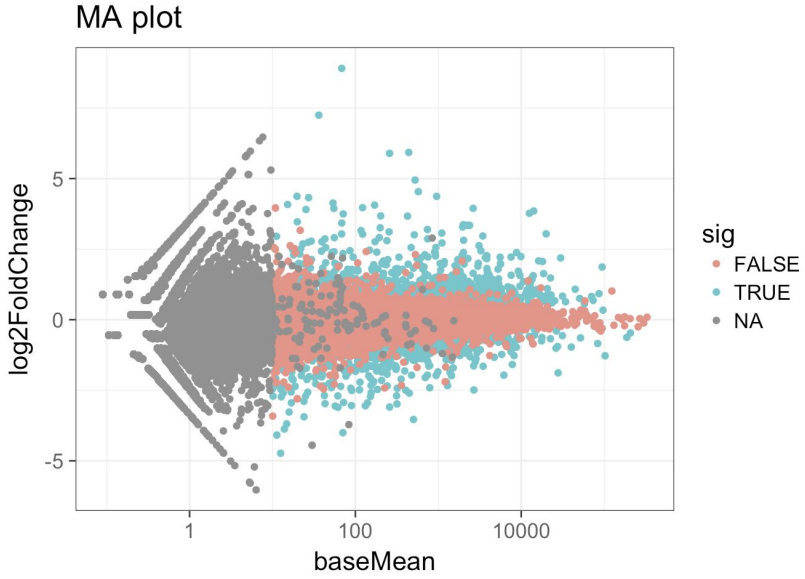
https://4va.github.io/biodatasci/r-rnaseq-airway.html



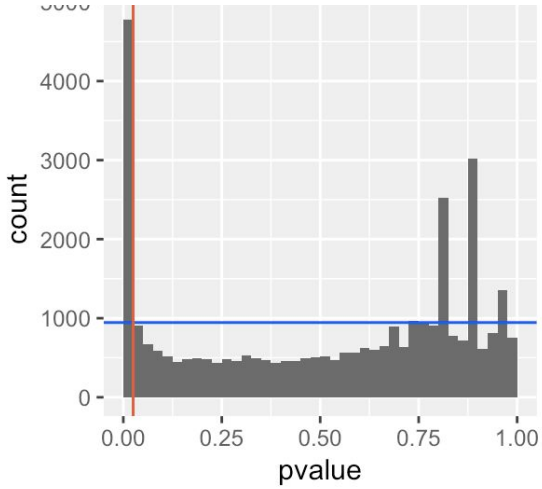
False Discovery Rate control

- Need to protect against the multiple testing problem
- Also, want practical (not just statistical) significance

Benjamini Hochberg: Reject as many hypotheses as possible, given constraint on area of the bottom left rectangle.



https://4va.github.com/biodatasci/r-rnaseq-airway.html



http://web.stanford.edu/class/bios221/book/Chap-Testing.html

False Discovery Rate control

- Need to protect against the multiple testing problem
- Also, want practical (not just statistical) significance

DESeq2 p-values are adjusted according to Benjamini-Hochberg.

$$\boxed{K_{ij} \sim \mathrm{GP}(\mu_{ij}, \alpha_i)}$$

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

$$\log_2(q_{ij}) = \sum_k x_{jk} \boxed{\beta_{ik}}$$

```
#' dds <- makeExampleDESeqDataSet()
#' dds <- estimateSizeFactors(dds)
#' dds <- estimateDispersions(dds)
#' dds <- nbinomWaldTest(dds)
#' res <- results(dds)
```

https://github.com/mikelove/DESeq2/blob/master/R/core.R

Conclusion

- We've deconstructed some of the essential ideas in DESeq2
- You also now have some powerful tools at your disposal
  - Removal of batch effects (negative controls, technical replicates, size factor estimation)
  - Modeling for (overdispersed) count data
  - Information sharing and False Discovery Rate Control
- New technologies will need new analysis methods, but *fundamental principles change slowly*