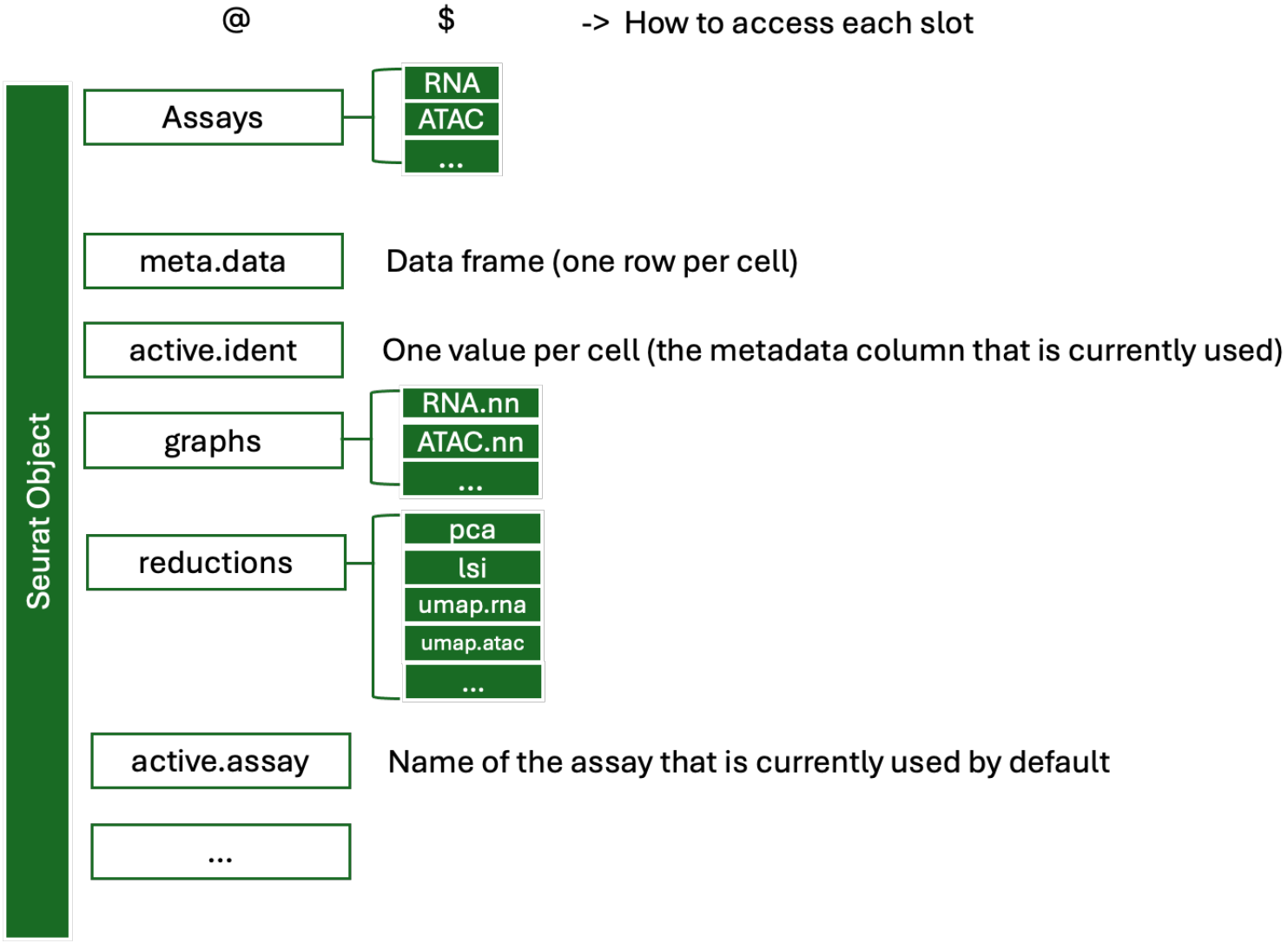
# From “raw” data to “clean”, normalised count matrices

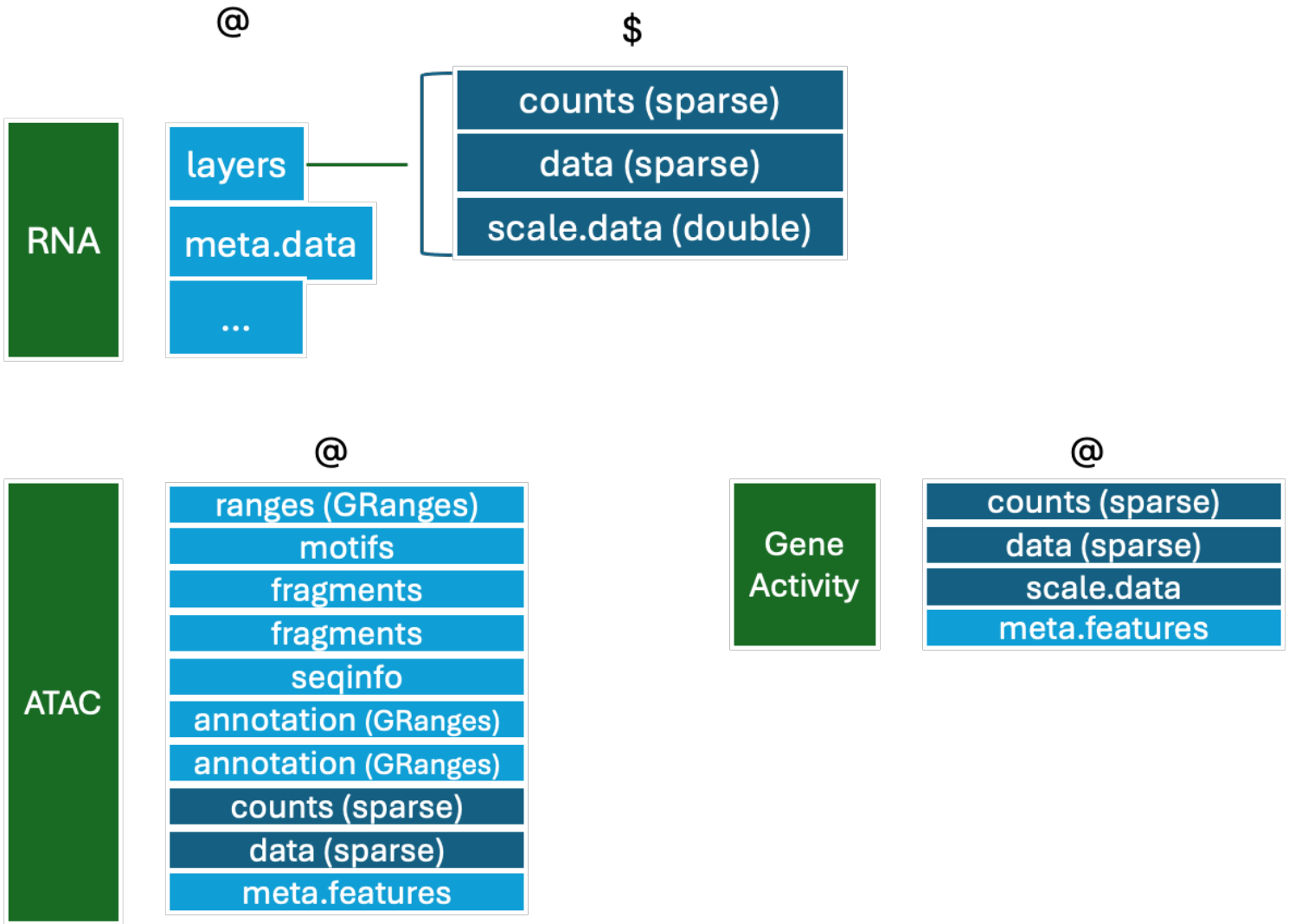
“Single-cell multiomic data analysis”
Summer Semester 2025
Antonio Scialdone

HELMHOLTZ
MUNICH

# Seurat object structure



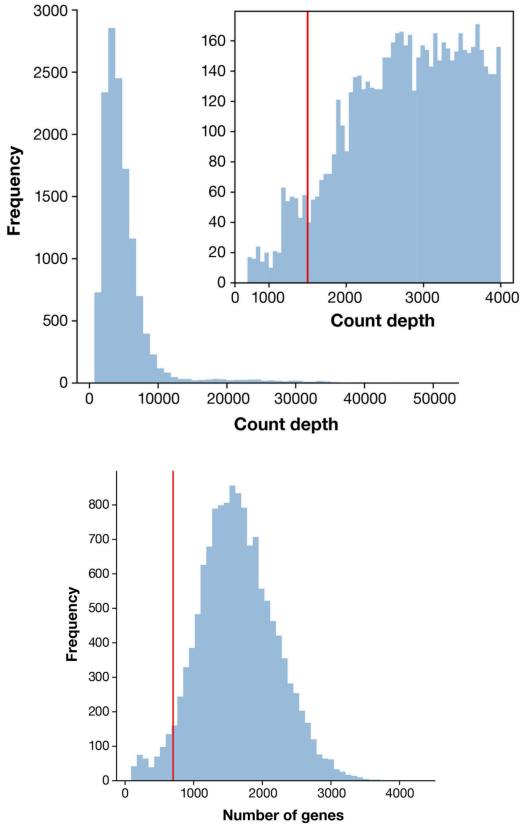
# Seurat object structure

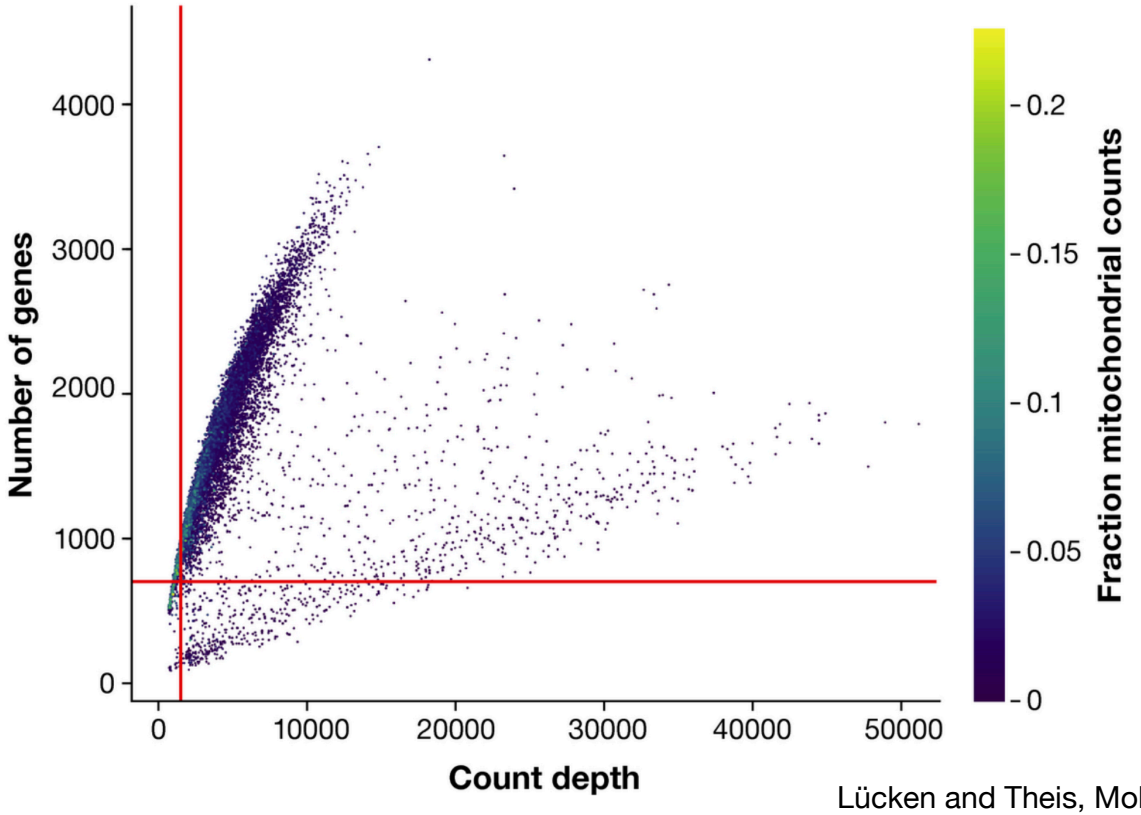


# Quality control - RNA-seq

The goal is removing “low quality cells”:
- Damaged cells
- Stressed/apoptotic cells
- Cells with too few counts to work with

Quality metrics:
- Number of UMIs (count depth)
- Number of genes
- Fraction of mitochondrial gene counts
- Fraction of ribosomal gene counts



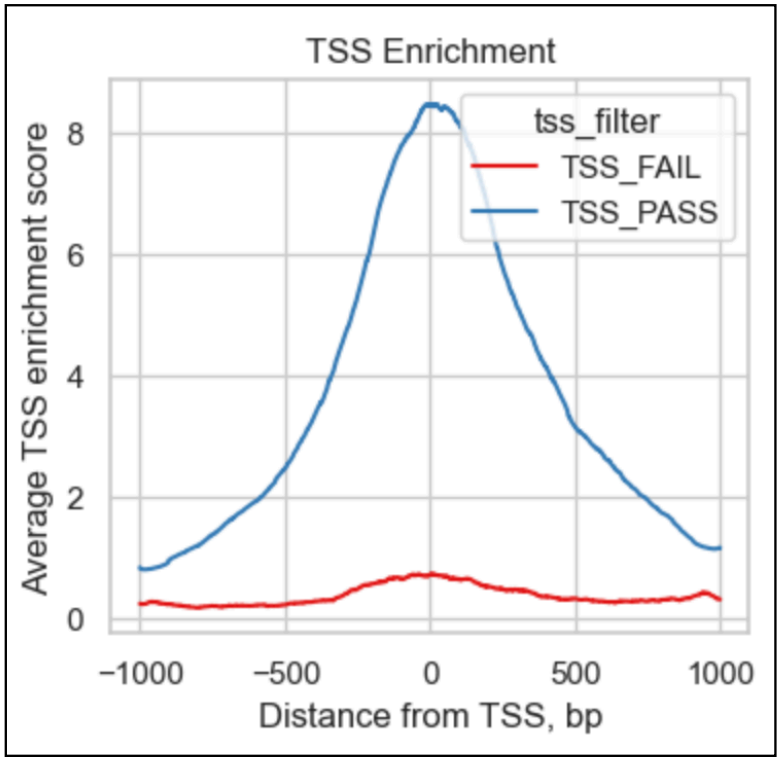

**Note:** one should pick outlier cells as “low quality cells” and use thresholds to filter…but be cautious!

Lücken and Theis, MolSySBio, 2019

# Quality control - ATAC-seq

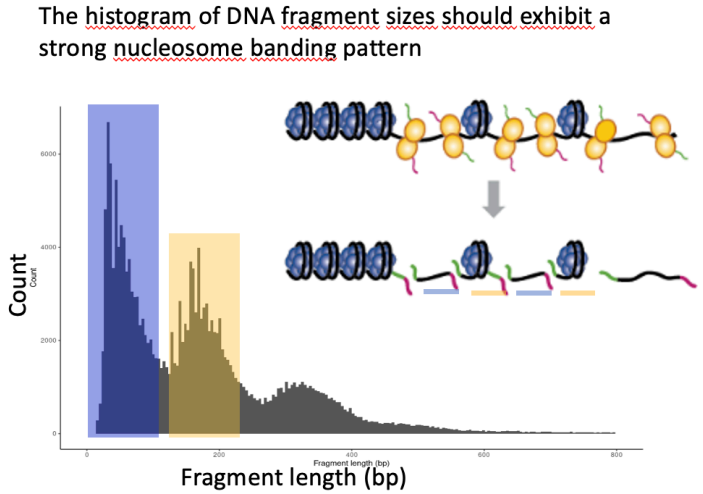
The goal is removing “low quality cells”:
- Damaged cells
- Stressed/apoptotic cells
- Cells with too few counts to work with

Quality metrics:
- Number of fragments in peaks (~sequencing depth)
- Fraction of fragments in peaks
- Fraction of reads overlapping ENCODE blacklisted regions
- TSS enrichment (~signal-to-noise measure)
- Nucleosome signal

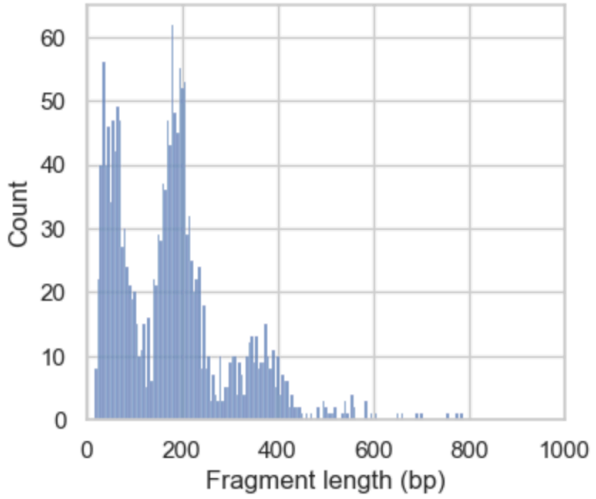



## Pass

The histogram of DNA fragment sizes should exhibit a strong nucleosome banding pattern




## Fail



# Quality control - multiplets removal

Multiplets (Doublets):

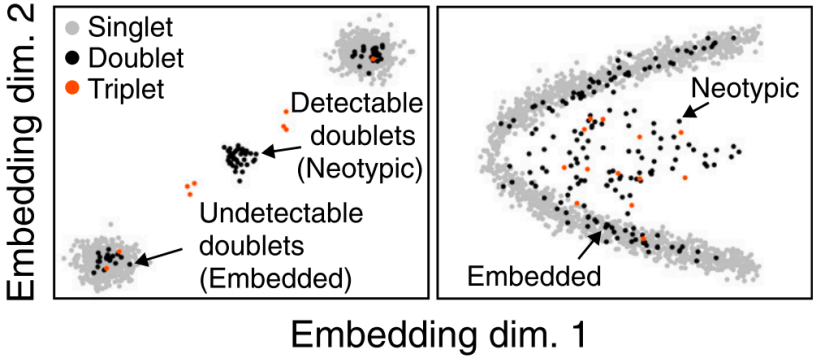
- Combinations of two cells with the same barcode (from same or different cells - **embedded** vs **neotypic**)
  - problems mainly from neotypic doublets -> fake clusters and “bridges” in trajectories
- Typically “aim” for ~5-10% doublets in droplet-based approaches

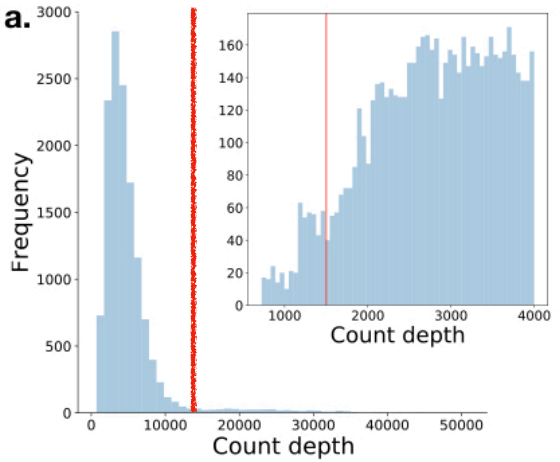
Simple solution 1: include maximum thresholds for counts & genes

- Issue: doublets don’t necessarily have more counts/genes

Simple solution 2: Look for cells with marker genes from different cell types

- Issue: “poly-hormonal cells”, unknown markers/ cell types

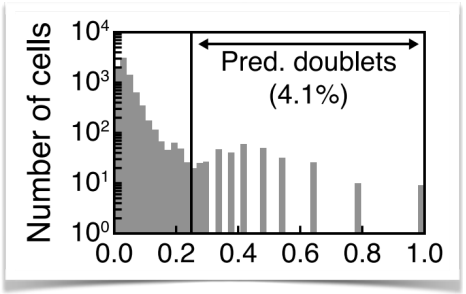



Wollock et al., Cell Systems (2019)

Lücken and Theis, Mol Sys Biol (2019)

# Quality control - multiplets removal



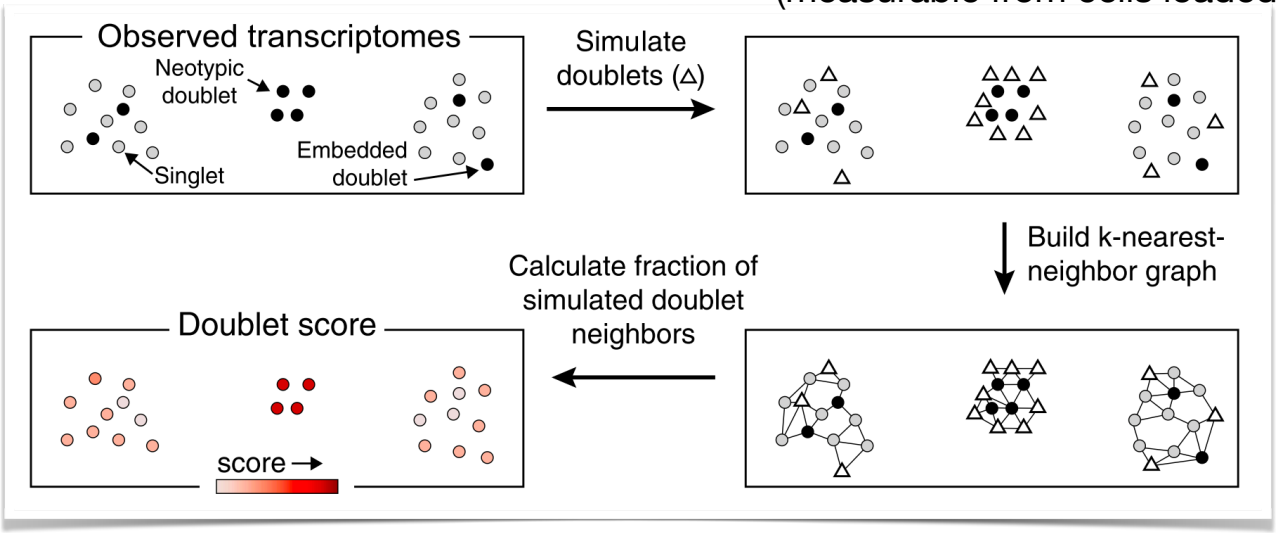

Solution 3: Automated doublet detection - Scrublet

Concept:

simulate doublets by random sampling and find similar real cells

Method:

1. Calculate distances between cells and simulated doublets
2. build a k-nearest neighbour graph connecting k most similar cells to each cell
3. doublet score for real cells based on fraction of simulated doublet neighbours
4. threshold doublet score based on fraction of expected doublets (measurable from cells loaded onto flowchip)




Assumptions:

- doublets are rare
- Singlets are available for every doublet
- Doublets are a random sampling event
  - imperfect tissue dissociation
  - clumping of “sticky cells”

Wollock et al., Cell Systems (2019)

# Quality control - ambient RNA

Library preparation (tissue dissociation, etc.) can cause cells to break, resulting in free floating mRNA in the cell suspension ("soup" or "ambient RNA")

- Tissue-specific problem due to differing dissociation protocols
- **Problems**: soup is sample specific, so different samples may have different soup
- **Assumption**: empty droplets have only soup
- **Approach**: model the background profile of empty droplets and remove this profile from each cell
- **Methods**: SoupX (Young and Behjati, GigaScience 2020), CellBender (Fleming et al, NatMethods 2023)

$n_{gc}$ - UMI counts per gene, g, in cell, c

$b_g$ - background expression per gene, g

$D$ - set of empty droplets

$m_{gc}$ - cell endogenous counts

$o_{gc}$ - counts from the background

$\rho_c$ - background contamination fraction

$$N_c = \sum_g n_{gc}$$

The SoupX approach:

$$b_g = \frac{\sum_{c \in D} n_{gc}}{\sum_{g \in G} \sum_{c \in D} n_{gc}}$$

$$n_{gc} = m_{gc} + o_{gc}$$

$$o_{gc} = N_c \rho_c b_g$$

can be estimated, e.g., from "negative" markers

Fleming et al, NatMethods (2023)

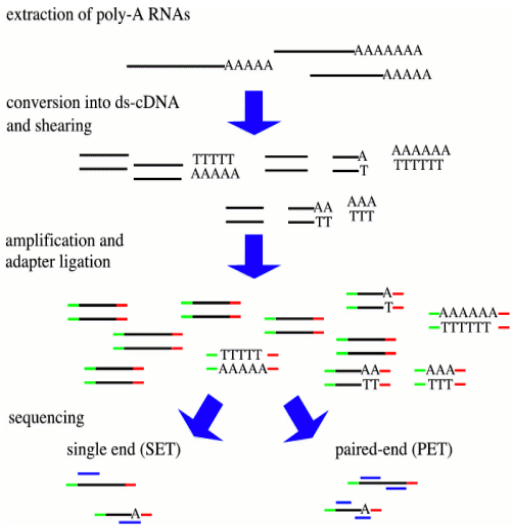
Young and Behjati, GigaScience (2020)

# Data normalisation

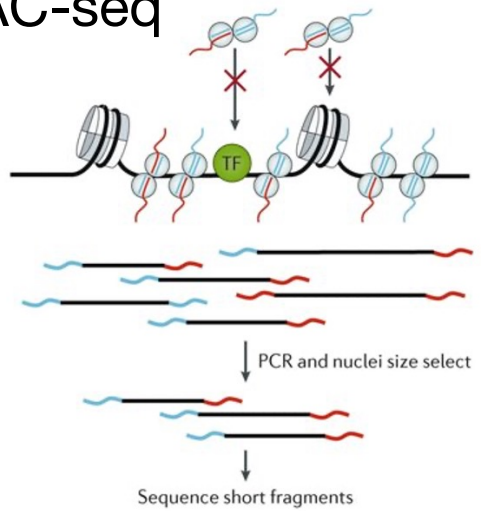
A **measurement** in RNA/ATAC-seq is the result of a **random sampling** (e.g., a UMI count is produced by an mRNA molecule that was captured, reverse transcribed, and sequenced)

Differences in UMI/peak counts could be due to “real” differences in mRNA levels/ chromatin accessibility…OR to **random sampling**

## RNA-seq




## ATAC-seq




The goal of **data normalisation** is to make data across cells comparable by mitigating the effects of random sampling

# Data normalisation - RNA-seq

## Counts per Million (CPM)

$$x_{gc} = n_{gc}\frac{K}{\sum_g n_{gc}}$$

size factor=$s_c$

- $x_{gc}$ - normalised expression values
- $n_{gc}$ - UMI counts per gene, g, in cell, c
- $K$ - constant: factors of 10 or median count depth across dataset

## CPM variant

$$x_{gc} = n_{gc}\frac{K}{\sum_{g\in G'} n_{gc}}$$

- $G'$ - set of genes with fewer than X% of total counts

Exclude genes that are highly expressed in size factor estimate

Possible issue:

- Highly expressed genes skew cell size estimation

Toy example

**Raw counts**

| Gene.Name | Rep1 | Rep2 | Rep3 |
|---|---|---|---|
| A | 10.00 | 12.00 | 30.00 |
| B | 20.00 | 25.00 | 60.00 |
| C | 5.00 | 8.00 | 15.00 |
| D | 0.00 | 0.00 | 1.00 |
| Total | 35.00 | 45.00 | 106.00 |

**CPM**

| Gene.Name | Rep1 | Rep2 | Rep3 |
|---|---|---|---|
| A | 0.29 | 0.27 | 0.28 |
| B | 0.57 | 0.55 | 0.57 |
| C | 0.14 | 0.18 | 0.14 |
| D | 0.00 | 0.00 | 0.01 |
| Total | 1.00 | 1.00 | 1.00 |

$\cdot 10^6$

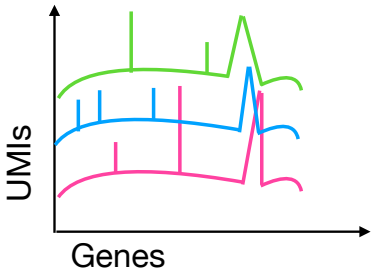
$\cdot 10^6$

# Normalization by global scaling: Scran pooling

Concept: Match expression profiles of cells to assess how much sequencing "attention" placed on a cells

Challenge: 0 values ("dropouts")

Solution: combine cells randomly to match profiles in "pseudo-bulk" fashion

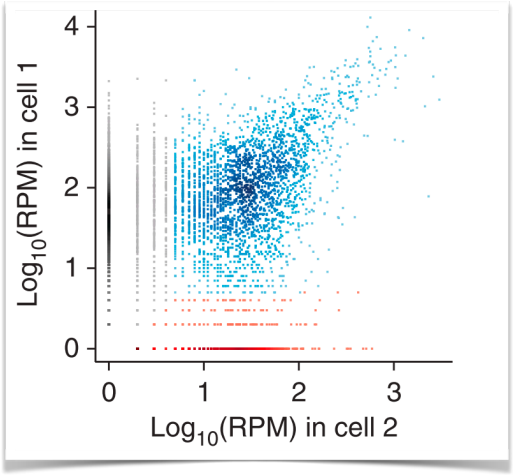



$$E(x_{gc}) = \theta_c \lambda_{g0} t_c^{-1}$$

biological cell size

Cell-specific bias
(size factor)

expected transcript count




Kharchenko, Silberstein, and Scadden, Nat. Meth. (2014)

Lun, Bach, and Marioni, Genome Biol. (2016)

# Normalization by global scaling: Scran pooling

Concept: Match expression profiles of cells to assess how much sequencing "attention" placed on a cells

Challenge: 0 values ("dropouts")

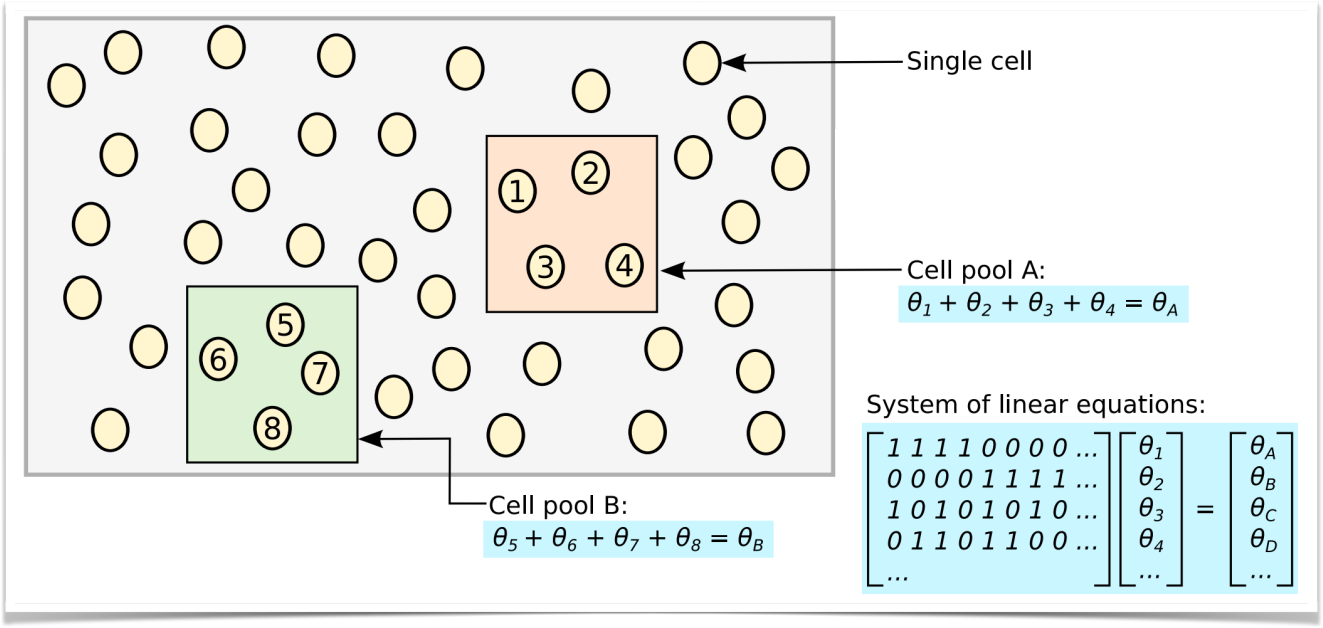
Solution: combine cells randomly to match profiles in "pseudo-bulk" fashion

$$E(x_{gc}) = \theta_c \lambda_{g0} t_c^{-1}$$

biological cell size

Cell-specific bias (size factor)

expected transcript count




Assumption:

- Less than 50% differentially expressed genes between cells
- Always need a "reference pseudo-cell"
  - not comparable between runs!

Lun, Bach, and Marioni, Genome Biol. (2016)

# Normalization by model fitting

If we can model scRNA-seq data, we understand what random variation looks like in this data

- can add technical factors to regression models
- departures from random & technical variation are biological signals
- residuals of model fits can be used as normalized expression values

$$n_{gc} \sim D_g(\mu, \phi)$$

model counts as coming from distribution $D_g$

$$\hat{n}_{gc} = f_D(N_c, \dots)$$

Build a regression model with technical covariate $N_c$ to account for differences in sequencing depth (with $\epsilon \sim D_g$)

$$x_{gc} = r_{gc} = n_{gc} - \hat{n}_{gc}$$

use residuals $r_{gc}$ as normalised expression values

$n_{gc}$ - UMI counts per gene, g, in cell, c

$\mu$ - parameter 1: e.g., mean

$\phi$ - parameter 2: e.g., dispersion

$D_g$ - Distribution per gene, g

$N_c$ - Count depth

$r_{gc}$ - residuals of model regression

$x_{gc}$ - normalised gene expression

Residuals have mean 0, are not strictly positive, but should be normally distributed (given good model fit)

# Normalization by model fitting

How do we model scRNA-seq data?

## Count distributions: Poisson vs Negative Binomial (NB)

$$P(X = k \mid \lambda) = \frac{\lambda^k e^{-k}}{k!}$$

$$\text{Var} = \mu = \lambda$$

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

continuous version:

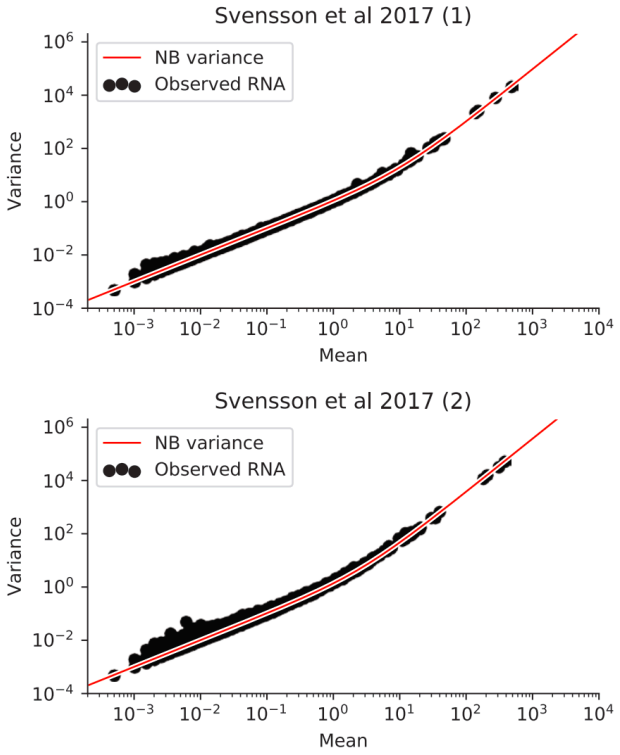
$$P(X = k \mid \mu, r) = \frac{\Gamma(r + k)}{k!\Gamma(r)}\left(\frac{\mu}{r + \mu}\right)^k\left(\frac{r}{r + \mu}\right)^r$$

$$\text{Var} = \mu + \frac{\mu^2}{r} \qquad \phi = \frac{1}{r}$$

(over)dispersion parameter

- NB is an over-dispersed Poisson distribution, droplet-based scRNA-seq data shown to follow this distribution

Disclaimer: droplet-based = UMI methods




distributions of free RNA added to droplets, no cells

More advanced versions:

- scTransform (Hafemeister and Satija, Genome Biol. 2019)
- GLM-PCA (Townes et al., Genome Biol. 2019)

Svensson, Nature Biotech. (2020)

# Data normalisation - ATAC-seq

Term Frequency - Inverse Document Frequency (TF-IDF) normalisation

- Technique from natural language processing to evaluate the importance of words in a sentence
- Term Frequency **(TF)**: frequency of a word in a document
- Inverse Document Frequency **(IDF)**: inverse document frequency of the word across a set of documents
- Term ~ Peak; Document ~ Cells

Cells

Peaks $\begin{pmatrix} c_{i,j} \end{pmatrix}$

$c_{i,j}$ = count for peak $i$ in cell $j$

$N$ = Number of cells

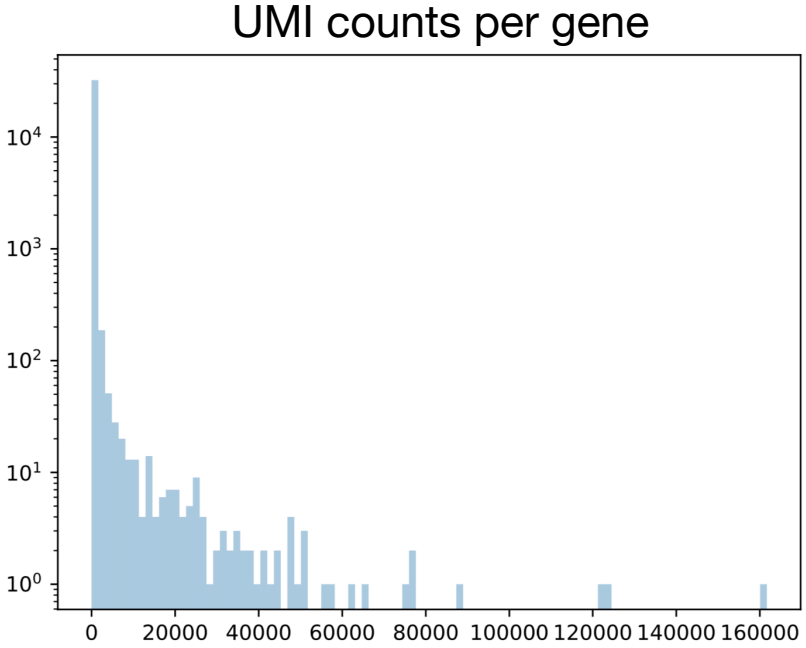
$$TF_{i,j} = \frac{c_{i,j}}{\sum_k c_{k,j}}$$

$$IDF_i = \frac{N}{\sum_k c_{i,k}}$$

$n_{i,j}$ = **normalised** count for peak $i$ in cell $j$ = $\log\left[1 + (TF_{i,j} \times IDF_i) \times 10^4\right]$

# Variance stabilization - log transformation or scaling

- Even after normalisation some genes are very highly expressed
- These genes would dominate downstream calculations as variance scales with the mean in scRNA-seq data (simplest case Poisson: variance = mean)
- idea: transform expression values to give less weight to high-variance genes




~3000 Peripheral blood mononuclear cells, 10X Genomics

Solution 1: log transformation $log(x_{gc} + 1)$

- pseudocount for $x_{gc} = 0$
- benefit: differences are now log-fold changes

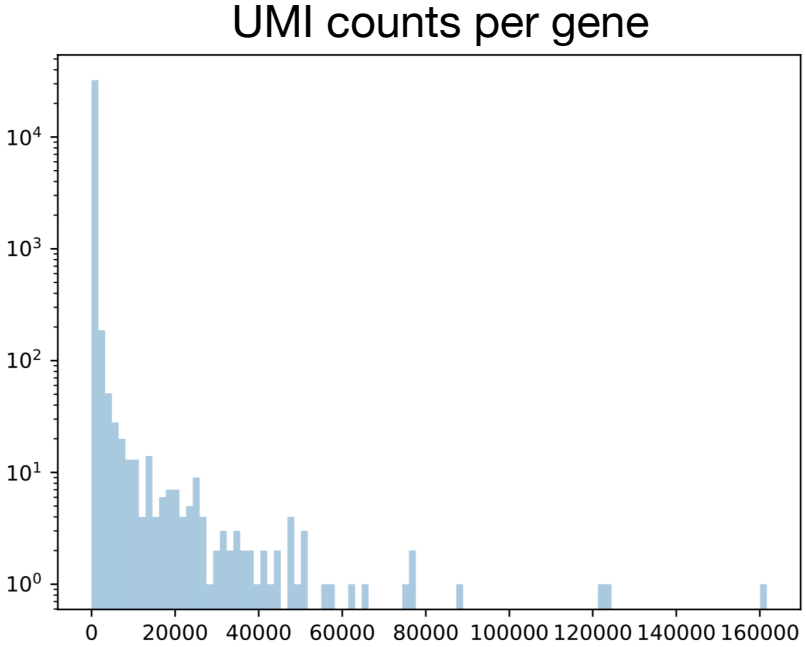
  $log(A) - log(B) = log(\frac{A}{B})$
- BUT:
  - pseudocount different effect on low and highly expressed genes

# Variance stabilization - log transformation or scaling

- Even after normalisation some genes are very highly expressed
- These genes would dominate downstream calculations as variance scales with the mean in scRNA-seq data (simplest case Poisson: variance = mean)
- idea: transform expression values to give less weight to high-variance genes

Solution 2: scaling (z-scores) $\frac{x_{gc} - \mu_g}{\sigma_g}$

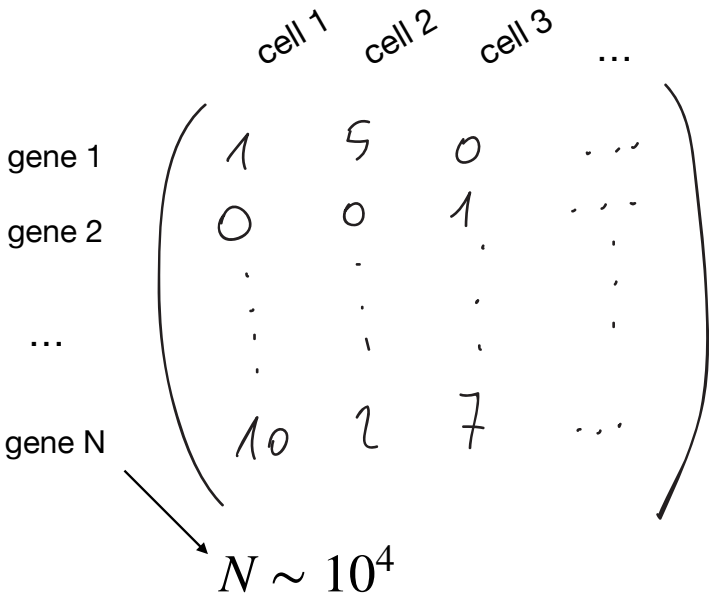
- benefit: all genes have the same net effect; mean=0, variance = 1
- BUT:
  - expression scale has biological meaning?

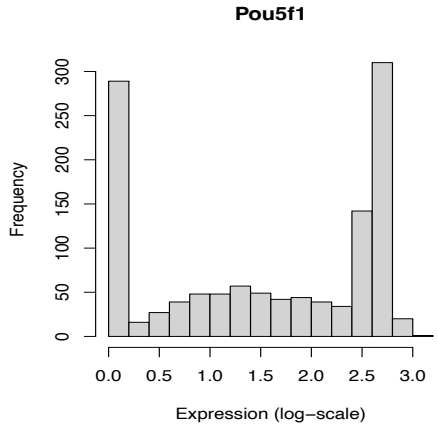


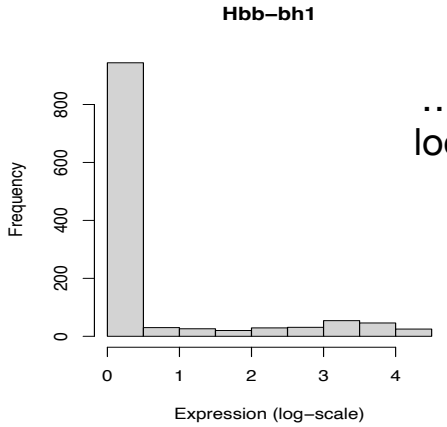

~3000 Peripheral blood mononuclear cells, 10X Genomics

Log transformation is standardly used, scaling occasionally (as well)

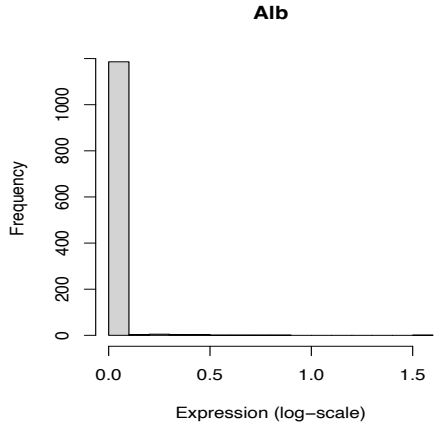
# Feature selection - selecting “interesting genes"

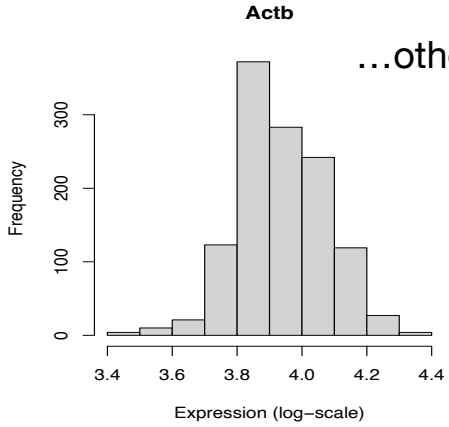



…some genes look interesting




…others less so

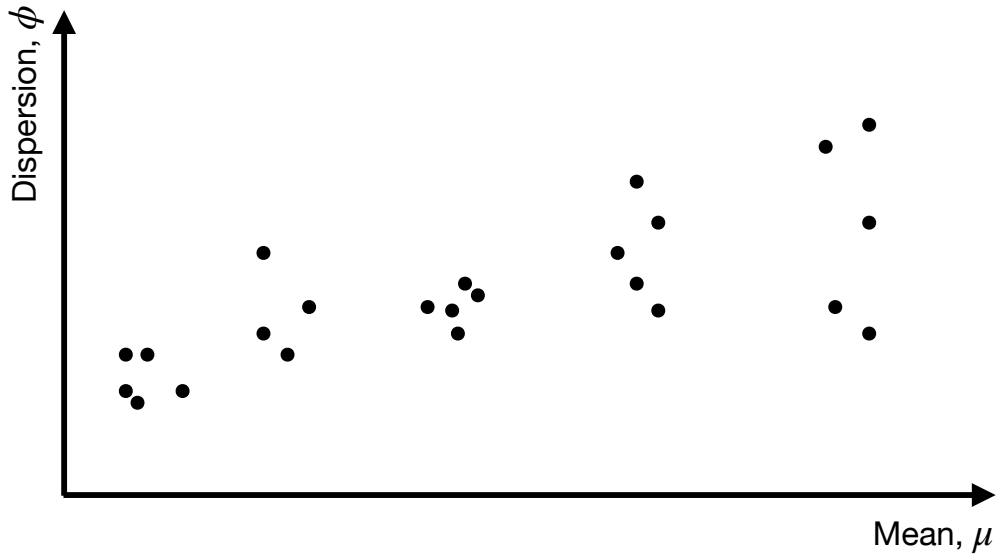
# Feature Selection - Highly variable genes

Why select features?

- Signal-to-noise ratio (Not all genes contain the same information)
- “Curse of dimensionality”
- Computational & storage efficiency

Approach: select genes with higher variance than expected from mean-variance relationship model (“Highly variable genes”)

- Relationship of gene dispersion ($\frac{\text{Var}}{\mu}$) to mean



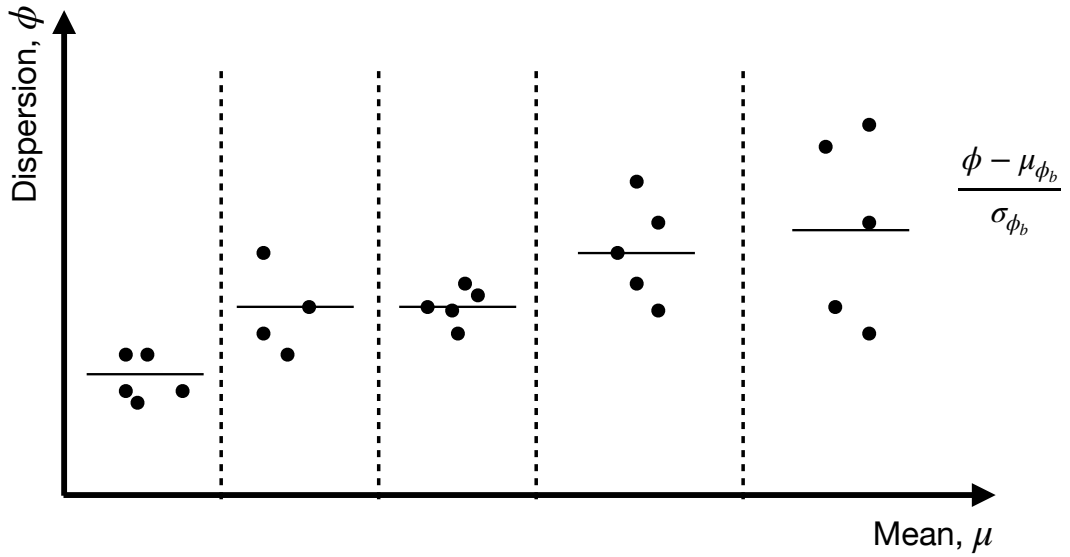
# Feature Selection - Highly variable genes

Why select features?

- Signal-to-noise ratio (Not all genes contain the same information)
- “Curse of dimensionality”
- Computational & storage efficiency

Approach: select genes with higher variance than expected from mean-variance relationship model (“Highly variable genes”)

- Relationship of gene dispersion ($\frac{\text{Var}}{\mu}$) to mean
- bin dispersions by means
- normalize dispersions (z-scores)



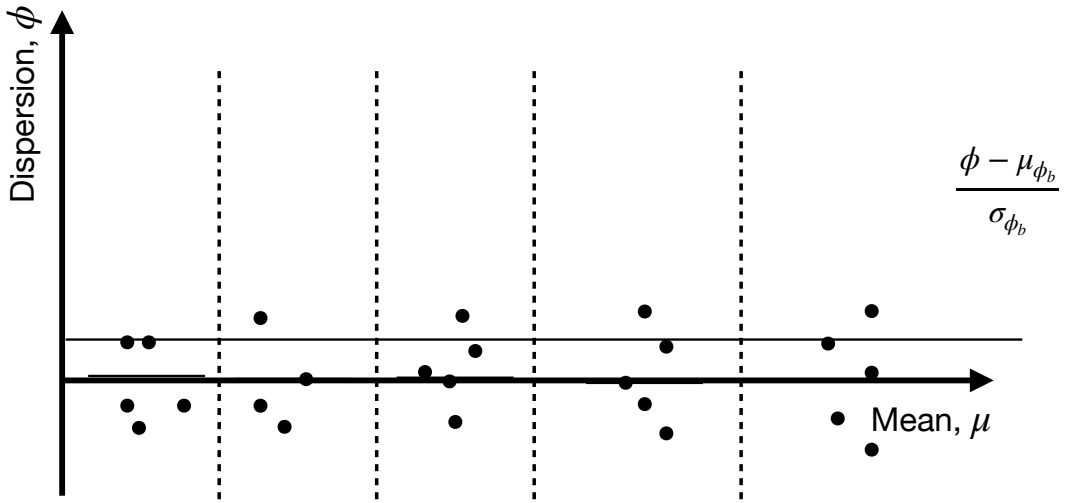
# Feature Selection - Highly variable genes

Why select features?

- Signal-to-noise ratio (Not all genes contain the same information)
- “Curse of dimensionality”
- Computational & storage efficiency

Approach: select genes with higher variance than expected from mean-variance relationship model (“Highly variable genes”)

- Relationship of gene dispersion ($\frac{\text{Var}}{\mu}$) to mean
- bin dispersions by means
- normalize dispersions (z-scores)
- select top genes based on normalized dispersions



# Feature Selection - ATAC-esq

- ATAC-seq is largely binary, which makes it hard to select “variable” features as we do in RNA-seq

- One approach is to remove those features that are present in fewer than *n* cells

- On top of this, one can still compute a **“variability score”** to rank features based on their variability in the  populations of cells. For example, in *episcanpy*, the score is such that maximum variable features (score =1) are those that are not zero in 50% of the cells; and the least variable (score =0) are those that are always 0 or non-zero.