Statistical Methods for single cell data analysis (focusing on technology)

Yongjin Park, UBC Path + Stat, BC Cancer

11 March 2024

Next Lectures – single cell genomics

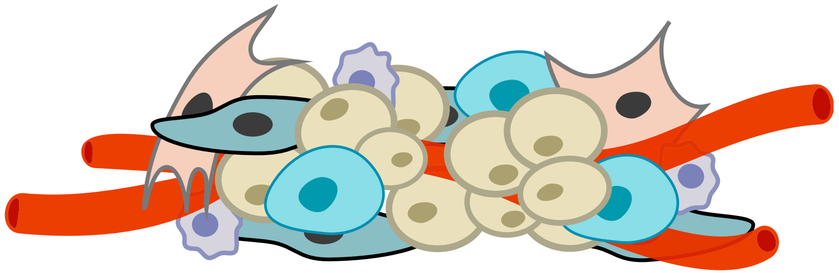
- Lecture 16: technology and data normalization
- Lecture 17: unsupervised learning -1
- Lecture 18: unsupervised learning -2

Today’s lecture

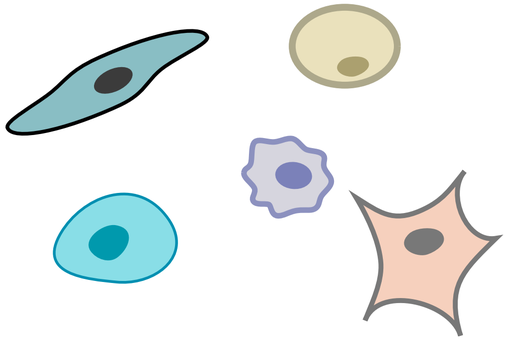
1. Single-cell sequencing technology
2. Basic quality control
3. Additional Q/C tools
4. Doublet detection in single-cell data
5. Data normalization across many batches

Droplet-based single-cell sequencing technology

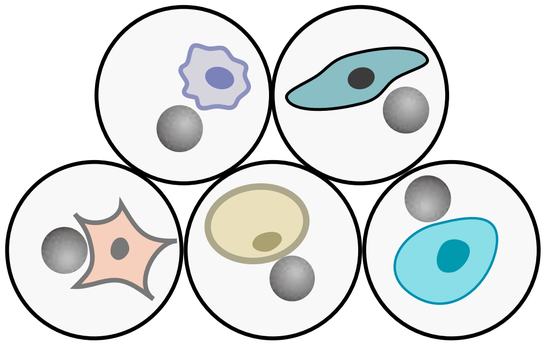
tissue sample



a mixture of cells



one drop = one cell

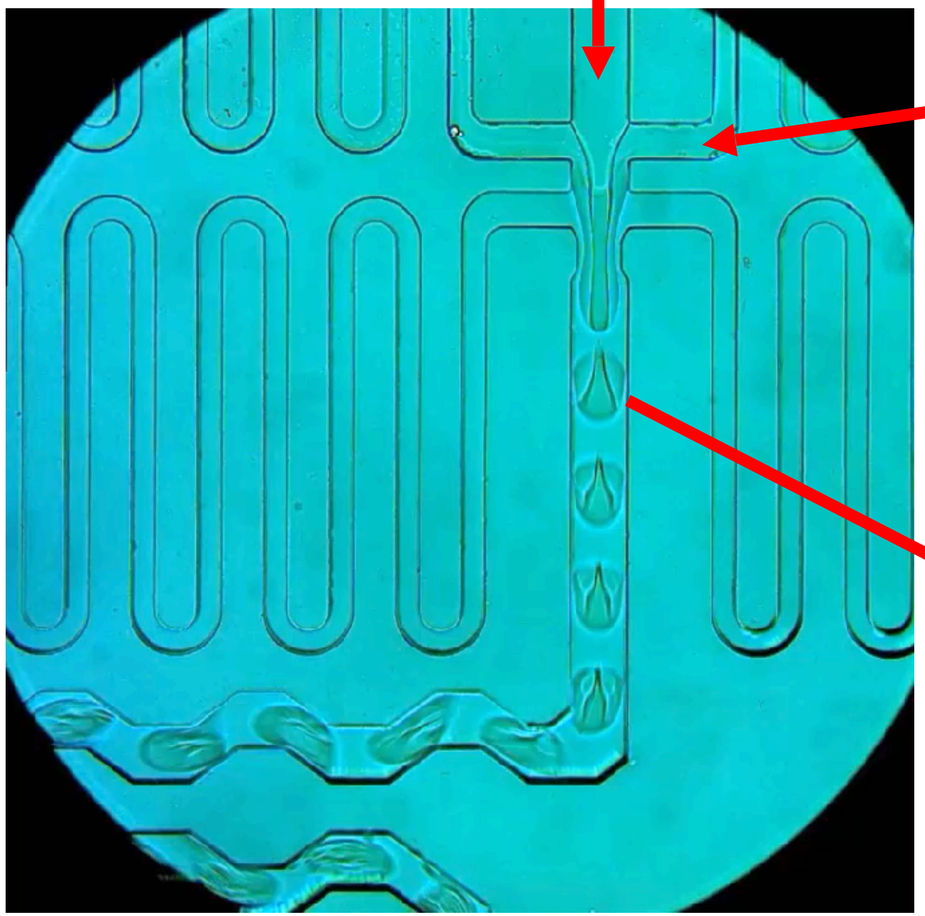


Macosko *et al.*, *Cell* (2015)

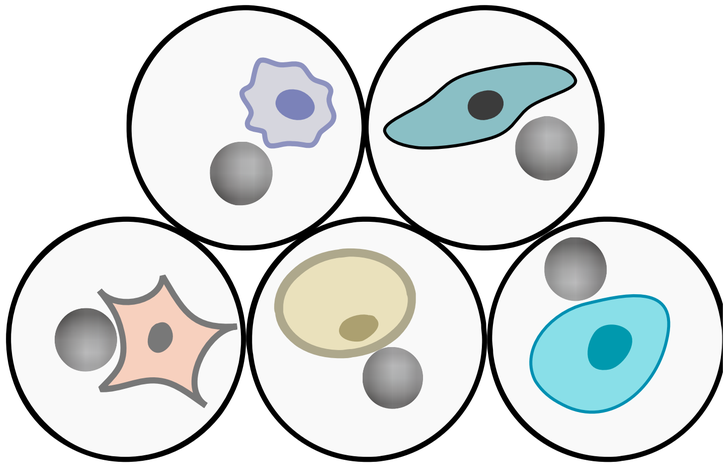
Drop-seq Idea #1: How do we capture each cell in each drop?

Microparticle

Oil flow



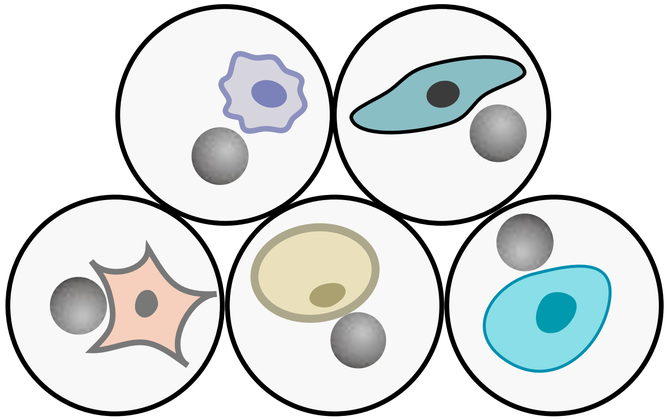
Cells



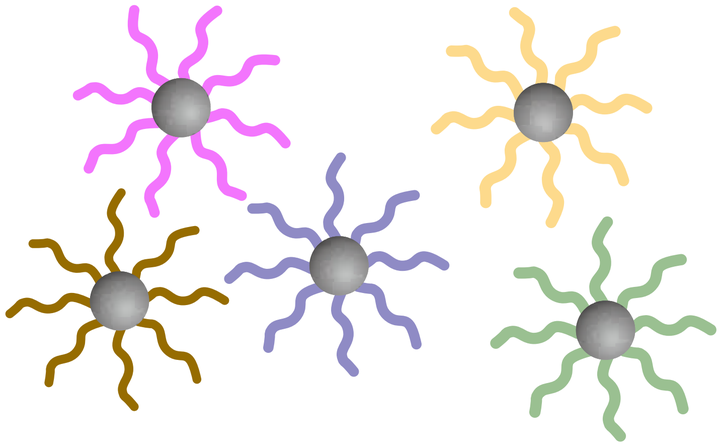
Makosco *et al.* Cell (2015)

Drop-seq idea 2: Massively-parallel sequencing followed by cell-specific barcoding

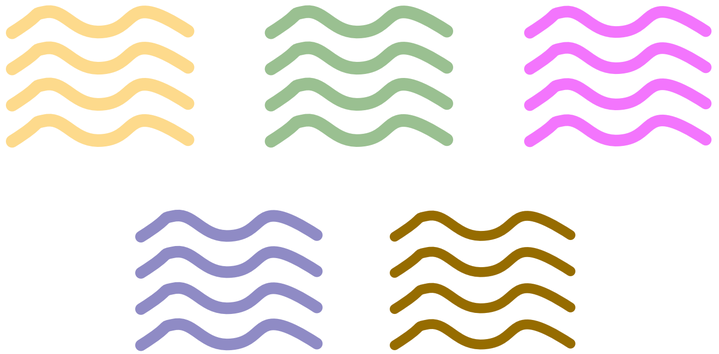
one drop = one cell



all the genes attached to the microbeads

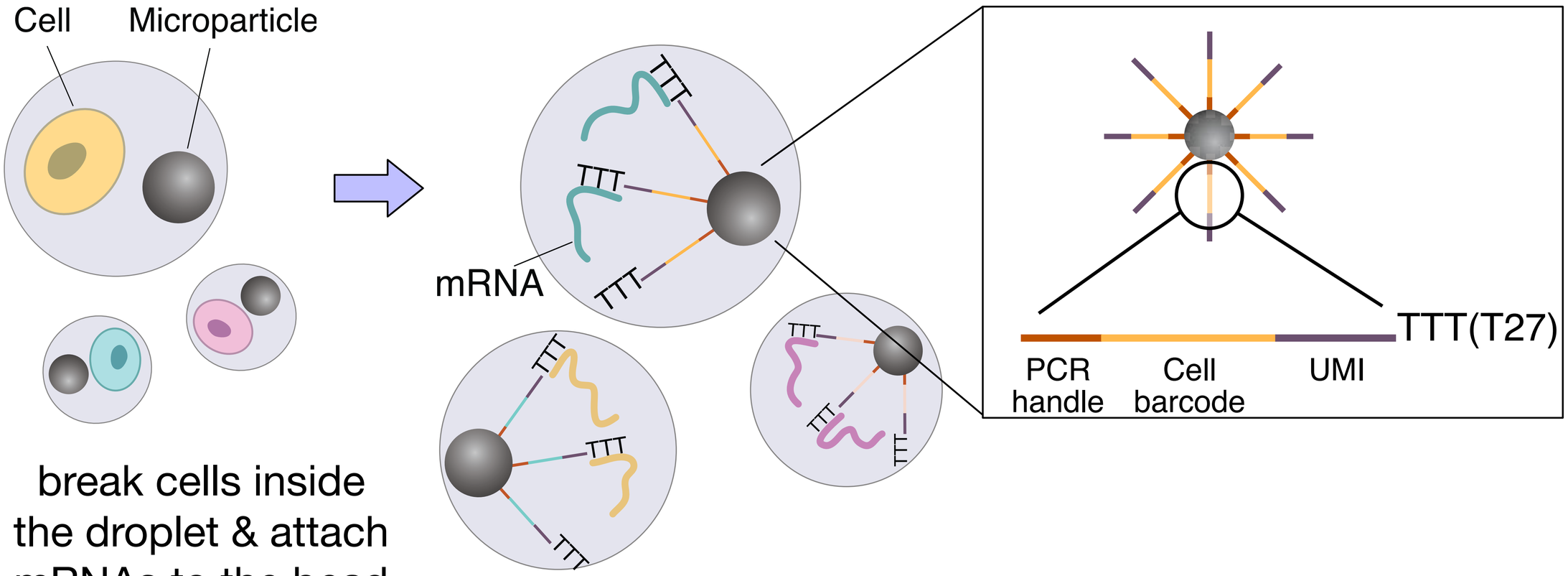


Massively parallel sequencing by mixing them all



Macosko *et al.*, *Cell* (2015)

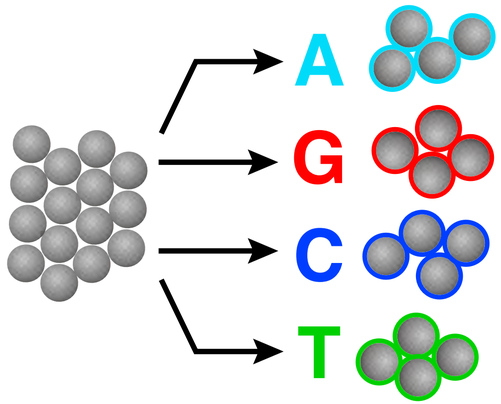
Drop-seq idea 3: How do we keep track of mRNA short reads' membership to a certain droplet?



break cells inside the droplet & attach mRNAs to the bead

Macosko *et al.*, *Cell* (2015)

How do we construct millions of unique barcodes? Use DNA as a hashing function!

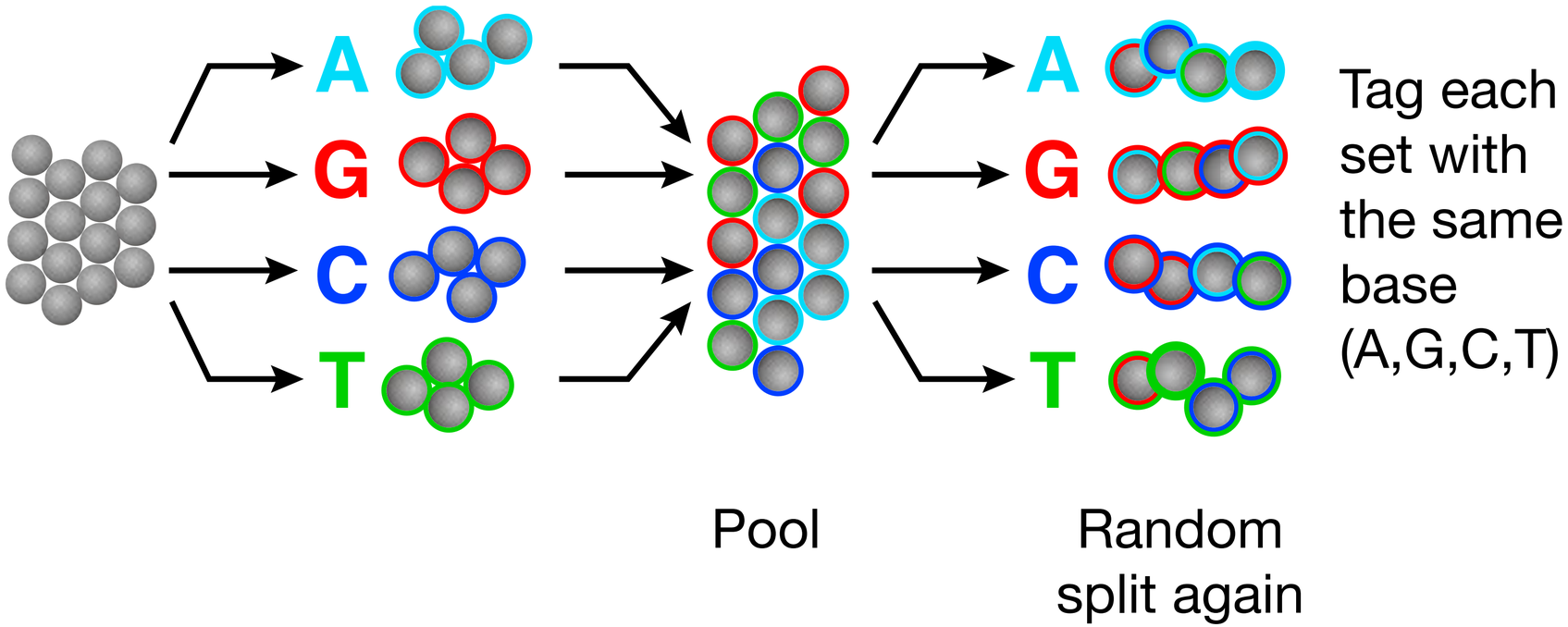


Randomly split the beads into 4 sets

Tag each set with the same base (A,G,C,T)

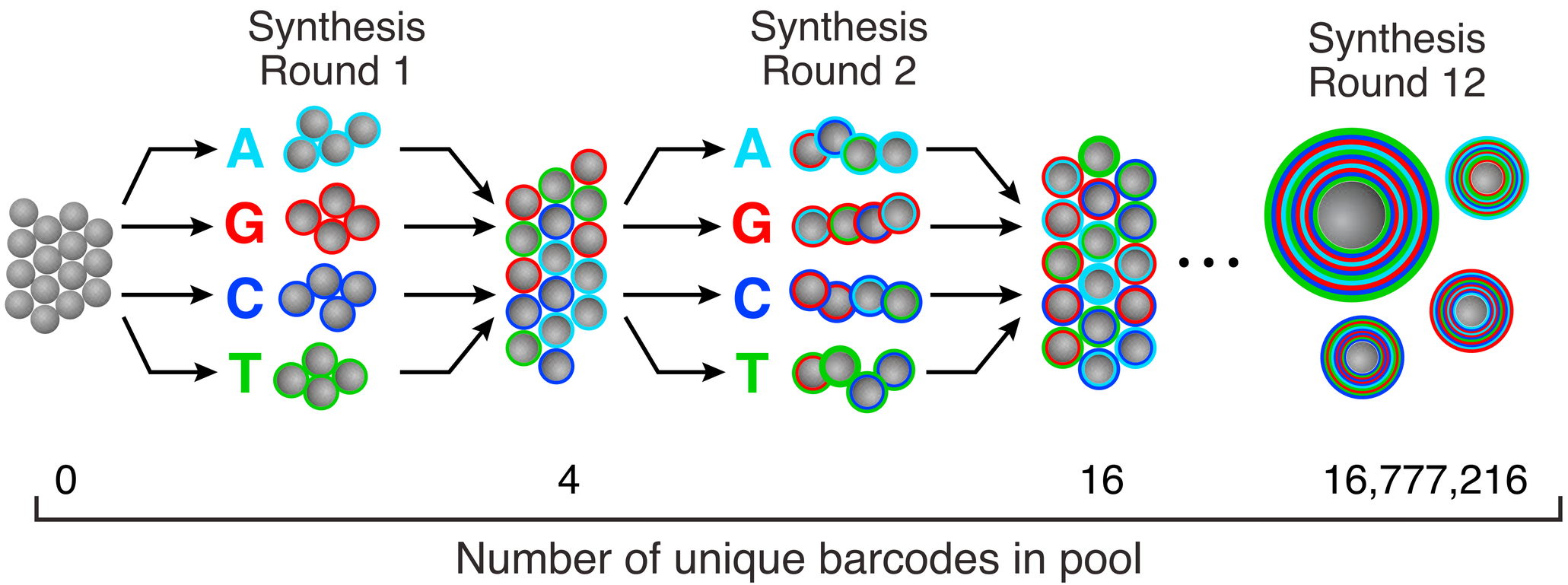
Macosko *et al.*, *Cell* (2015)

How do we construct millions of unique barcodes? Use DNA as a hashing function!



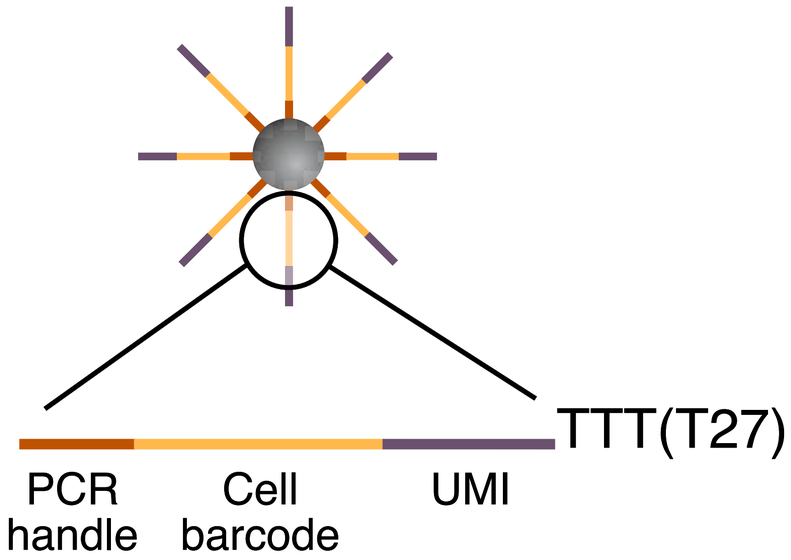
Macosko *et al.*, *Cell* (2015)

How do we construct millions of unique barcodes? Use DNA as a hashing function!

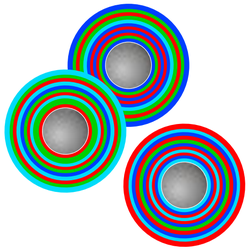


Macosko *et al.*, *Cell* (2015)

The lengths of barcode sequences determine data dimensionality



4^{12} 4^{8}



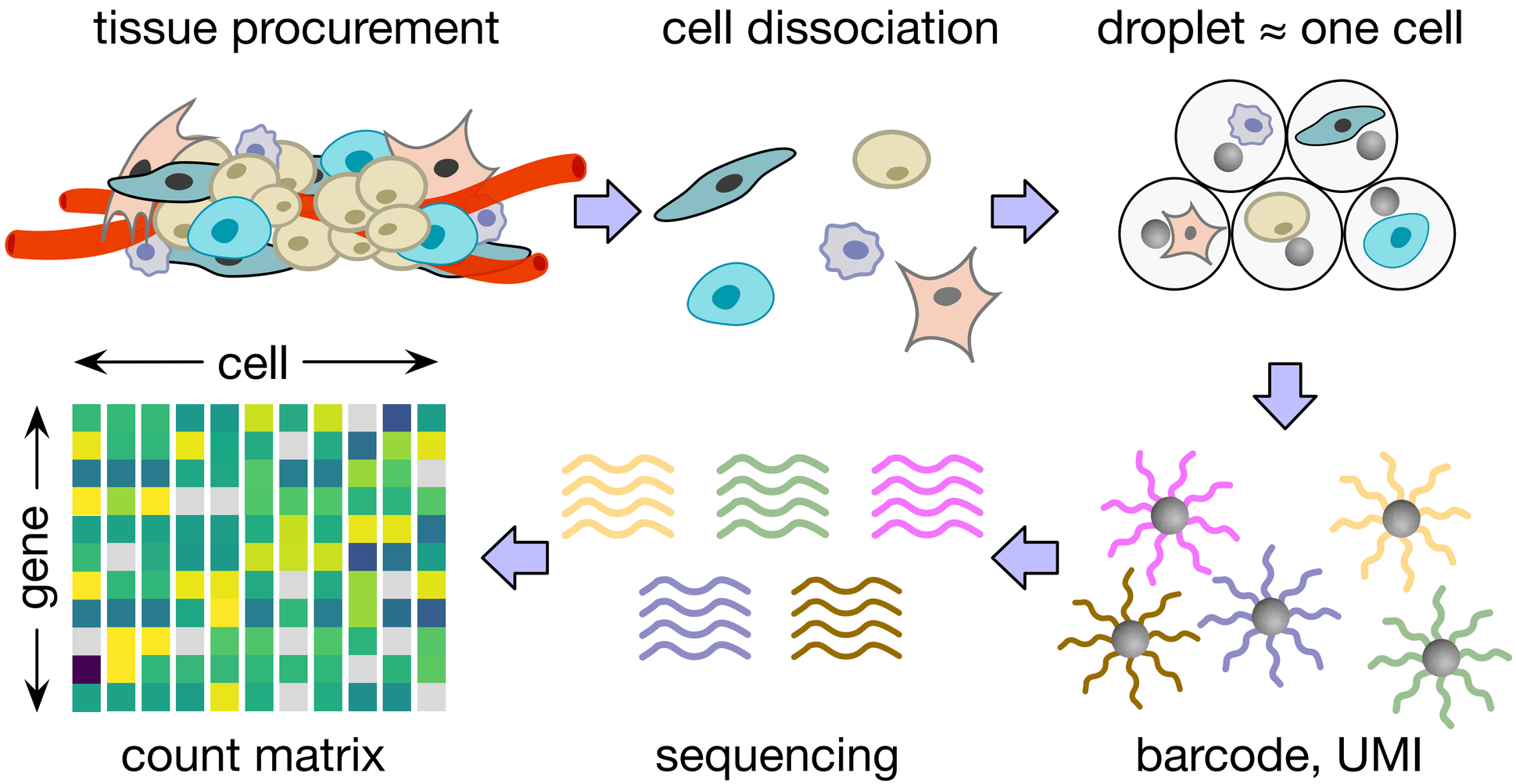
+ C T A G × 8 rounds of synthesis

- Millions of the same **cell barcode** per bead
- 4^8 different **molecular barcodes** (UMIs) per bead

Technically, we can build up to a 65,536 × 16,777,216, gene × cell expression matrix in one single-cell RNA-seq experiment.

Macosko *et al.*, *Cell* (2015)

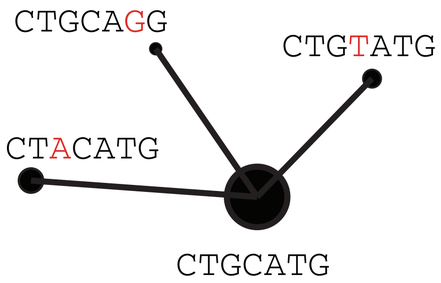
Droplet-based single-cell sequencing pipeline



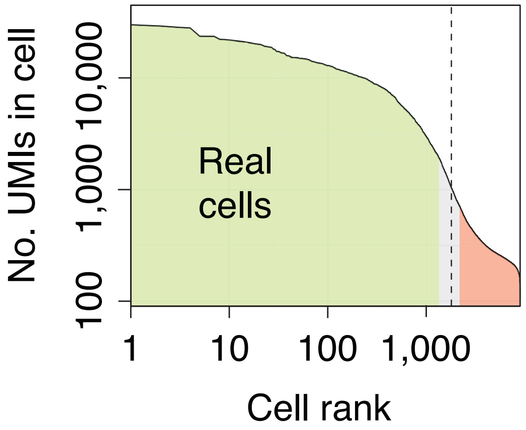
Macosko *et al.*, *Cell* (2015)

Overview of single-cell data analysis

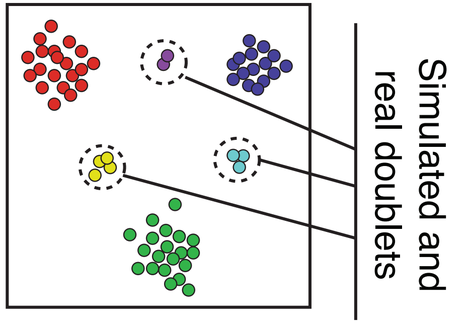
Alignment and molecular counting



Cell filtering and quality control



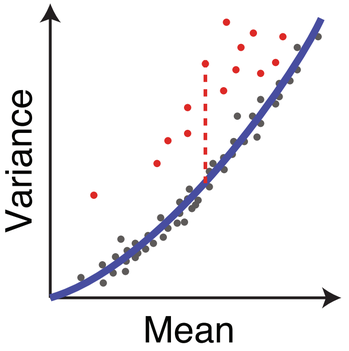
Doublet scoring



Cell size estimation

Cells	c_1	c_2	c_3
$Gene_1$	2	4	20
$Gene_2$	1	2	10
$Gene_3$	3	6	30
Cell depth:	6	12	60

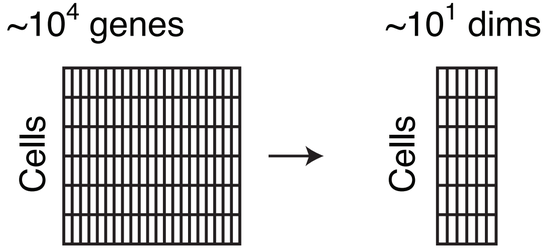
Gene variance analysis



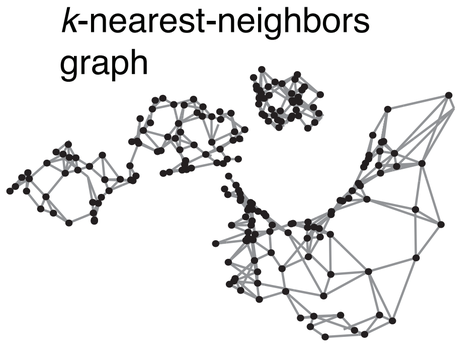
Karchenko, *Nature Methods* (2021)

Overview of single-cell data analysis cont'd

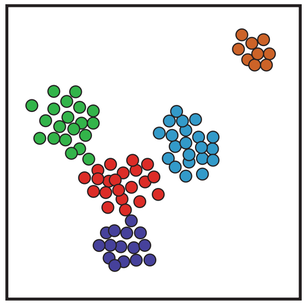
Reduction to a medium-dimensional space



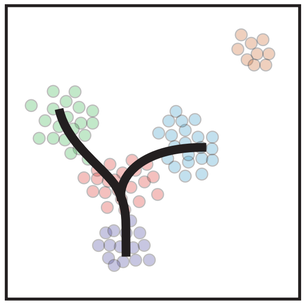
Manifold representation



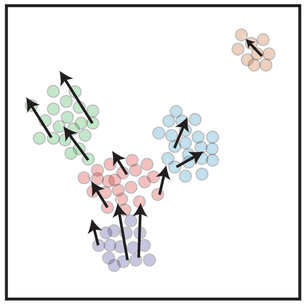
Clustering and differential expression



Trajectories



Velocity estimation

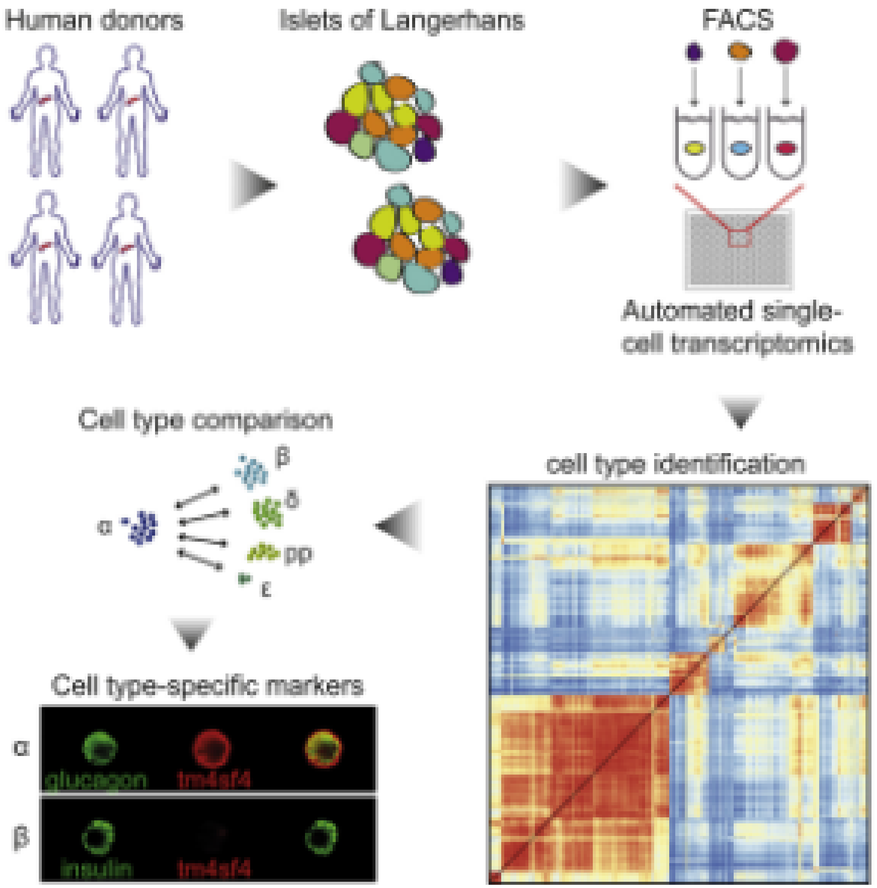


Karchenko, *Nature Methods* (2021)

Today’s lecture

1. Single-cell sequencing technology
2. Basic quality control
3. Additional Q/C tools
4. Doublet detection in single-cell data
5. Data normalization across many batches

Example: human pancreatic cells (Muraro *et al.* 2016)



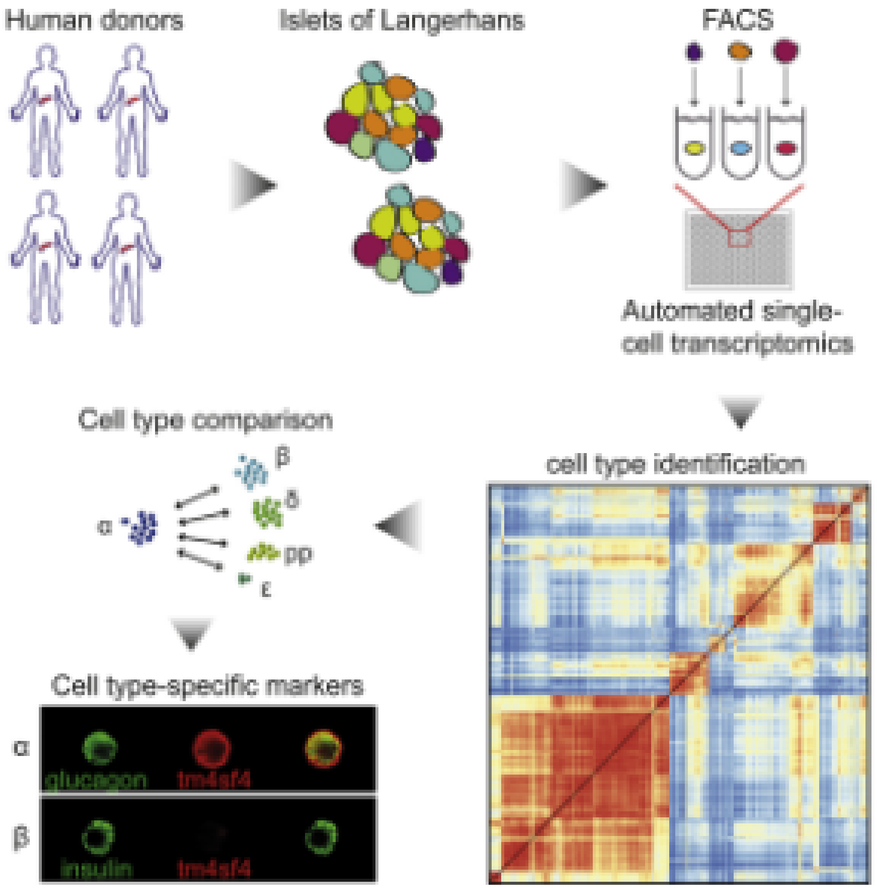
It's a sparse matrix...

```
X[1:10, 1:5]
```

```
## 10 x 5 sparse Matrix of class "dgCMatrix"
##          D28-1_1 D28-1_2 D28-1_3 D28-1_4 D28-1_5
## A1BG-AS1 .       .       .       .              .
## A1BG     .       .       1.00196 1.00196        .
## A1CF     6.07143 .       2.00785 6.07143        .
## A2M-AS1  .       .       .       .              .
## A2ML1    .       .       .       .              .
## A2M      .       5.04947 .       .              .
## A4GALT   .       1.00196 .       .              .
## A4GNT    .       .       .       .              .
## AAAS     1.00196 .       .       1.00196        .
## AACSP1   .       .       .       .              .
```

- What are the rows and columns?

Example: human pancreatic cells (Muraro *et al.* 2016)



It's a sparse matrix...

```
X[1:10, 1:5]
```

```
## 10 x 5 sparse Matrix of class "dgCMatrix"
##          D28-1_1 D28-1_2 D28-1_3 D28-1_4 D28-1_5
## A1BG-AS1 .       .       .       .              .
## A1BG     .       .       1.00196 1.00196        .
## A1CF     6.07143 .       2.00785 6.07143        .
## A2M-AS1  .       .       .       .              .
## A2ML1    .       .       .       .              .
## A2M      .       5.04947 .       .              .
## A4GALT   .       1.00196 .       .              .
## A4GNT    .       .       .       .              .
## AAAS     1.00196 .       .       1.00196        .
## AACSP1   .       .       .       .              .
```

- rows: 19,140 genes
- columns: 3,072 cells

Computing row-wise (genes) statistics

- Statistics across cells (columns) per gene (row)

```
head(row.scores)[, .(gene, nnz, mean, sd, cv, sum, sum.sq)]
```

```
##          gene   nnz          mean          sd        cv         sum        sum.sq
##        <char> <num>         <num>       <num>     <num>       <num>         <num>
## 1:  A1BG-AS1     3 0.0009784766 0.03130105 31.989576    3.00588      3.011772
## 2:      A1BG   256 0.0975982472 0.35768881  3.664910  299.82181    422.169739
## 3:      A1CF  1525 1.8421998024 3.05790520  1.659920 5659.23779 39141.703125
## 4:   A2M-AS1    60 0.0208792929 0.15205784  7.282710   64.14119     72.345612
## 5:     A2ML1    53 0.0176138654 0.13416946  7.617264   54.10979     56.235519
## 6:       A2M   148 0.2410295606 2.15662813  8.947567  740.44281  14461.827148
```

- `nnz`: number of non-zero elements
- `mean`: average; `sd`: standard deviation; `cv`: coefficient of variation
- `sum`: sum of all the counts; `sum.sq`: sum of all the squared counts

Computing column-wise (cells) statistics

- Statistics across genes (rows) per cell (column)

```
head(col.scores)
```

```
##       name   nnz       mean         sd        cv        sum     sum.sq
##     <char> <num>      <num>      <num>     <num>      <num>      <num>
## 1: D28-1_1  5448 1.22457922 19.0589027 15.563634 23438.4453 6980786.50
## 2: D28-1_2  6464 1.67512119  9.8313141  5.869017 32061.8203 1903582.62
## 3: D28-1_3  5212 0.90166962 12.9791527 14.394577 17257.9570 3239686.75
## 4: D28-1_4  7318 1.78534663 10.8671150  6.086838 34171.5352 2321212.75
## 5: D28-1_5  4666 0.84562463  5.1858549  6.132573 16185.2559  528393.56
## 6: D28-1_6   210 0.02899498  0.9003856 31.053156   554.9639   15531.96
```

- `nnz`: number of non-zero elements
- `mean`: average; `sd`: standard deviation; `cv`: coefficient of variation
- `sum`: sum of all the counts; `sum.sq`: sum of all the squared counts

QC #1. Column filtering to ensure the quality of cells

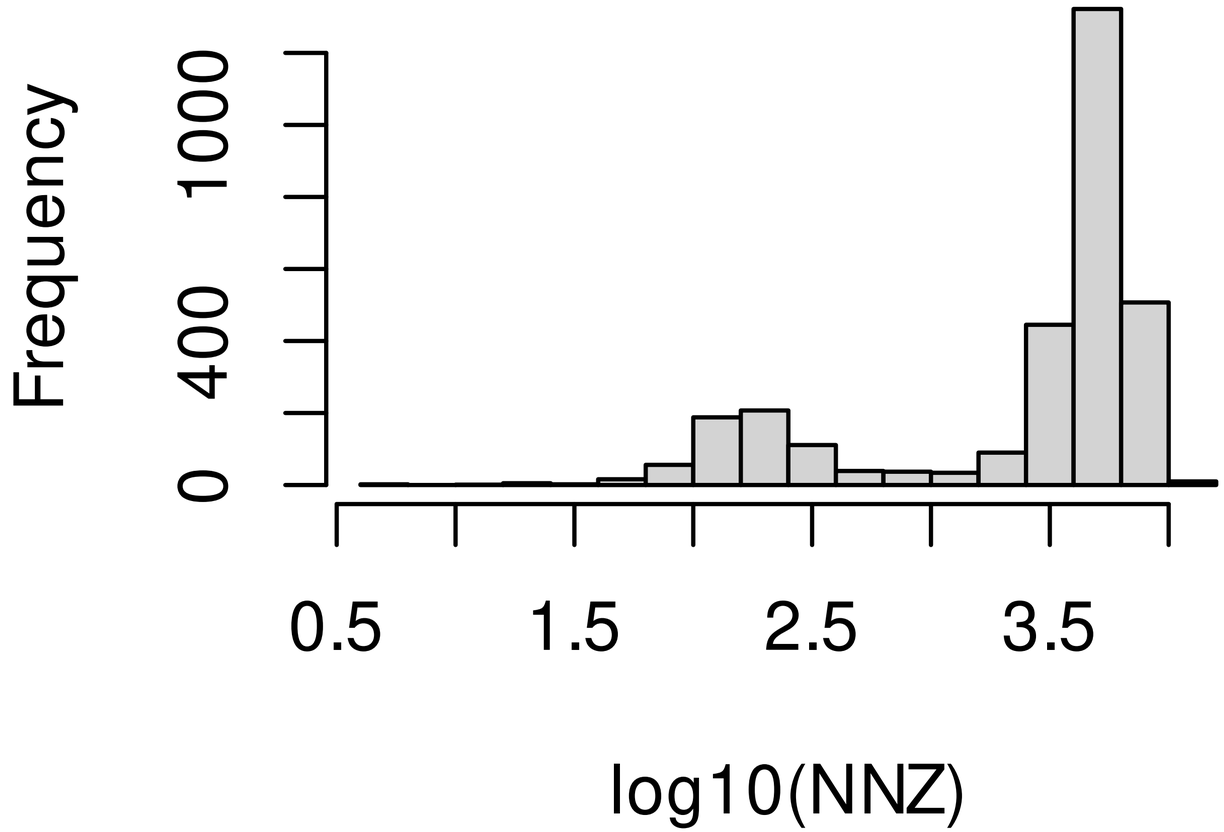
Why cell Q/C?

- Too few genes expressed:
- Unusual high expressions:

QC #1. Column filtering to ensure the quality of cells

Why cell Q/C?

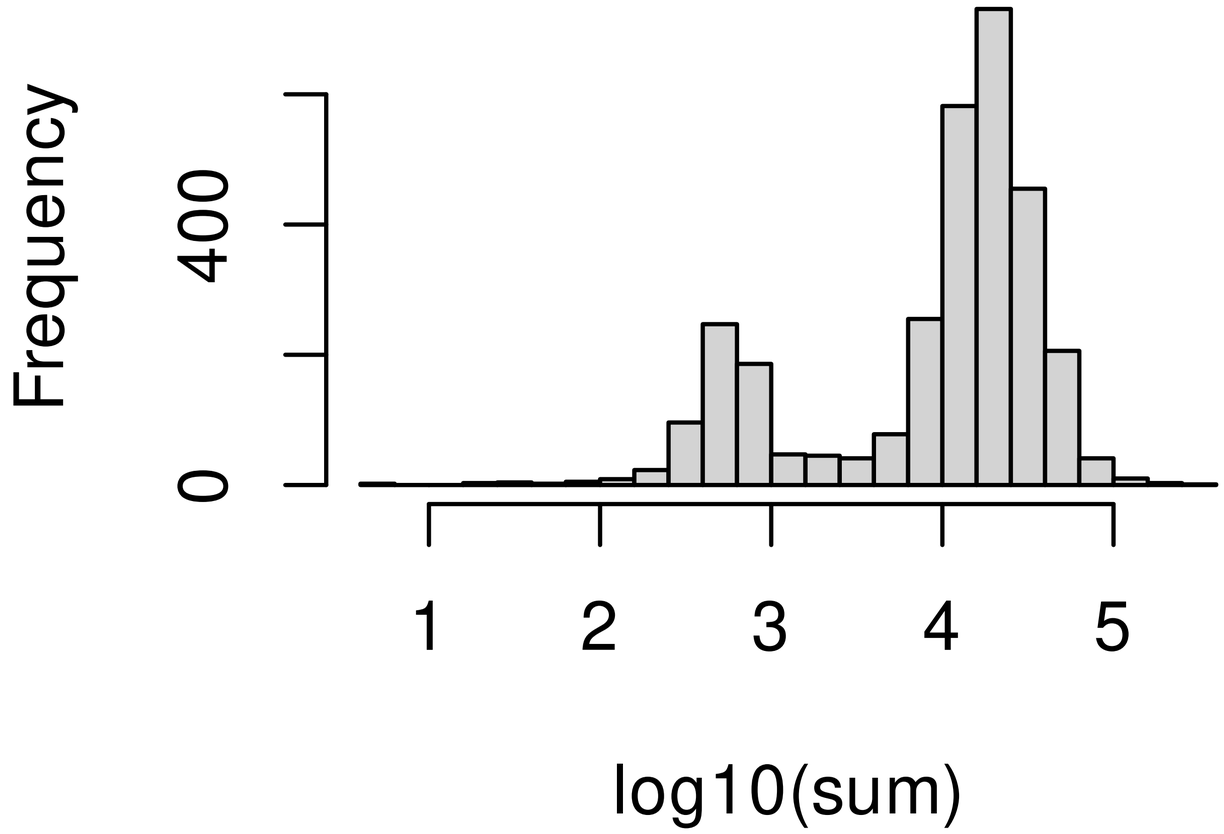
- Too few genes expressed: dying, ambient cells, too many technical drop out
- Unusual high expressions:



QC #1. Column filtering to ensure the quality of cells

Why cell Q/C?

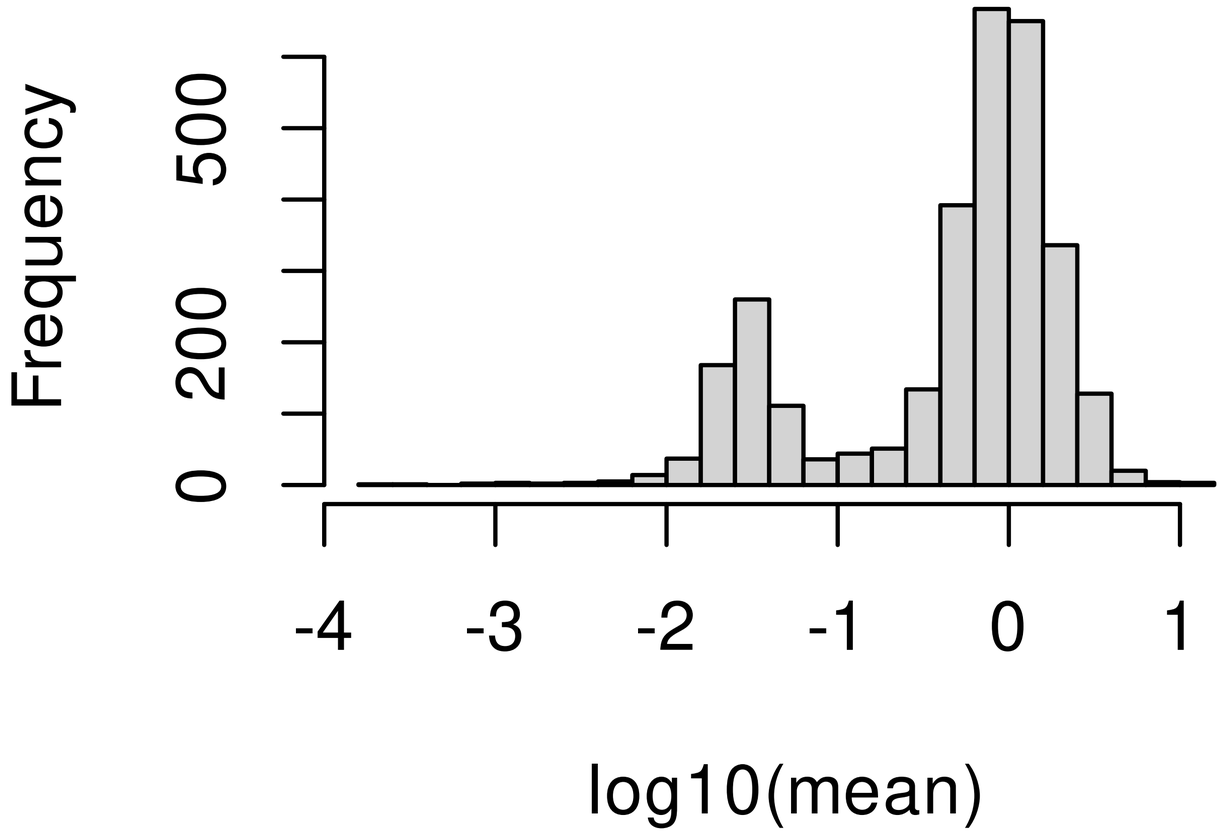
- Too few genes expressed: dying, ambient cells, too many technical drop out
- Unusual high expressions: doublets, high ribosomal genes (background activities), disequilibrium



QC #1. Column filtering to ensure the quality of cells

Why cell Q/C?

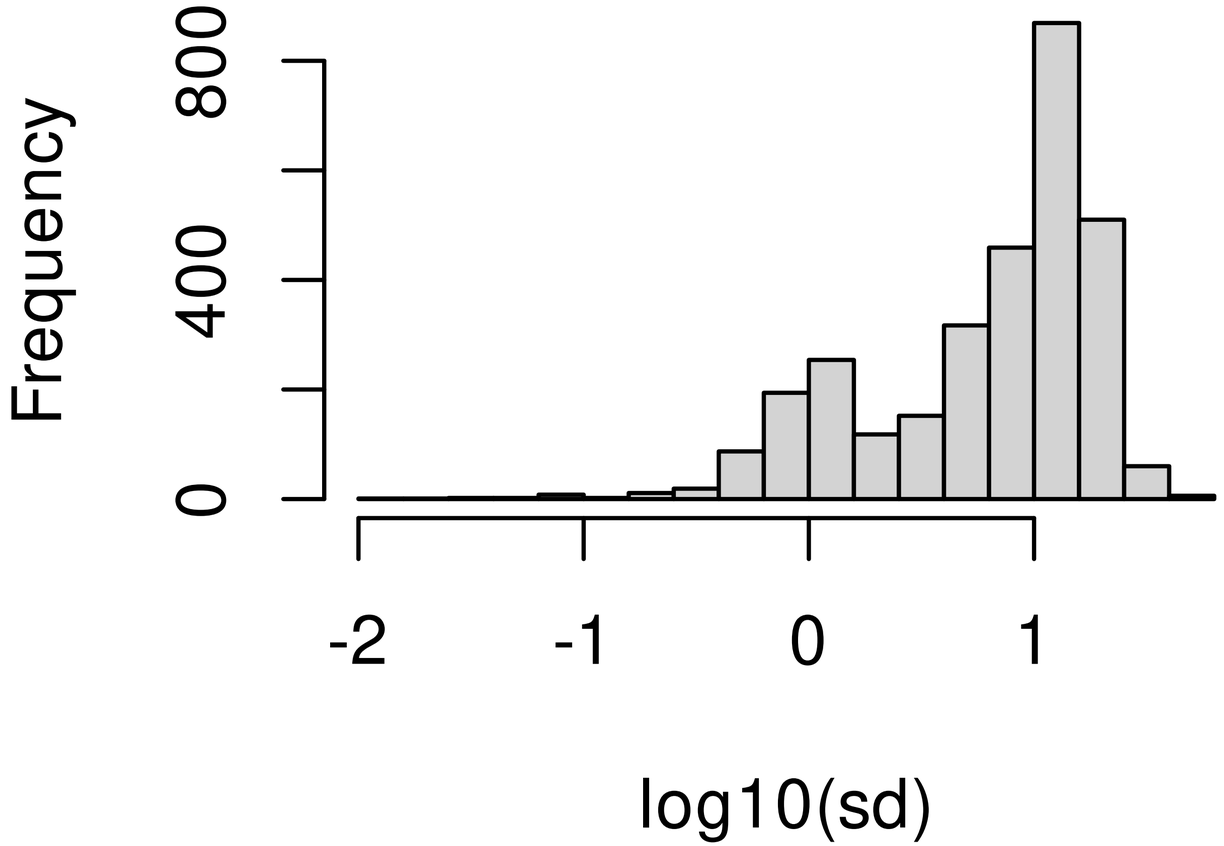
- Too few genes expressed: dying, ambient cells, too many technical drop out
- Unusual high expressions:



QC #1. Column filtering to ensure the quality of cells

Why cell Q/C?

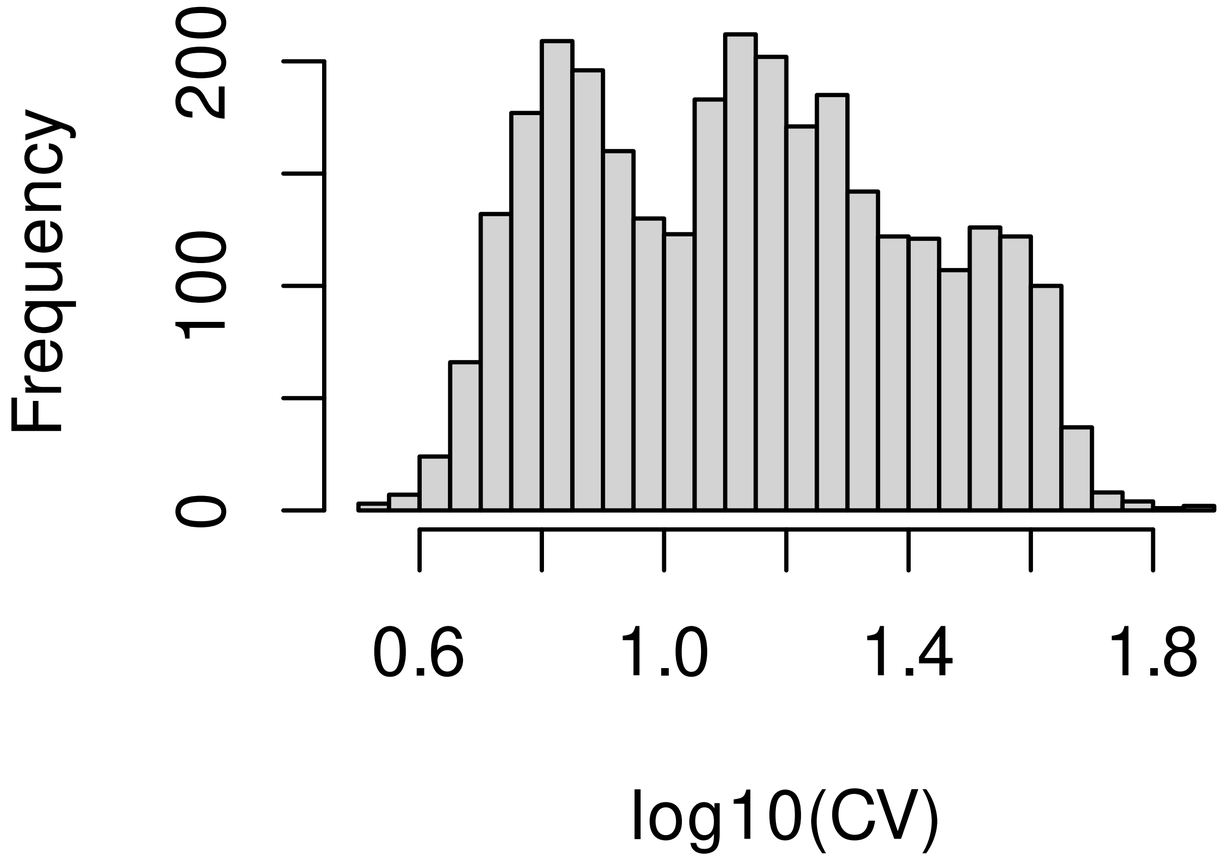
- Too few genes expressed: dying, ambient cells, too many technical drop out
- Unusual high expressions:



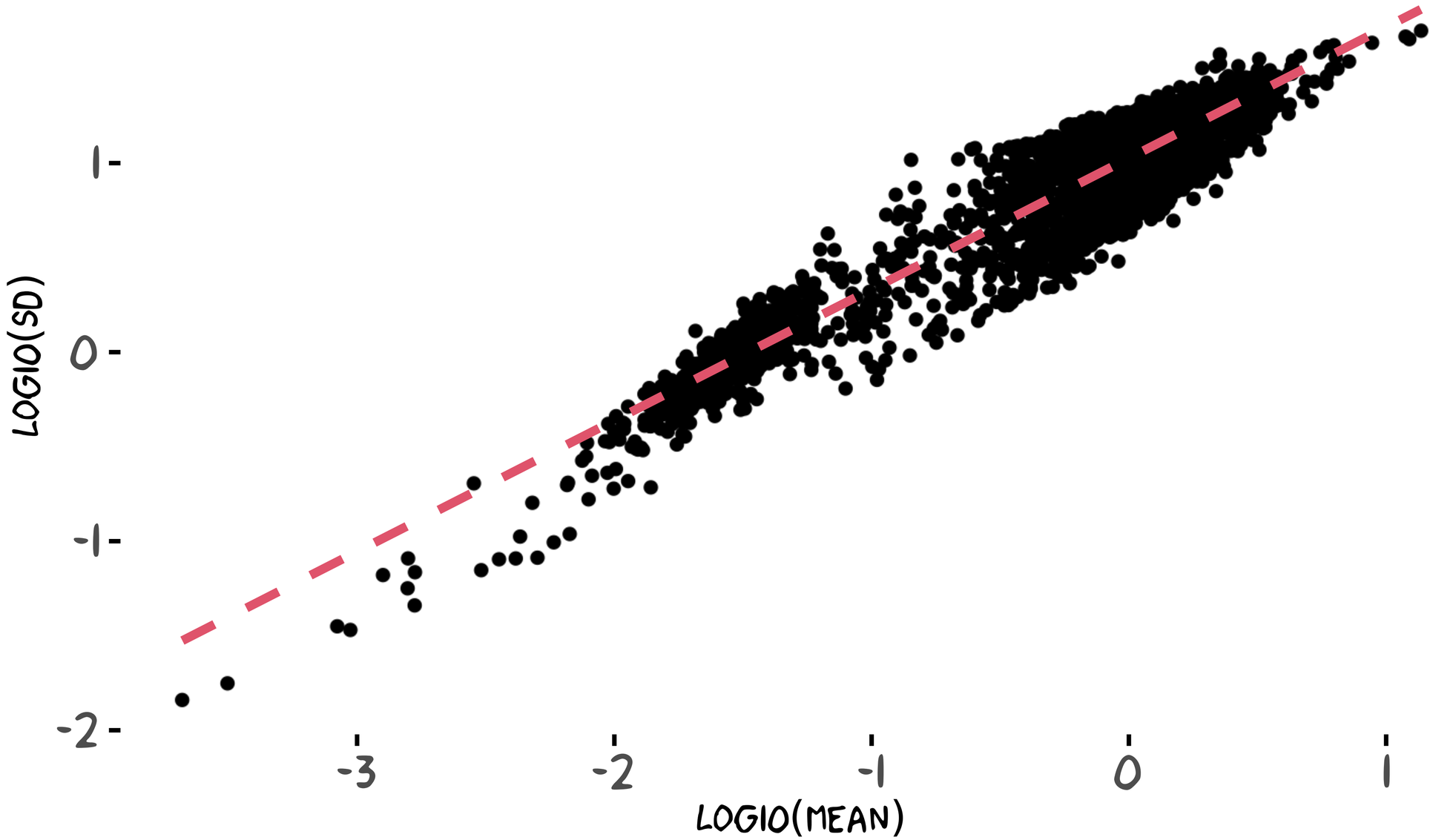
QC #1. Column filtering to ensure the quality of cells

Why cell Q/C?

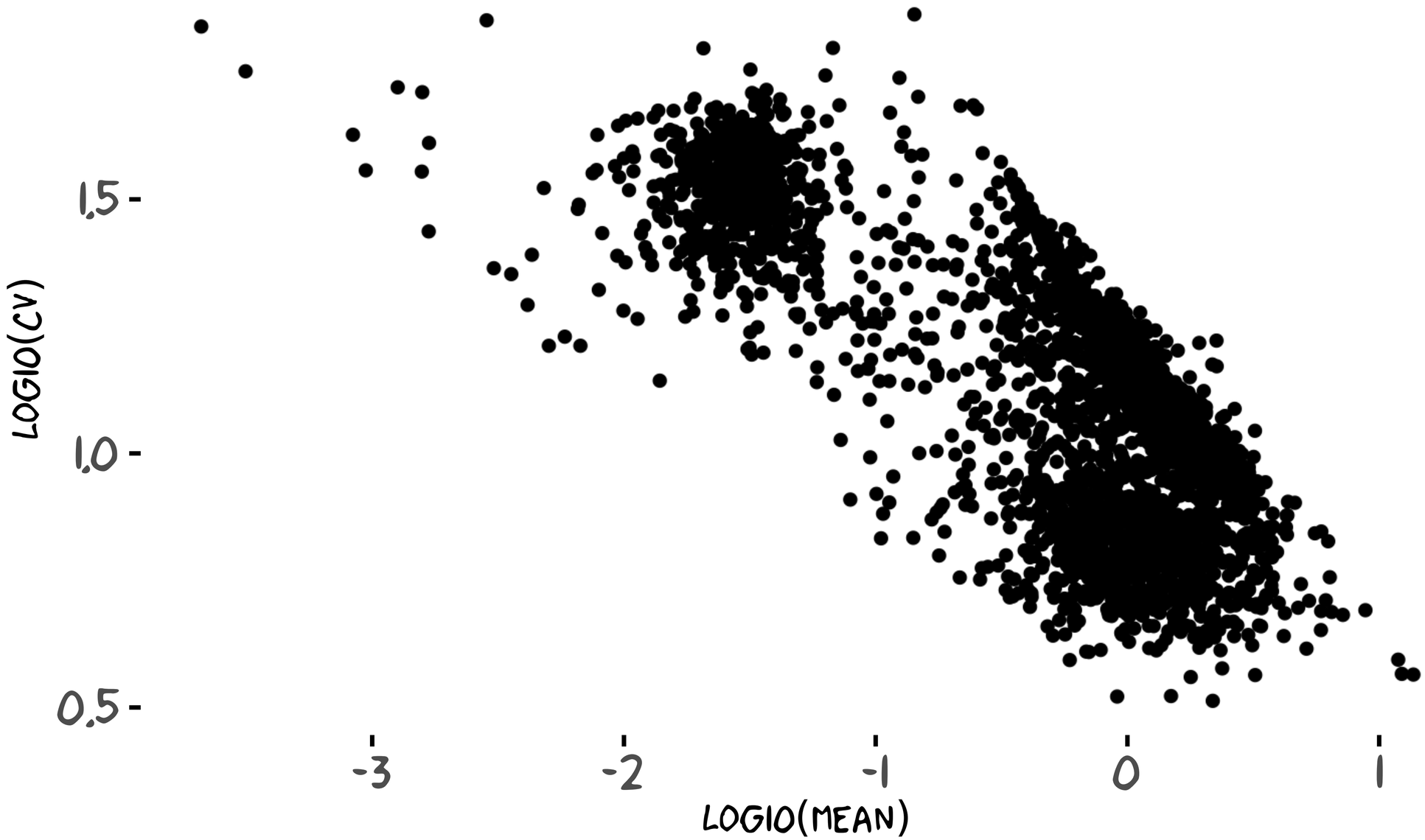
- Too few genes expressed: dying, ambient cells, too many technical drop out
- Unusual high expressions:



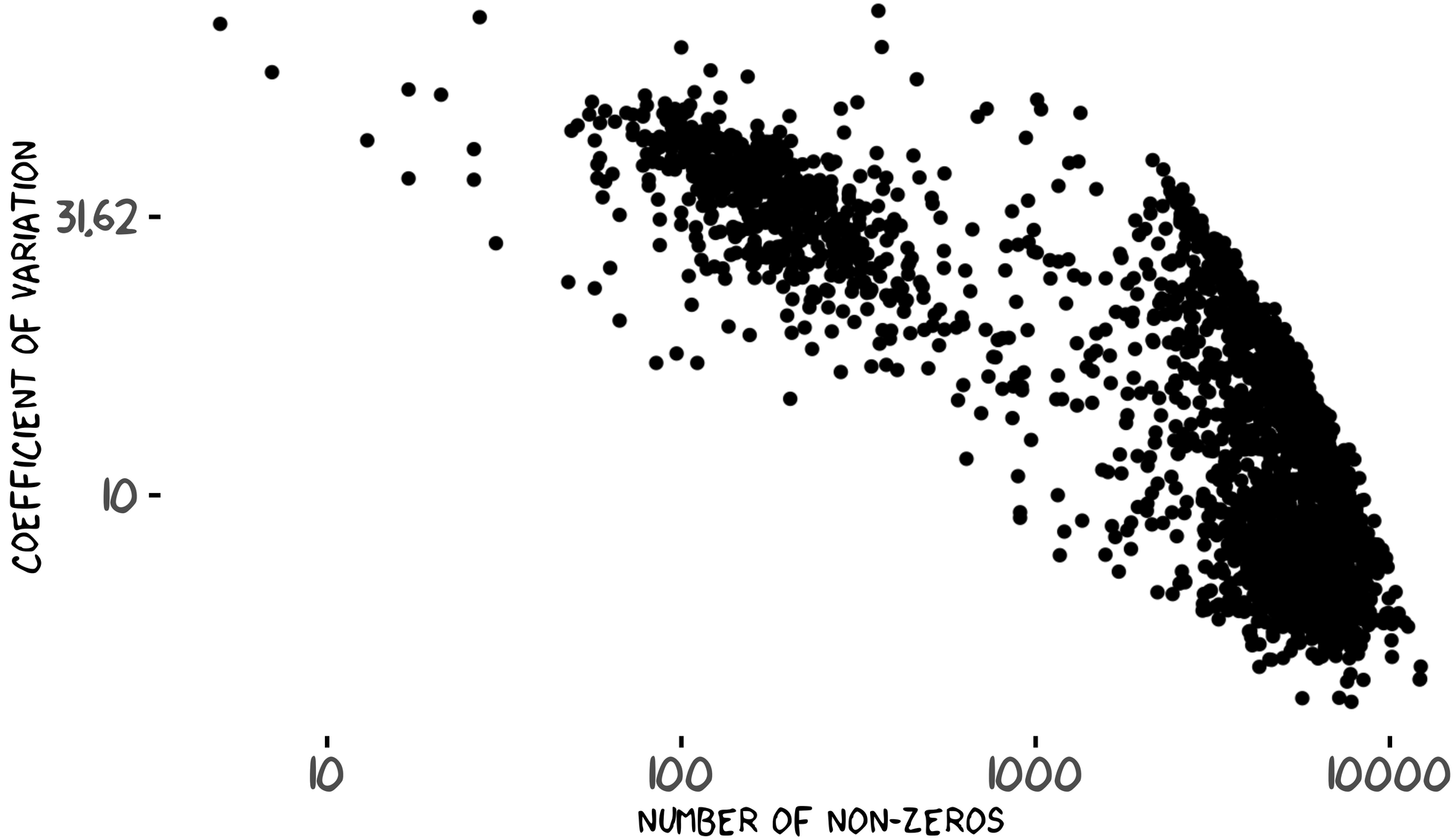
Mean-variance relationship of cell statistics



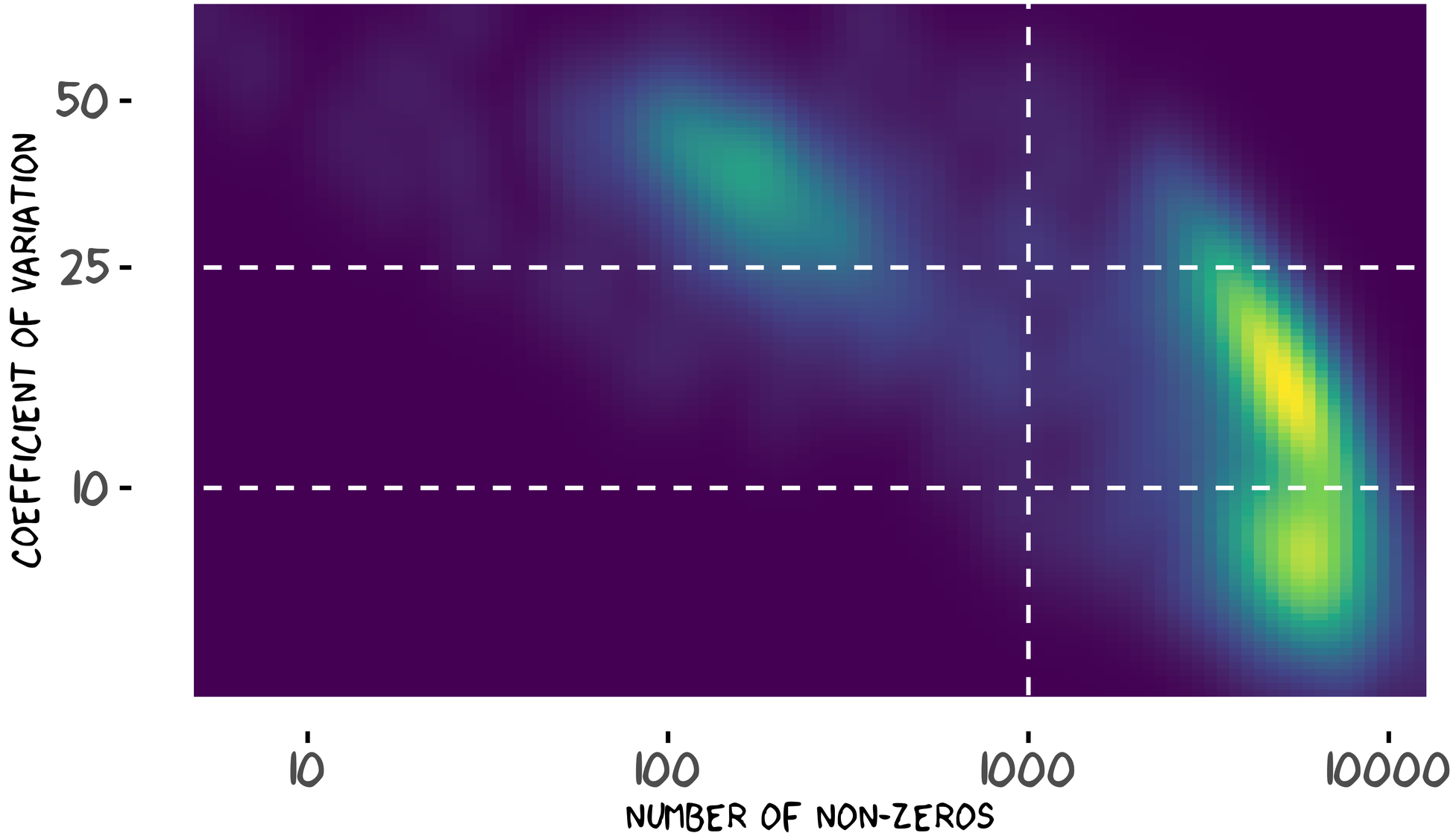
Mean-variance relationship of cell statistics



Robust Q/C by putting multiple scores together



Robust Q/C by putting multiple scores together



Can we systematically dissect “axes of variation?”

Unsupervised Learning Methods considered in single-cell analysis:

1. **Principal** Component Analysis (PCA)

Can we systematically dissect “axes of variation?”

Unsupervised Learning Methods considered in single-cell analysis:

1. **Principal** Component Analysis (PCA)
2. (more general) matrix factorization (lect 17)
3. (Variational) autoencoder (lect 18)

Can we systematically dissect “axes of variation?”

Unsupervised Learning Methods considered in single-cell analysis:

1. **Principal** Component Analysis (PCA)
2. (more general) matrix factorization (lect 17)
3. (Variational) autoencoder (lect 18)
4. GPT (Generative Pretrained Transformer; *ask ChatGPT*)

What is Principal Component Analysis?

PCA: Consider this multivariate linear model

$$\begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{pmatrix} = \begin{pmatrix} U_{11} & \cdots & U_{1k} \\ U_{21} & \cdots & U_{2k} \\ \vdots & \vdots & \vdots \\ U_{p1} & \cdots & U_{pk} \end{pmatrix} \begin{pmatrix} V_1 \\ \vdots \\ V_k \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_p \end{pmatrix}$$

or

$\mathbf{x} = U\mathbf{v} + \epsilon$

- If we **knew** U, we would be able to solve weights V.
- And vice versa…

PCA: Consider this multivariate linear model

For many columns, $X = UV + E$, or

$$\begin{pmatrix} X_{11} & \cdots & X_{1n} \\ X_{21} & \cdots & X_{2n} \\ \vdots & & \vdots \\ X_{p1} & \cdots & X_{pn} \end{pmatrix} = \begin{pmatrix} U_{11} & \cdots & U_{1k} \\ U_{21} & \cdots & U_{2k} \\ \vdots & \vdots & \vdots \\ U_{p1} & \cdots & U_{pk} \end{pmatrix} \begin{pmatrix} V_{11} & \cdots & V_{1n} \\ \vdots & & \vdots \\ V_{k1} & \cdots & V_{kn} \end{pmatrix} + \cdots$$

- If we **knew** U, we would be able to solve weights V.
- And vice versa…

PCA: What is a projection matrix?

Suppose we knew V. For each gene (row) g, we solve the following regression.

$$\mathbf{x}_g^\top \sim V\mathbf{u}_g^\top + \epsilon I$$

Least square solution:

$$\min_{\mathbf{u}_g} \|\mathbf{x}_g^\top - V\mathbf{u}_g^\top\|$$

PCA: What is a projection matrix?

Suppose we knew V. For each gene (row) g, we solve the following regression.

$$\mathbf{x}_g^\top \sim V\mathbf{u}_g^\top + \epsilon I$$

$$\begin{pmatrix} X_{g1} \\ \vdots \\ X_{gn} \end{pmatrix} \sim \begin{pmatrix} V_{11} & \cdots & V_{1k} \\ \vdots & \vdots & \vdots \\ V_{n1} & \cdots & V_{nk} \end{pmatrix} \begin{pmatrix} U_{g1} \\ \vdots \\ U_{gk} \end{pmatrix} + \begin{pmatrix} \epsilon \\ \vdots \\ \epsilon \end{pmatrix}$$

Least square solution:

$$\min_{\mathbf{u}_g} \|\mathbf{x}_g^\top - V\mathbf{u}_g^\top\|$$

PCA: What is a projection matrix?

Suppose we knew V. For each gene (row) g, we solve the following regression.

$$\mathbf{x}_g^\top \sim V\mathbf{u}_g^\top + \epsilon I$$

$$\begin{pmatrix} X_{g1} \\ \vdots \\ X_{gn} \end{pmatrix} \sim \begin{pmatrix} V_{11} & \cdots & V_{1k} \\ \vdots & \vdots & \vdots \\ V_{n1} & \cdots & V_{nk} \end{pmatrix} \begin{pmatrix} U_{g1} \\ \vdots \\ U_{gk} \end{pmatrix} + \begin{pmatrix} \epsilon \\ \vdots \\ \epsilon \end{pmatrix}$$

Least square solution:

$$\widehat{\mathbf{u}}_g^\top = (V^\top V)^{-1} V^\top \mathbf{x}_g^\top$$

PCA: What is a projection matrix?

Plugging

$$\hat{\mathbf{u}}_g^\top = (V^\top V)^{-1} V^\top \mathbf{x}_g^\top$$

into the following prediction model:

$$\hat{\mathbf{x}}_g^\top = V \hat{\mathbf{u}}_g^\top$$

PCA: What is a projection matrix?

Plugging

$$\hat{\mathbf{u}}_g^\top = (V^\top V)^{-1} V^\top \mathbf{x}_g^\top$$

into the following prediction model:

$$\begin{aligned} \hat{\mathbf{x}}_g^\top &= V\hat{\mathbf{u}}_g^\top \\ &= V(V^\top V)^{-1} V^\top \mathbf{x}_g^\top \end{aligned}$$

PCA: What is a projection matrix?

Plugging

$$\hat{\mathbf{u}}_g^\top = (V^\top V)^{-1} V^\top \mathbf{x}_g^\top$$

into the following prediction model:

$$\begin{aligned} \hat{\mathbf{x}}_g^\top &= V \hat{\mathbf{u}}_g^\top \\ \hat{\mathbf{x}}_g &= \mathbf{x}_g \underbrace{V(V^\top V)^{-1} V^\top}_{n \times n \text{ projection matrix}} \end{aligned}$$

PCA: Projection can be done on the other side

A multivariate regression model for sample (column) j:

$$\mathbf{x}_j \sim U\mathbf{v}_j + \epsilon$$

The least-square solution for the V:

$$\hat{\mathbf{v}}_j = (U^\top U)^{-1} U^\top \mathbf{x}_j$$

Then we have

$$\hat{\mathbf{x}}_j = \underbrace{U(U^\top U)^{-1}U^\top}_{p \times p \text{ projection matrix}} \mathbf{x}_j$$

PCA = maximizing total variance of the projected data

Constrained optimization

Given the rank-1 factorization

$$\hat{X} = \mathbf{u}\mathbf{v}^\top$$

and the least square solution $\hat{\mathbf{v}}$

$$\hat{\mathbf{v}}^\top = \frac{\mathbf{u}^\top}{\mathbf{u}^\top\mathbf{u}} X$$

We want to maximize the variance of this projected vector $\mathbf{v}$

$$\hat{\mathbf{v}}^\top\hat{\mathbf{v}} = \frac{\mathbf{u}^\top}{\mathbf{u}^\top\mathbf{u}} XX^\top \frac{\mathbf{u}}{\mathbf{u}^\top\mathbf{u}} = \mathbf{u}^\top XX^\top\mathbf{u}$$

where we assume $\mathbf{u}$ is a unit vector.

PCA is an eigen value problem

PCA

Letting the covariance matrix $\hat{\Sigma} = XX^\top/(p-1)$, we want to find a unit vector $\mathbf{v}$ by

$$\max \mathbf{v}^\top \hat{\Sigma} \mathbf{v}$$

subject to $\mathbf{v}^\top \mathbf{v} = 1$.

Eigen value problem

Given the covariance matrix $\hat{\Sigma}$, we can resolve an eigen-value λ and the corresponding eigen-vector $\mathbf{v}$ such that

$$\hat{\Sigma}\mathbf{v} = \lambda \mathbf{v}$$

SVD: another equivalent method for PCA

Singular Value Decomposition

SVD identifies three matrices of X:

$$X = UDV^\top$$

where both U and V vectors are orthonormal, namely,

- $U^\top U = I, \mathbf{u}_k^\top \mathbf{u}_k = 1$ for all k,
- $V^\top V = I, \mathbf{v}_k^\top \mathbf{v}_k = 1$ for all k.

Covariance by SVD

Covariance across the columns (samples)

$$X^\top X/p = VD^2V^\top/p$$

Covariance across the rows (genes)

$$XX^\top/n = UD^2U^\top/n$$

Remark: standardized matrix

SVD: another equivalent method for PCA

We can confirm the equivalent relations by multiplying singular vectors to the covariance matrix:

$$\underbrace{\frac{(X^\top X)}{p-1}}_{\text{sample covariance}} \mathbf{v}_1 \propto (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) \begin{pmatrix} D_1^2 & 0 & \dots & \dots \\ 0 & D_2^2 & 0 & \dots \\ 0 & \dots & \ddots & 0 \\ 0 & \dots & 0 & D_k^2 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \vdots \\ \mathbf{v}_k^\top \end{pmatrix} \mathbf{v}_1$$

SVD: another equivalent method for PCA

We can confirm the equivalent relations by multiplying singular vectors to the covariance matrix:

$$\underbrace{\frac{(X^\top X)}{p-1}}_{\text{sample covariance}} \mathbf{v}_1 \propto (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) \begin{pmatrix} D_1^2 & 0 & \dots & \dots \\ 0 & D_2^2 & 0 & \dots \\ 0 & \dots & \ddots & 0 \\ 0 & \dots & 0 & D_k^2 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \vdots \\ \mathbf{v}_k^\top \end{pmatrix} \mathbf{v}_1$$

$$\propto (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) \begin{pmatrix} D_1^2 & 0 & \dots & \dots \\ 0 & D_2^2 & 0 & \dots \\ 0 & \dots & \ddots & 0 \\ 0 & \dots & 0 & D_k^2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

SVD: another equivalent method for PCA

We can confirm the equivalent relations by multiplying singular vectors to the covariance matrix:

$$\underbrace{\frac{(X^\top X)}{p-1}}_{\text{sample covariance}} \mathbf{v}_1 \propto (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) \begin{pmatrix} D_1^2 & 0 & \dots & \dots \\ 0 & D_2^2 & 0 & \dots \\ 0 & \dots & \ddots & 0 \\ 0 & \dots & 0 & D_k^2 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \vdots \\ \mathbf{v}_k^\top \end{pmatrix} \mathbf{v}_1$$

$$\propto (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) \begin{pmatrix} D_1^2 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

SVD: another equivalent method for PCA

We can confirm the equivalent relations by multiplying singular vectors to the covariance matrix:

$$\underbrace{\frac{(X^\top X)}{p-1}}_{\text{sample covariance}} \mathbf{v}_1 \propto (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) \begin{pmatrix} D_1^2 & 0 & \dots & \dots \\ 0 & D_2^2 & 0 & \dots \\ 0 & \dots & \ddots & 0 \\ 0 & \dots & 0 & D_k^2 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \vdots \\ \mathbf{v}_k^\top \end{pmatrix} \mathbf{v}_1$$

$$= \underbrace{\frac{D_1^2}{p-1}}_{\text{eigenvalue}} \underbrace{\mathbf{v}_1}_{\text{eigenvector}}$$

Learning PCA by singular value decomposition

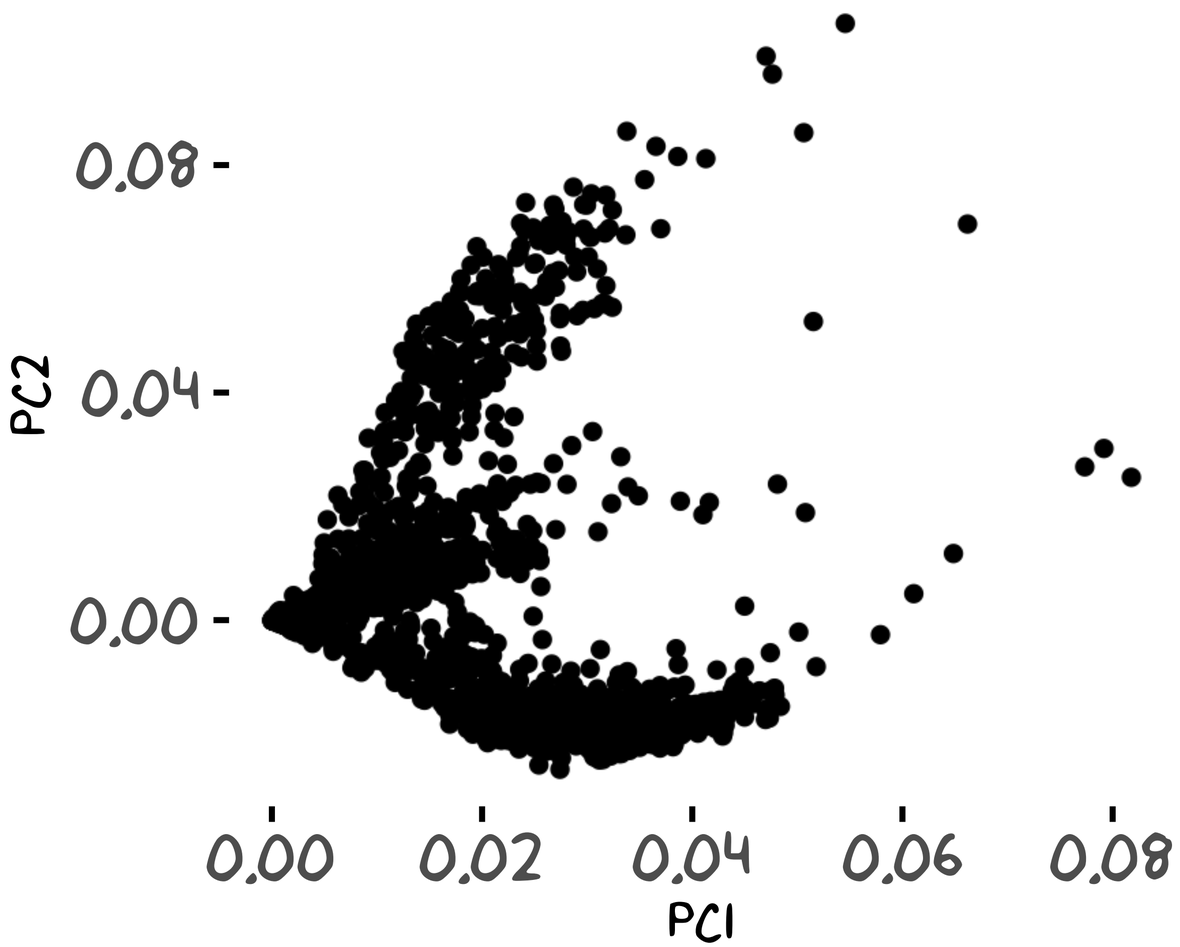
```
library(rsvd)
.svd <- rsvd(X, k = 3)
names(.svd)
```

```
## [1] "d" "u" "v"
```

```
dim(.svd$v)
```

```
## [1] 3072    3
```

- `rsvd` provides faster SVD



Learning PCA by singular value decomposition - 2

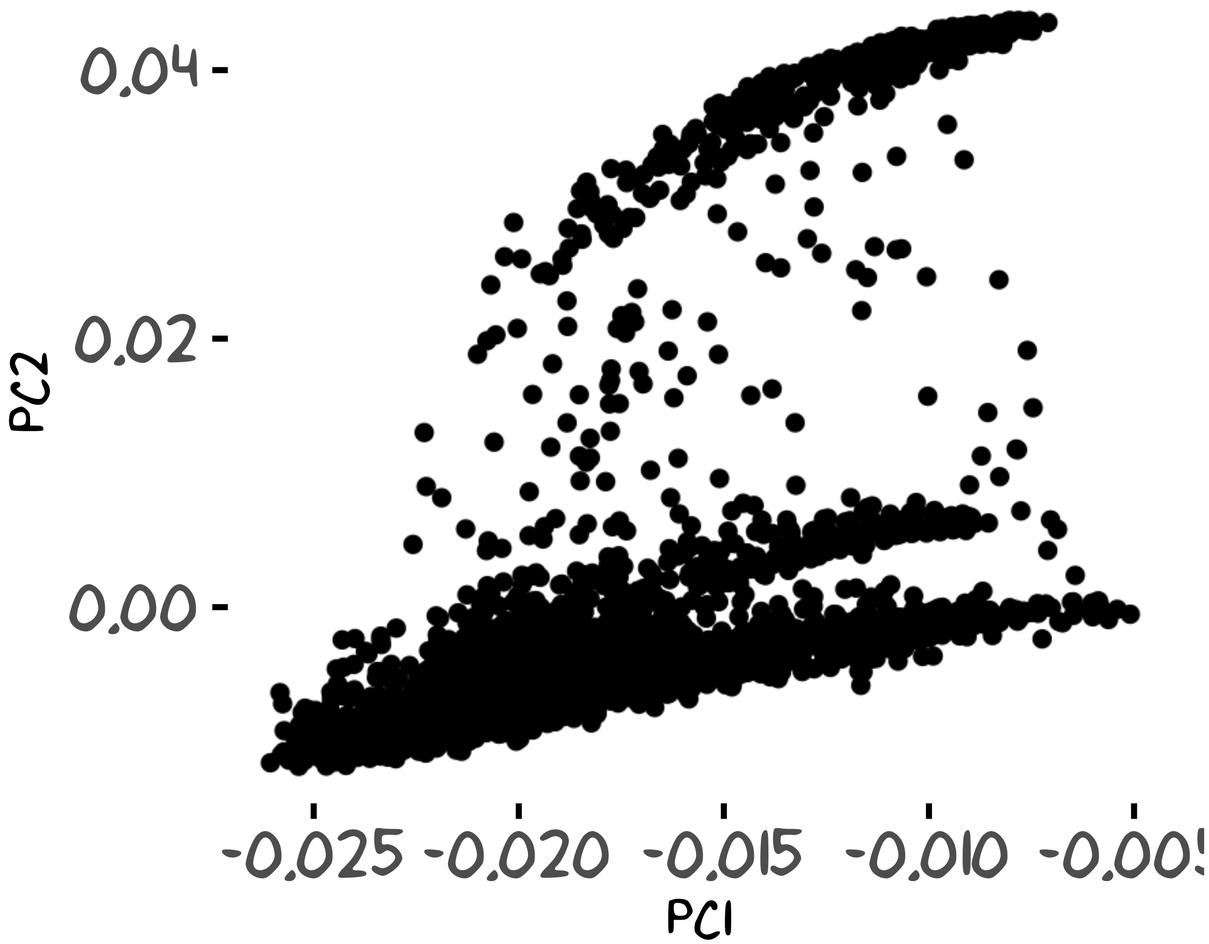
In single-cell data analysis, it is useful to transform the raw count values to more Gaussian-like ones.

```
X.n <- apply(X, 2, function(x) x/sum(x))
X.n <- X.n/10000
log.x <- apply(log1p(X.n), 2, scale)
.svd <- rsvd(log.x, k = 3)
names(.svd)
```

```
## [1] "d" "u" "v"
```

```
dim(.svd$v)
```

```
## [1] 3072    3
```



Learning PCA by singular value decomposition - 3

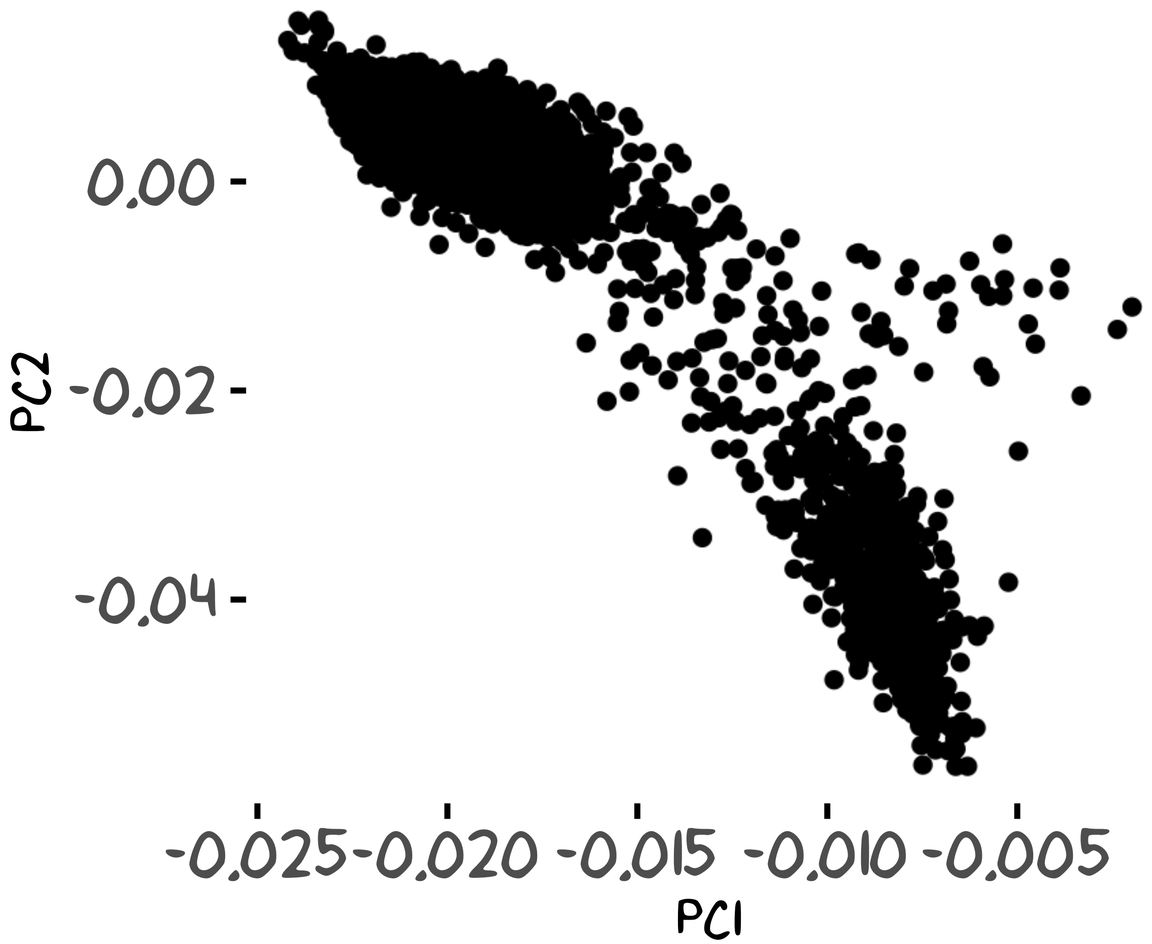
We can also add pseudo-count to make the SVD results smoother..

```
names(.svd)
```

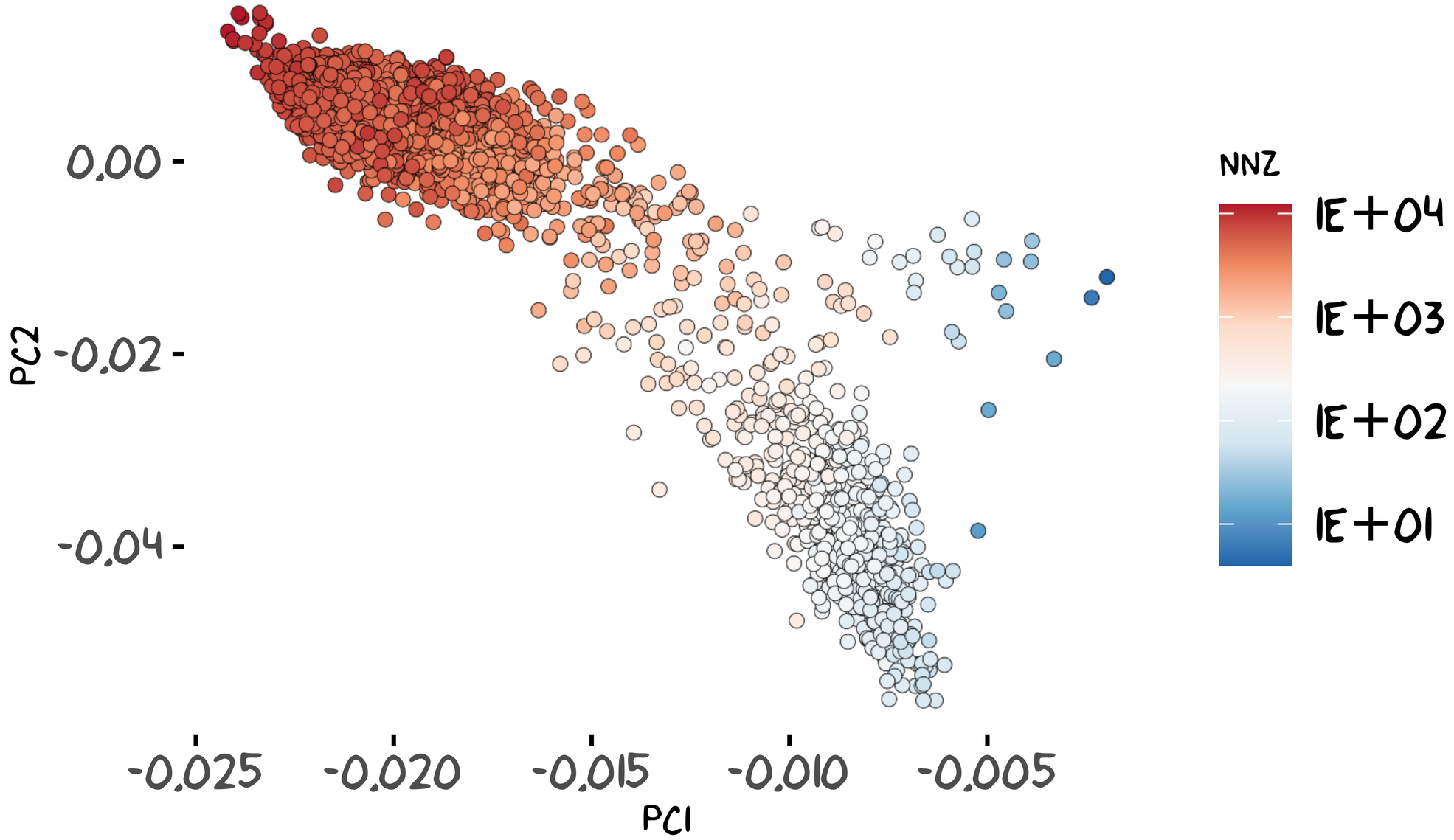
```
## [1] "U" "D" "V"
```

```
dim(.svd$V)
```

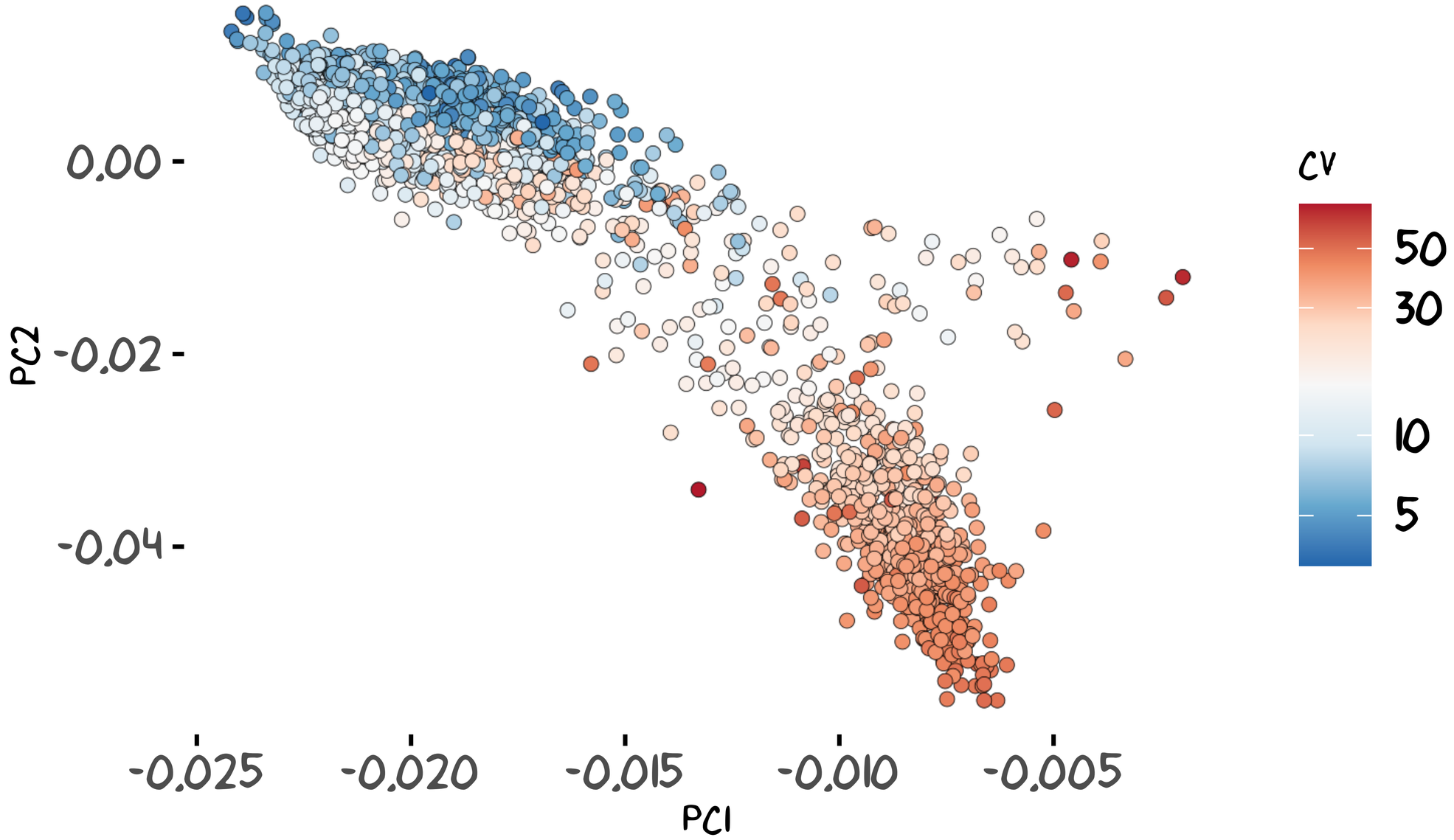
```
## [1] 3072    3
```



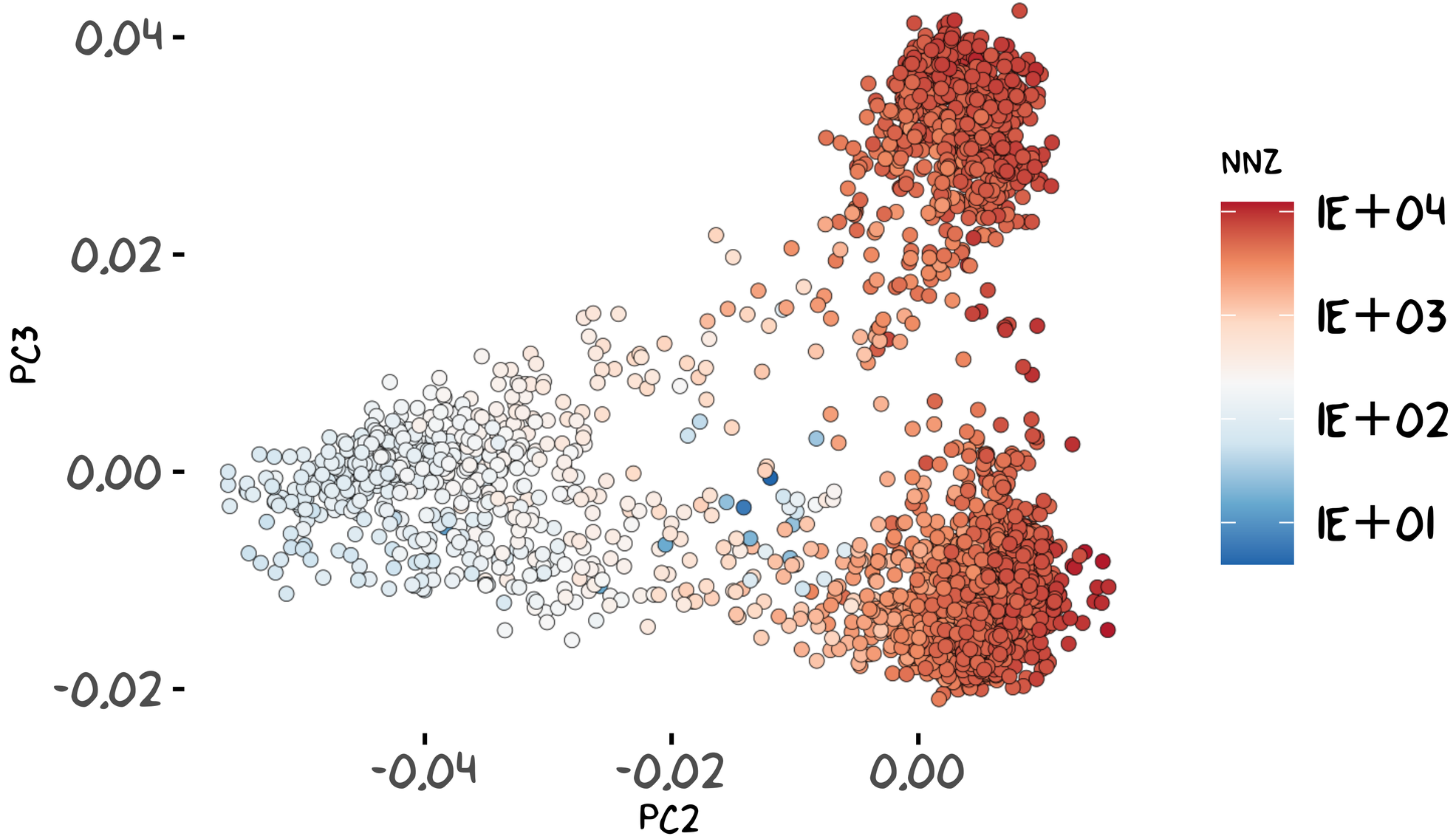
PCA to confirm/remvoe low-quality cells



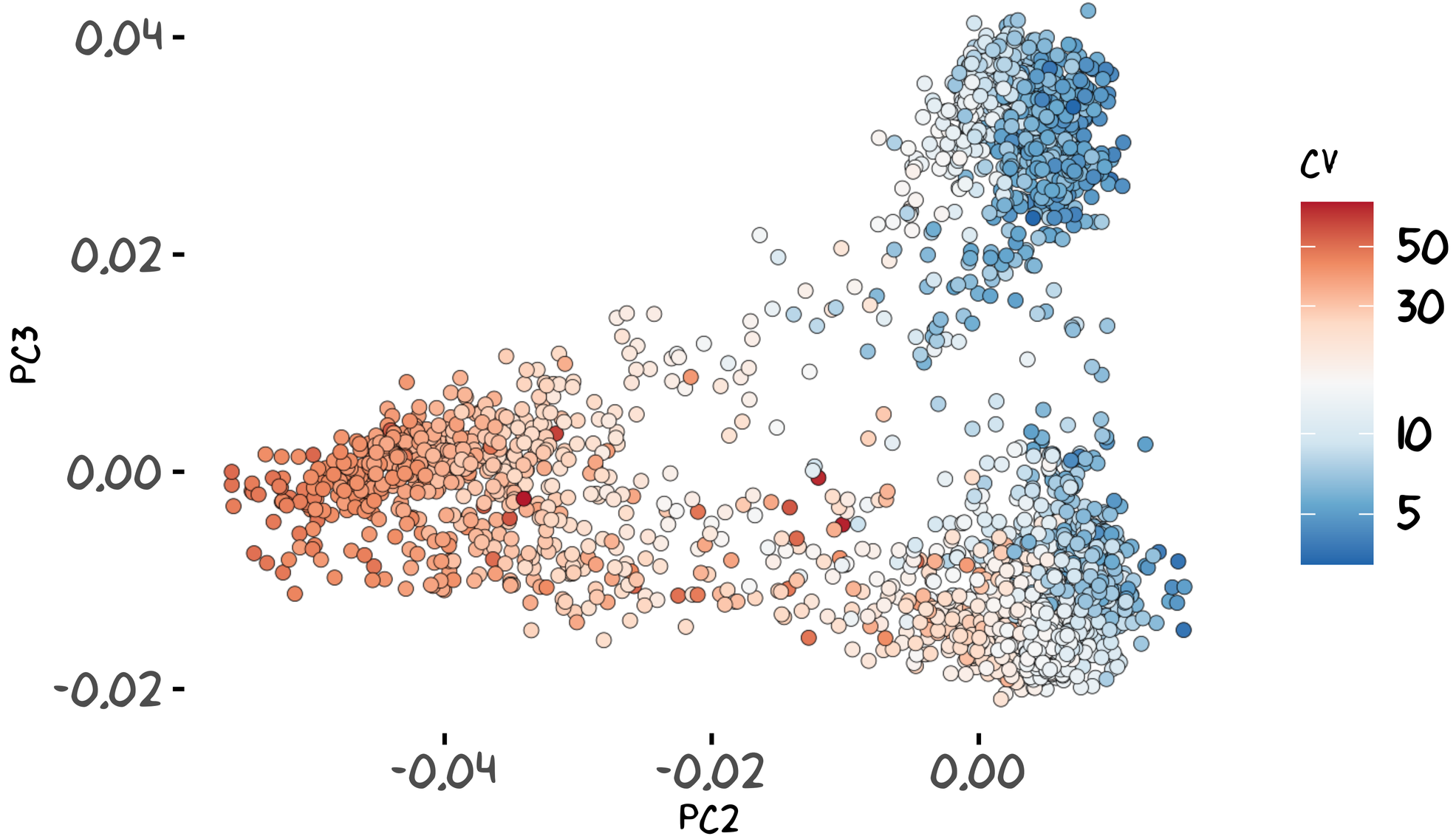
PCA to confirm/remvoe low-quality cells



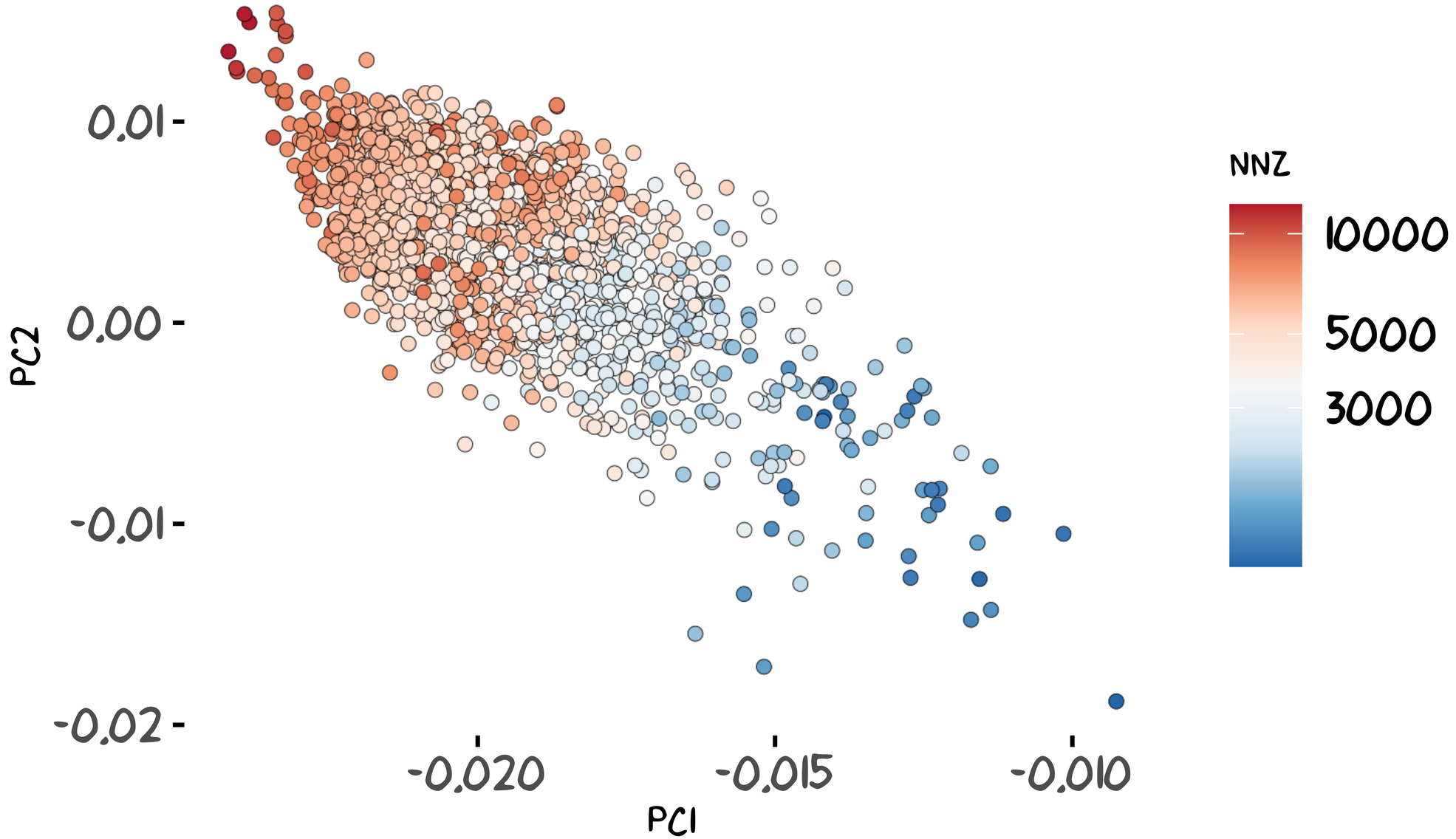
PCA to confirm/remvoe low-quality cells



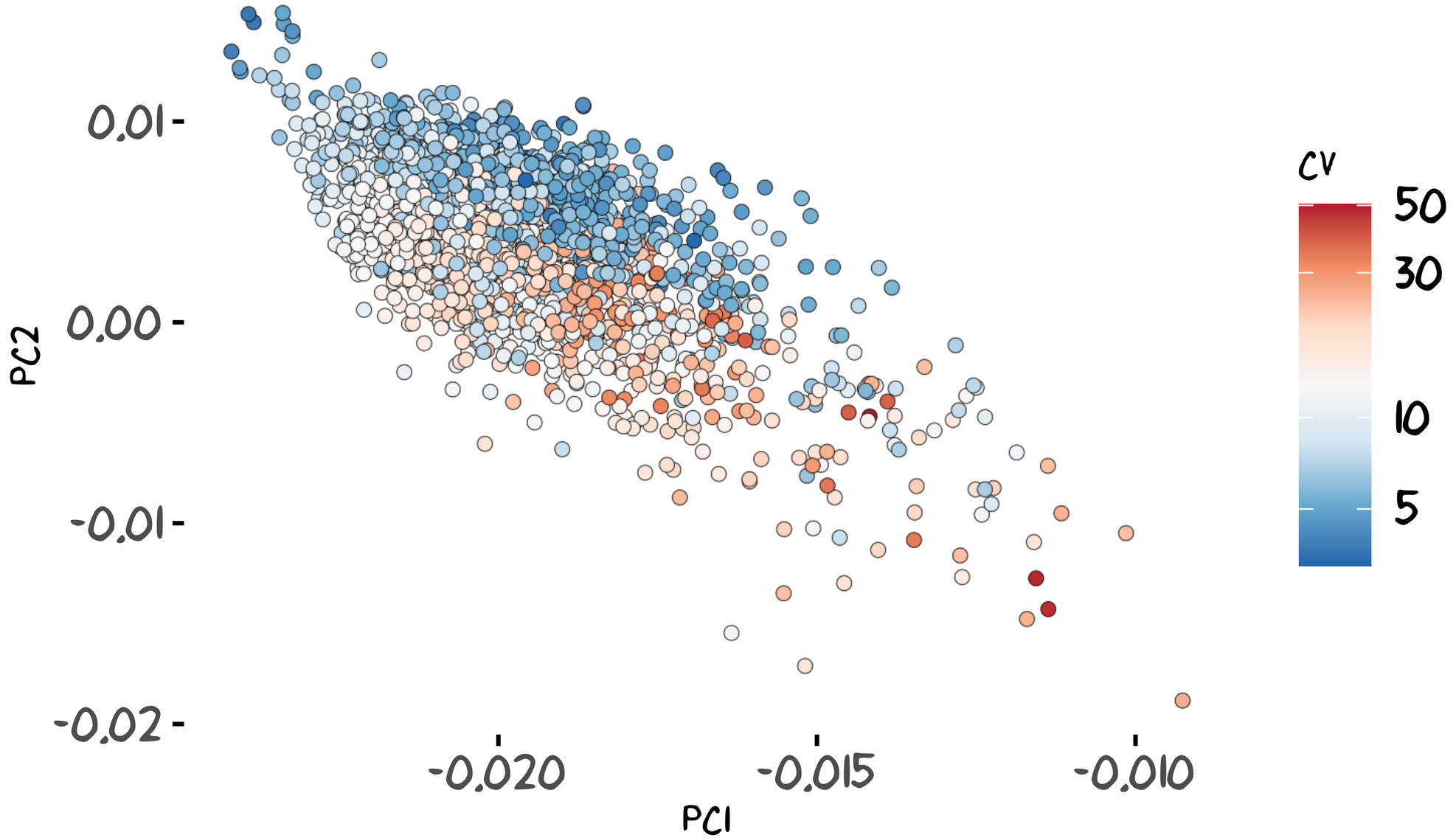
PCA to confirm/remvoe low-quality cells



After removing “dying” cells …



After removing “dying” cells …

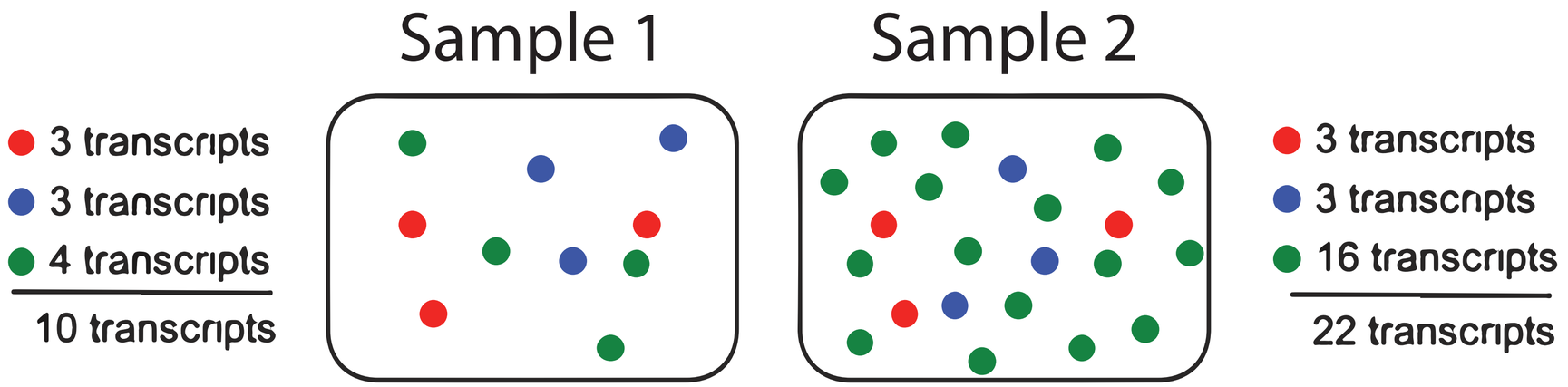


Should we normalize single-cell gene expression vectors?

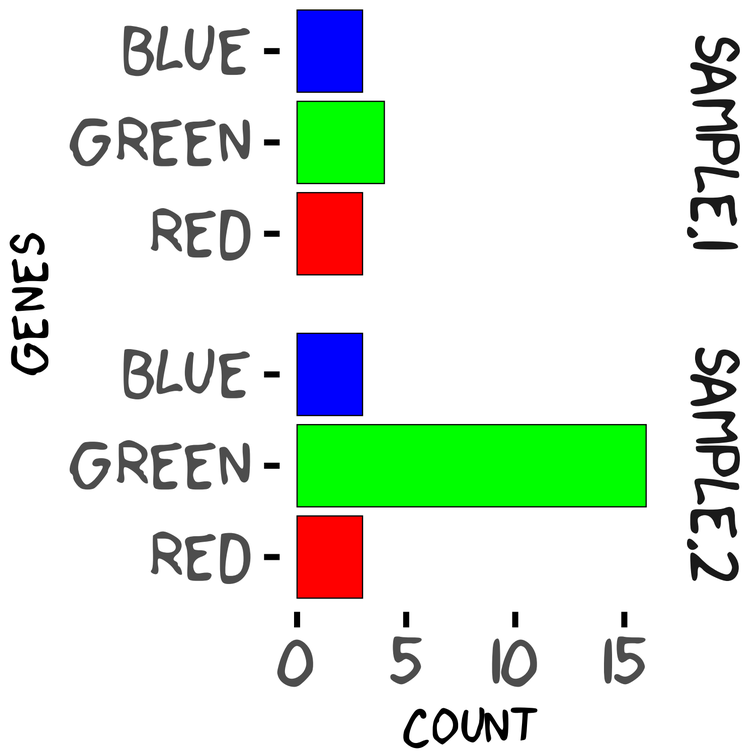
Recall: Should we always normalize within each sample?

What could go wrong with that?

Consider the following example:

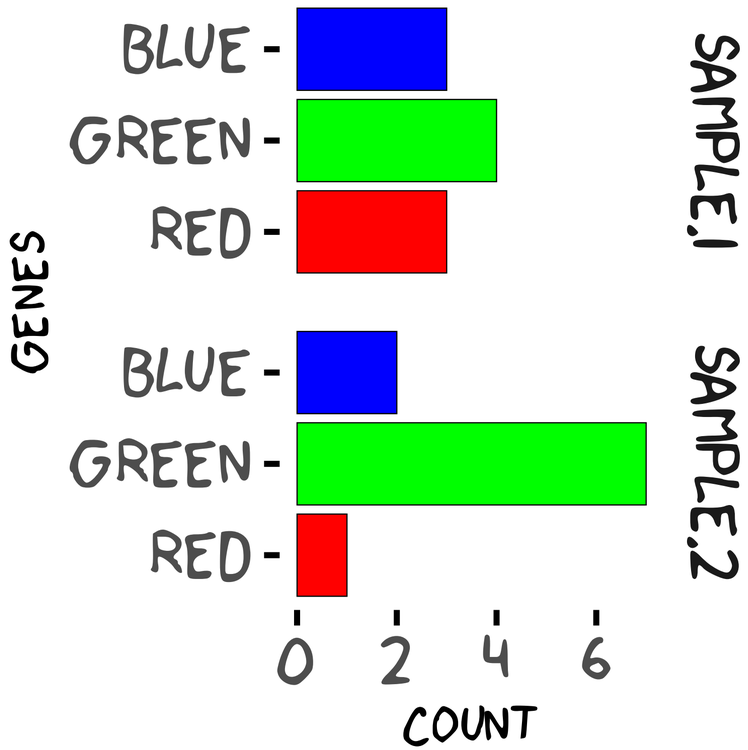


What if we sequence “deep” enough?




```
##       sample.1 sample.2
## Red          3        3
## Green        4       16
## Blue         3        3
```

What if we sequence *less* than the actual counts?

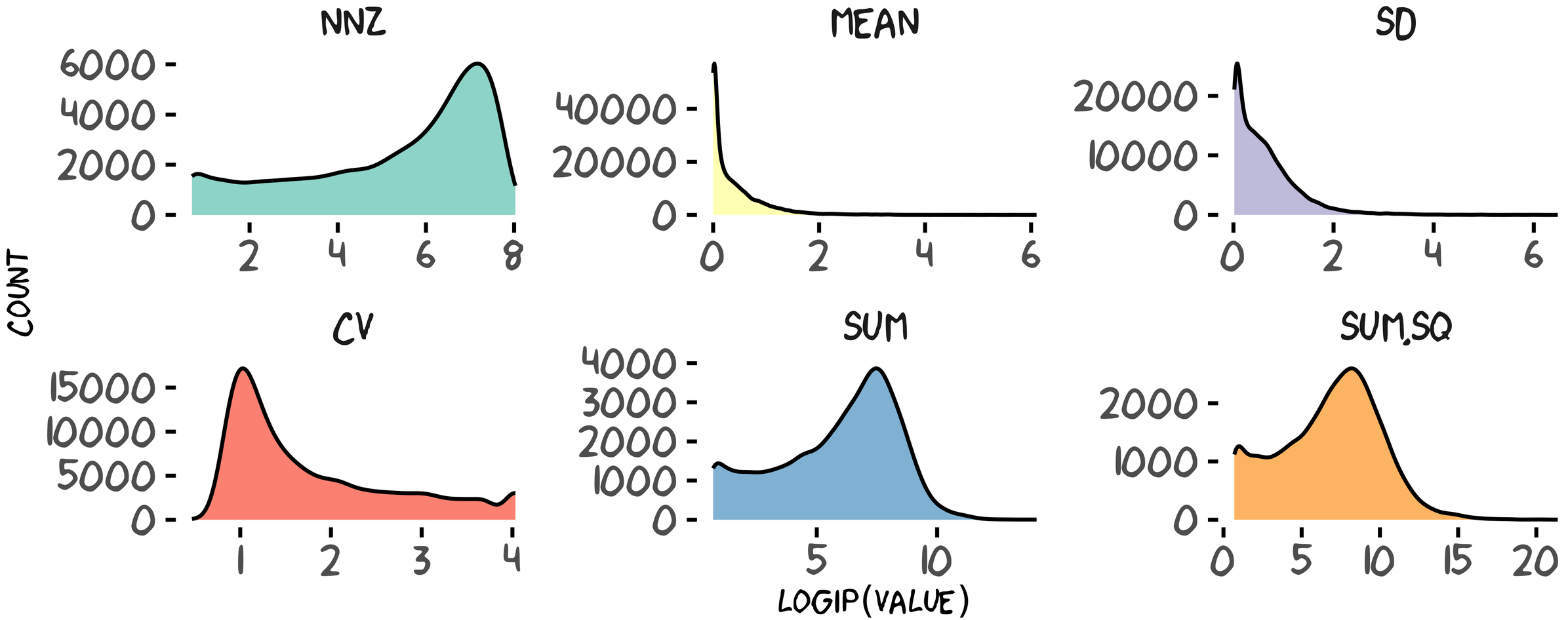



```
##       sample.1 sample.2
## Red          3        1
## Green        4        7
## Blue         3        2
```

Question

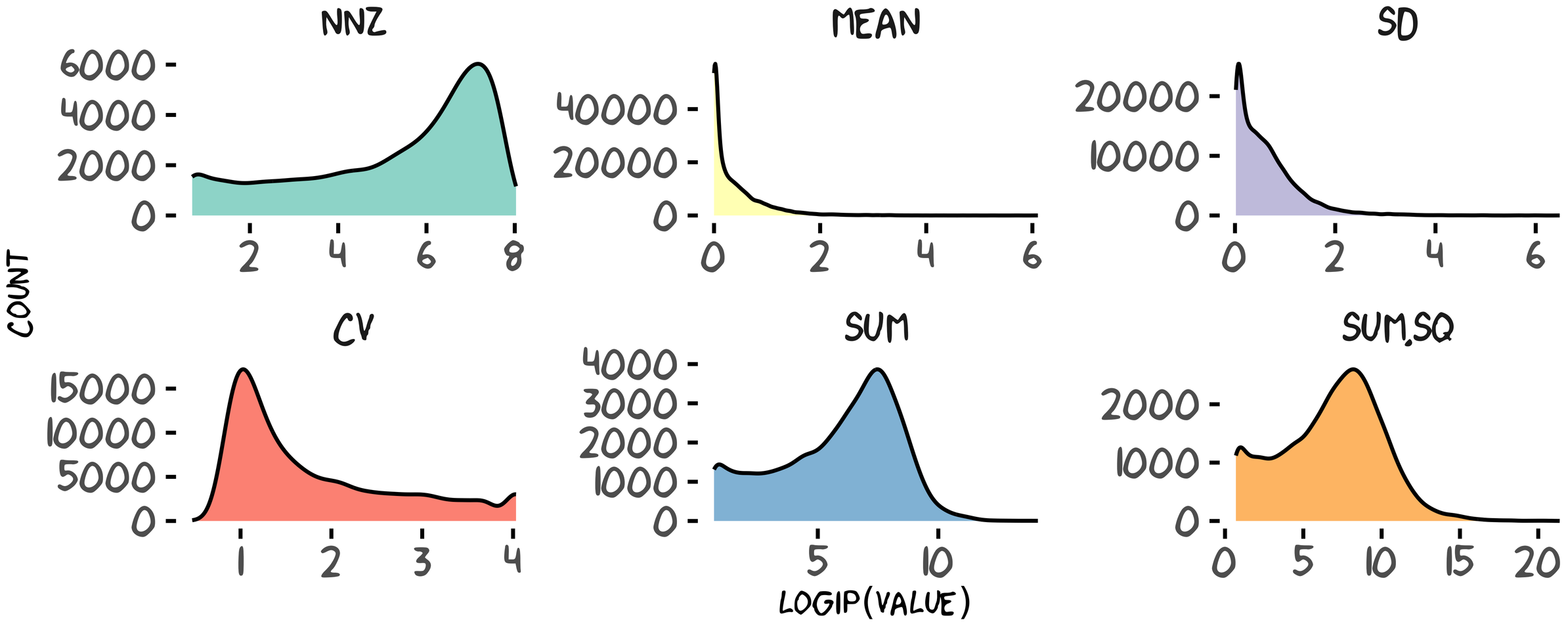
- Is it okay to normalize to have both samples have the same sequence depth?
- Are we comparing the gene “red” with the gene “green” or vice versa?

QC 2. Can we select informative rows/genes?



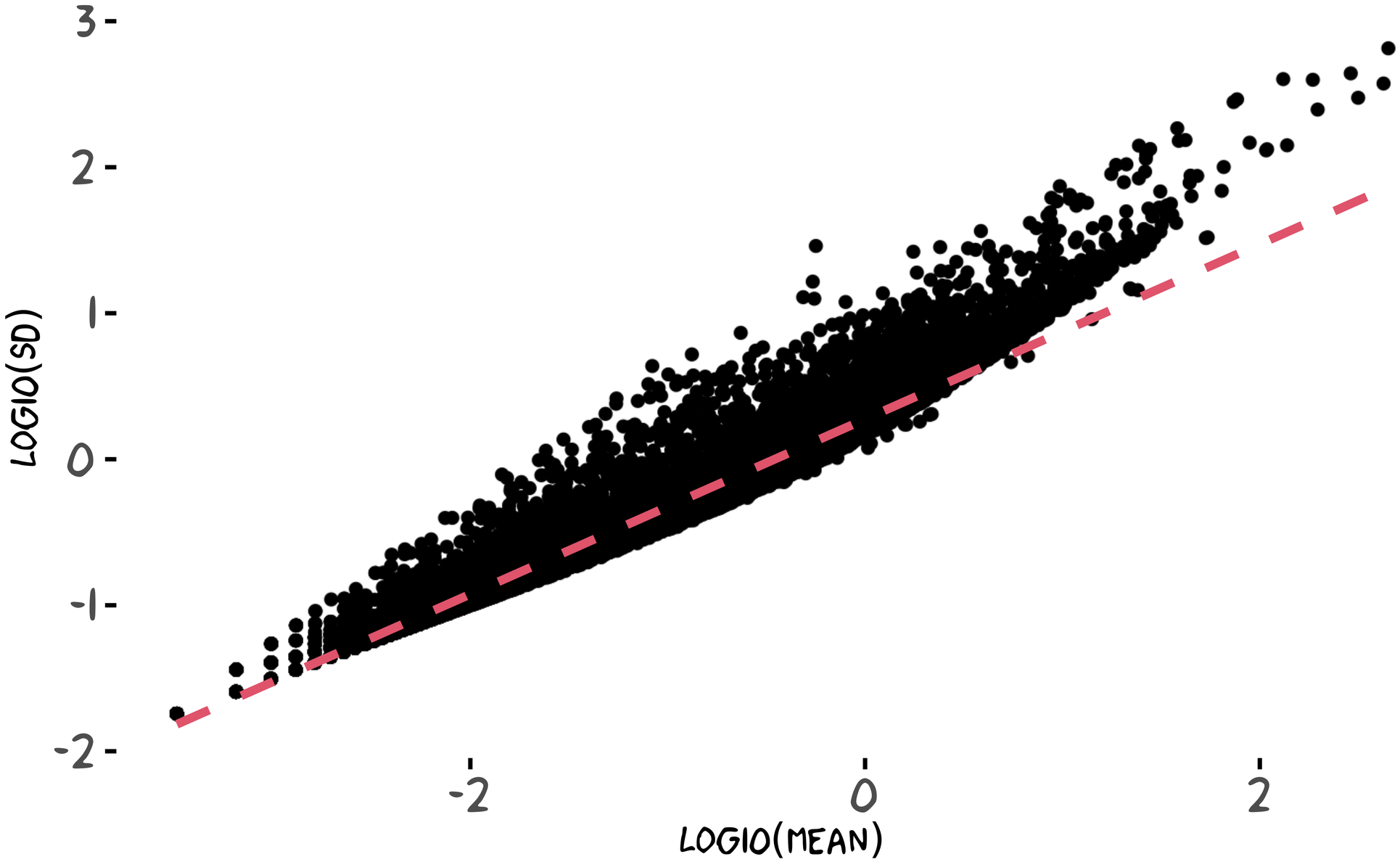
‘nnz‘: number of non-zero elements, ‘sd‘: standard deviation, ‘cv‘: coefficient of variation (‘sd‘/‘mean‘), ‘sum.sq‘: sum of squares.

QC 2. Can we select informative rows/genes?

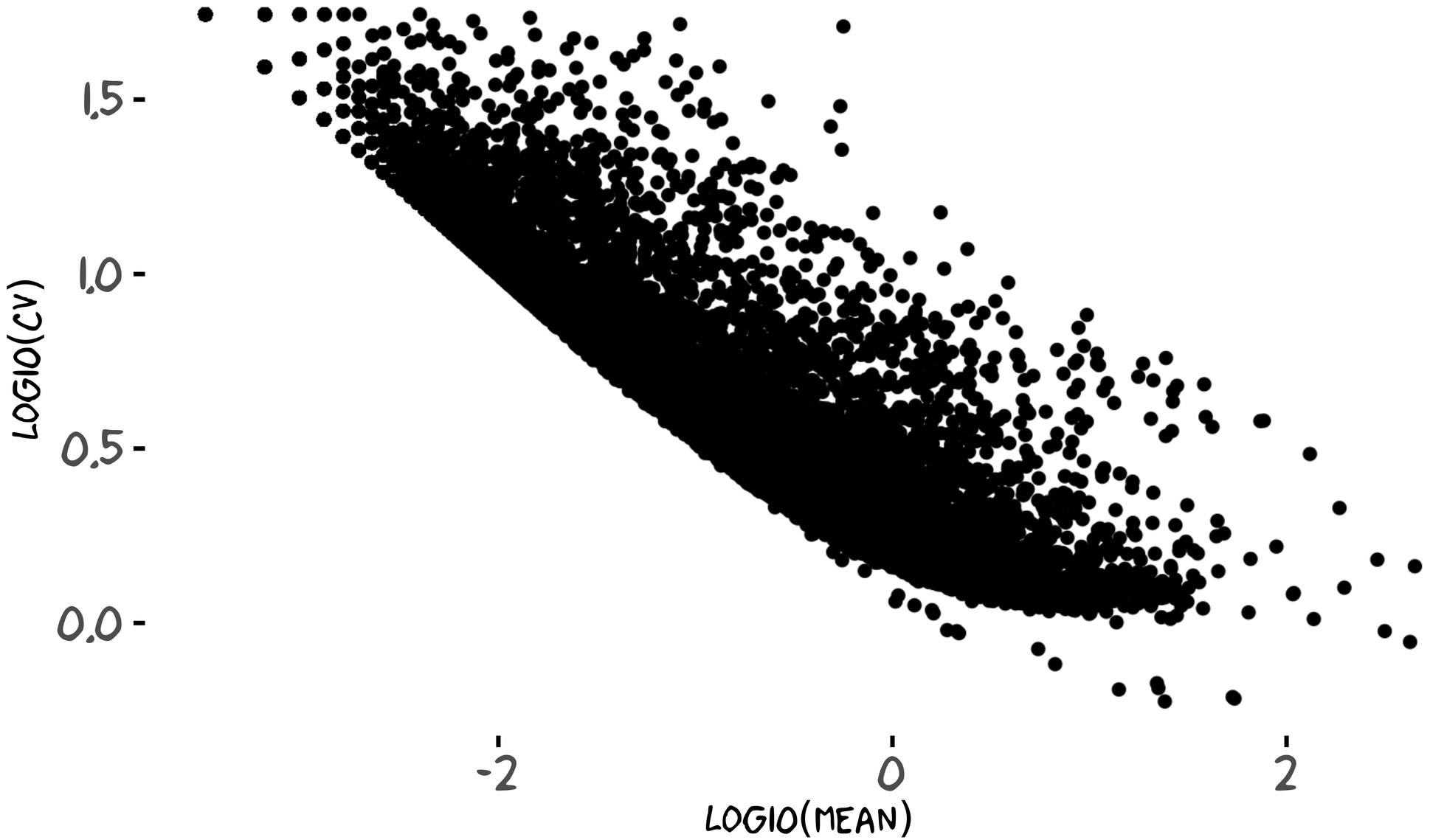


We may select only highly variable genes for the subsequent analysis

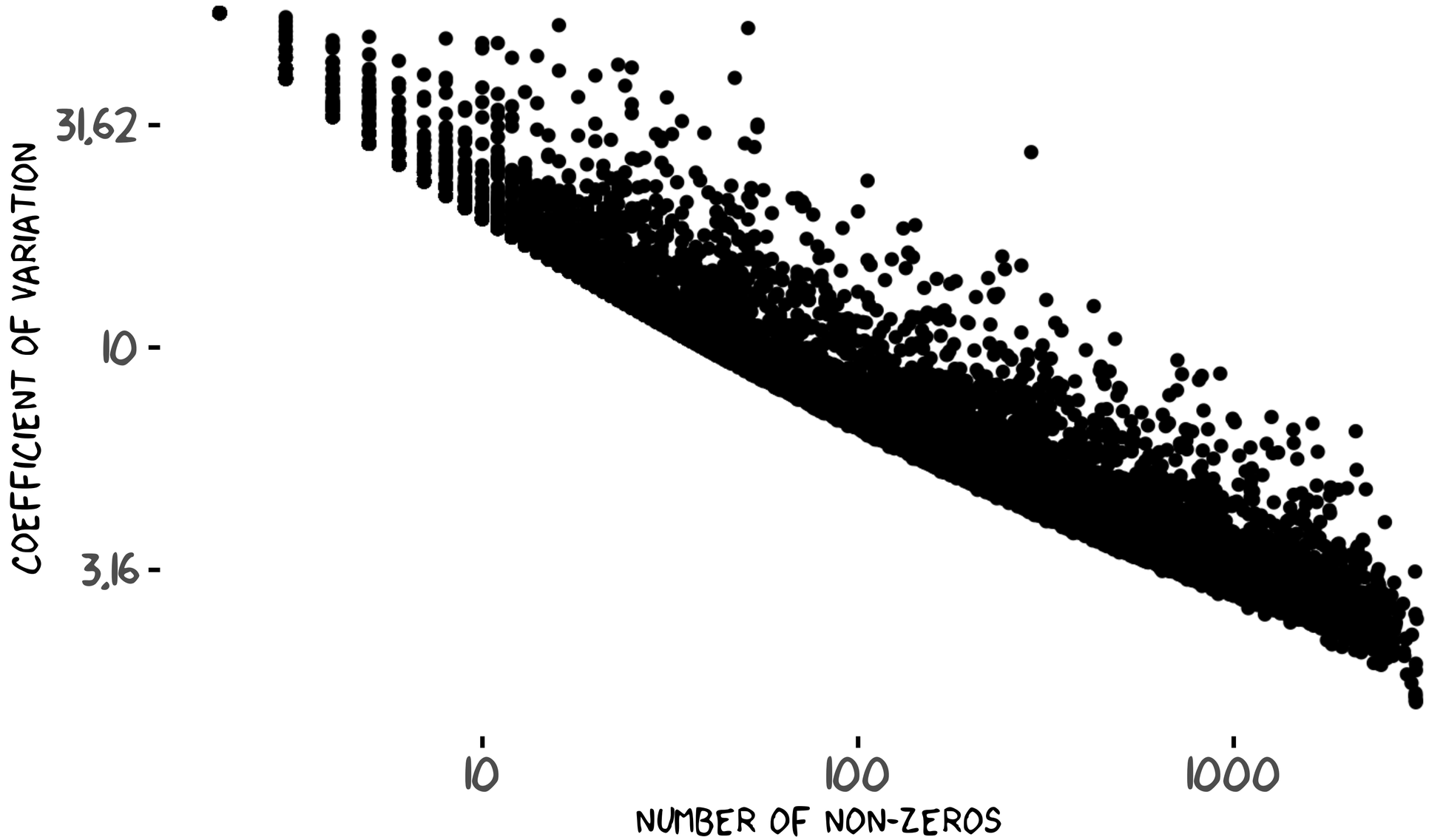
Mean-variance relationship of genes



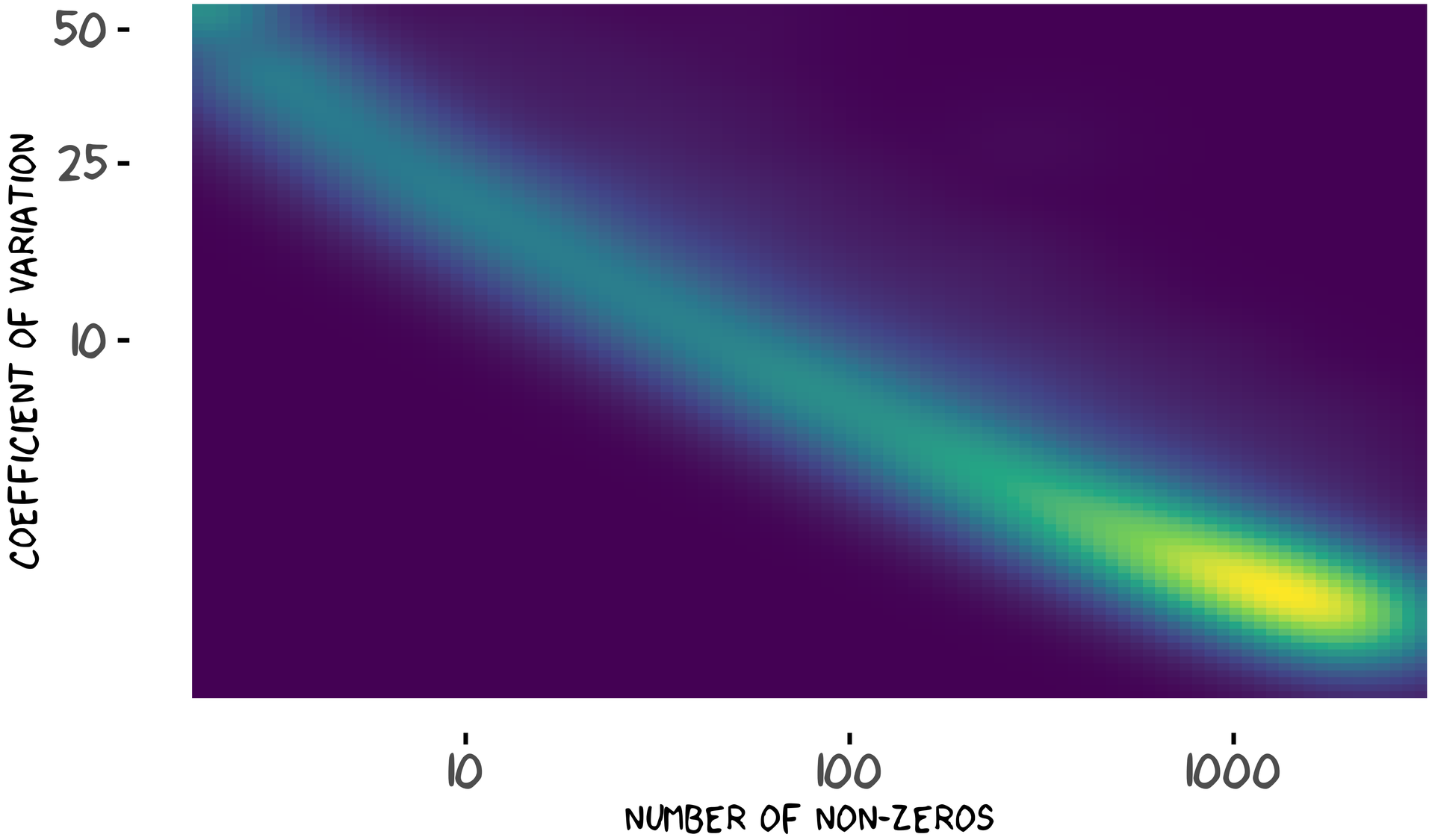
Mean-variance relationship of genes



Gene Q/C: Any obvious cutoff?



Gene Q/C: Any obvious cutoff?



Can we use the count data as they are?

- In bulk RNA-seq, we transformed the count data to adjust size factors that vary across samples.
- In single-cell data analysis, we generally care a lot about zero values, not knowing whether they come from actual “no expression” or technical drop-out.
- Should we take some sort of `voom` transformation?

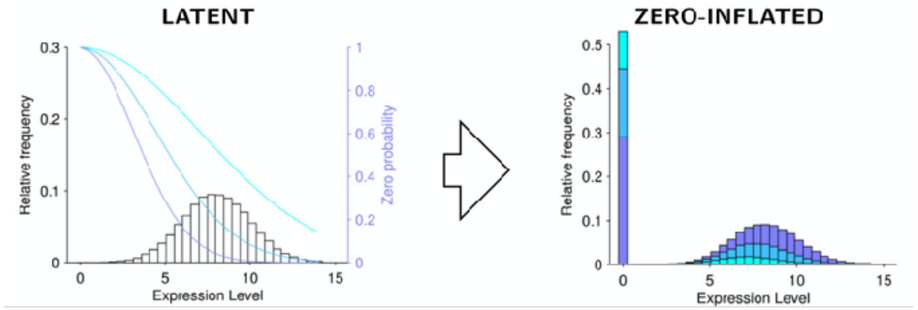
Digression

Single-cell data is noisy but…

Lots of zeros…

ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis

Emma Pierson[1] and Christopher Yau[1,2*]



2015

ZI-Poisson

A general and flexible method for signal extraction from single-cell RNA-seq data

Davide Risso[1], Fanny Perraudeau[2], Svetlana Gribkova[3], Sandrine Dudoit[2,4] & Jean-Philippe Vert[5,6,7,8]

2018

ZI-NB

Deep generative modeling for single-cell transcriptomics

Romain Lopez[1], Jeffrey Regier[1], Michael B. Cole[2], Michael I. Jordan[1,3] and Nir Yosef[1,4,5*]

2019

ZI-VAE

Digression

Don’t over-model it!

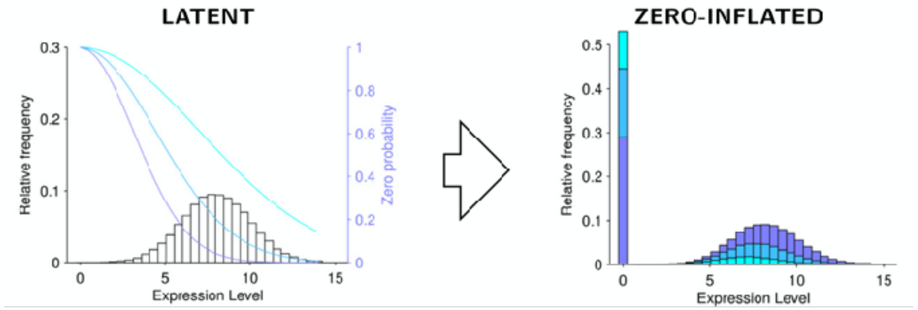
E.g., zero-inflation in scRNA-seq

ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis

2015

ZI-Poisson

Emma Pierson[1] and Christopher Yau[1,2*]



A general and flexible method for signal extraction from single-cell RNA-seq data

2018

ZI-NB

Davide Risso[1], Fanny Perraudeau[2], Svetlana Gribkova[3], Sandrine Dudoit[2,4] & Jean-Philippe Vert[5,6,7,8]

Deep generative modeling for single-cell transcriptomics

2019

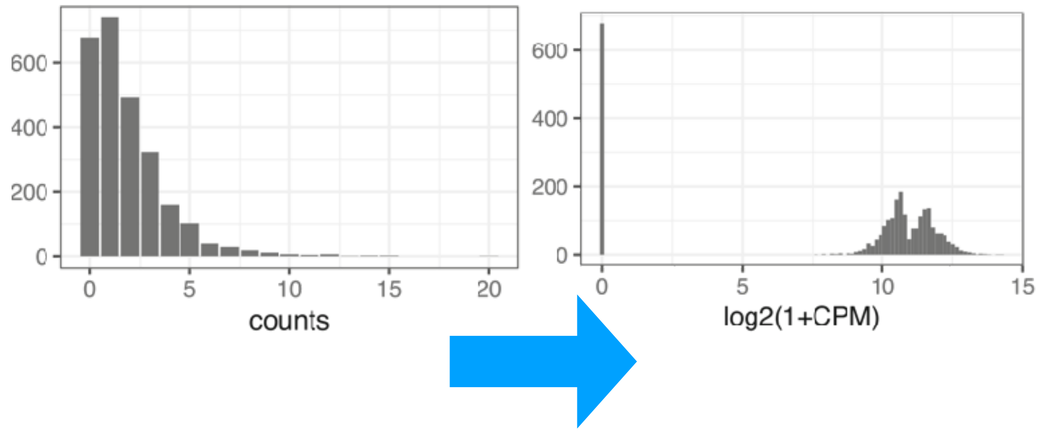
ZI-VAE

Romain Lopez[1], Jeffrey Regier[1], Michael B. Cole[2], Michael I. Jordan[1,3] and Nir Yosef[1,4,5*]

Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model

2019

F. William Townes[1,2], Stephanie C. Hicks[3], Martin J. Aryee[1,4,5,6] and Rafael A. Irizarry[1,7*]



Digression

Don’t over-model it!

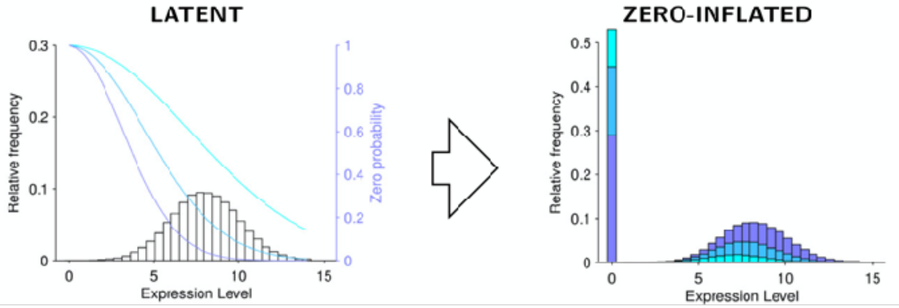
E.g., zero-inflation in scRNA-seq

ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis

2015

ZI-Poisson

Emma Pierson[1] and Christopher Yau[1,2*]



A general and flexible method for signal extraction from single-cell RNA-seq data

2018

ZI-NB

Davide Risso[1], Fanny Perraudeau[2], Svetlana Gribkova[3], Sandrine Dudoit[2,4] & Jean-Philippe Vert[5,6,7,8]

Deep generative modeling for single-cell transcriptomics

2019

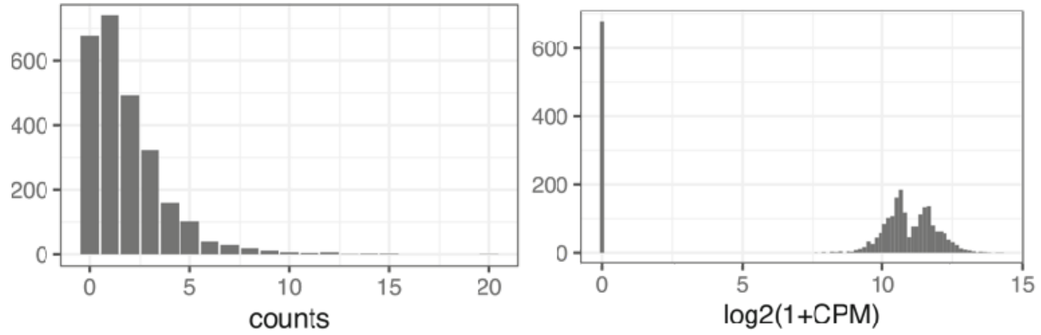
ZI-VAE

Romain Lopez[1], Jeffrey Regier[1], Michael B. Cole[2], Michael I. Jordan[1,3] and Nir Yosef[1,4,5*]

Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model

2019

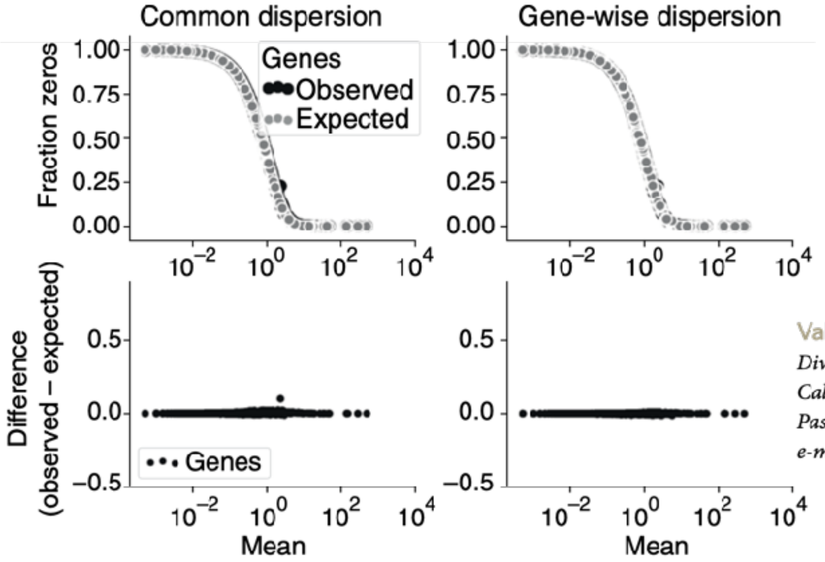
F. William Townes[1,2], Stephanie C. Hicks[3], Martin J. Aryee[1,4,5,6] and Rafael A. Irizarry[1,7*]



Droplet scRNA-seq is not zero-inflated

2020

Solution (brain endogenous RNA and ERCC spike-ins)



Valentine Svensson
Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, CA, USA.
e-mail: v@nxn.se

Simple transformation will work fine

nature methods

Analysis

https://doi.org/10.1038/s41592-023-01814-1

Comparison of transformations for single-cell RNA-seq data

Received: 25 August 2021

Accepted: 11 February 2023

Published online: 10 April 2023

Check for updates

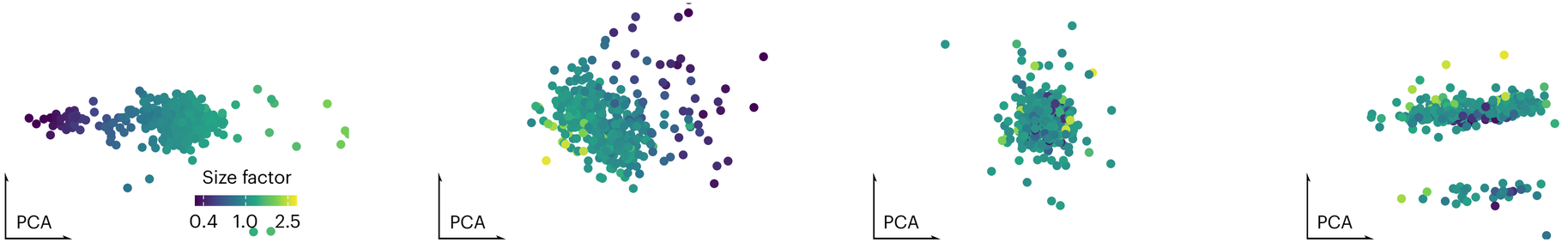
Constantin Ahlmann-Eltze [1,2] ✉ & Wolfgang Huber [1]

The count table, a numeric matrix of genes × cells, is the basic input data structure in the analysis of single-cell RNA-sequencing data. A common preprocessing step is to adjust the counts for variable sampling efficiency

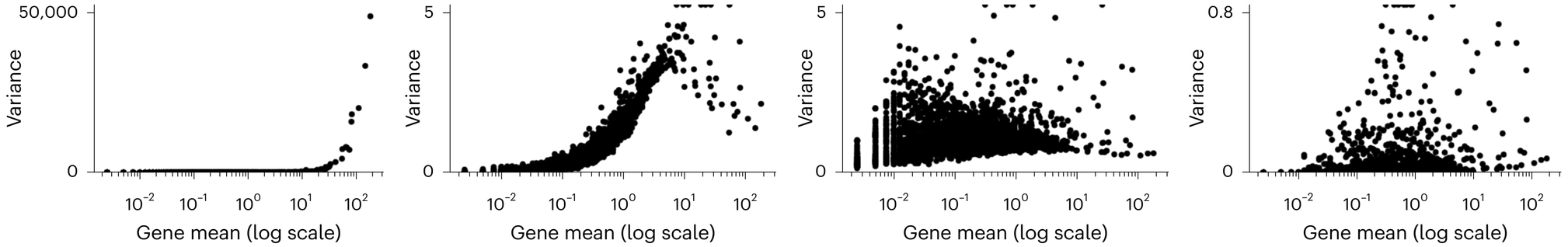
log(x + 1) robust, little affected by size factor

Raw counts	Delta method	GLM residual	Latent expression
Y	$\log(Y/s + 1)$	$\frac{Y - \mu}{\sqrt{\mu + \alpha\mu^2}}$	$Y \sim \text{Poisson}(M)$ $M \sim \text{logNormal}(\mu, \sigma^2)$

a Confounding effect of size factors on PCA embedding of droplets encapsulating a homogeneous RNA solution



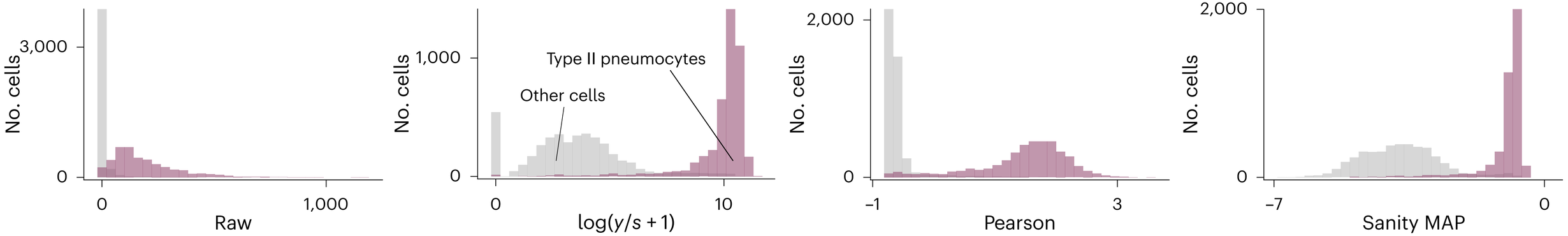
b Mean–variance relation for 2,597 genes of the 10x hematopoietic cell dataset

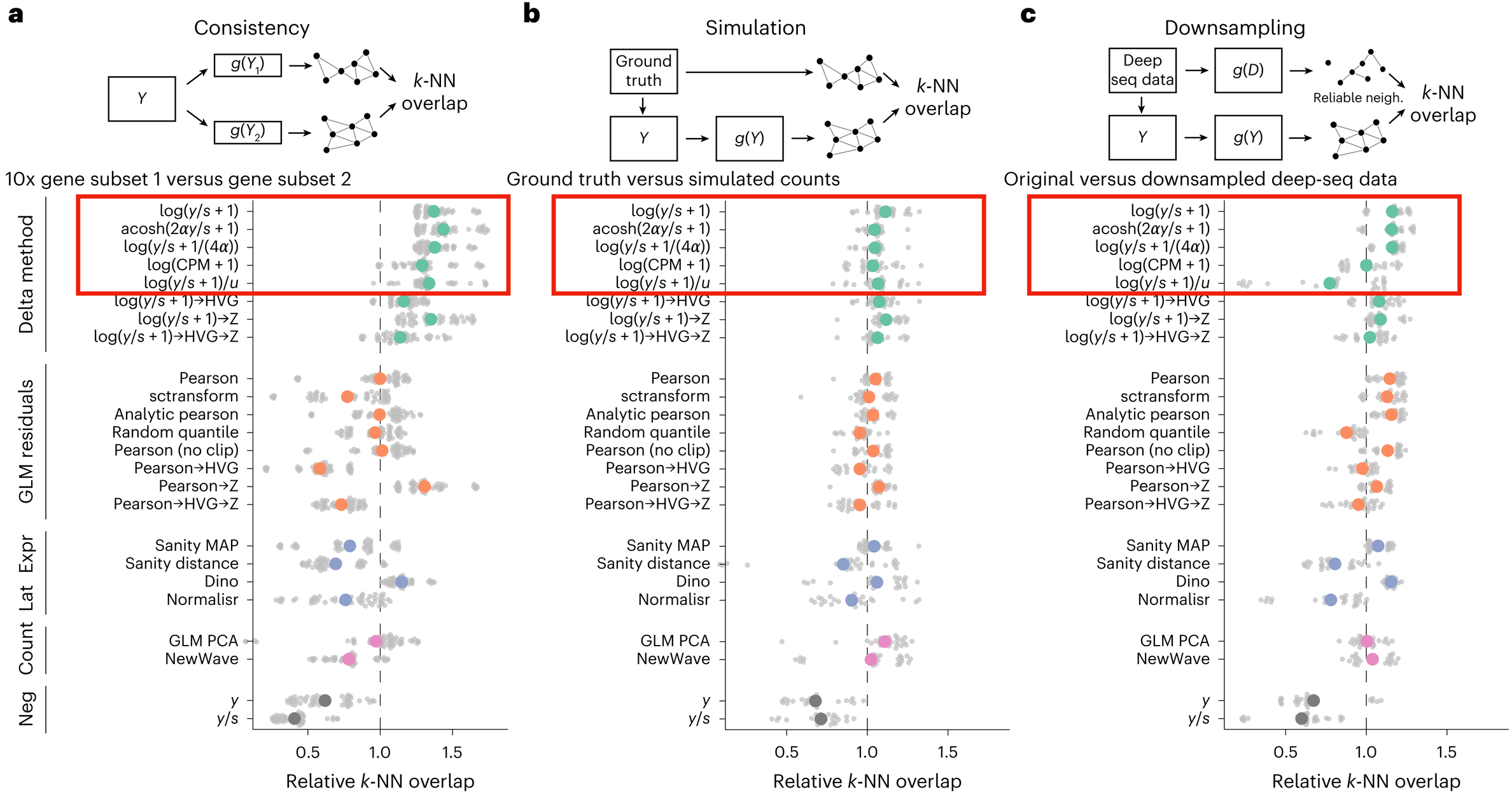


log(x + 1) robust, little affected by size factor

$$Y \qquad \log(Y/s + 1) \qquad \frac{Y - \mu}{\mu + \alpha\mu^2} \qquad Y \sim \text{Poisson}(\lambda) \quad \log\lambda \sim \mathcal{N}(\mu, \sigma^2)$$

c Distribution of a single gene (*Sftpc*) with a bimodal expression pattern in lung epithelium

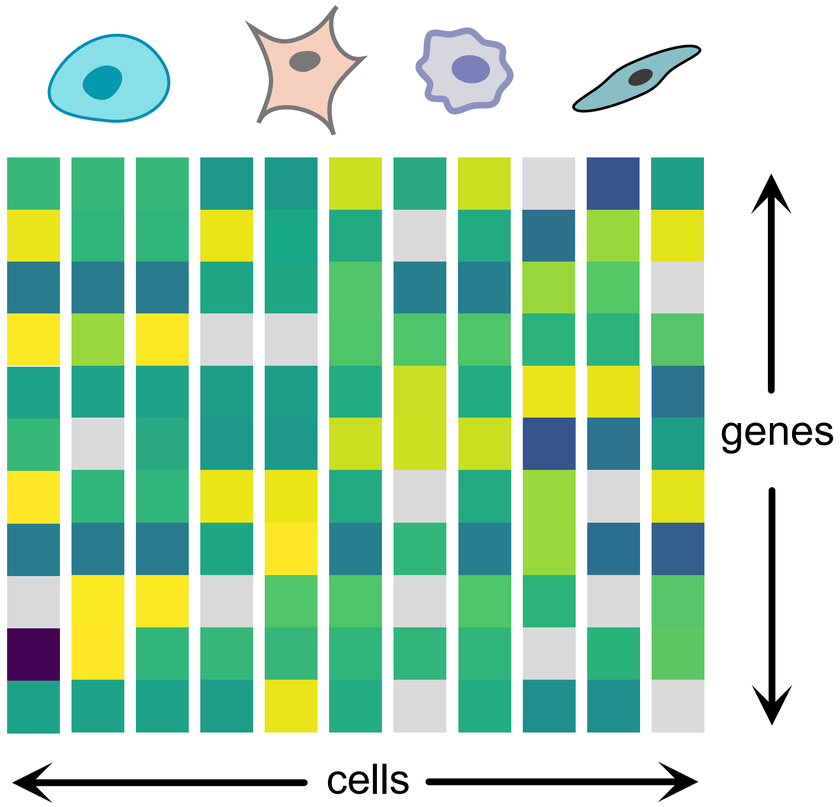



a
Consistency
Y
g(Y1)
g(Y2)
k-NN overlap
10x gene subset 1 versus gene subset 2
b
Simulation
Ground truth
Y
g(Y)
k-NN overlap
Ground truth versus simulated counts
c
Downsampling
Deep seq data
g(D)
Reliable neigh.
Y
g(Y)
k-NN overlap
Original versus downsampled deep-seq data
Delta method
GLM residuals
Expr
Lat
Count
Neg
log(y/s + 1)
acosh(2αy/s + 1)
log(y/s + 1/(4α))
log(CPM + 1)
log(y/s + 1)/u
log(y/s + 1)→HVG
log(y/s + 1)→Z
log(y/s + 1)→HVG→Z
Pearson
sctransform
Analytic pearson
Random quantile
Pearson (no clip)
Pearson→HVG
Pearson→Z
Pearson→HVG→Z
Sanity MAP
Sanity distance
Dino
Normalisr
GLM PCA
NewWave
y
y/s
0.5
1.0
1.5
Relative k-NN overlap

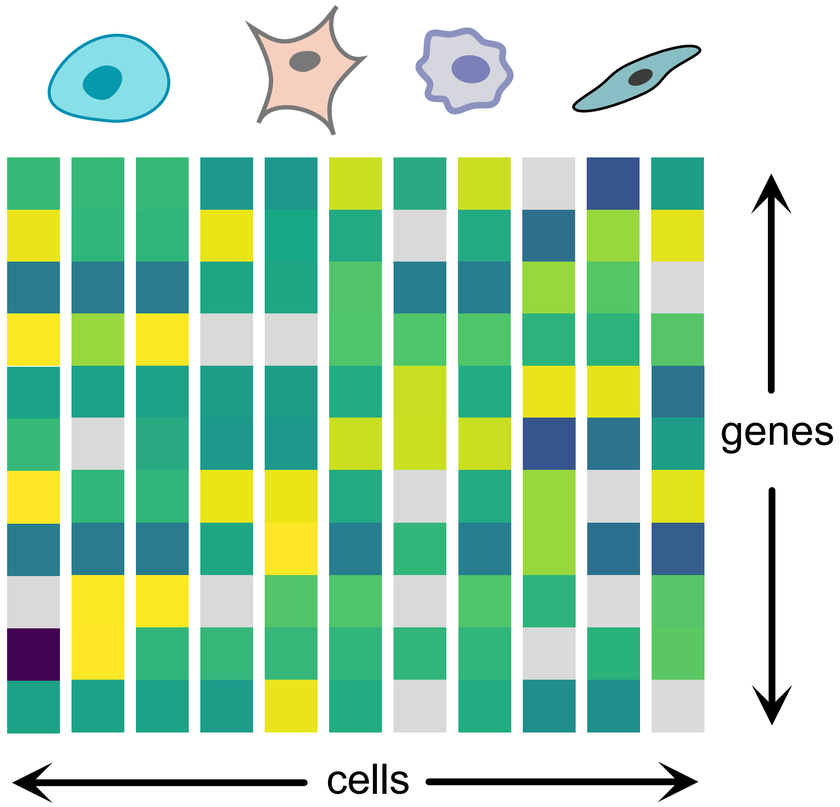
Today’s lecture

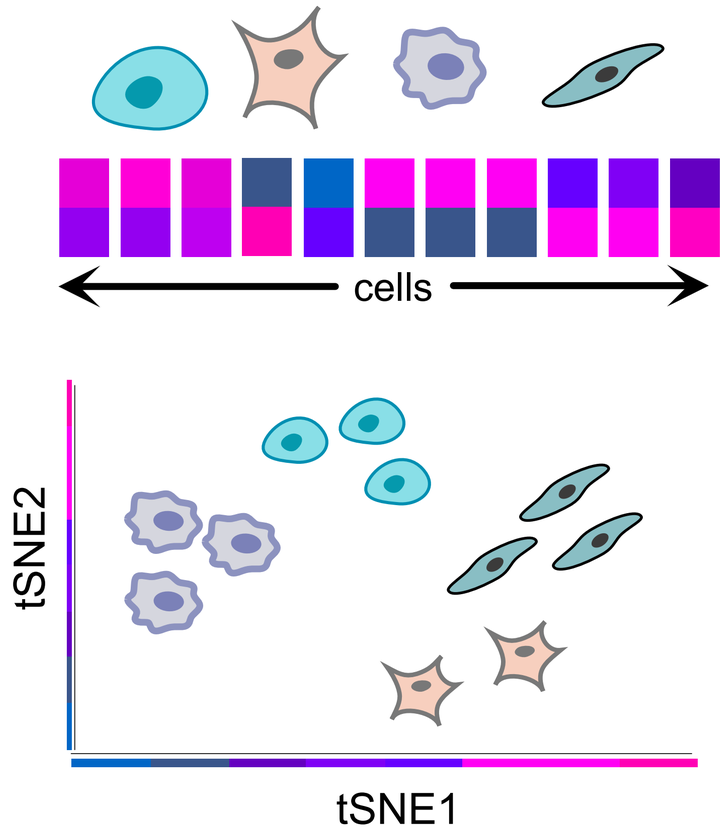
1. Single-cell sequencing technology
2. Basic quality control
3. Additional Q/C tools
4. Doublet detection in single-cell data
5. Data normalization across many batches

It’s useful to show high-dimensional data in 2D



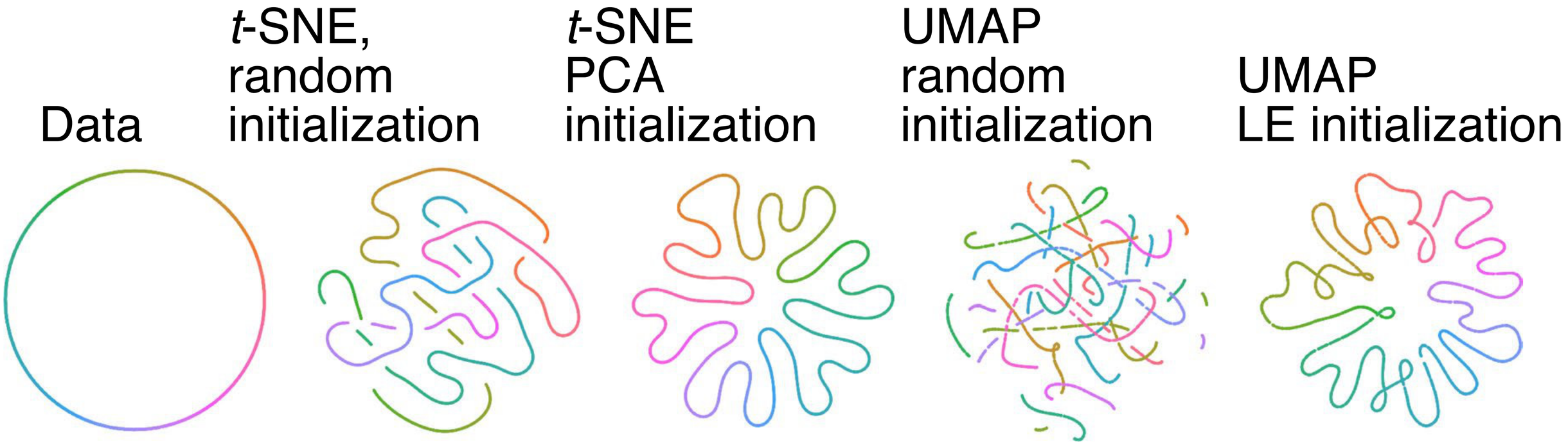
It’s useful to show high-dimensional data in 2D





`tSNE`: t-distributed Stochastic Neighbourhood Embedding (Van der Maaten & Hinton, 2008).

Warning: Don’t over interpret 2D embedding



Kobak and Berens, *Nature Biotech* (2021)

Don’t over interpret 2D embedding results

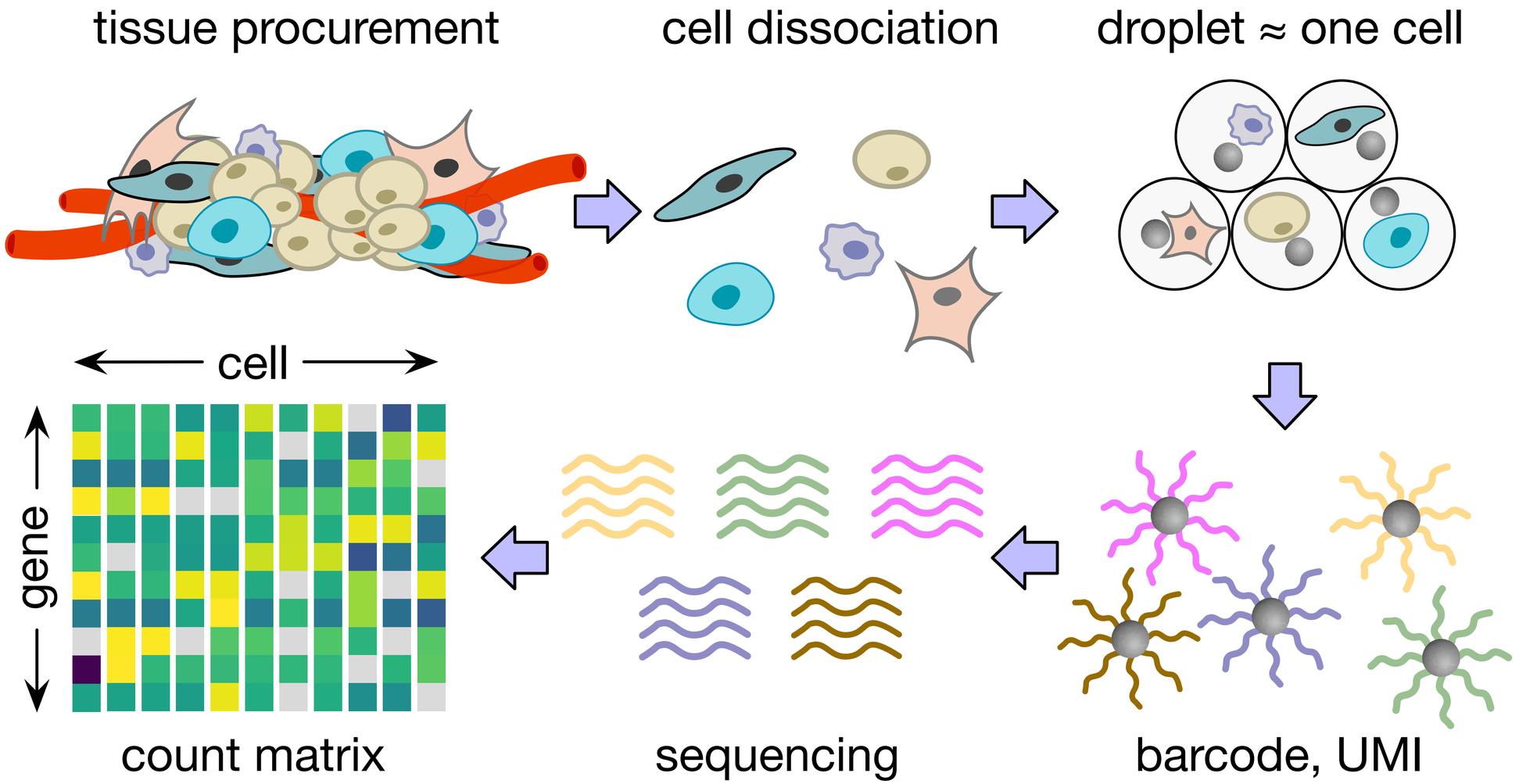
Check out these papers:

- Initialization is critical for preserving global data structure in both t-SNE and UMAP
- The art of using t-SNE for single-cell transcriptomics
- Dimensionality reduction for visualizing single-cell data using UMAP

Today’s lecture

1. Single-cell sequencing technology
2. Basic quality control
3. Additional Q/C tools
4. **Doublet detection in single-cell data**
5. Data normalization across many batches

What if we capture more than one cell in a droplet?



Macosko *et al.*, *Cell* (2015)

What is a doublet in single-cell data?

Biological/technical definition:

- One or more cells captured (usually at most two cells by chance)
- Thus, multiple cells accidental share the same cell barcode sequence
- Not so clear in general… since we missed the chance to assign different tags to different cells encapsulated in the same droplet.

Statistical definition:

- If we could find marker genes of multiple cell types are simultaneously expressed…
- An unvetted approach: Find ambiguous/intermediate coordinates in PCA/tSNE/UMAP (after removing ambient cells).

Can we create artificial doublets?

A straightforward definition (used in `DoubletFinder`):

For each cell i:

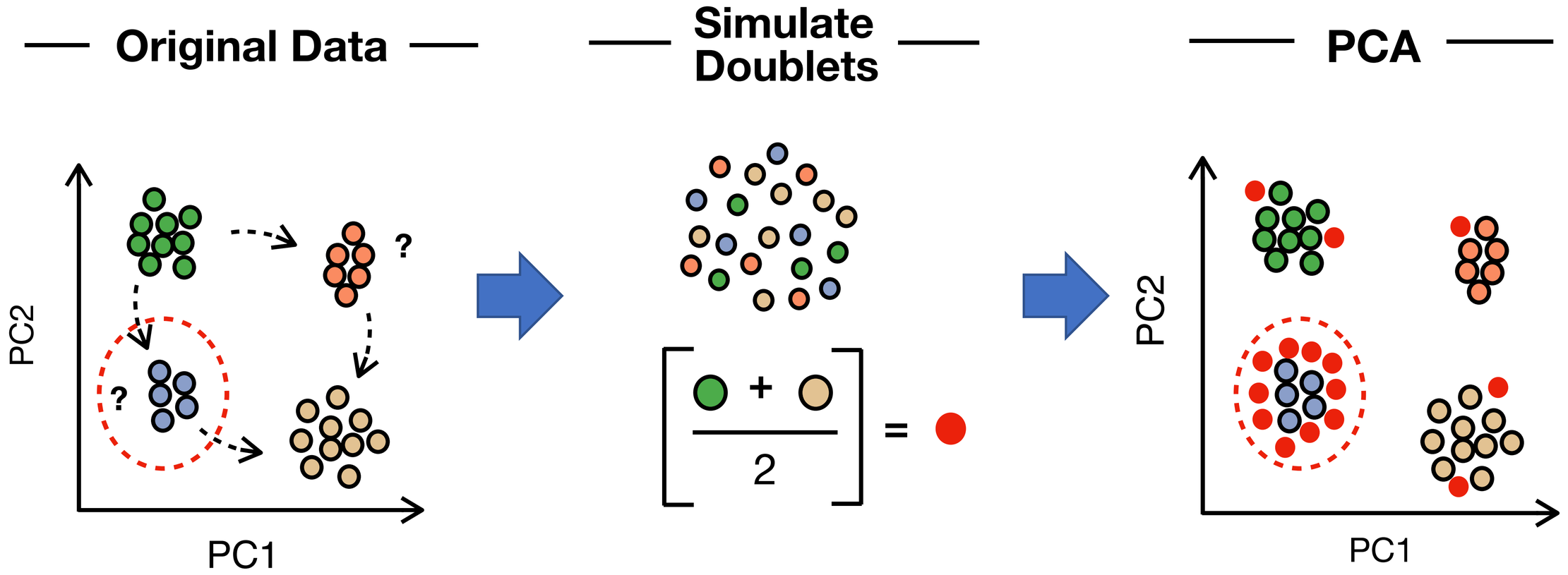
- Take some other j by random selection
- Create an artificial doublet

$$\tilde{\mathbf{x}} \leftarrow \frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j)$$

Some thought questions:

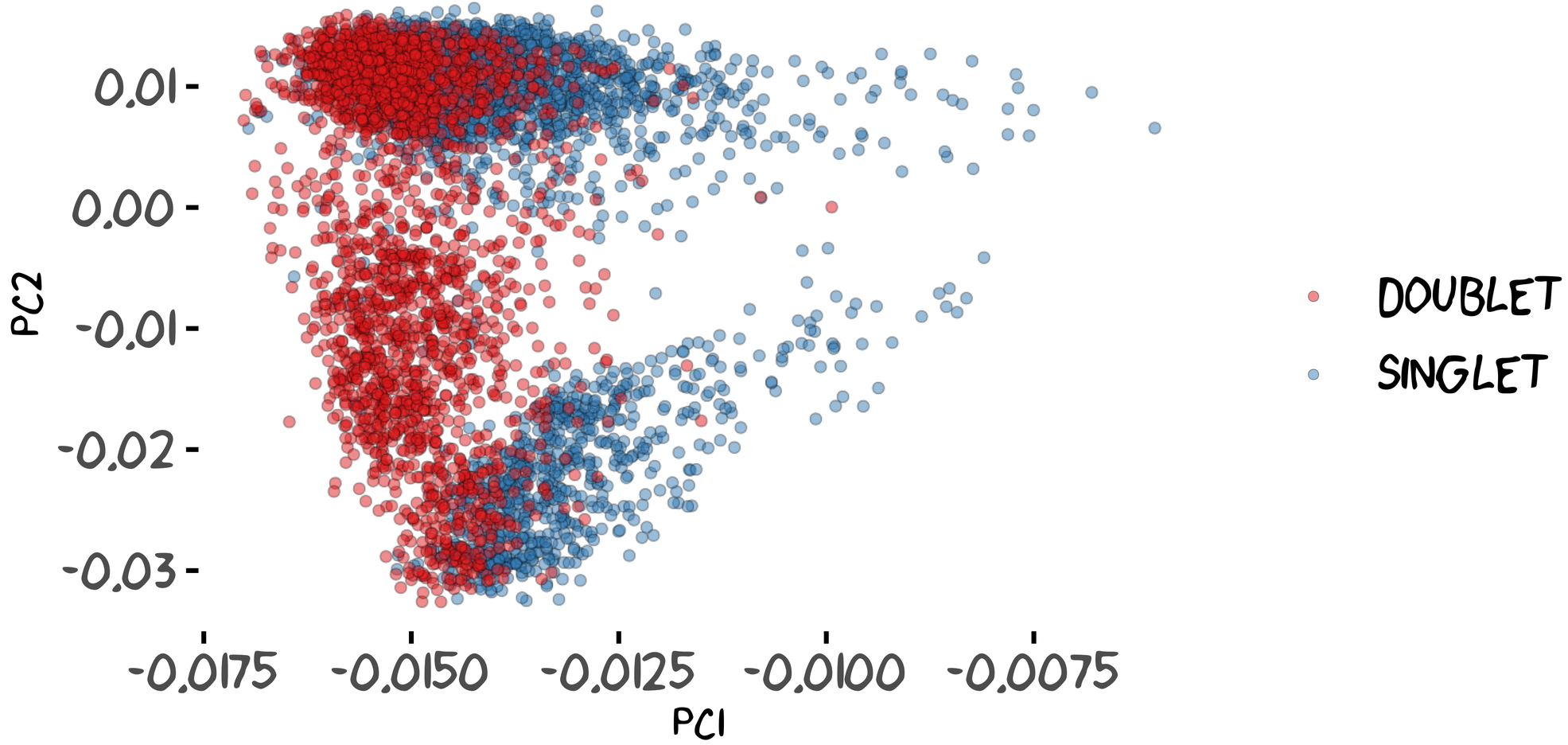
- Doublets within the same cell type?

k-Nearest Neighbour classification for doublet detection

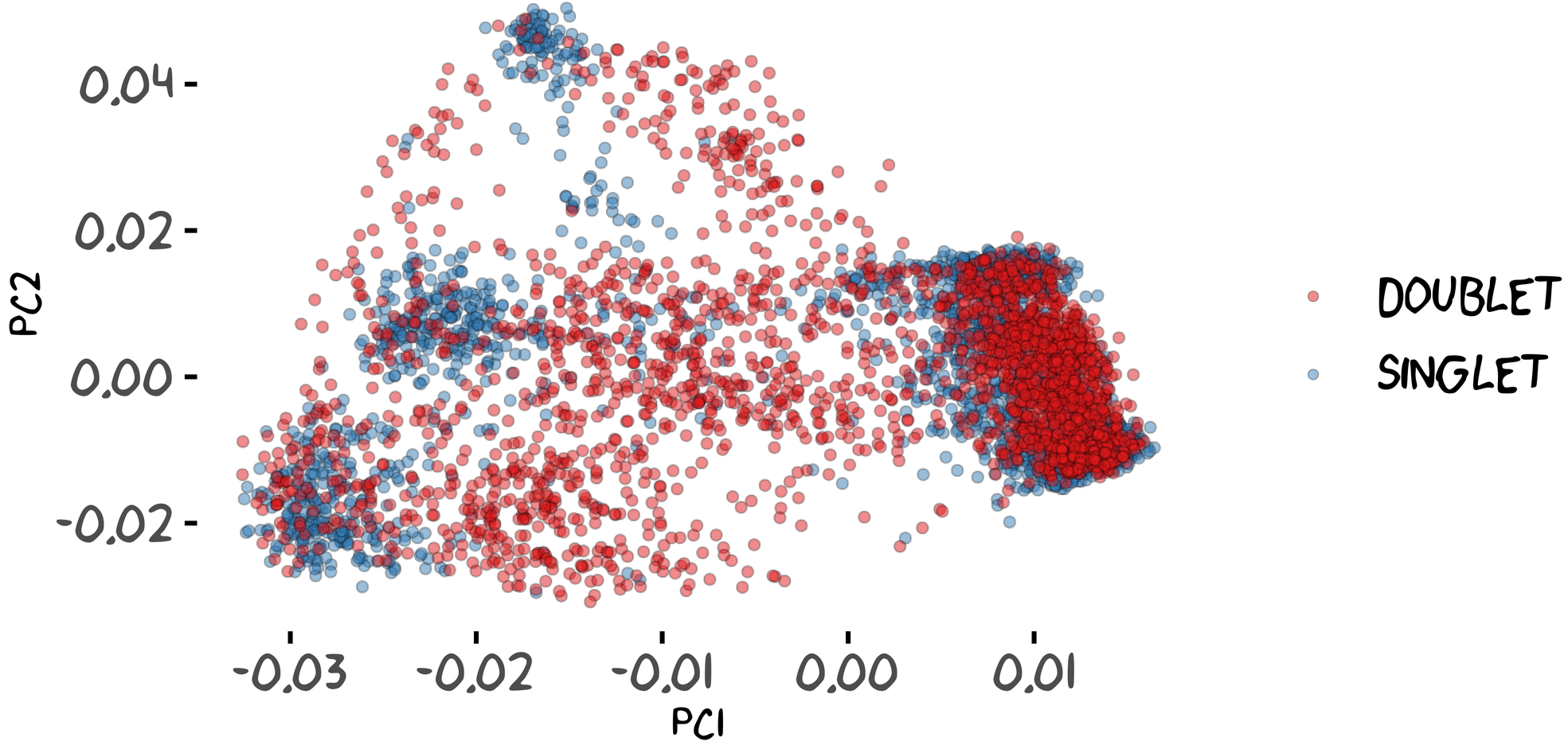


McGinnis et al. Cell Systems (2019)

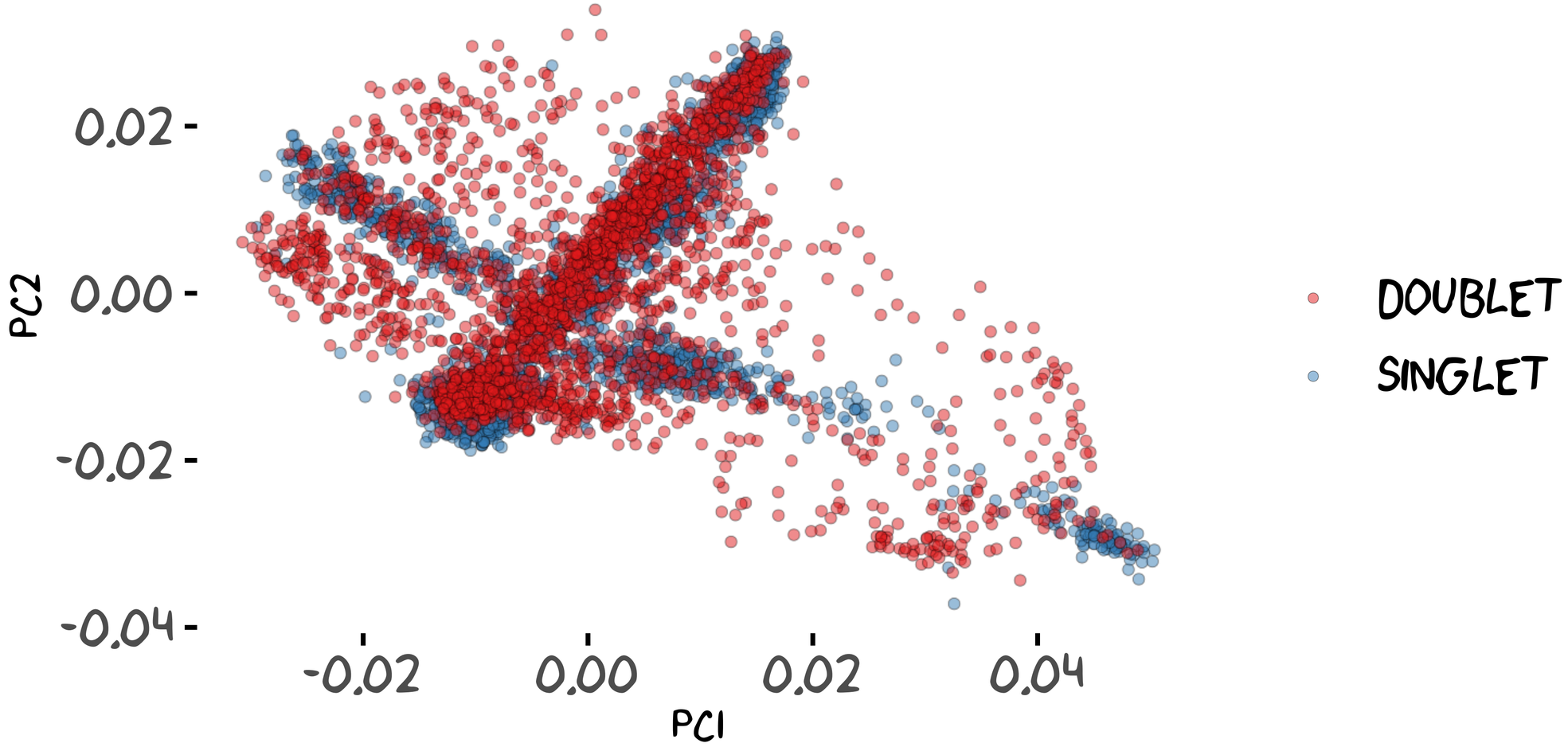
Can you tell the difference by a quick visual inspection?



Can you tell the difference by a quick visual inspection?



Can you tell the difference by a quick visual inspection?



Can we design a classifier to distinguish singlets vs. doublets?

k-Nearest Neighbour classification for doublet detection

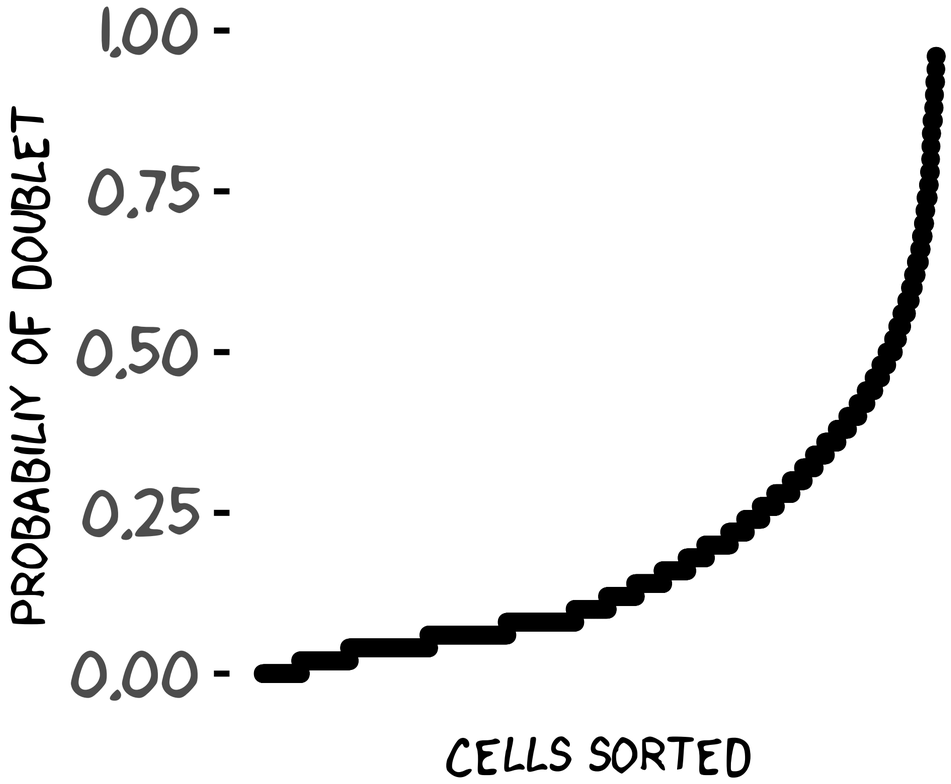
- Step 1. Create artificial doublets, $\tilde{\mathbf{x}}$
- Step 2. Mix them with the original cells and perform PCA
- Step 3. Find nearest neighbours of the original cells (using #PC=10)
- Step 4. Count the number of doublets in the neighbourhood

k-Nearest Neighbour classification of doublets

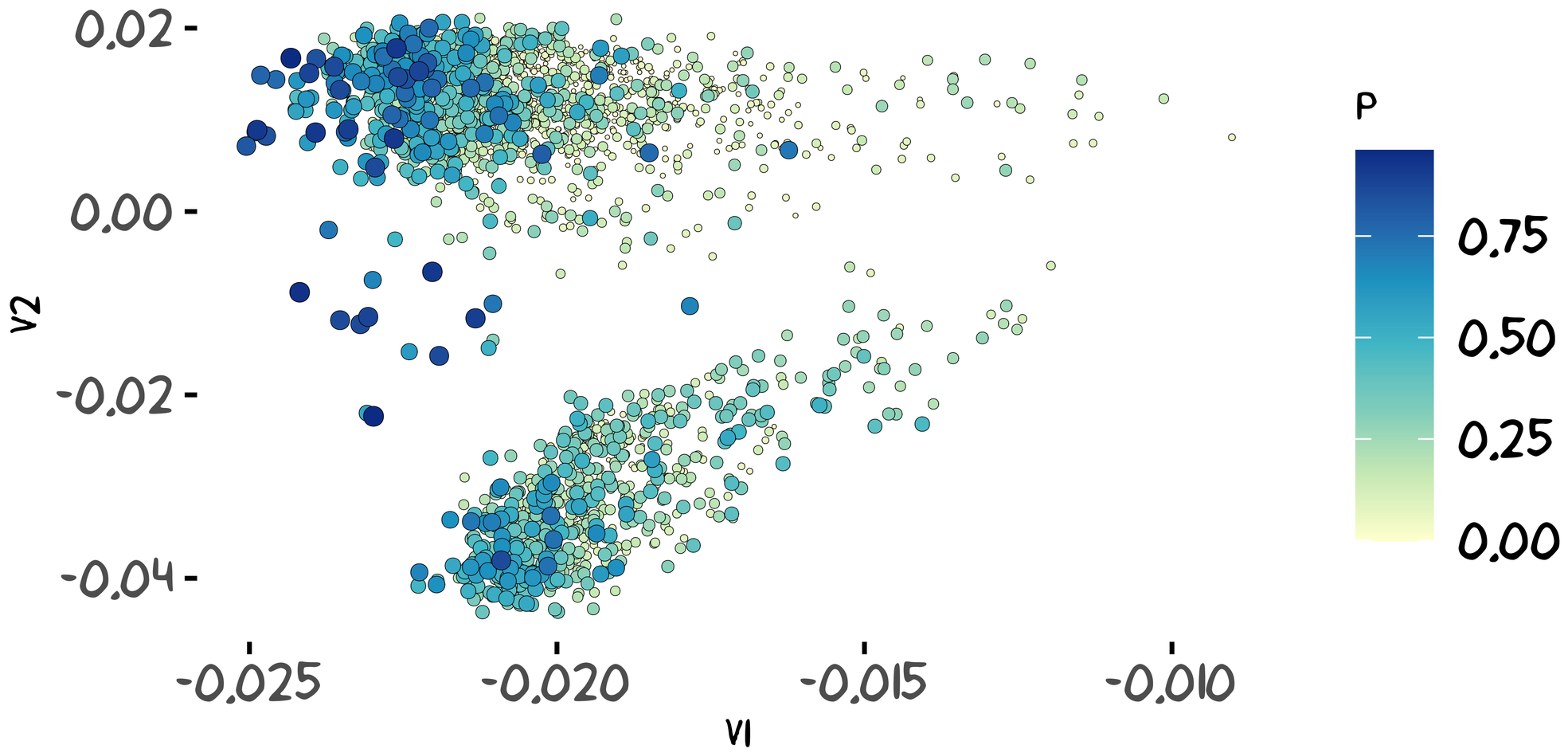
- **Q: How many of my neighbours are indeed a doublet?**

$$\hat{P}_i = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} I\{j \text{ is a doublet}\}$$

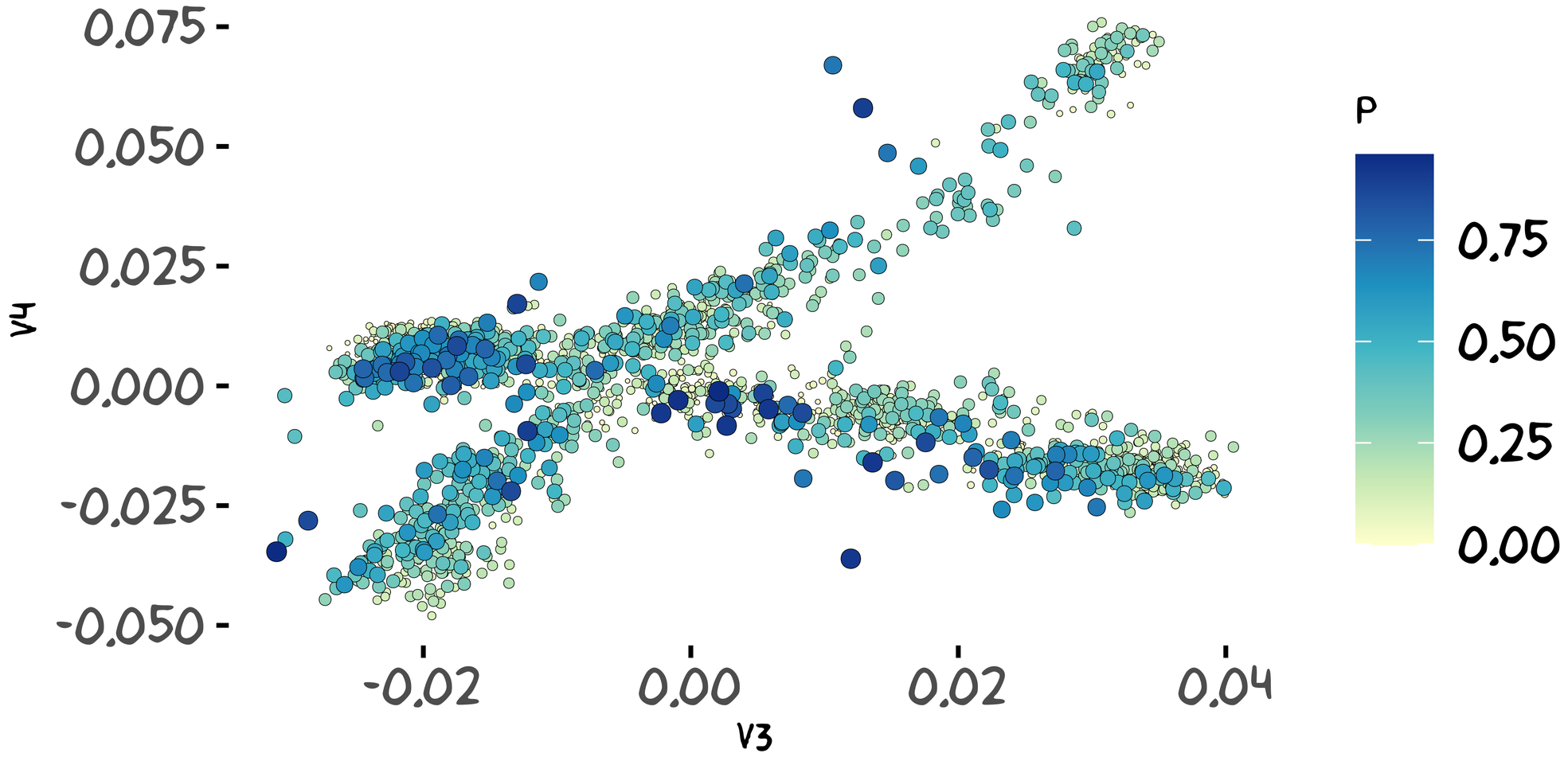
Key assumption: There is a principal component that can set apart hidden doublets from the most of singlets.



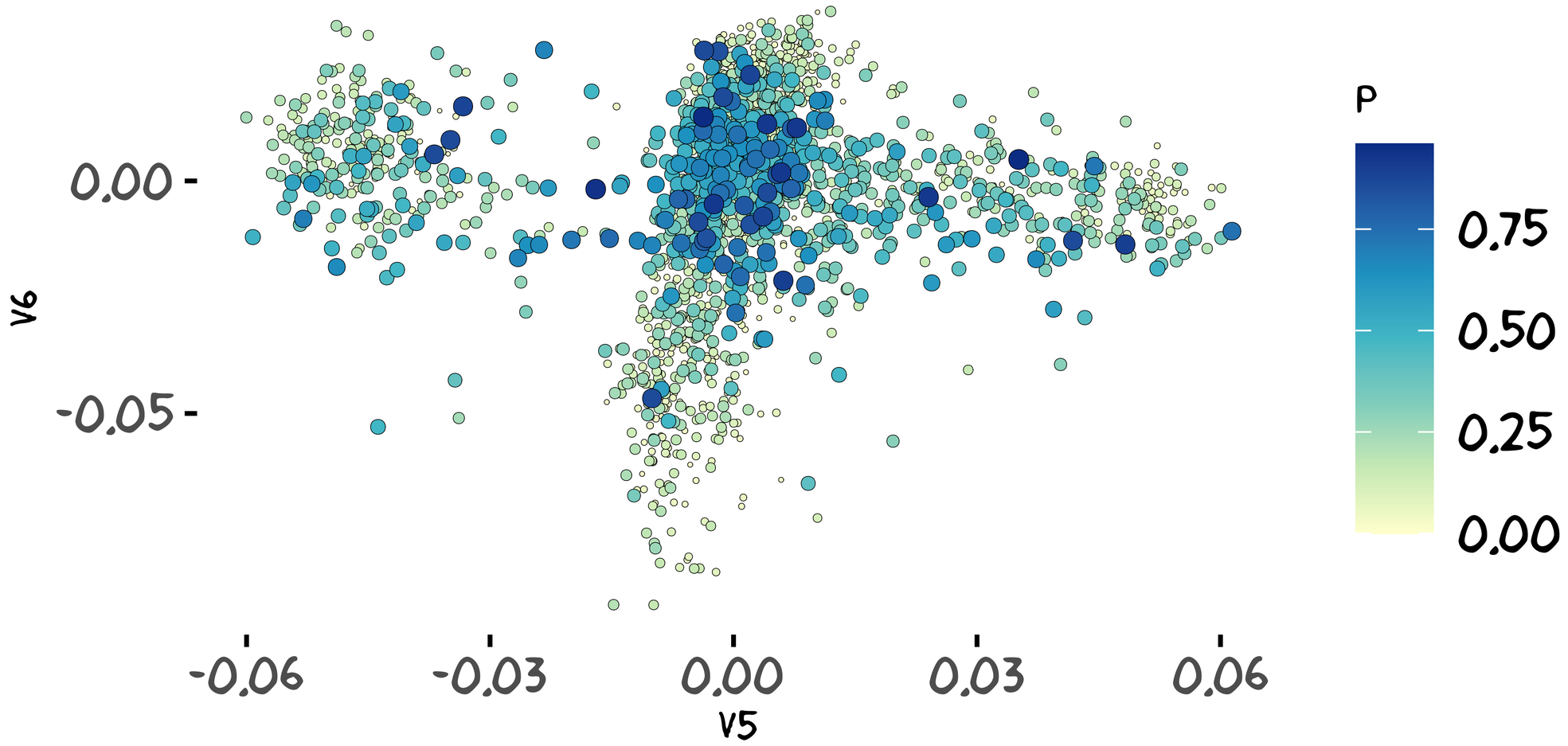
Doublets enriched in the outskirts of clusters



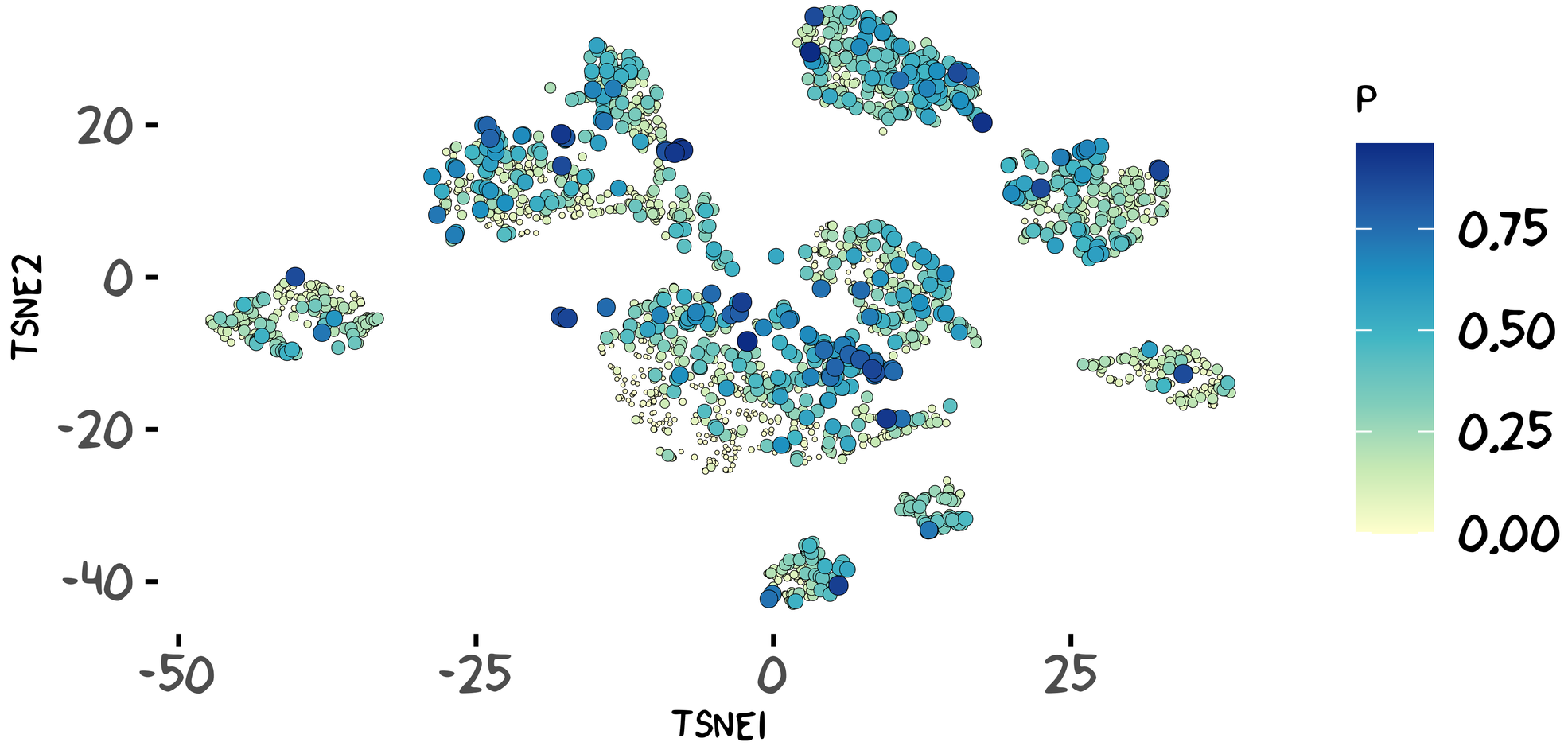
Doublets enriched in the outskirts of clusters



Doublets enriched in the outskirts of clusters



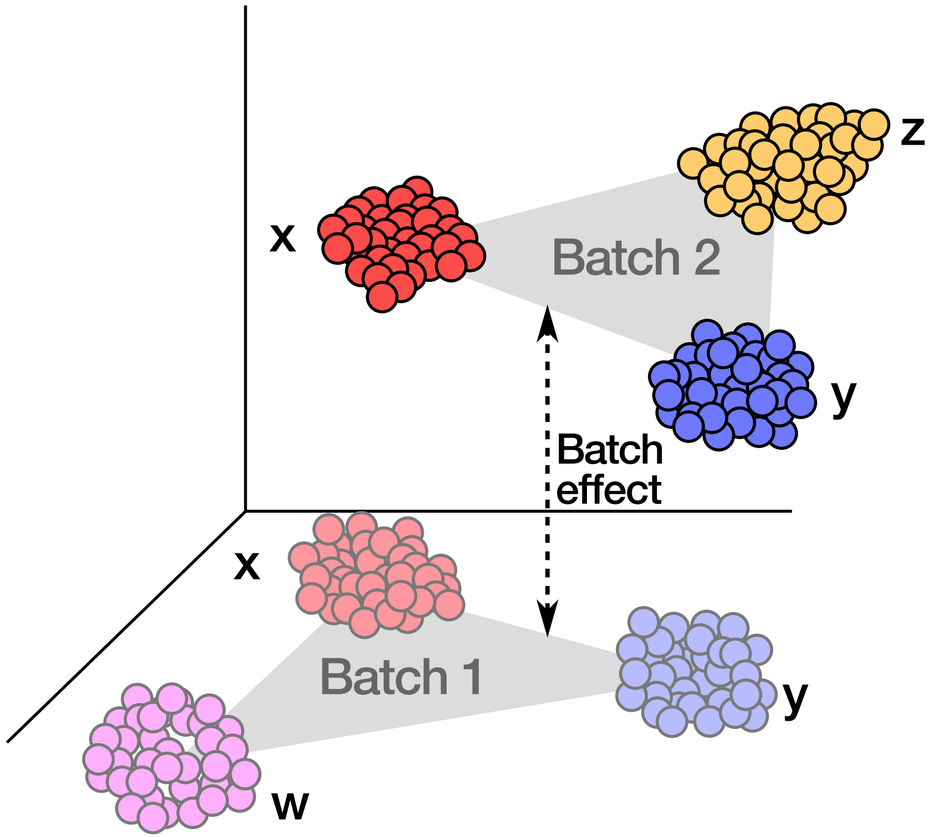
Doublets enriched in the outskirts of clusters



Today’s lecture

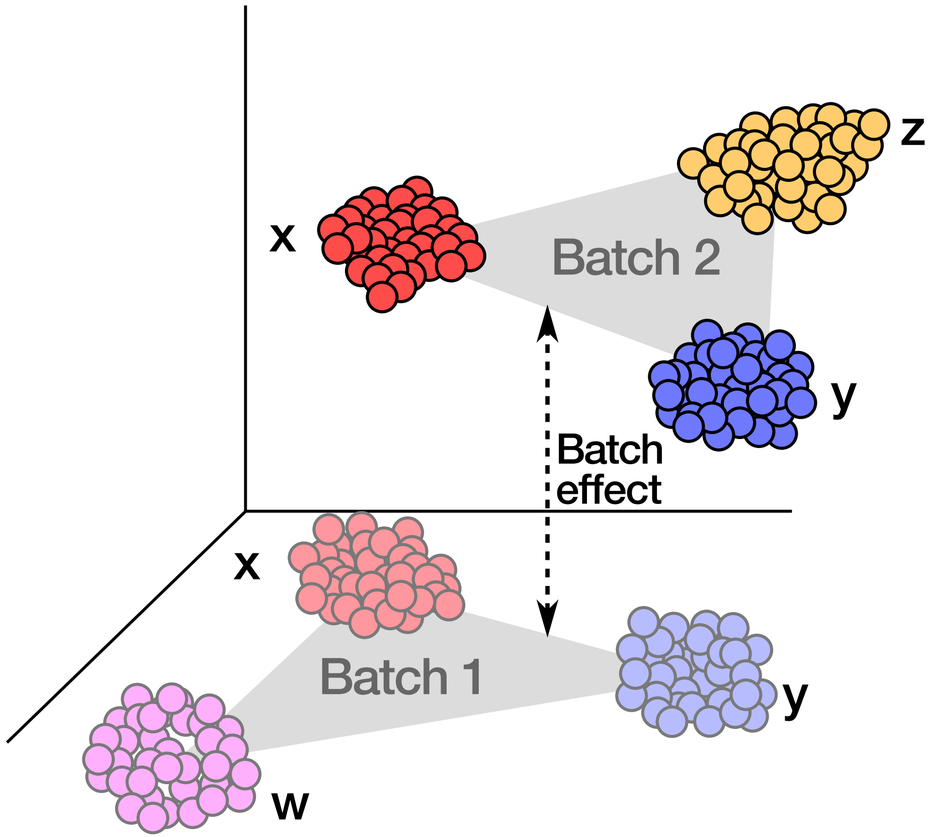
1. Single-cell sequencing technology
2. Basic quality control
3. Additional Q/C tools
4. Doublet detection in single-cell data
5. Data normalization across many batches

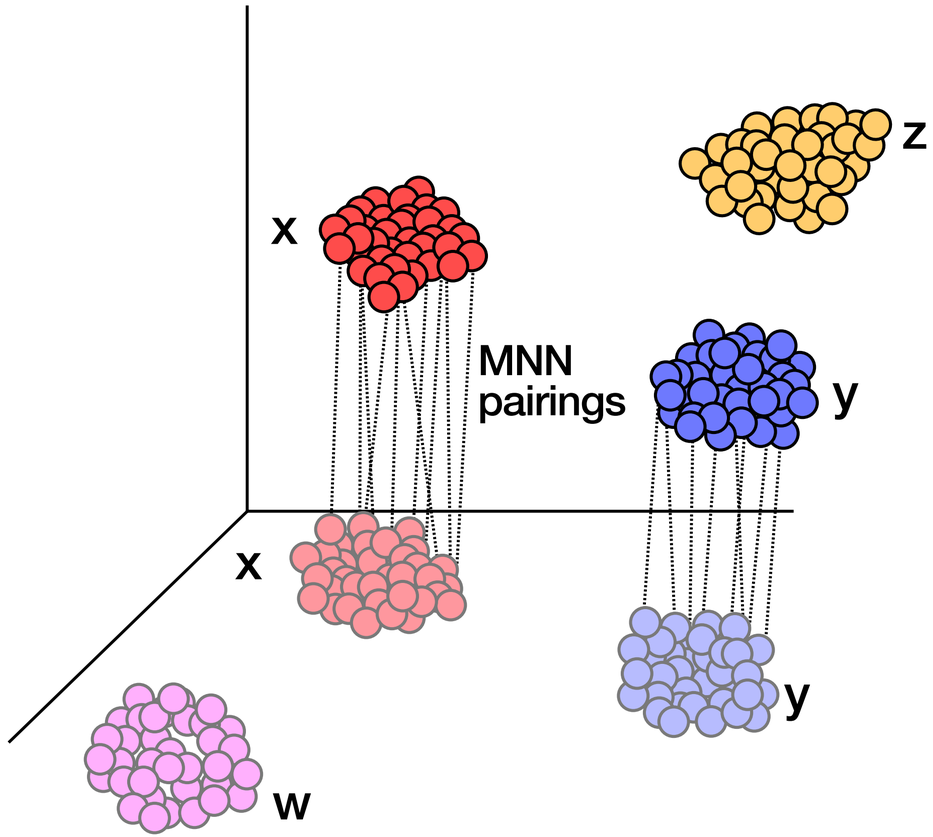
Batch normalization aims to minimize the difference between nearest cells across different batches



Haghverdi, .., and Marioni, *Nature Biotechnology* (2018)

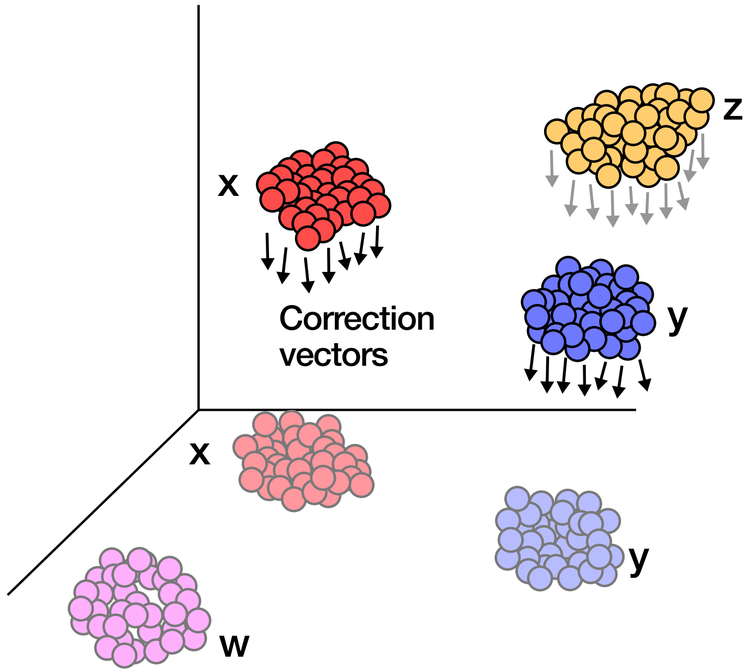
Batch normalization aims to minimize the difference between nearest cells across different batches

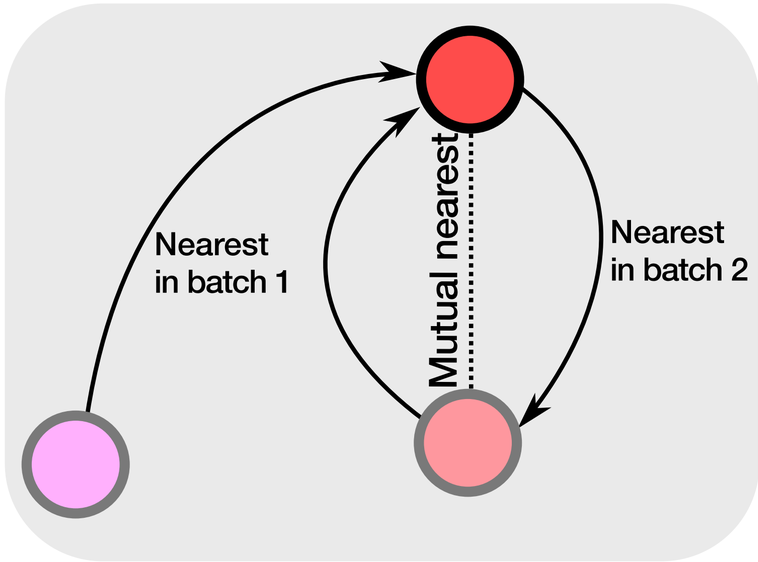




Haghverdi, .., and Marioni, *Nature Biotechnology* (2018)

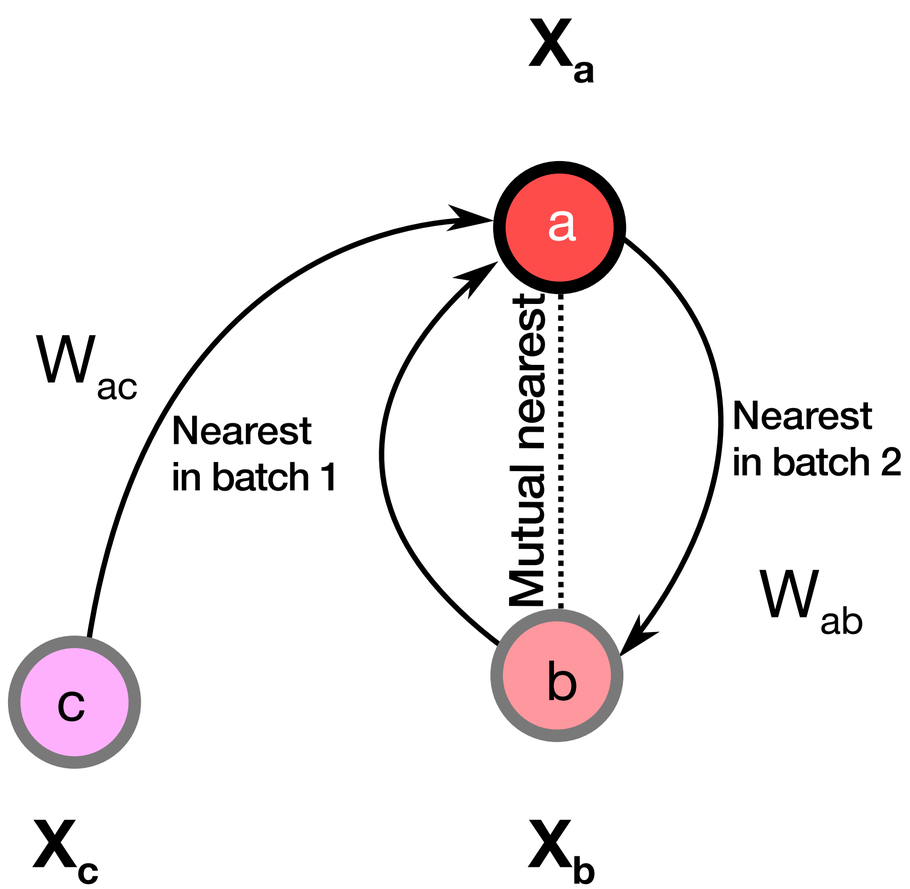
Batch normalization aims to minimize the difference between nearest cells across different batches





Haghverdi, .., and Marioni, *Nature Biotechnology* (2018)

Batch normalization aims to minimize the difference between nearest cells across different batches



What is the gap Δ between the batches?

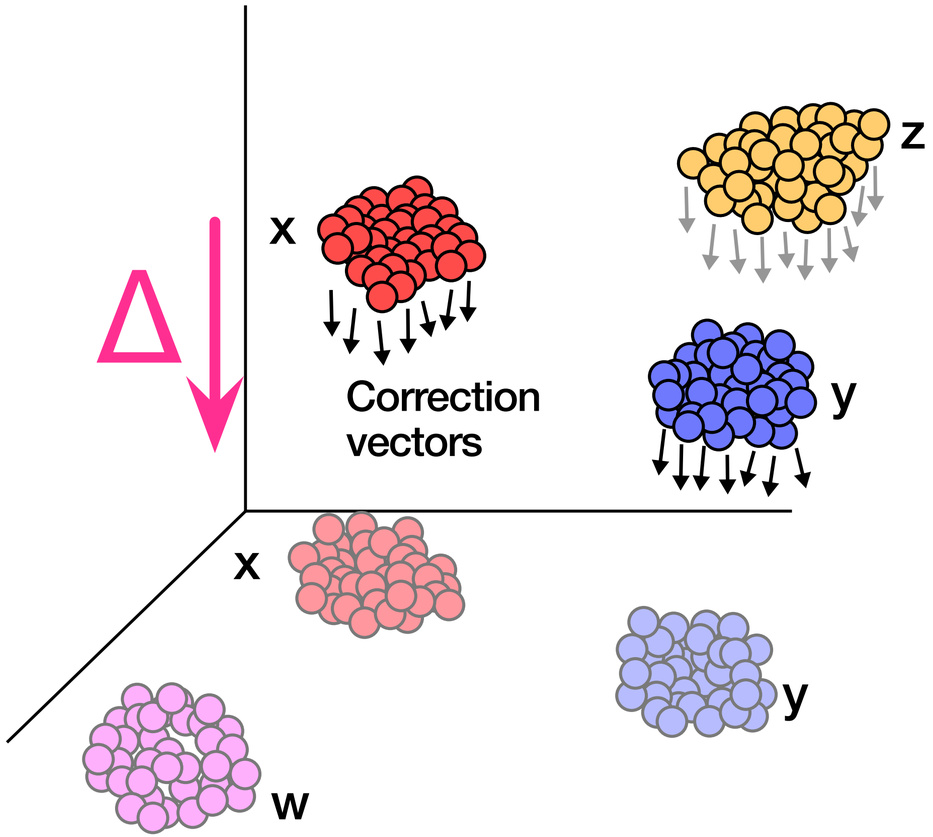
$$\min_{\Delta} \sum_{a,b} W_{ab} \|\mathbf{x}_a - \mathbf{x}_b - \Delta\|_2$$

A key assumption:

$$0 \approx W_{ac} < W_{ab}$$

Haghverdi and Marioni, *Nature Biotechnology* (2018)

Batch normalization aims to minimize the difference between nearest cells across different batches



What is the gap Δ between the batches?

$$\min_{\Delta} \sum_{a,b} W_{ab} \|\mathbf{x}_a - \mathbf{x}_b - \Delta\|_2$$

Fixed point (local) optimal solution:

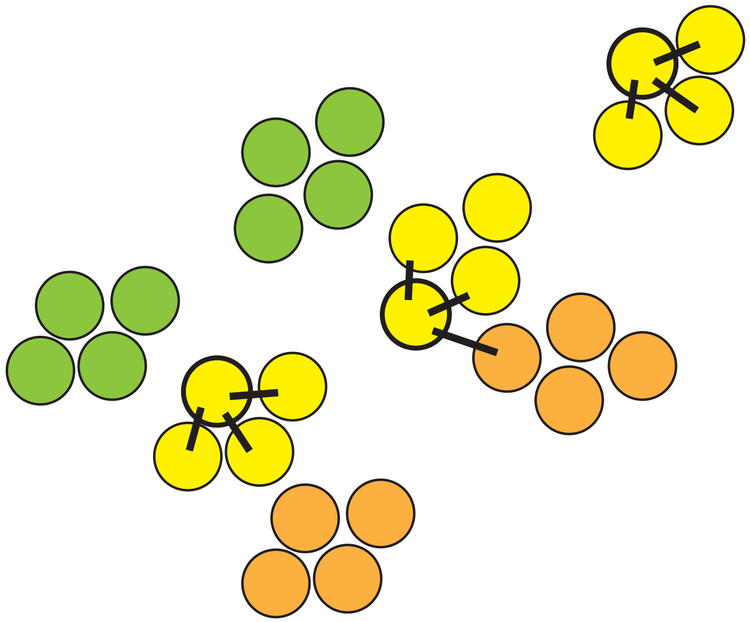
$$\Delta \leftarrow \frac{\sum_{a,b} W_{ab}(\mathbf{x}_a - \mathbf{x}_a)}{\sum_b W_{ab}}$$

Haghverdi, .., and Marioni, *Nature Biotechnology* (2018)

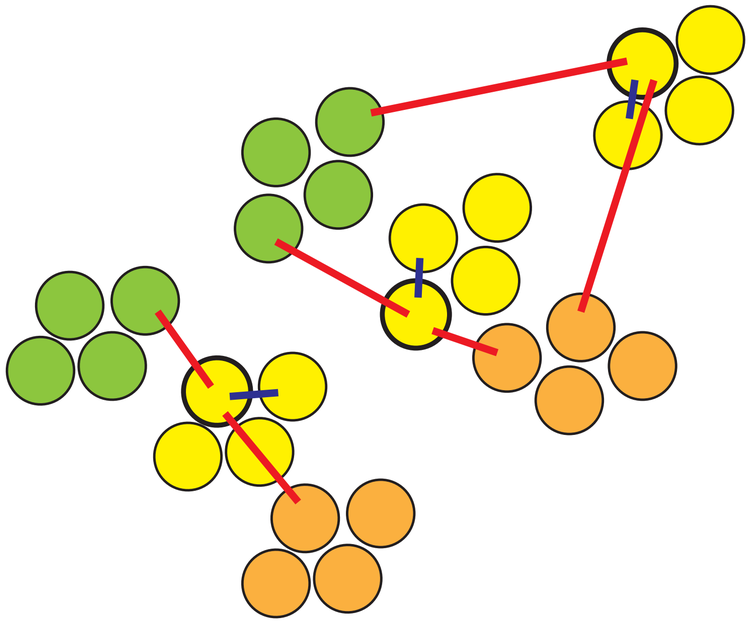
A batch-balancing k-nearest neighbour graph

BBKNN method strikes balance between over- and under-normalization

K-Nearest Neighbour



Batch Balanced K-Nearest Neighbour

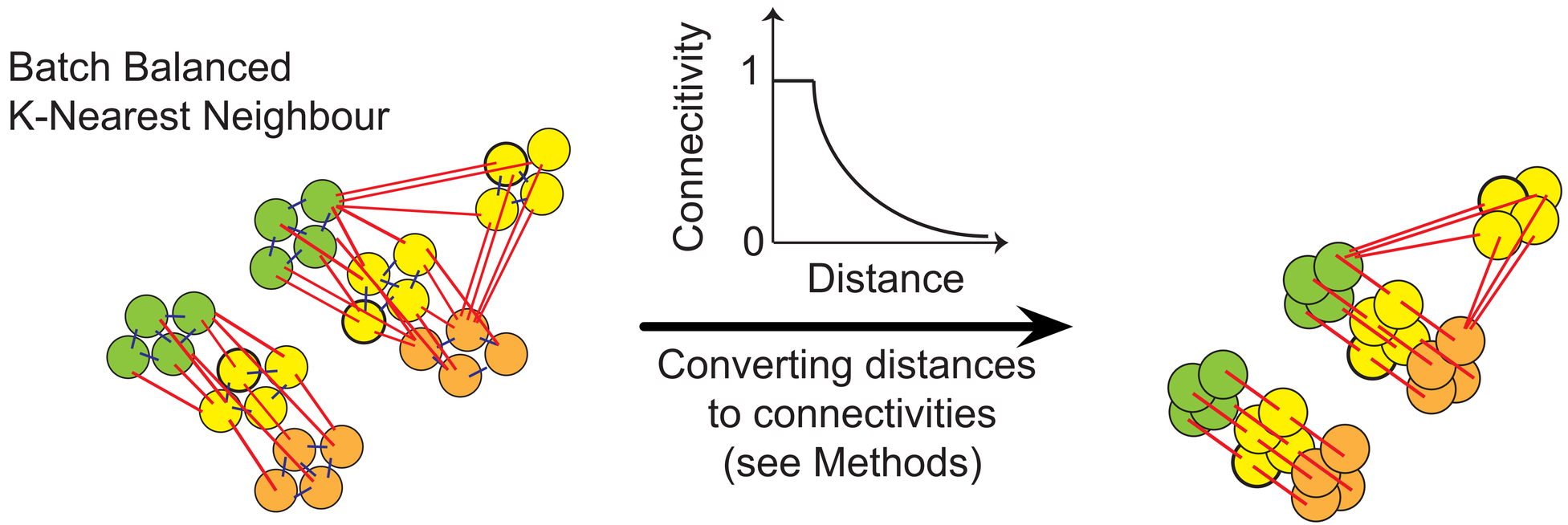


What kind of differences in due to inter-batch, technical discrepancy, not inter-cell-type divergence?

Polanski, .., Teichmann *Bioinformatics* (2019)

A batch-balancing k-nearest neighbour graph

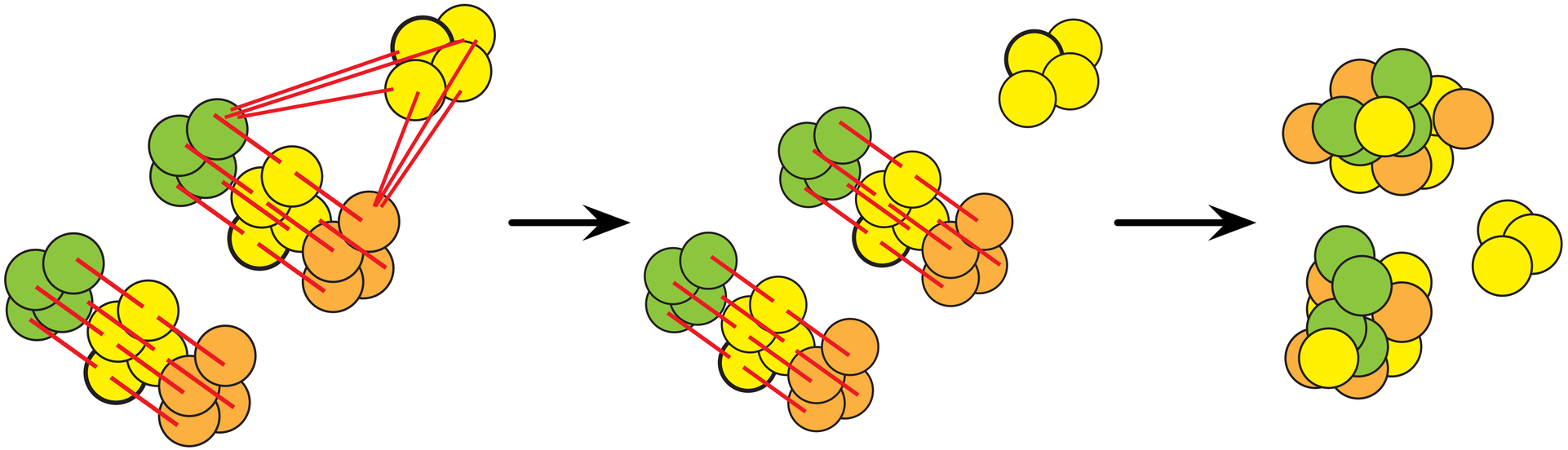
BBKNN method strikes balance between over- and under-normalization



Polanski, .., Teichmann *Bioinformatics* (2019)

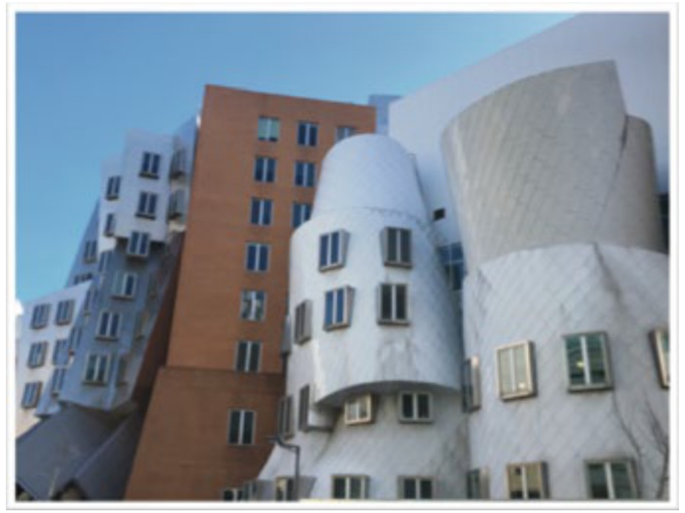
A batch-balancing k-nearest neighbour graph

BBKNN method strikes balance between over- and under-normalization



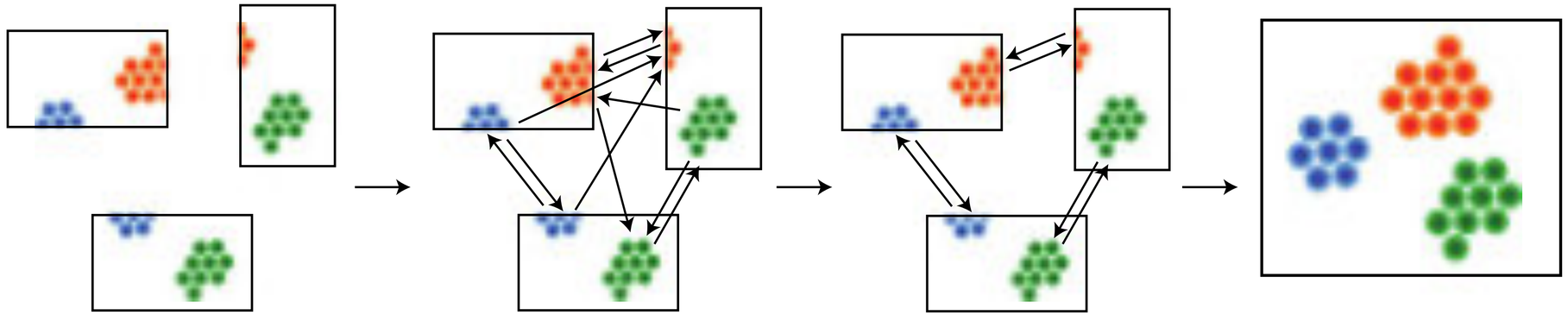
Polanski, .., Teichmann *Bioinformatics* (2019)

snapshots
of many data



panorama stitched
together

How do we integrate multiple samples/batches?



Collect many single-cell RNA-seq experiments

Find nearest neighbours across data sets

Keep mutually neighbouring cell pairs

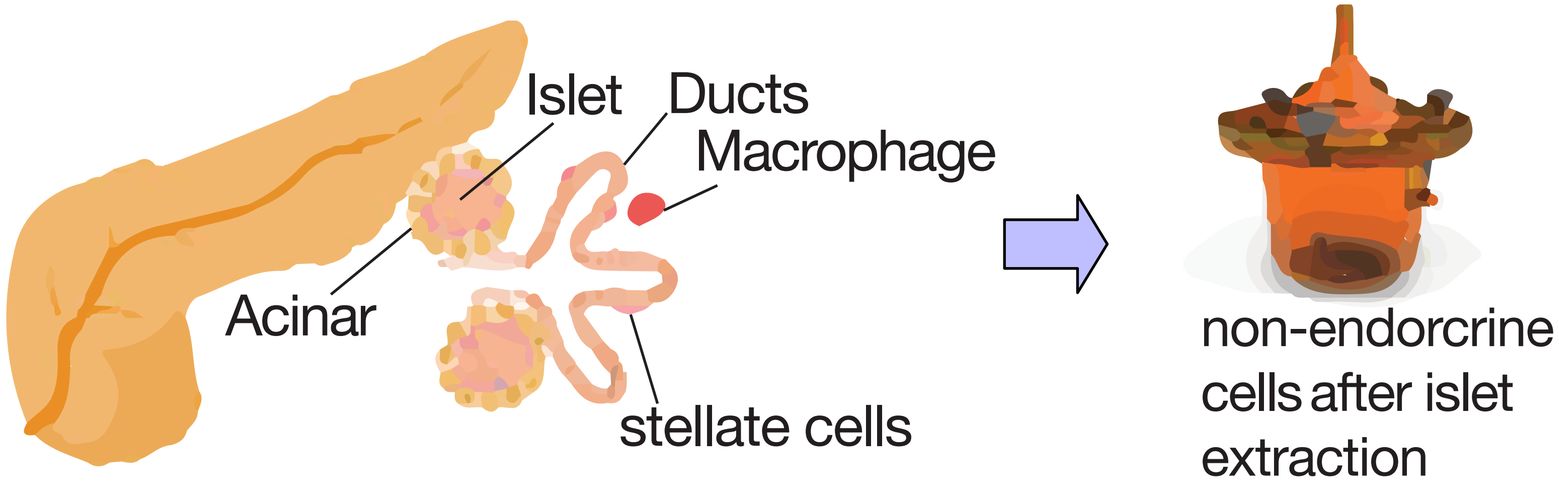
Create single-cell panorama

Scanorama: mutual nearest neighbourhood-based data integration

Hie, Bryson, Berger *Nature Biotechnology* (2019)

Example: pancreatic cells across three batches

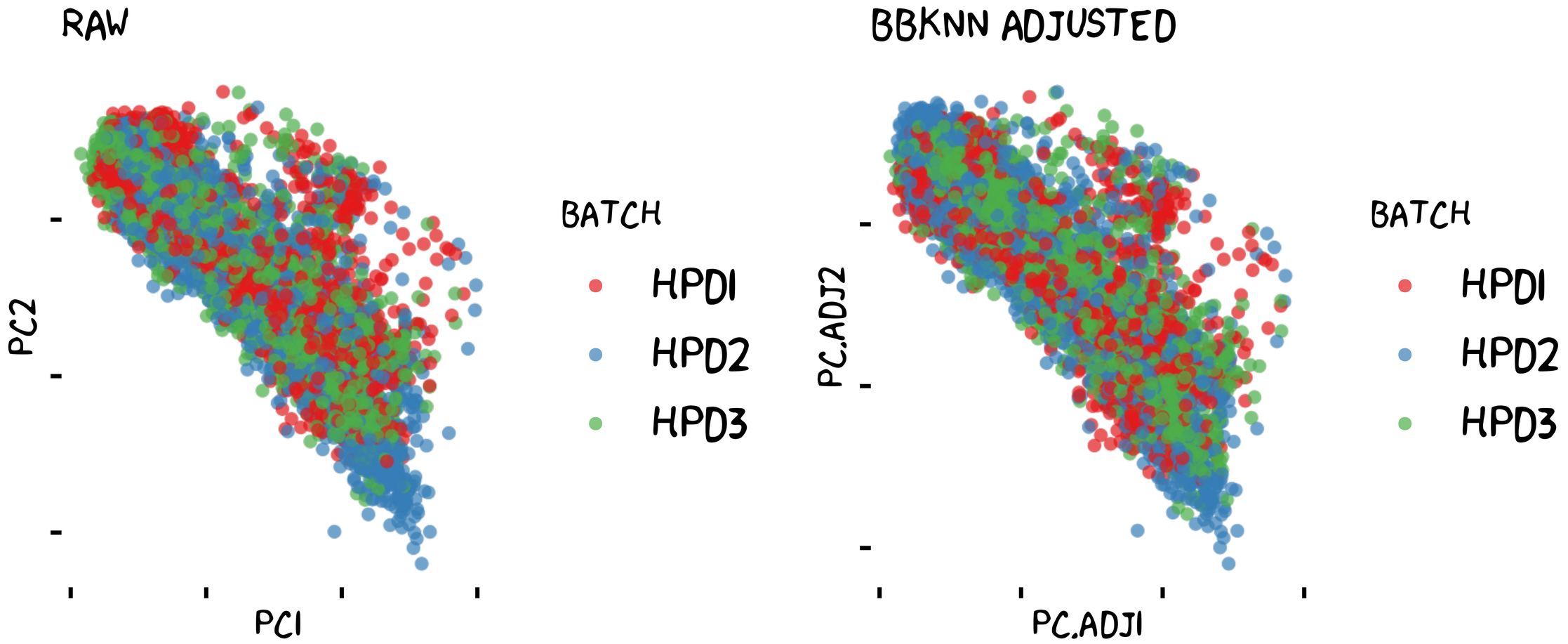
Single-cell RNA-seq data from three donors (three batches)



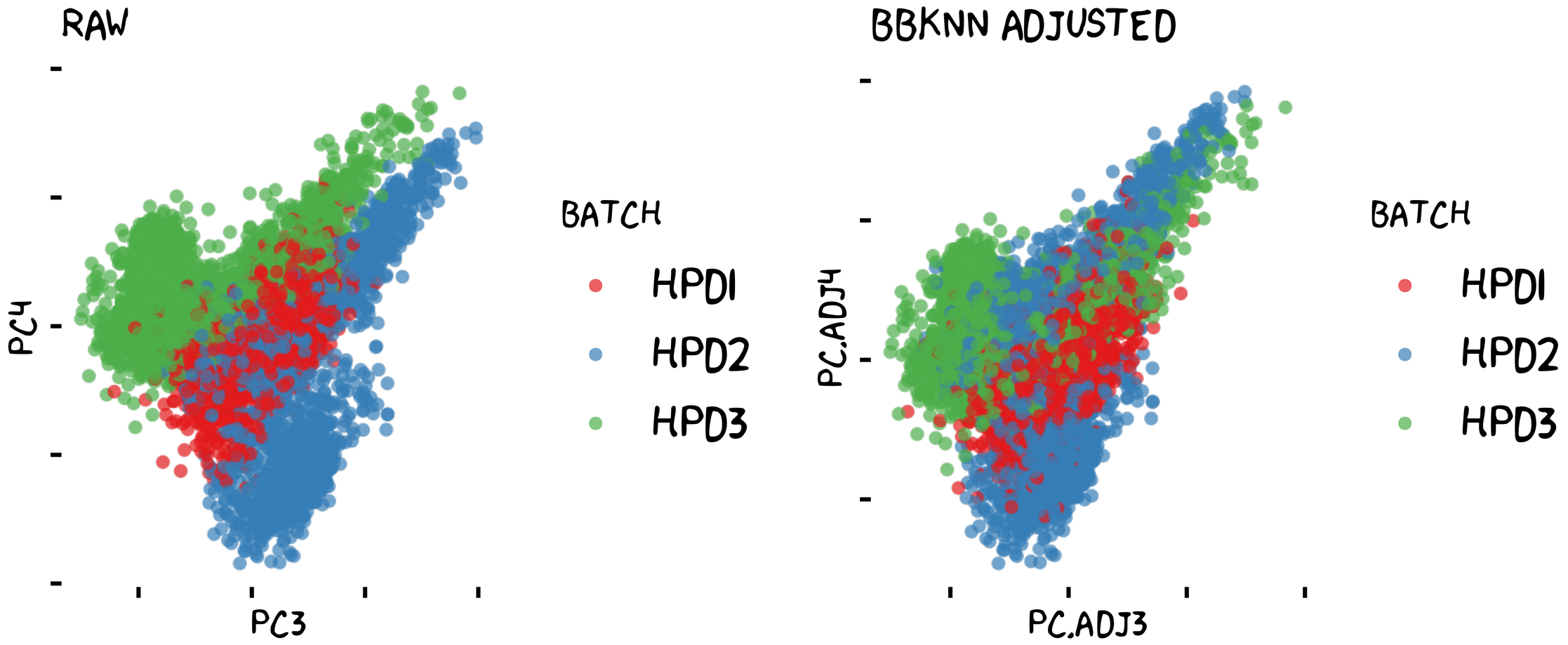
Goal: Remove potential batch effects across different donors. (1) Construct BBKNN graphs between cells; (2) compute average discrepancy Δ between batches in the PC space; (3) adjust them.

Qadir, *et al.*, *PNAS* (2020)

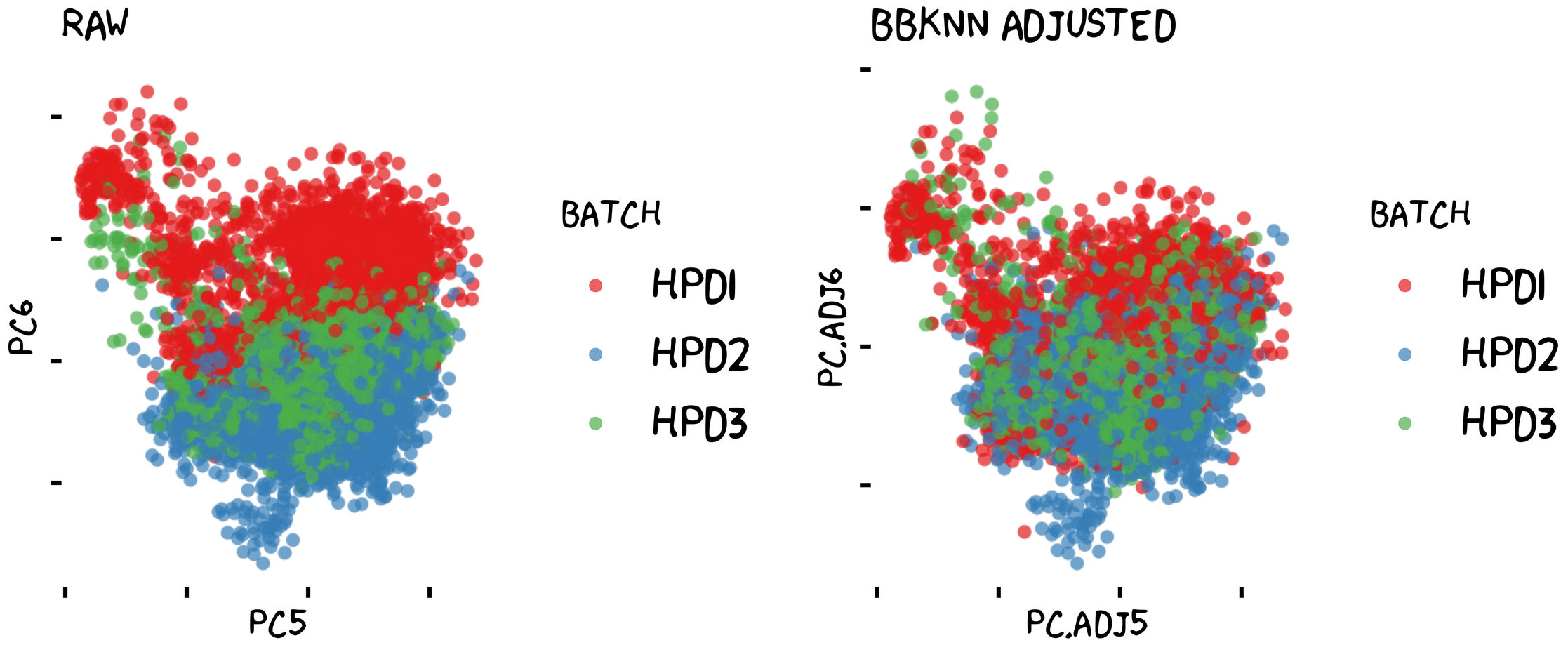
BBKNN-guided normalization



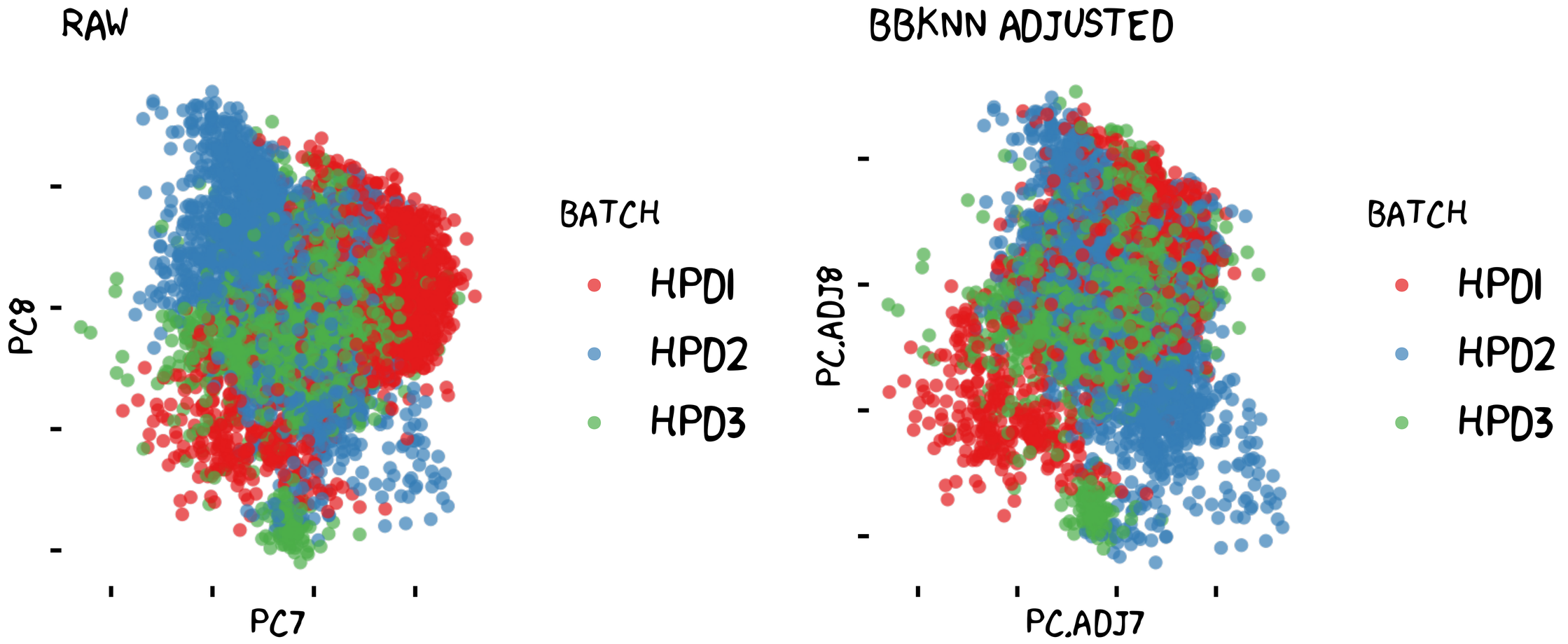
BBKNN-guided normalization



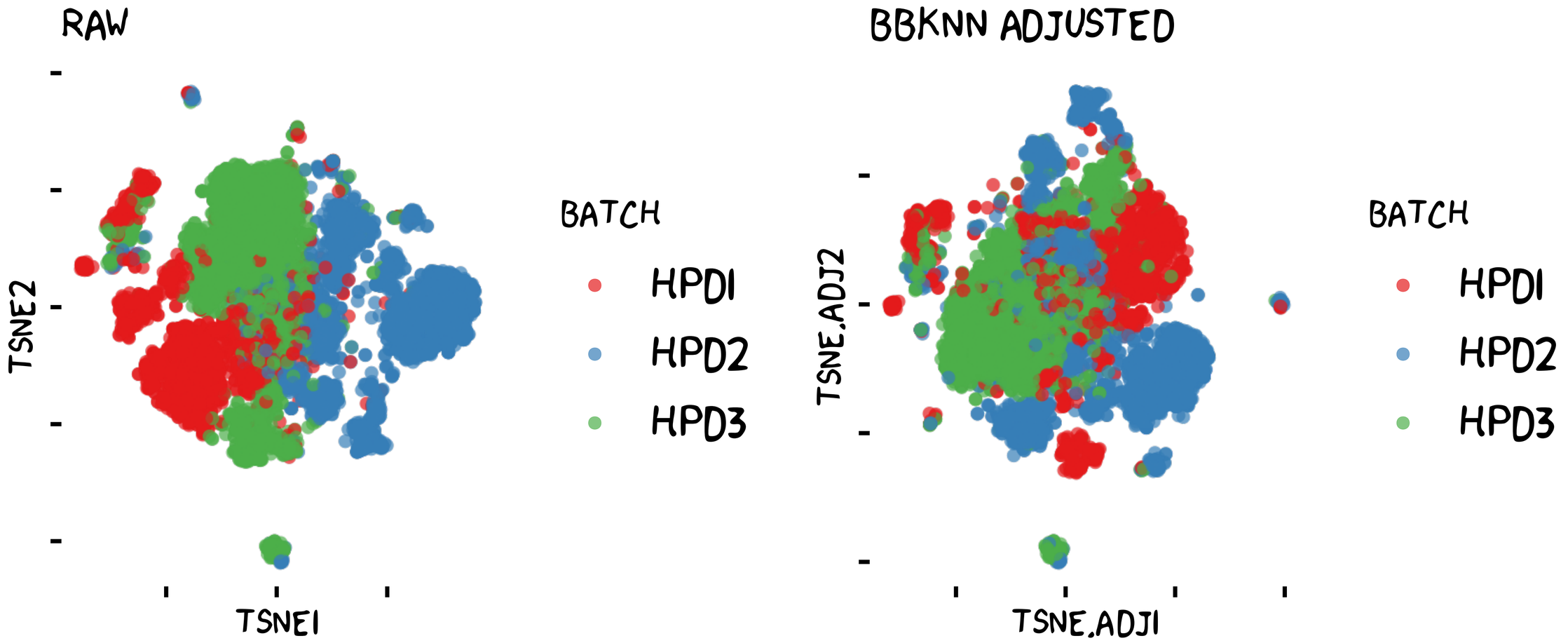
BBKNN-guided normalization



BBKNN-guided normalization



BBKNN-guided normalization



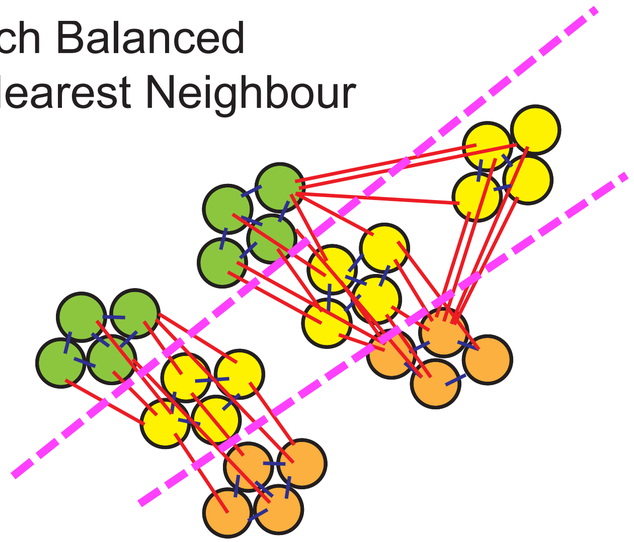
Discussions

- What can we do with BBKNN graphs?
- Why do we need batch normalization?
- Is it possible to over-correct the differences?
- Is it also possible to under-correct the differences?

Graph-based clustering of cells

Where is the graph?

Batch Balanced
K-Nearest Neighbour



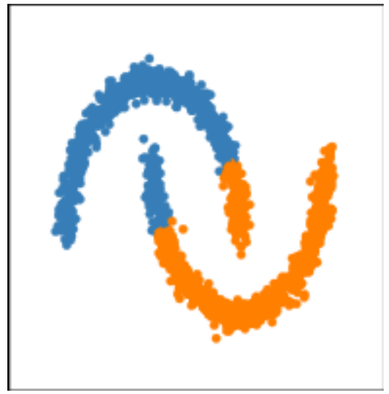
Most existing scRNA-seq toolboxes aim to produce best possible k-NN graph

Polanski *et al*. Bioinformatics (2019)

And why graph-based?

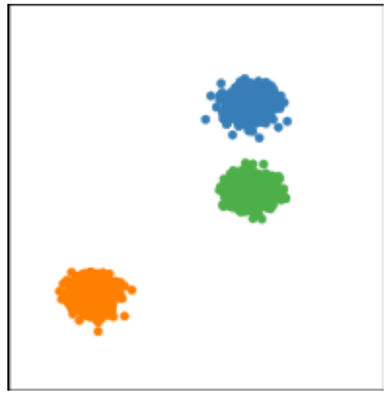
k-means

spectral clustering based on k-NN graph

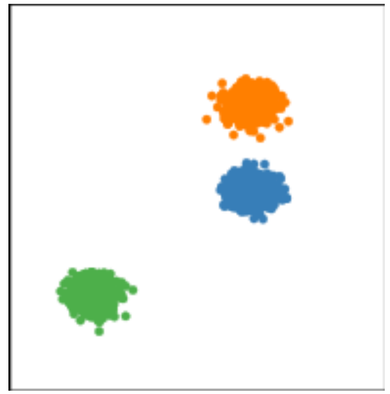


vs.

We may not know the distrib. of scRNA-seq



vs.



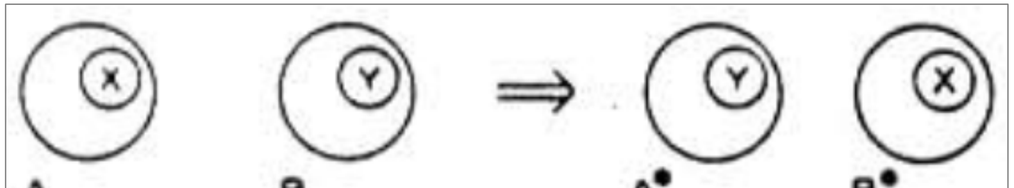
It's hard to assume that single-cell data can be faithfully modeled by a multivar. linear Gaussian model

https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

Digression

Graph clustering/partitioning is a classic discrete optimization problem

Two-cut problems (vertex set A and B)



Kernighan & Lin (1970)

$A^{\bullet} = A - X + Y$

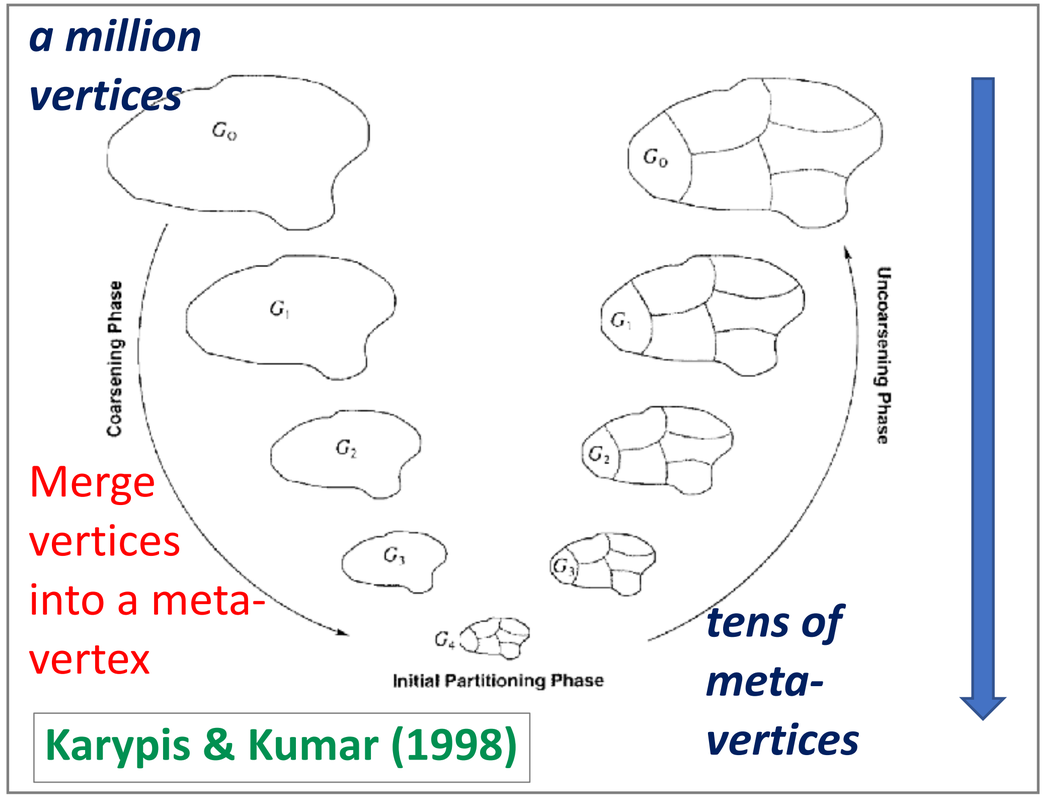
$B^{\bullet} = B - Y + X$

Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming

0.87856-approximation to opt

Goemans & Williamson (1995)

k-metis algorithm (heavily used in SciPy)



Graph contraction to simplify for high-level greedy move

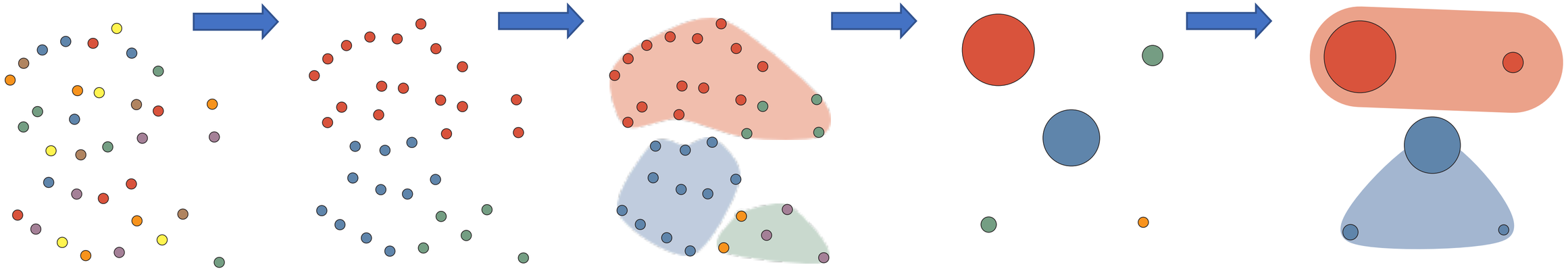
Leiden algorithm to resolve densely-connected community of nodes (cells)

Initial move (greedy update)

Initial partition

Create meta-vertices

Refine vertex assignment (locally)



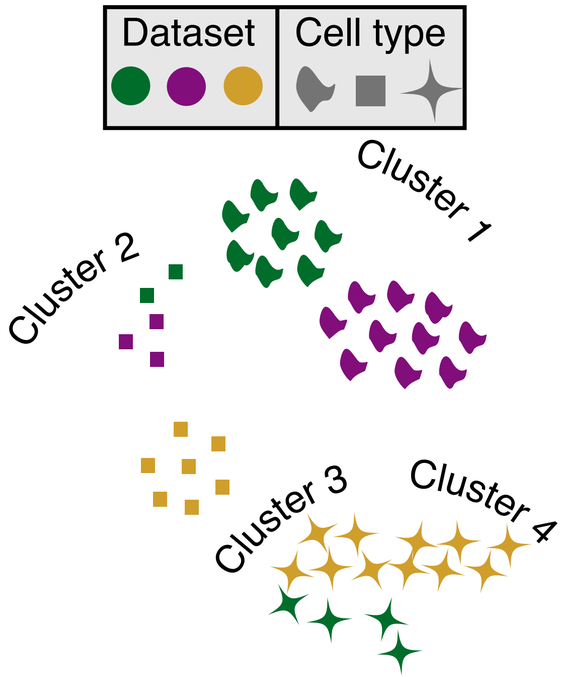
From Louvain to Leiden: guaranteeing well-connected communities

V.A. Traag,* L. Waltman, and N.J. van Eck
Centre for Science and Technology Studies, Leiden University, the Netherlands
(Dated: October 22, 2018)

Louvain, Belgium

Leiden, Netherland

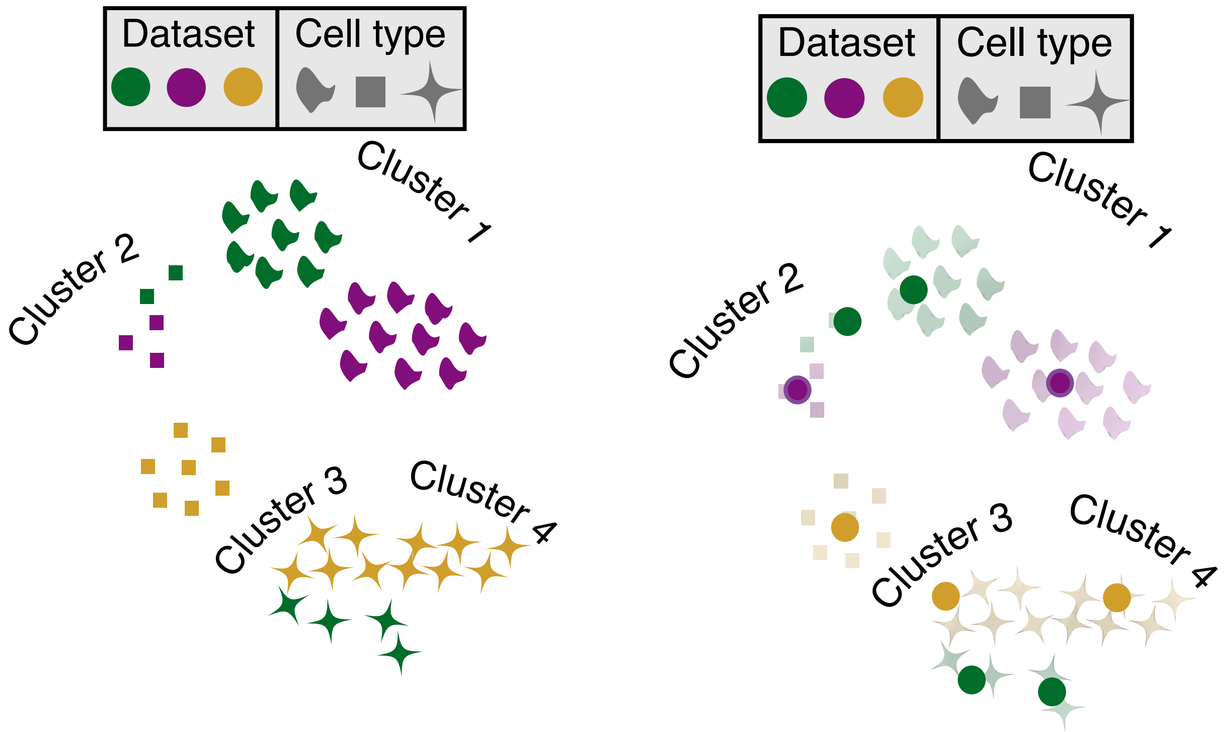
Harmony: clustering-based data normalization



Soft assign cells to clusters, favoring mixed dataset representation

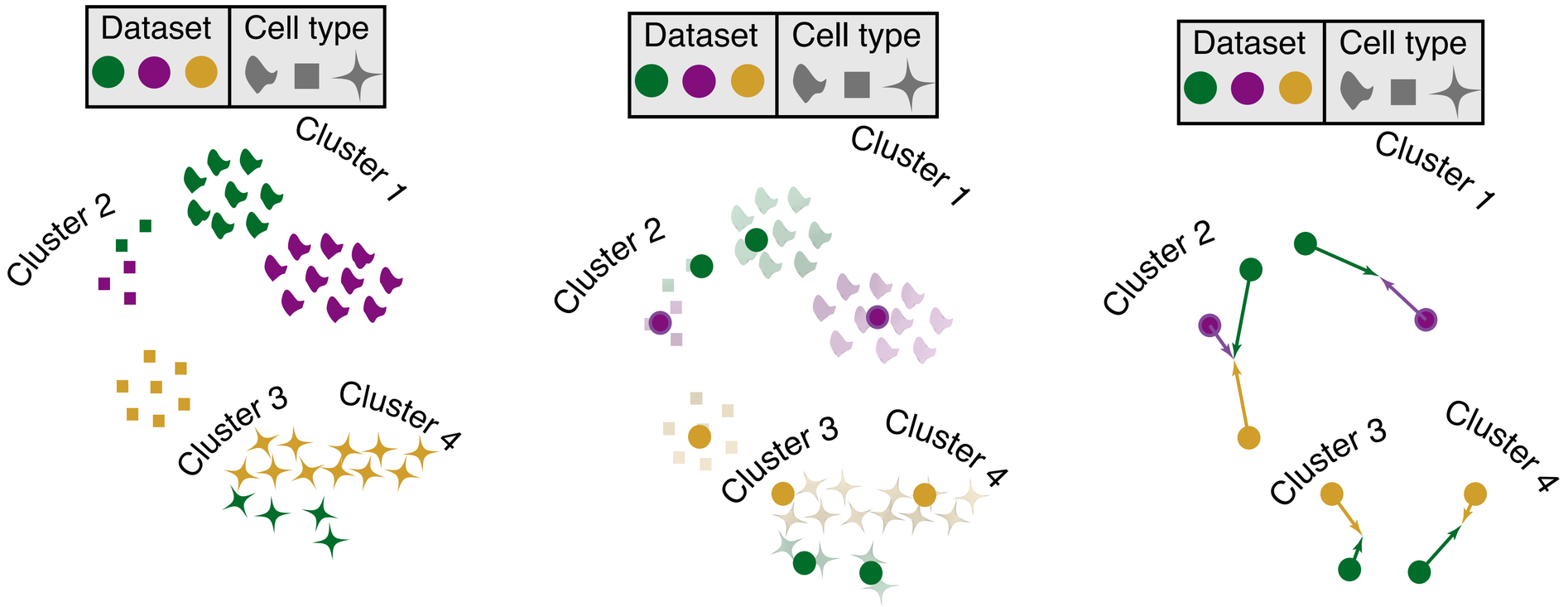
Korsunsky, .., Loh, Raychaudhuri *Nature Methods* (2019)

Harmony: clustering-based data normalization



Korsunsky, .., Loh, Raychaudhuri *Nature Methods* (2019)

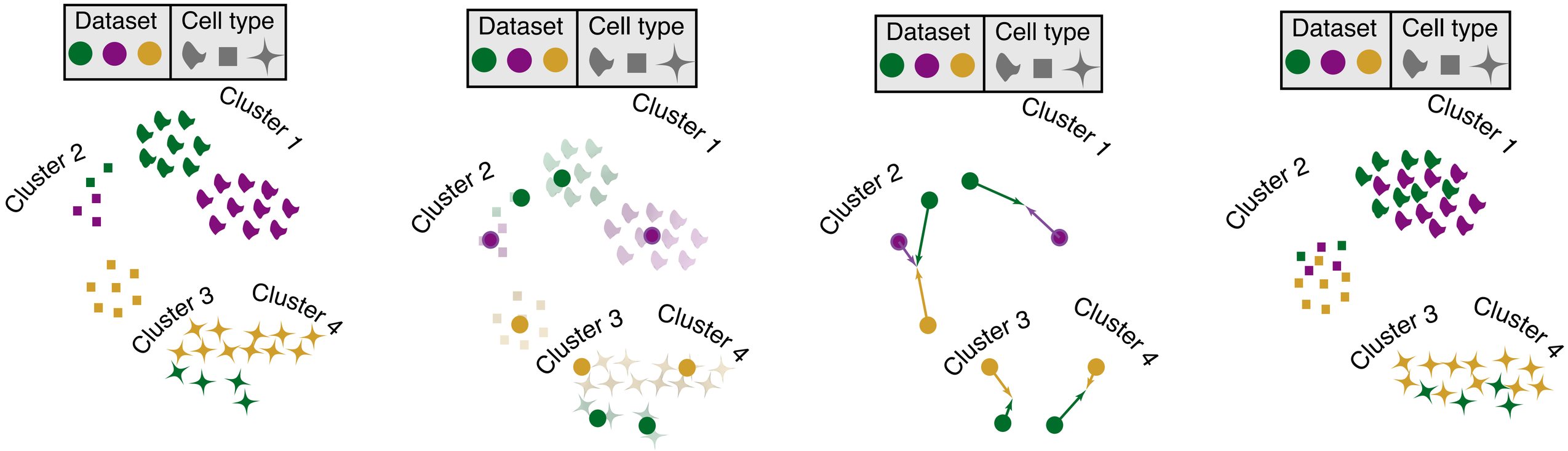
Harmony: clustering-based data normalization



Soft assign cells to clusters, favoring mixed dataset representation → Get cluster centroids for each dataset → Get dataset correction factors for each cluster

Korsunsky, .., Loh, Raychaudhuri *Nature Methods* (2019)

Harmony: clustering-based data normalization



Korsunsky, .., Loh, Raychaudhuri *Nature Methods* (2019)