Avoiding double dipping in the analysis of single-cell RNA sequencing data

Anna Neufeld
UW Combi Seminar
January 17, 2024

What is double dipping?

Classical statistical methods assume that we only ever test pre-specified hypotheses about pre-specified models.

What is double dipping?

Classical statistical methods assume that we only ever test pre-specified hypotheses about pre-specified models.

In reality, we explore our data, fit several models, evaluate these models, select our favorite model, then test hypotheses about this model.

What is double dipping?

Classical statistical methods assume that we only ever test pre-specified hypotheses about pre-specified models.

In reality, we explore our data, fit several models, evaluate these models, select our favorite model, then test hypotheses about this model.

Double Dipping: Using the same data for two tasks, such as:

1. Fitting and evaluating a model.
2. Generating and testing a null hypothesis.

We can often avoid double dipping through sample splitting

	Feature 1	Feature 2
Obs. 1	12	6
Obs. 2	31	8
Obs. 3	11	31
Obs. 4	22	34

We can often avoid double dipping through sample splitting

	Feature 1	Feature 2
Obs. 1	12	6
Obs. 2	31	8
Obs. 3	11	31
Obs. 4	22	34

Train

	Feature 1	Feature 2
Obs. 1	12	6
Obs. 2	31	8

Test

	Feature 1	Feature 2
Obs. 3	11	31
Obs. 4	22	34

We can often avoid double dipping through sample splitting

	Feature 1	Feature 2
Obs. 1	12	6
Obs. 2	31	8
Obs. 3	11	31
Obs. 4	22	34

Train

	Feature 1	Feature 2
Obs. 1	12	6
Obs. 2	31	8

Fit model.

Test

	Feature 1	Feature 2
Obs. 3	11	31
Obs. 4	22	34

We can often avoid double dipping through sample splitting

	Feature 1	Feature 2
Obs. 1	12	6
Obs. 2	31	8
Obs. 3	11	31
Obs. 4	22	34

Train

	Feature 1	Feature 2
Obs. 1	12	6
Obs. 2	31	8

Fit model.

Test

	Feature 1	Feature 2
Obs. 3	11	31
Obs. 4	22	34

Evaluate model.

We can often avoid double dipping through sample splitting

	Feature 1	Feature 2
Obs. 1	12	6
Obs. 2	31	8
Obs. 3	11	31
Obs. 4	22	34

Train

	Feature 1	Feature 2
Obs. 1	12	6
Obs. 2	31	8

Select hypothesis.

Test

	Feature 1	Feature 2
Obs. 3	11	31
Obs. 4	22	34

We can often avoid double dipping through sample splitting

	Feature 1	Feature 2
Obs. 1	12	6
Obs. 2	31	8
Obs. 3	11	31
Obs. 4	22	34

Train

	Feature 1	Feature 2
Obs. 1	12	6
Obs. 2	31	8

Select hypothesis.

Test

	Feature 1	Feature 2
Obs. 3	11	31
Obs. 4	22	34

Test hypothesis.

Outline

1. **Motivation: settings where sample splitting doesn't work**
2. Poisson thinning
3. Data thinning
4. Application to human fetal cell atlas data
5. Application to cardiomyocyte differentiation data
6. Ongoing work

Single cell RNA-sequencing

	Gene 1	Gene 2	Gene 3
Cell 1	18	0	22
Cell 2	4	0	5
Cell 3	2	0	0
Cell 4	29	15	17

Single cell RNA-sequencing

	Gene 1	Gene 2	Gene 3
Cell 1	18	0	22
Cell 2	4	0	5
Cell 3	2	0	0
Cell 4	29	15	17

Examples of Questions

1. Which genes are differentially expressed across cell types?
2. Which genes are differentially expressed along a cellular differentiation trajectory?

Single cell RNA-sequencing

	Gene 1	Gene 2	Gene 3
Cell 1	18	0	22
Cell 2	4	0	5
Cell 3	2	0	0
Cell 4	29	15	17

Examples of Questions

1. Which genes are differentially expressed across cell types?
2. Which genes are differentially expressed along a cellular differentiation trajectory?

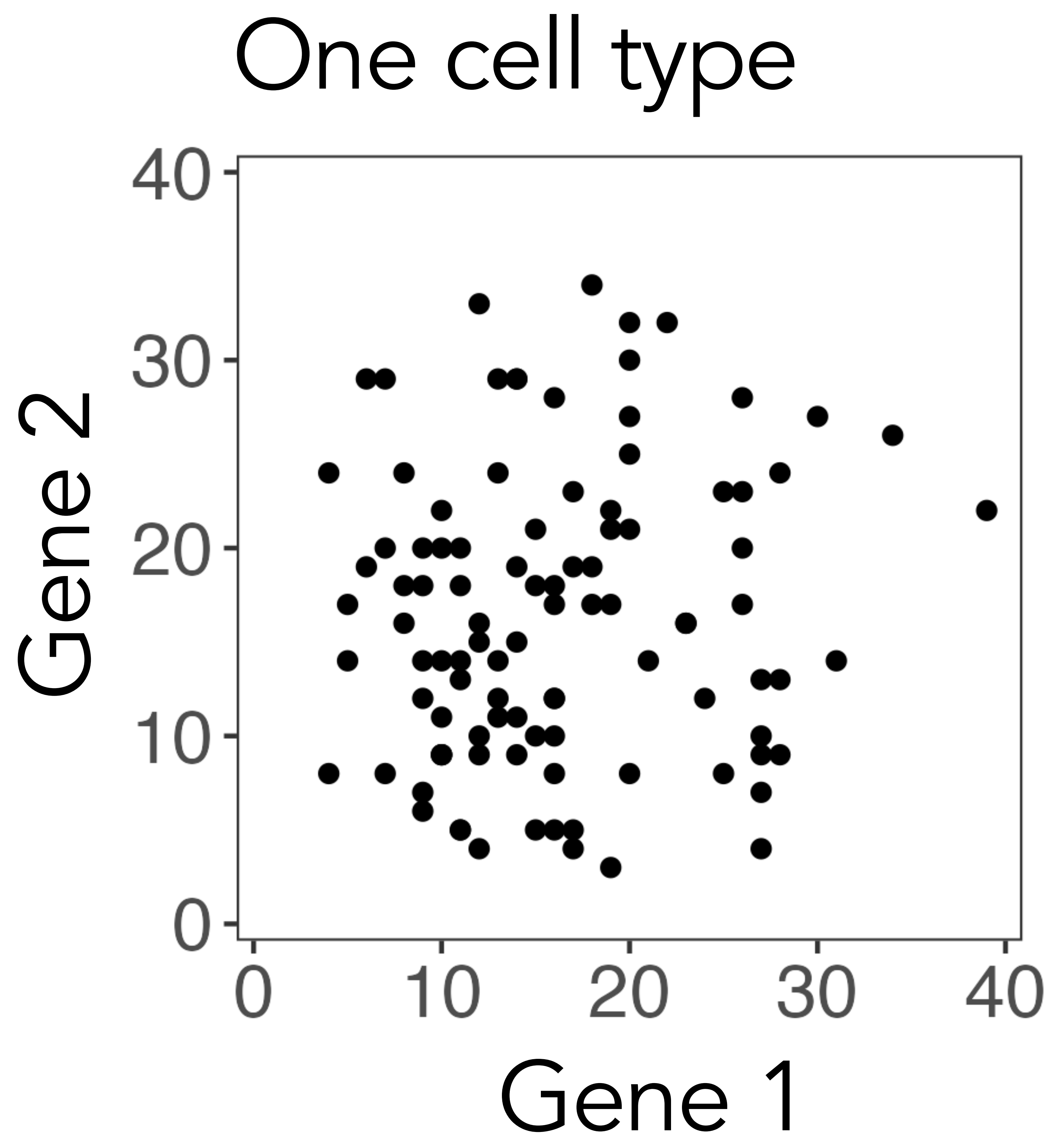
Examples of Challenges

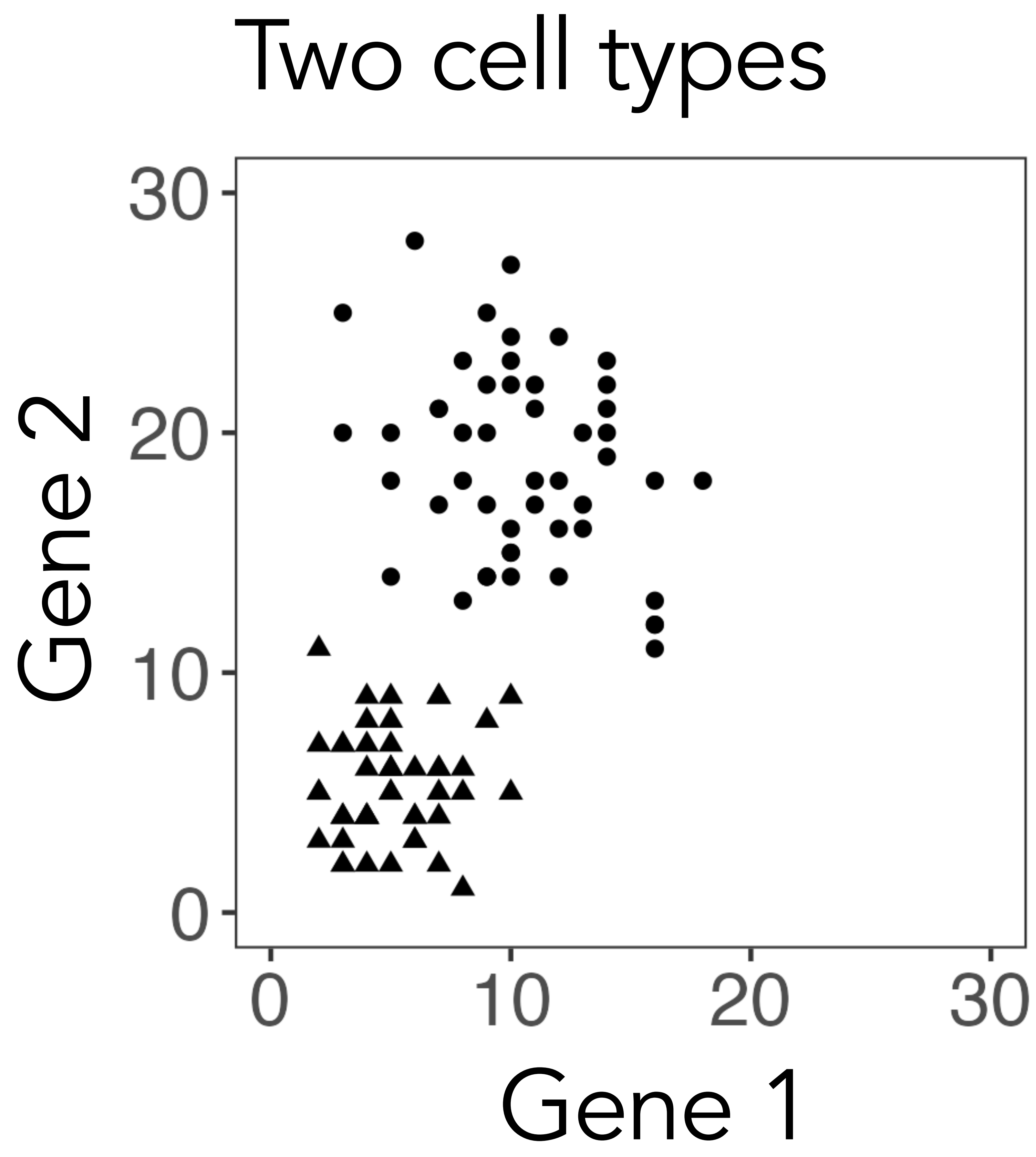
1. Cell type and cell trajectory are unobserved and must be estimated.
2. Number of cell types or topology of trajectory not necessarily known in advance.

Two instances where double dipping arises

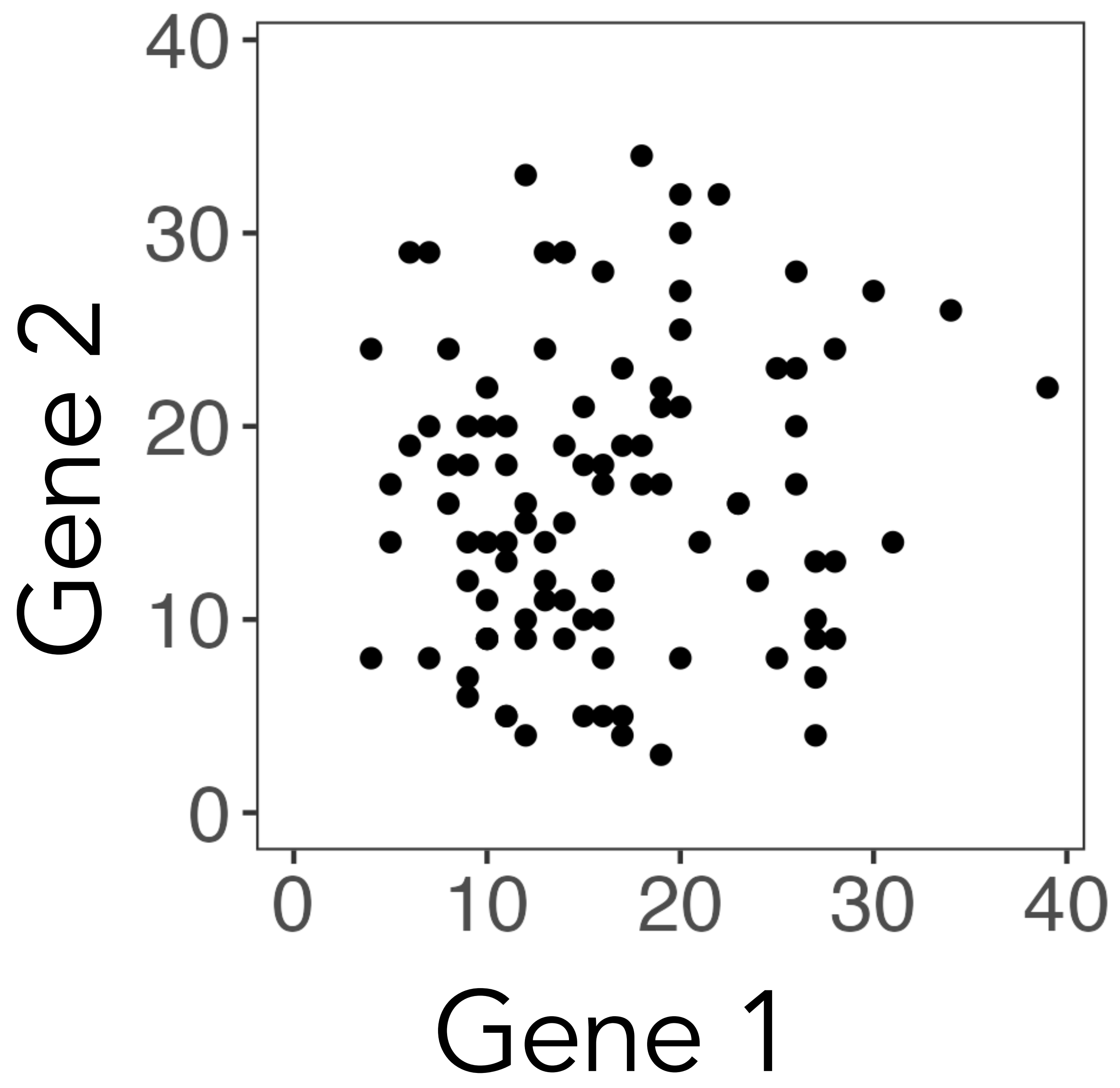
1. **Model selection for latent variable models.**
   - "How many cell types exist in this data?"
   - We double dip if we use the same data to fit and evaluate the models.
2. **Inference after latent variable estimation.**
   - "Which genes are differentially expressed across cell types?"
   - We double dip if we use the same data to estimate the clusters and then test for differential expression.

Example 1: how many distinct cell types exist in a scRNA-seq dataset?

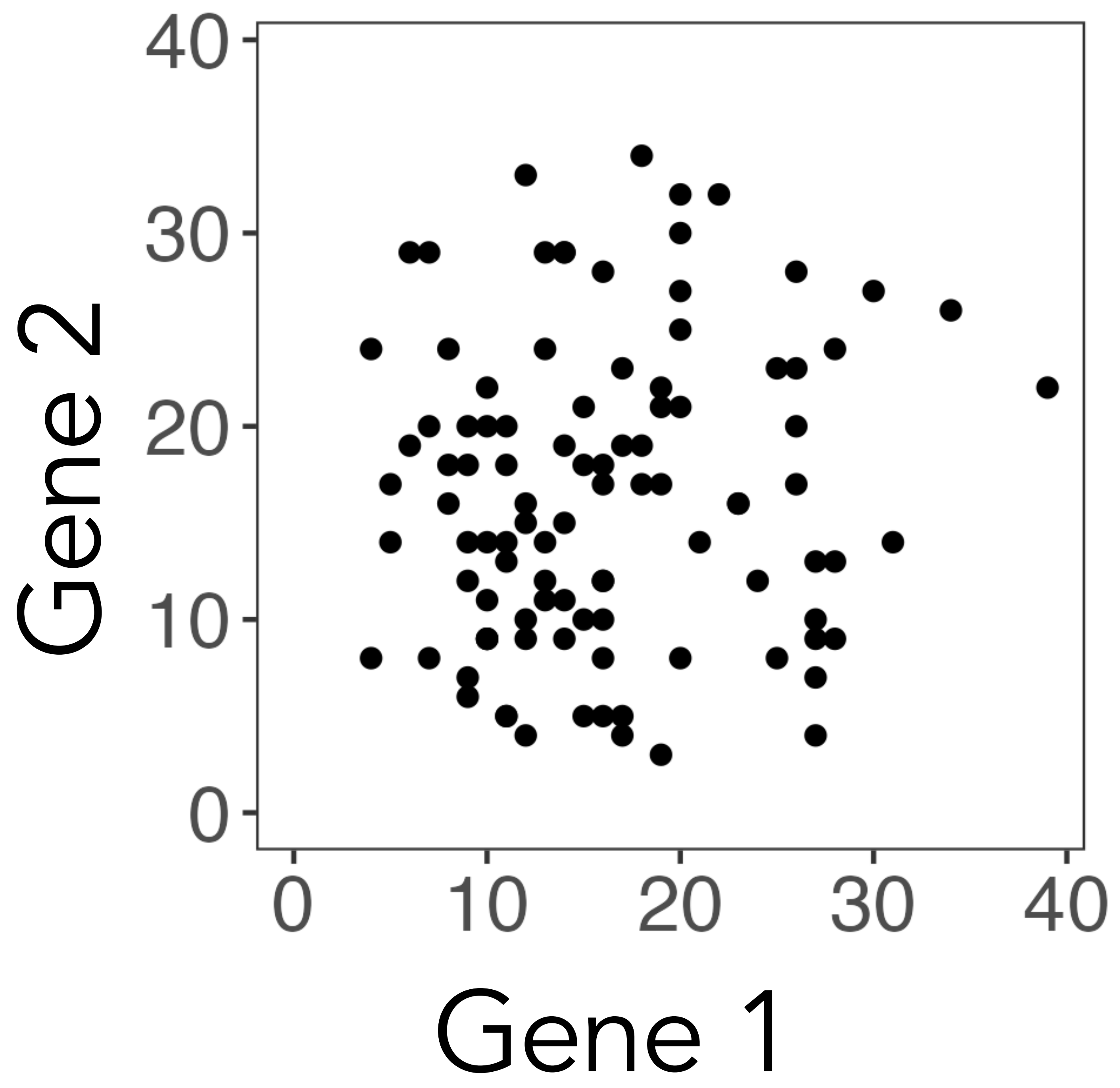




Example 1: how many distinct cell types exist in a scRNA-seq dataset?

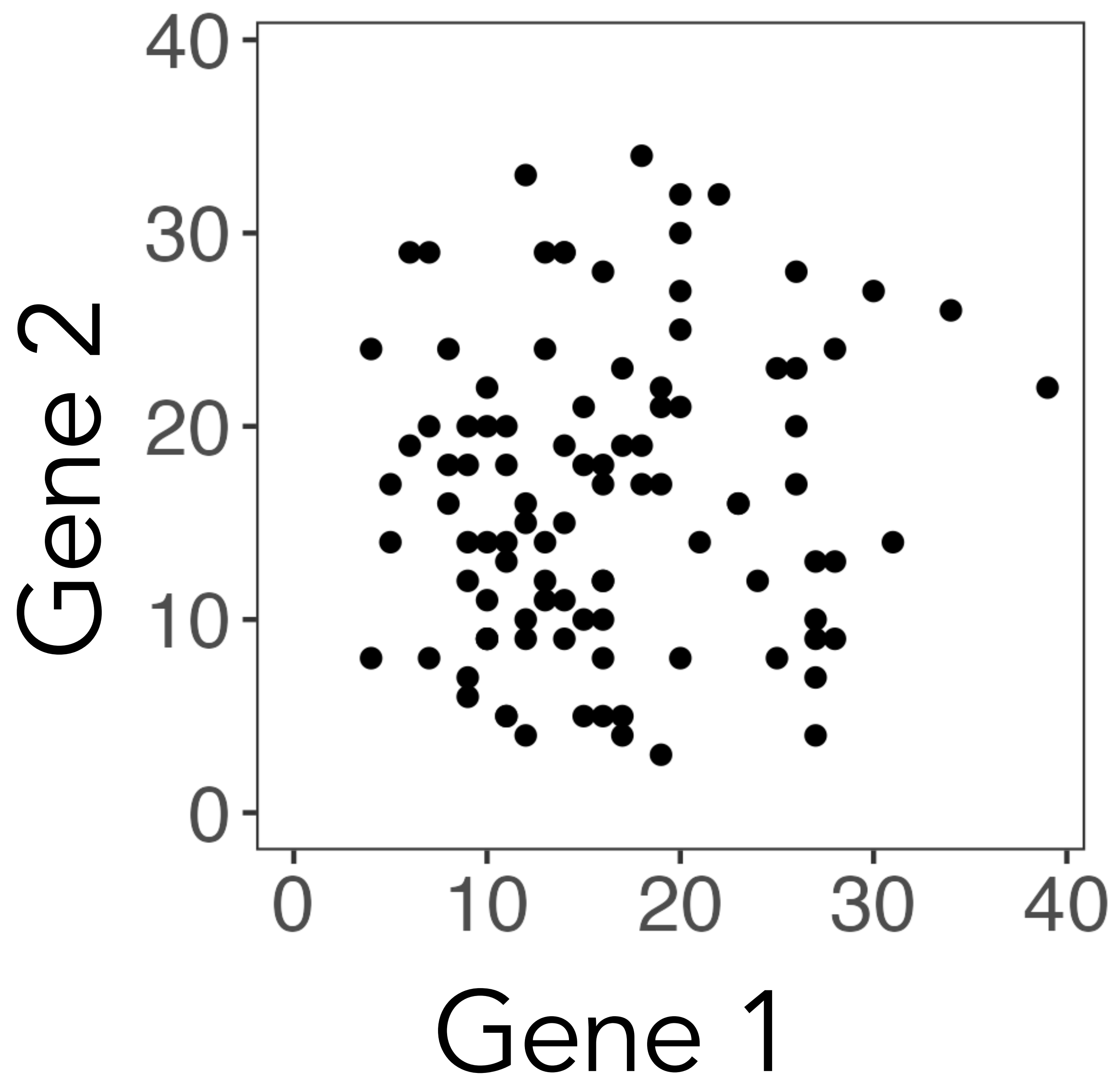


Example 1: how many distinct cell types exist in a scRNA-seq dataset?



Goal: how many clusters are in this data?

Example 1: how many distinct cell types exist in a scRNA-seq dataset?



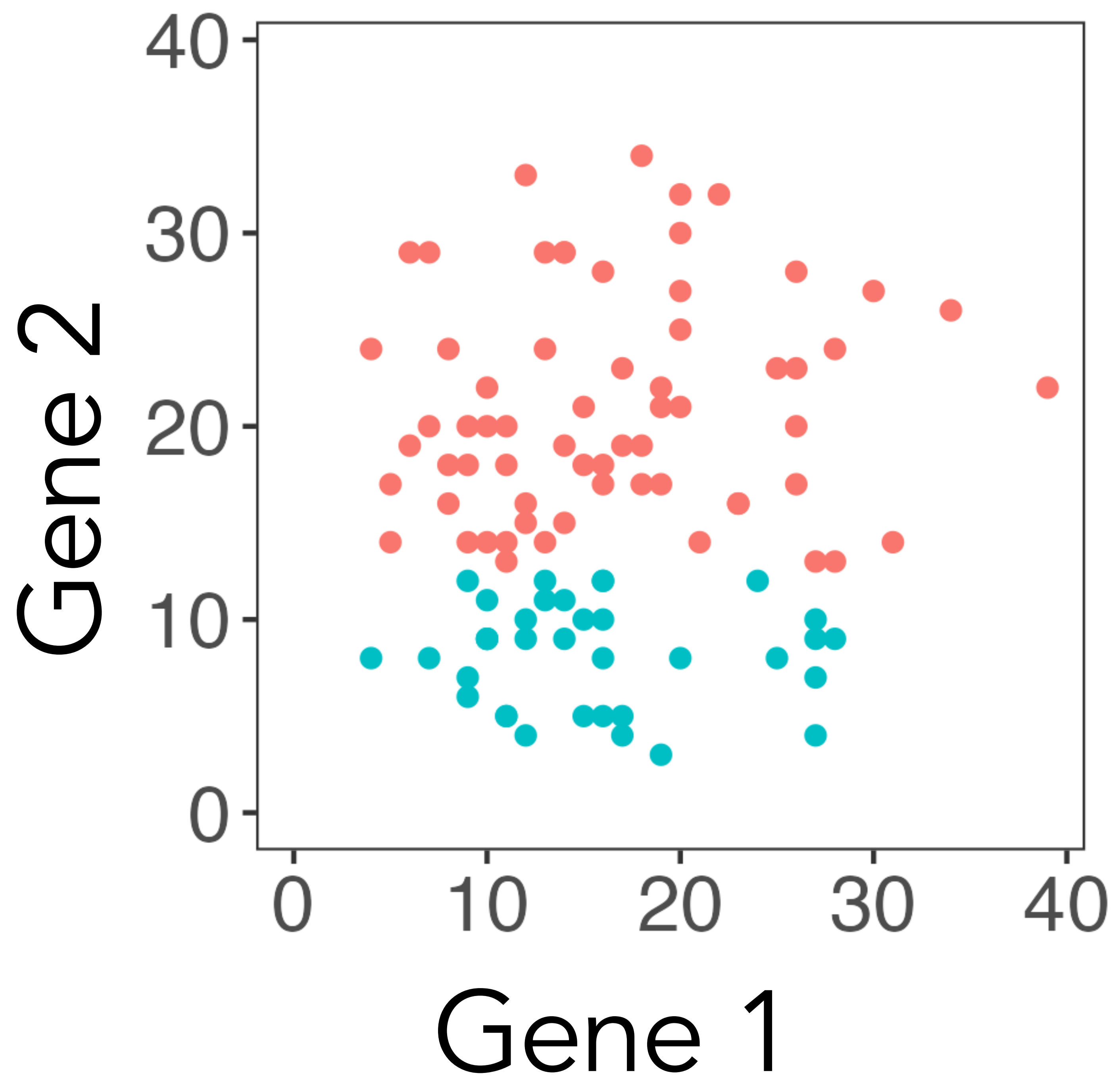
Goal: how many clusters are in this data?

For several values of k:

Step 1: fit a model with k clusters.

Step 2: evaluate model using a loss function.

Example 1: how many distinct cell types exist in a scRNA-seq dataset?



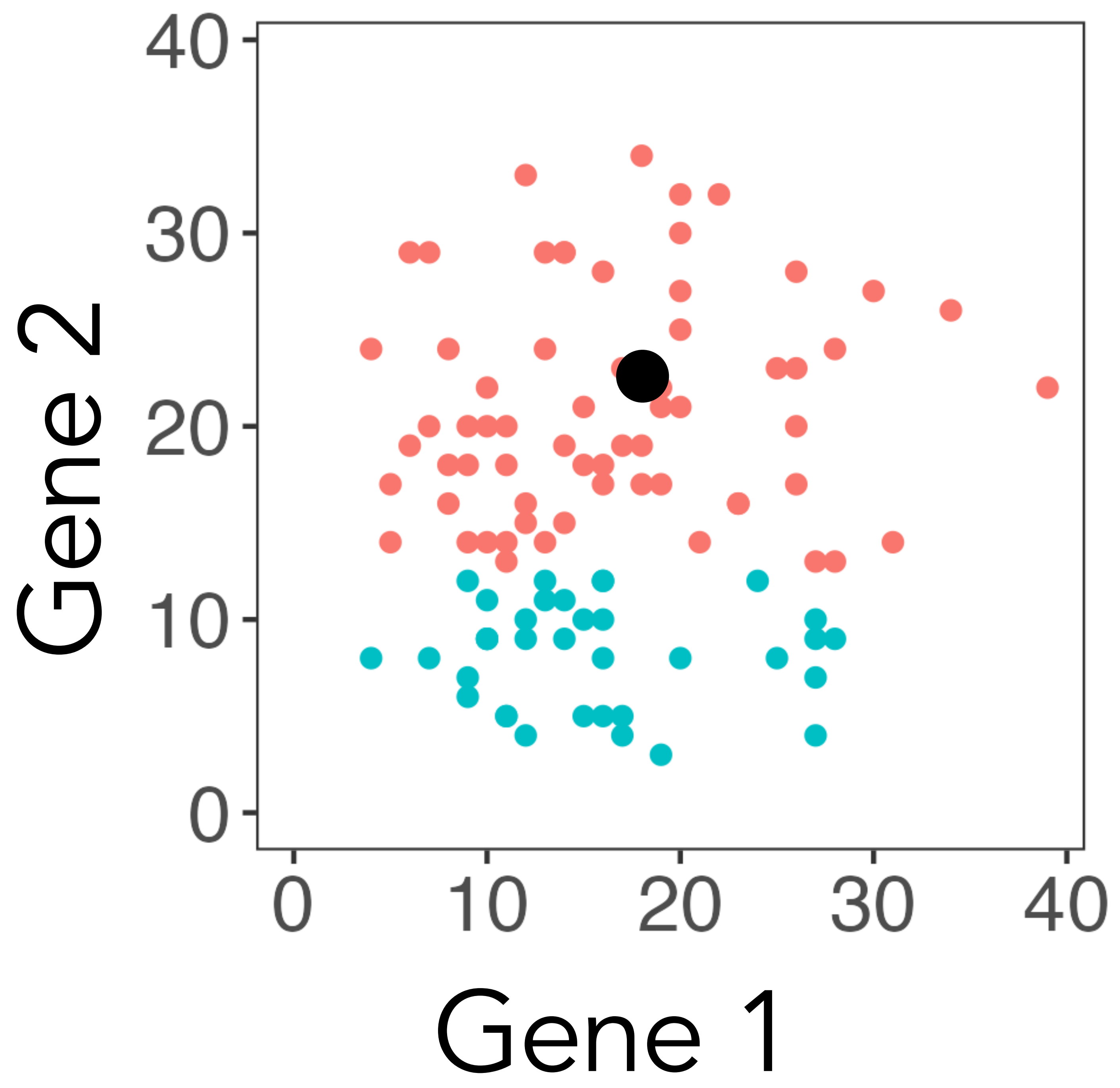
Goal: how many clusters are in this data?

For several values of k:

Step 1: fit a model with k clusters.

Step 2: evaluate model using a loss function.

Example 1: how many distinct cell types exist in a scRNA-seq dataset?



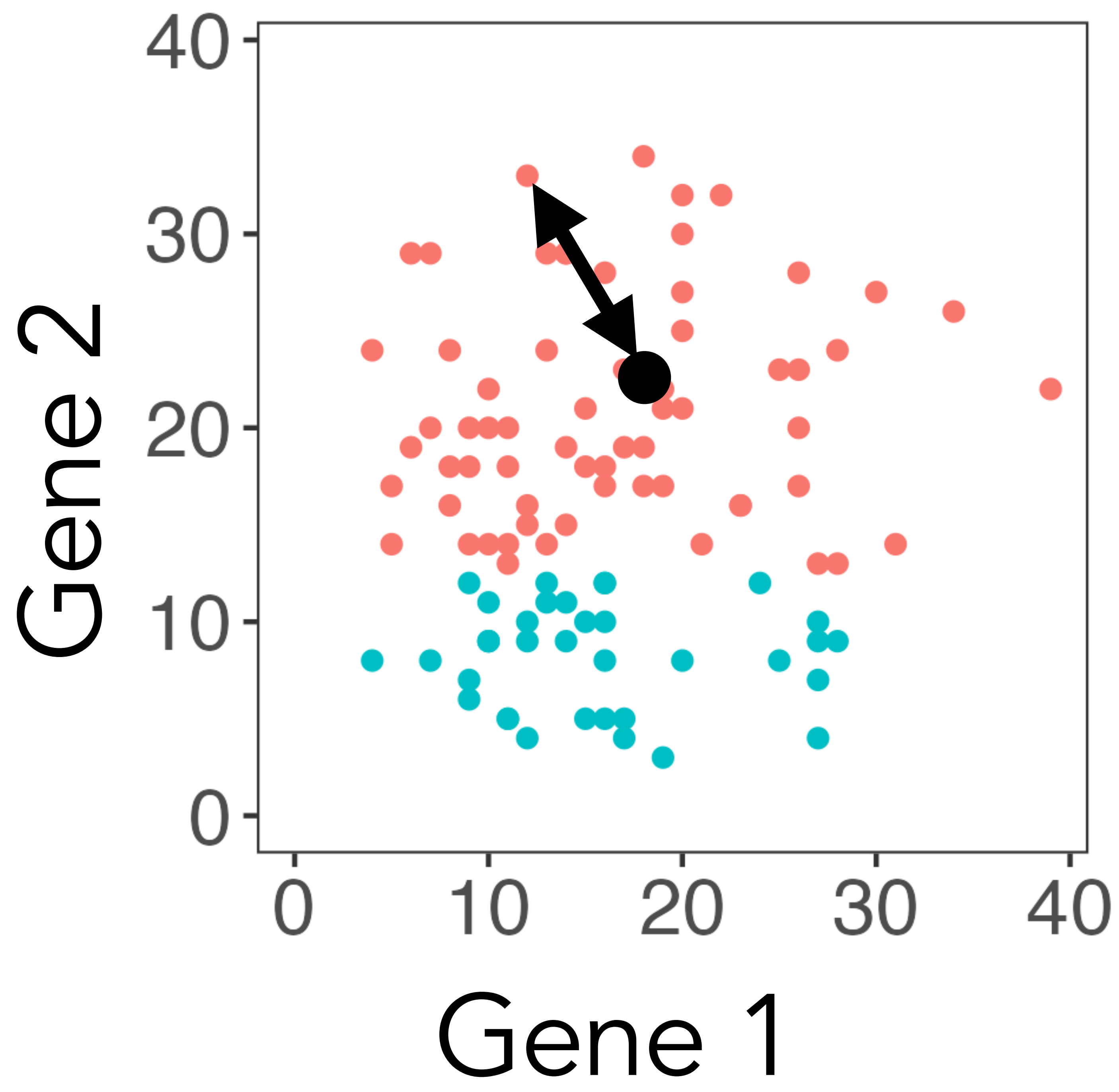
Goal: how many clusters are in this data?

For several values of k:

Step 1: fit a model with k clusters.

Step 2: evaluate model using a loss function.

Example 1: how many distinct cell types exist in a scRNA-seq dataset?



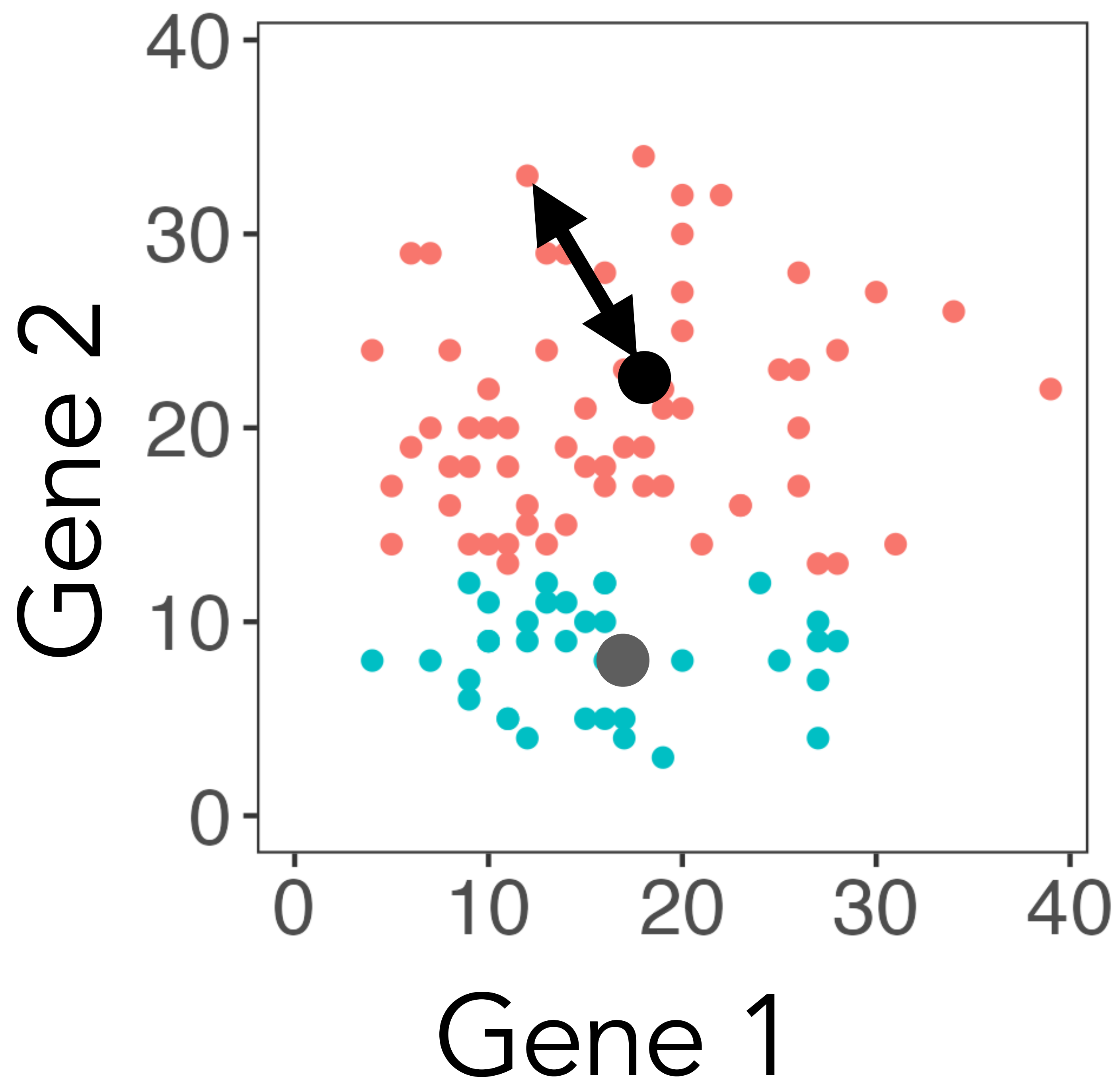
Goal: how many clusters are in this data?

For several values of k:

Step 1: fit a model with k clusters.

Step 2: evaluate model using a loss function.

Example 1: how many distinct cell types exist in a scRNA-seq dataset?



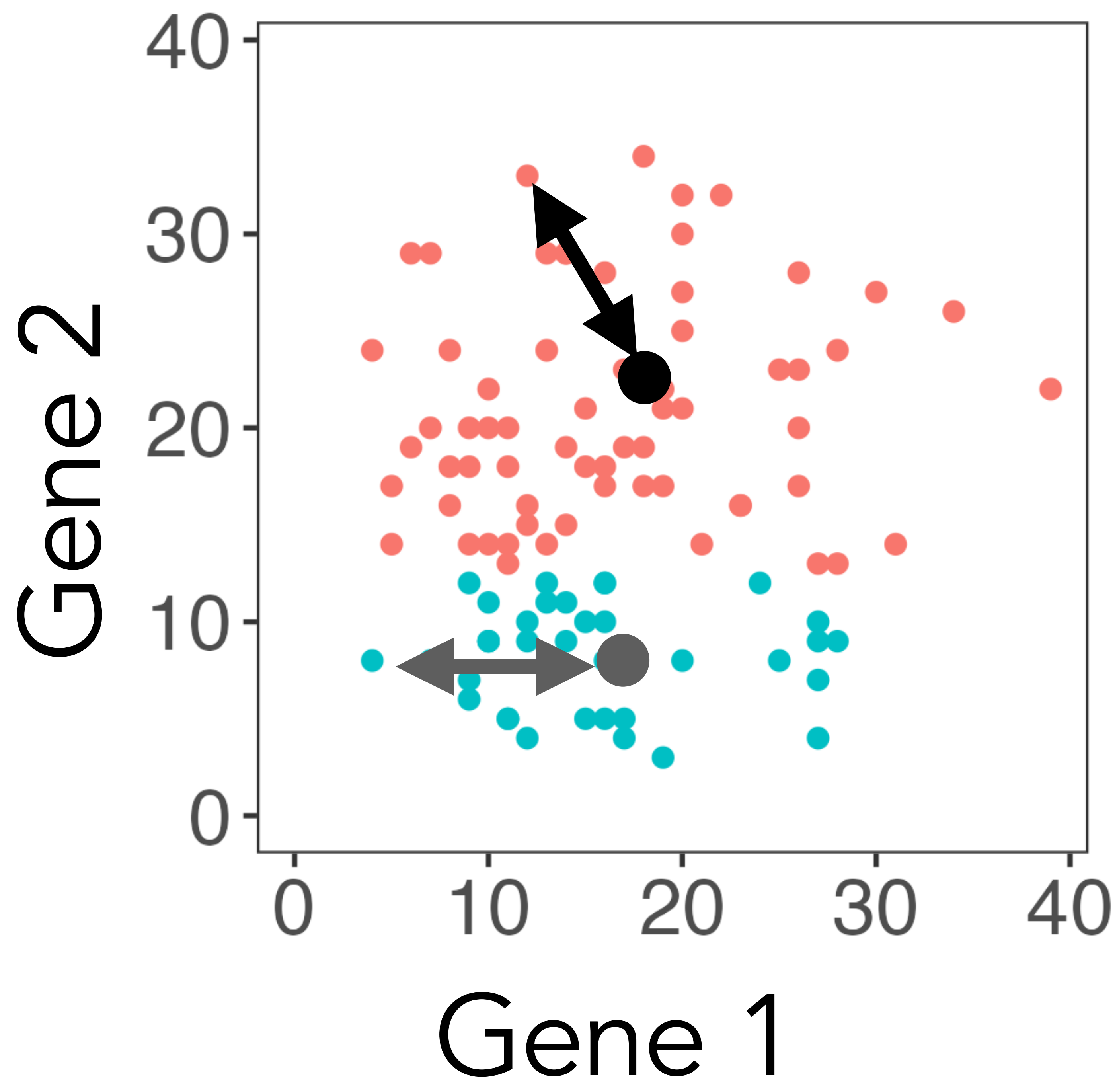
Goal: how many clusters are in this data?

For several values of k:

Step 1: fit a model with k clusters.

Step 2: evaluate model using a loss function.

Example 1: how many distinct cell types exist in a scRNA-seq dataset?



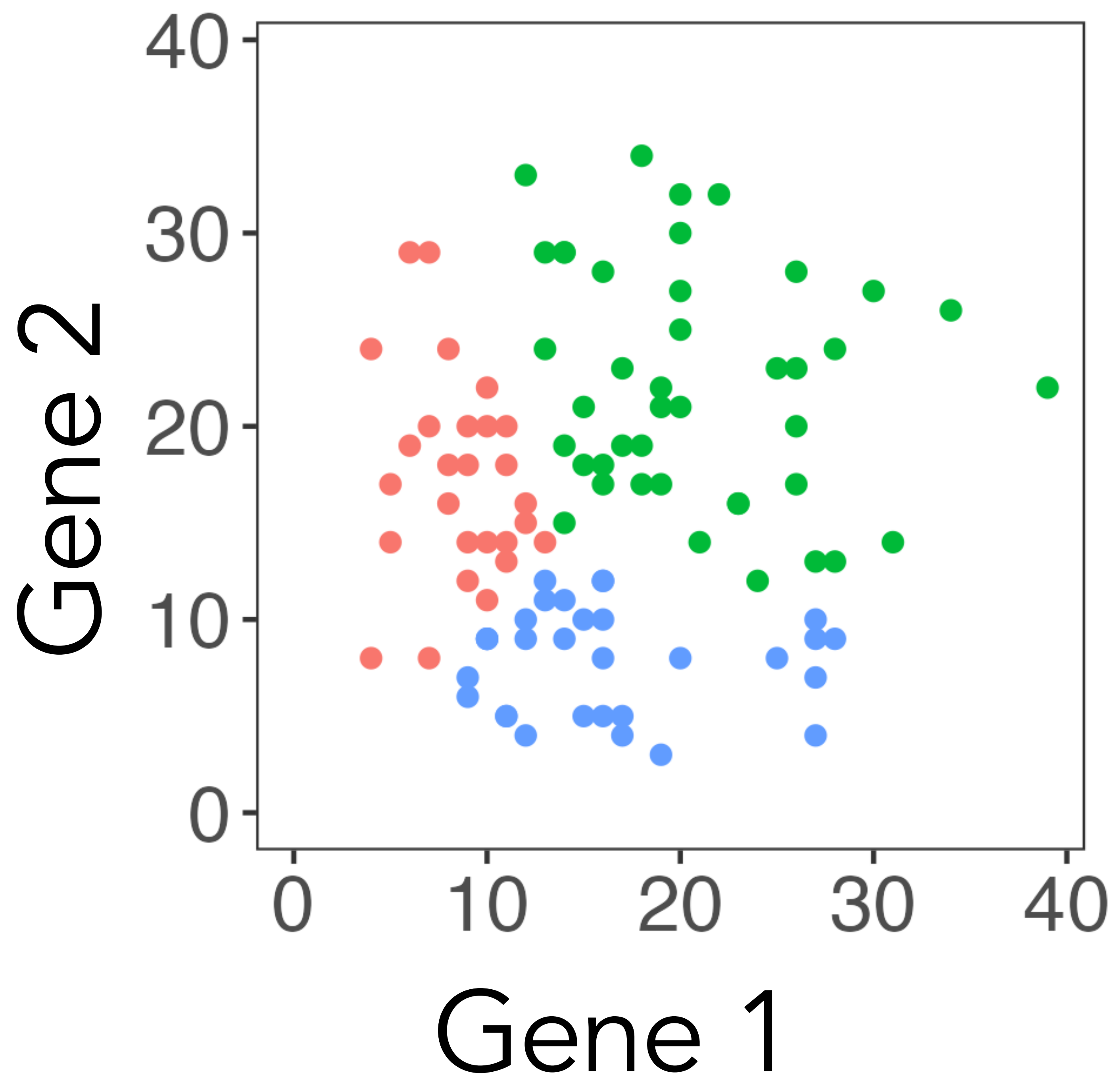
Goal: how many clusters are in this data?

For several values of k:

Step 1: fit a model with k clusters.

Step 2: evaluate model using a loss function.

Example 1: how many distinct cell types exist in a scRNA-seq dataset?



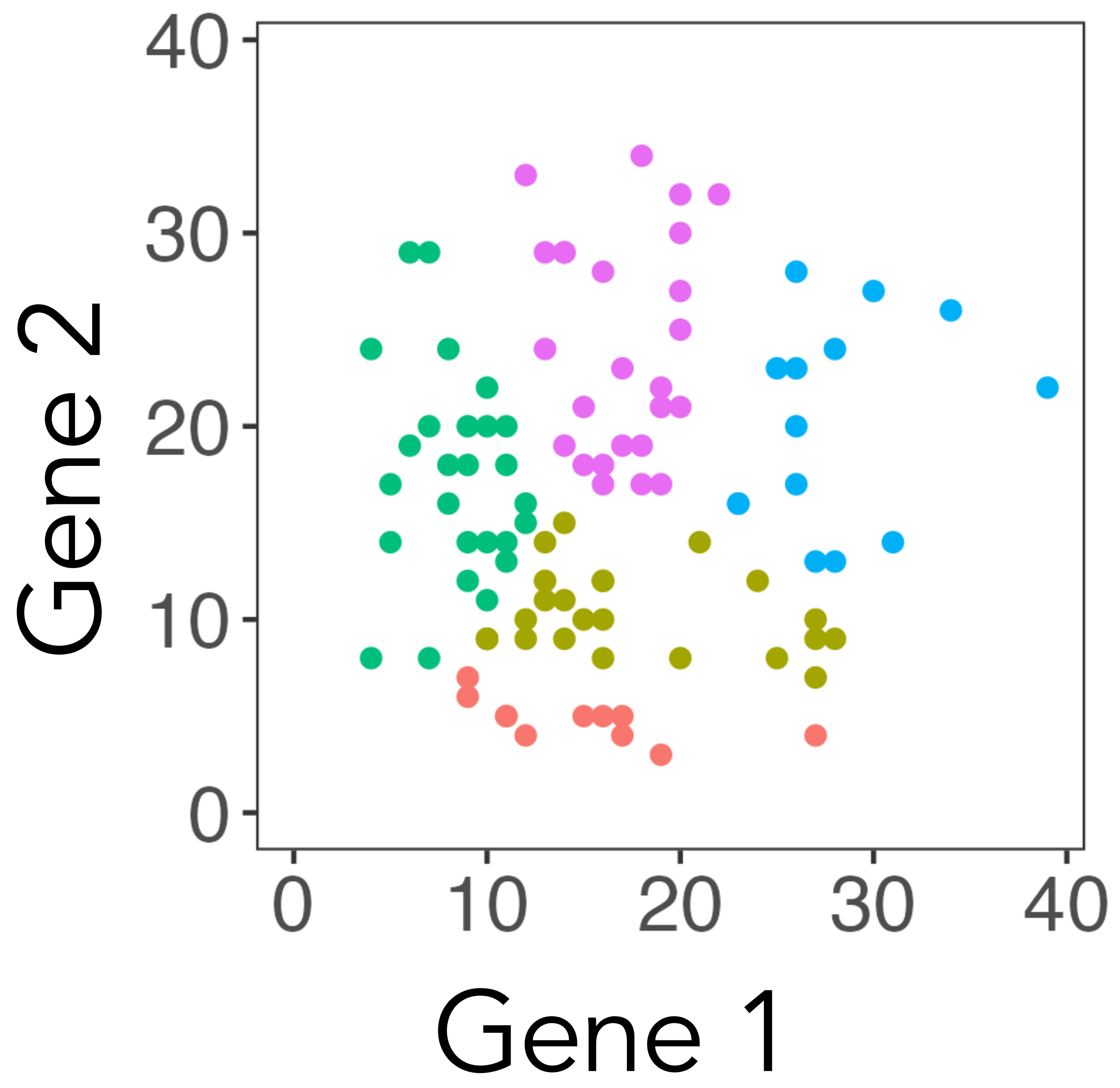
Goal: how many clusters are in this data?

For several values of k:

Step 1: fit a model with k clusters.

Step 2: evaluate model using a loss function.

Example 1: how many distinct cell types exist in a scRNA-seq dataset?



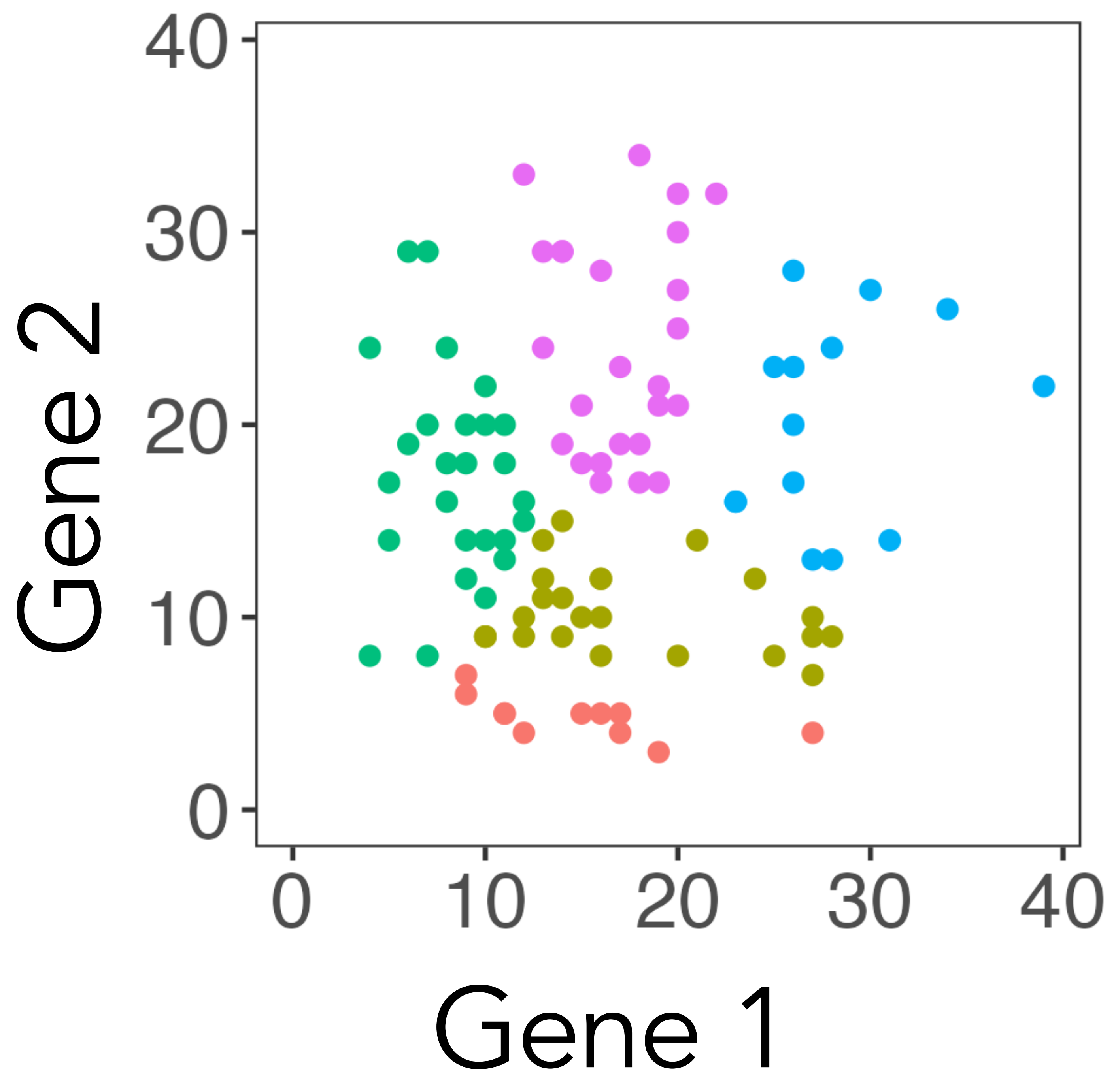
Goal: how many clusters are in this data?

For several values of k:

Step 1: fit a model with k clusters.

Step 2: evaluate model using a loss function.

Example 1: how many distinct cell types exist in a scRNA-seq dataset?

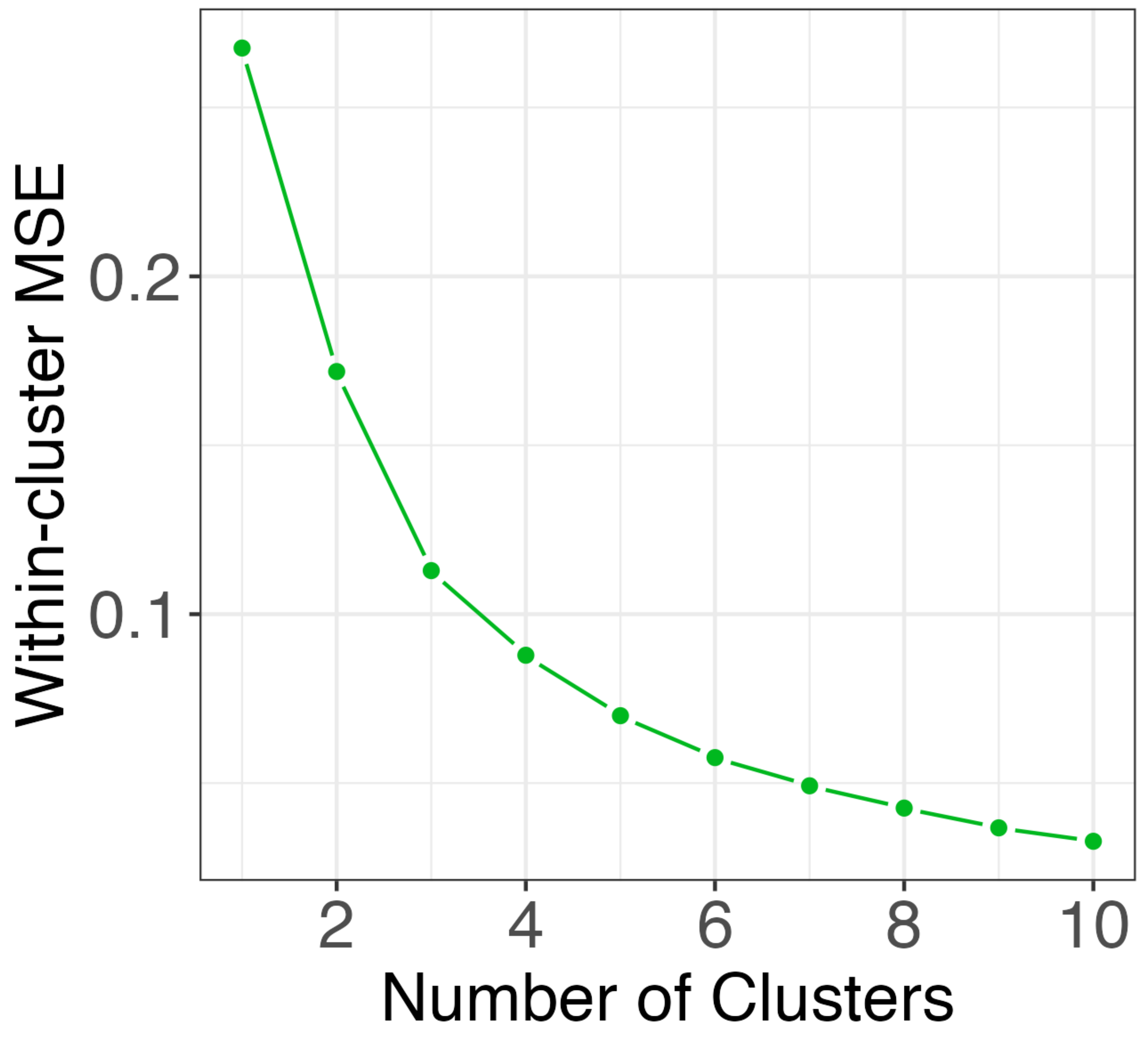


Goal: how many clusters are in this data?

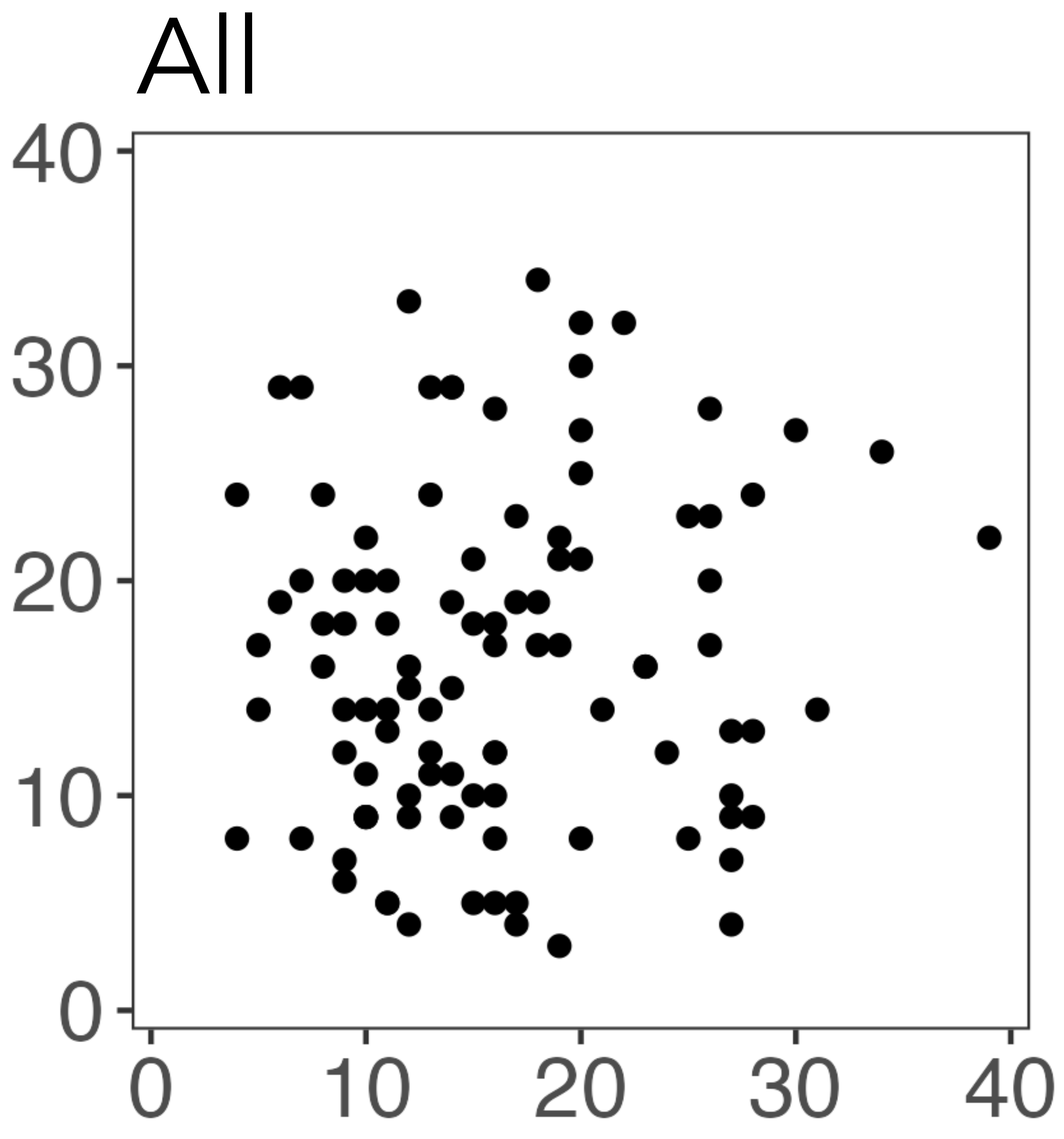
For several values of k:

Step 1: fit a model with k clusters.

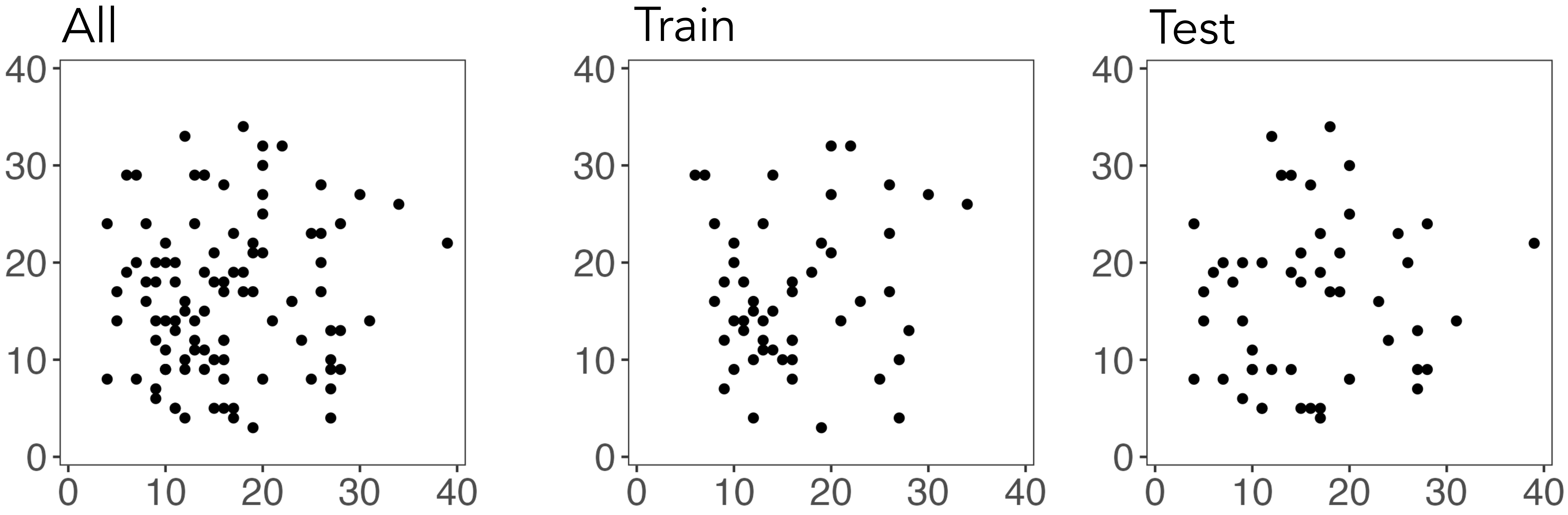
Step 2: evaluate model using a loss function.



Sample splitting cannot be used for example 1

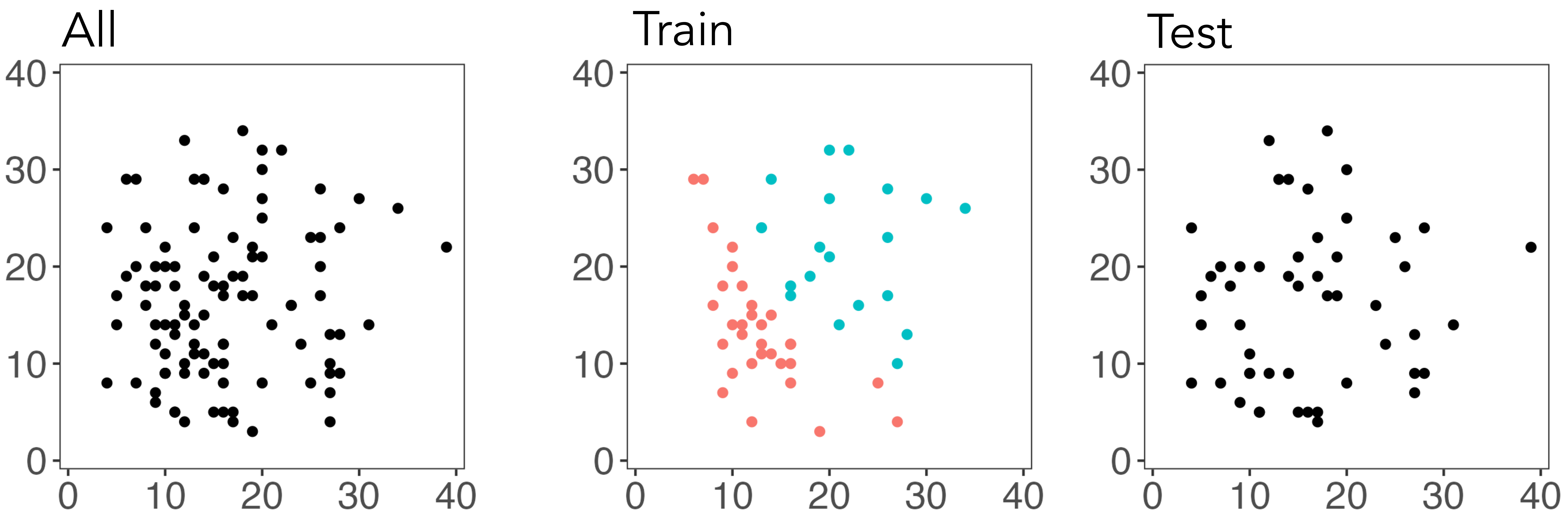


Sample splitting cannot be used for example 1



Step 1: split observations into train/test.

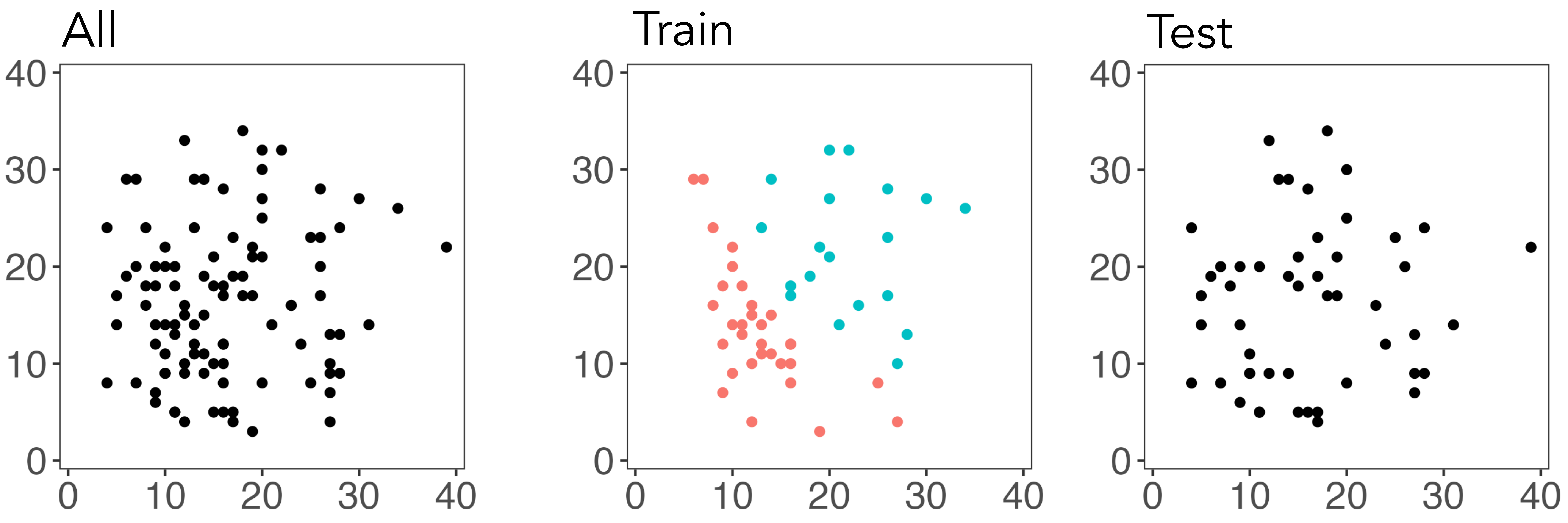
Sample splitting cannot be used for example 1



Step 1: split observations into train/test.

Step 2: cluster the training set.

Sample splitting cannot be used for example 1

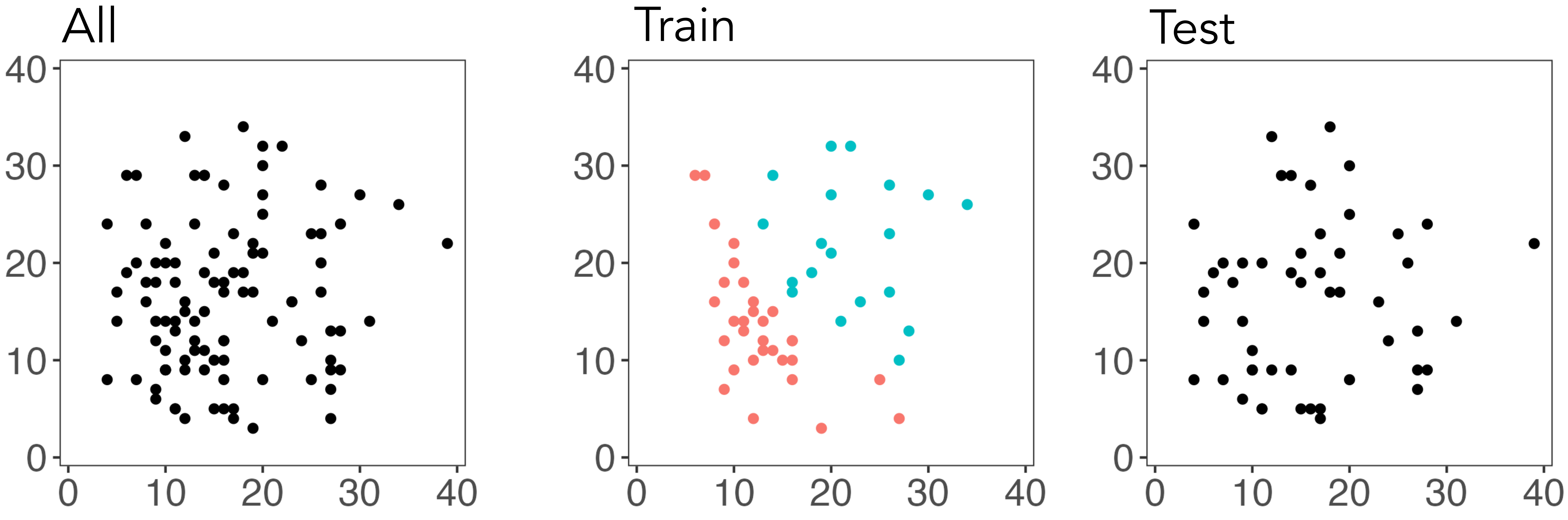


Step 1: split observations into train/test.

Step 2: cluster the training set.

Step 3: evaluate clusters using test set.

Sample splitting cannot be used for example 1



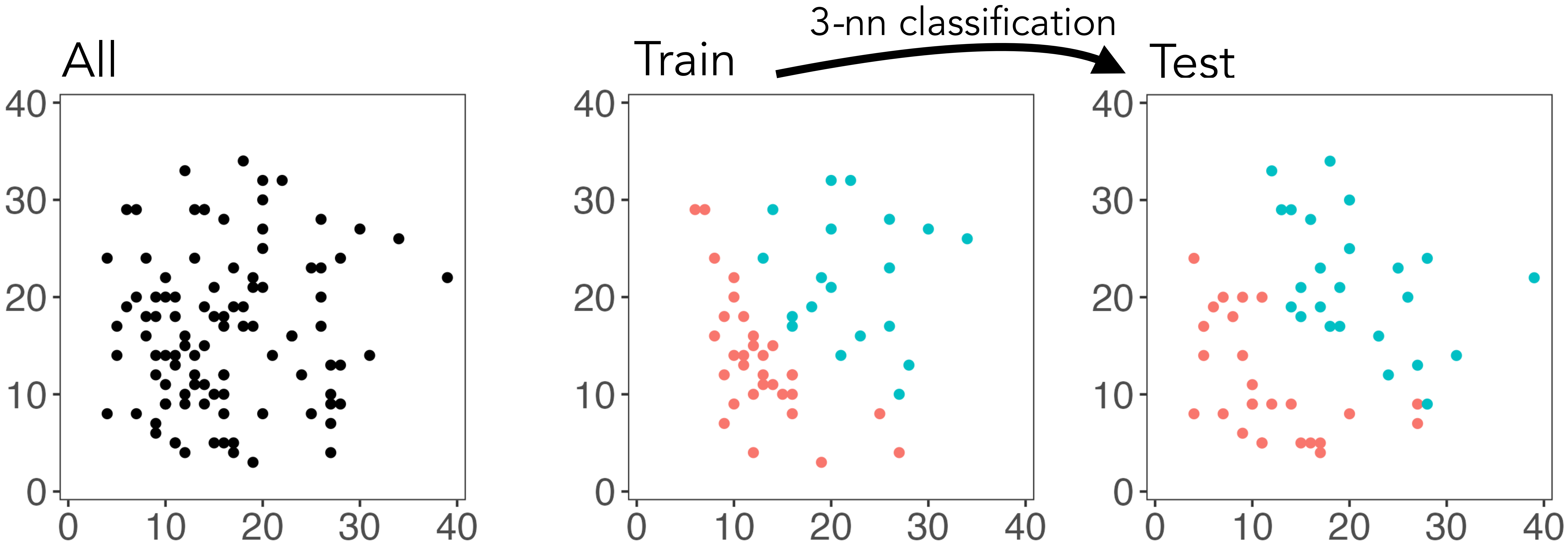
Step 1: split observations into train/test.

Step 2: cluster the training set.

Step 2.5: assign labels to observations in test set.

Step 3: evaluate clusters using test set.

Sample splitting cannot be used for example 1



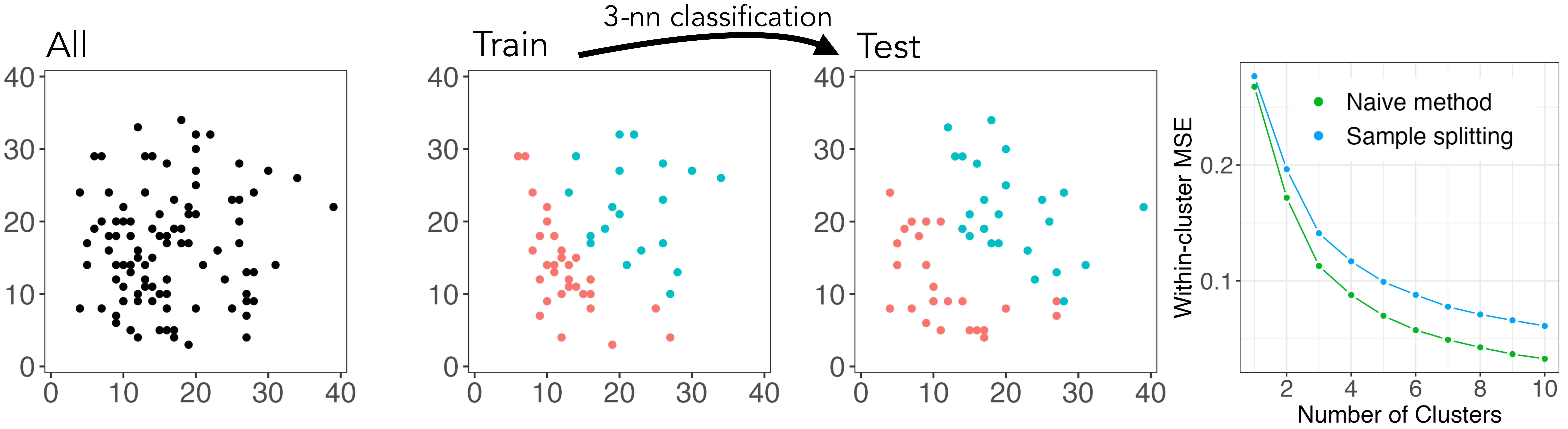
Step 1: split observations into train/test.

Step 2: cluster the training set.

Step 2.5: assign labels to observations in test set.

Step 3: evaluate clusters using test set.

Sample splitting cannot be used for example 1



Step 1: split observations into train/test.

Step 2: cluster the training set.

Step 2.5: assign labels to observations in test set.

Step 3: evaluate clusters using test set.

Example 1 remains a hard problem

Yu *et al. Genome Biology* (2022) 23:49
https://doi.org/10.1186/s13059-022-02622-0

Genome Biology

RESEARCH Open Access

Benchmarking clustering algorithms on estimating the number of cell types from single-cell RNA-sequencing data

Lijia Yu[1,2,3], Yue Cao[1,3], Jean Y. H. Yang[1,3] and Pengyi Yang[1,2,3*]

Abstract

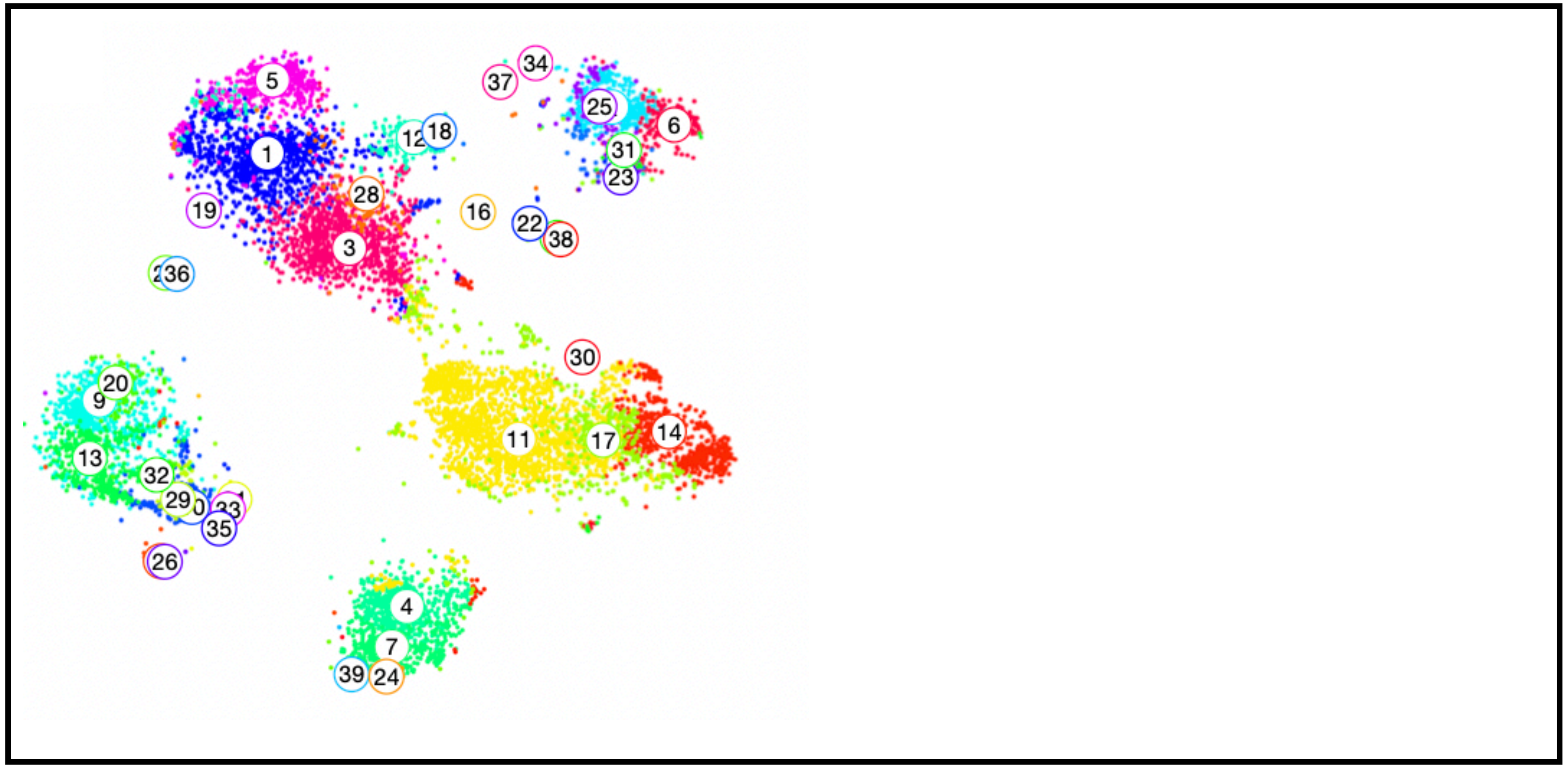
Background: A key task in single-cell RNA-seq (scRNA-seq) data analysis is to accurately detect the number of cell types in the sample, which can be critical for downstream analyses such as cell type identification. Various scRNA-seq data clustering algorithms have been specifically designed to automatically estimate the number of cell types through optimising the number of clusters in a dataset. The lack of benchmark studies, however, complicates the choice of the methods.

Example 2: which genes are differentially expressed across cell type?

A human liver cell atlas reveals heterogeneity and epithelial progenitors

Nadim Aizarani, Antonio Saviano, Sagar, Laurent Mailly, Sarah Durand, Josip S. Herman, Patrick Pessaux, Thomas F. Baumert & Dominic Grün

Nature **572**, 199–204 (2019) | Cite this article

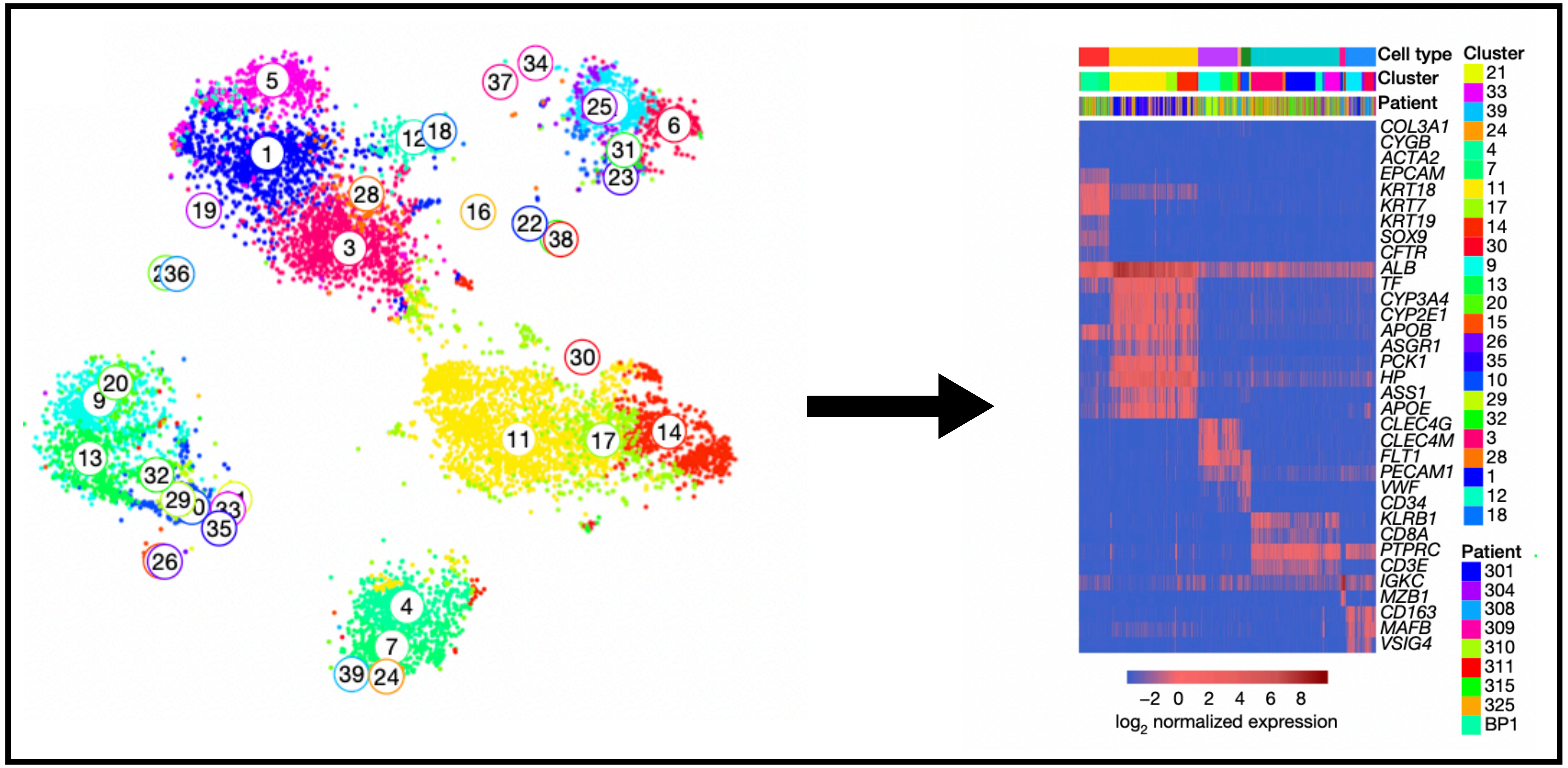


Example 2: which genes are differentially expressed across cell type?

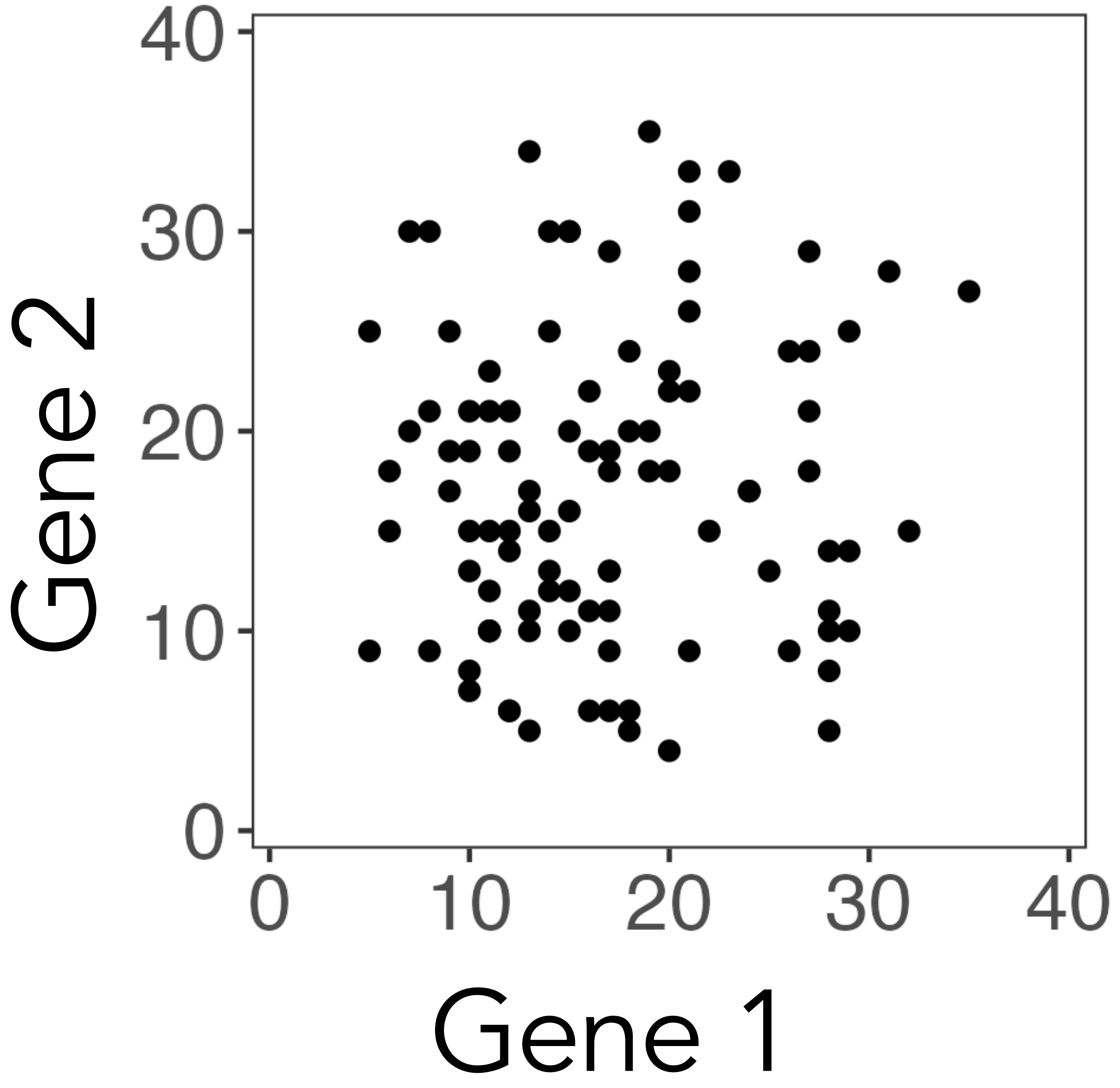
A human liver cell atlas reveals heterogeneity and epithelial progenitors

Nadim Aizarani, Antonio Saviano, Sagar, Laurent Mailly, Sarah Durand, Josip S. Herman, Patrick Pessaux, Thomas F. Baumert & Dominic Grün

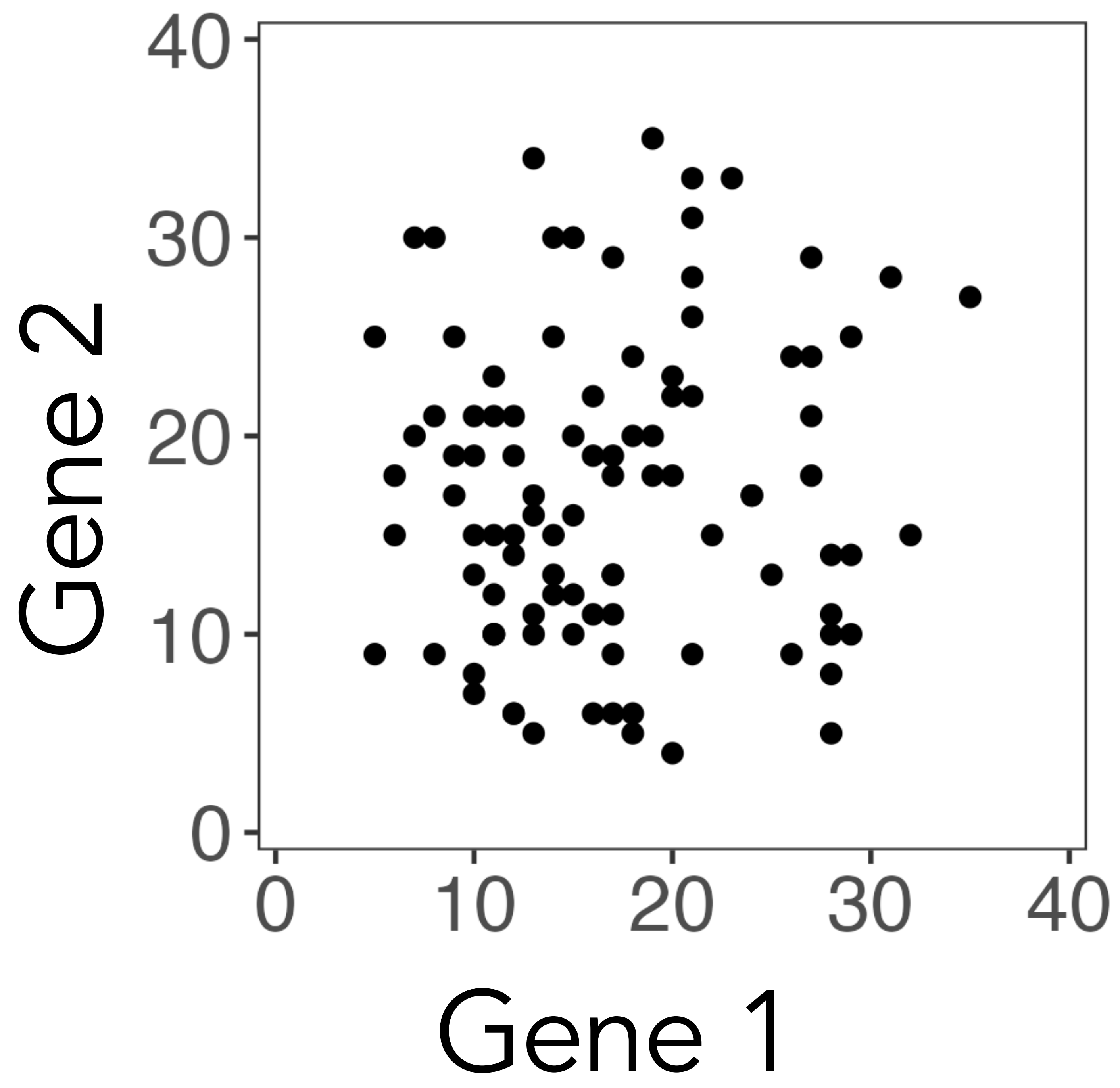
Nature **572**, 199–204 (2019) | Cite this article



Example 2: which genes are differentially expressed across cell types?

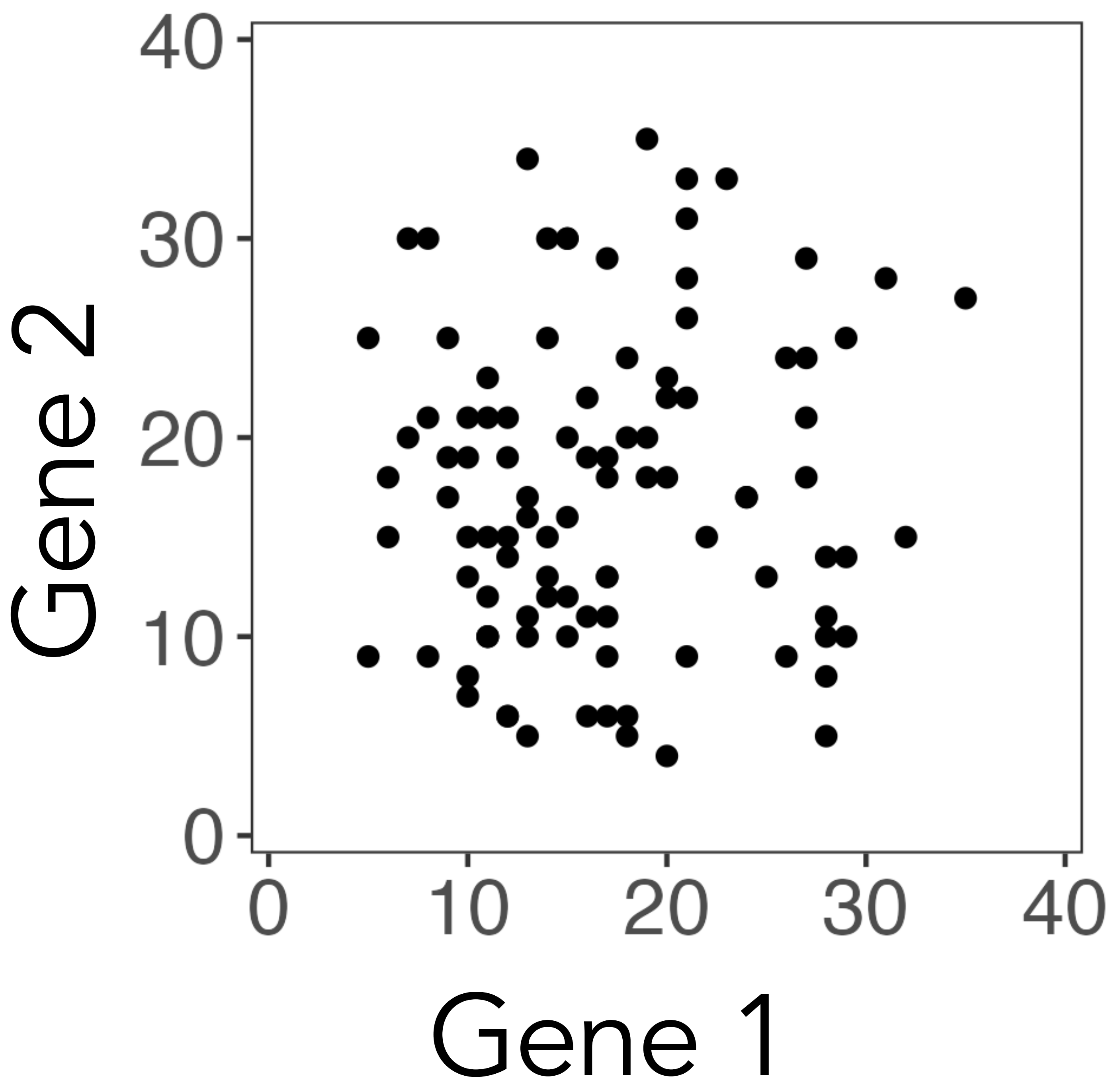


Example 2: which genes are differentially expressed across cell types?



Naive method:

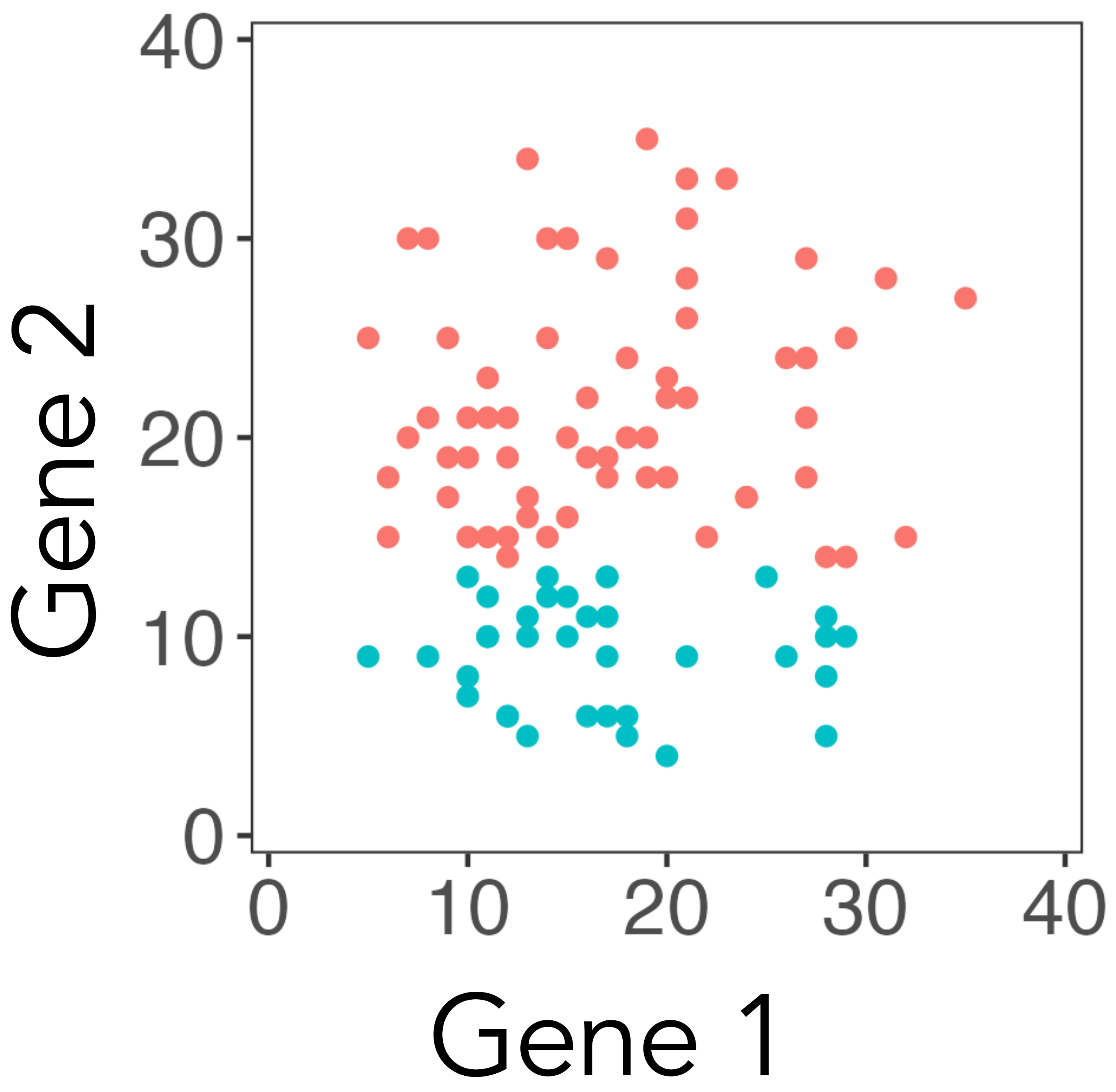
Example 2: which genes are differentially expressed across cell types?



Naive method:

Step 1: cluster the observations.

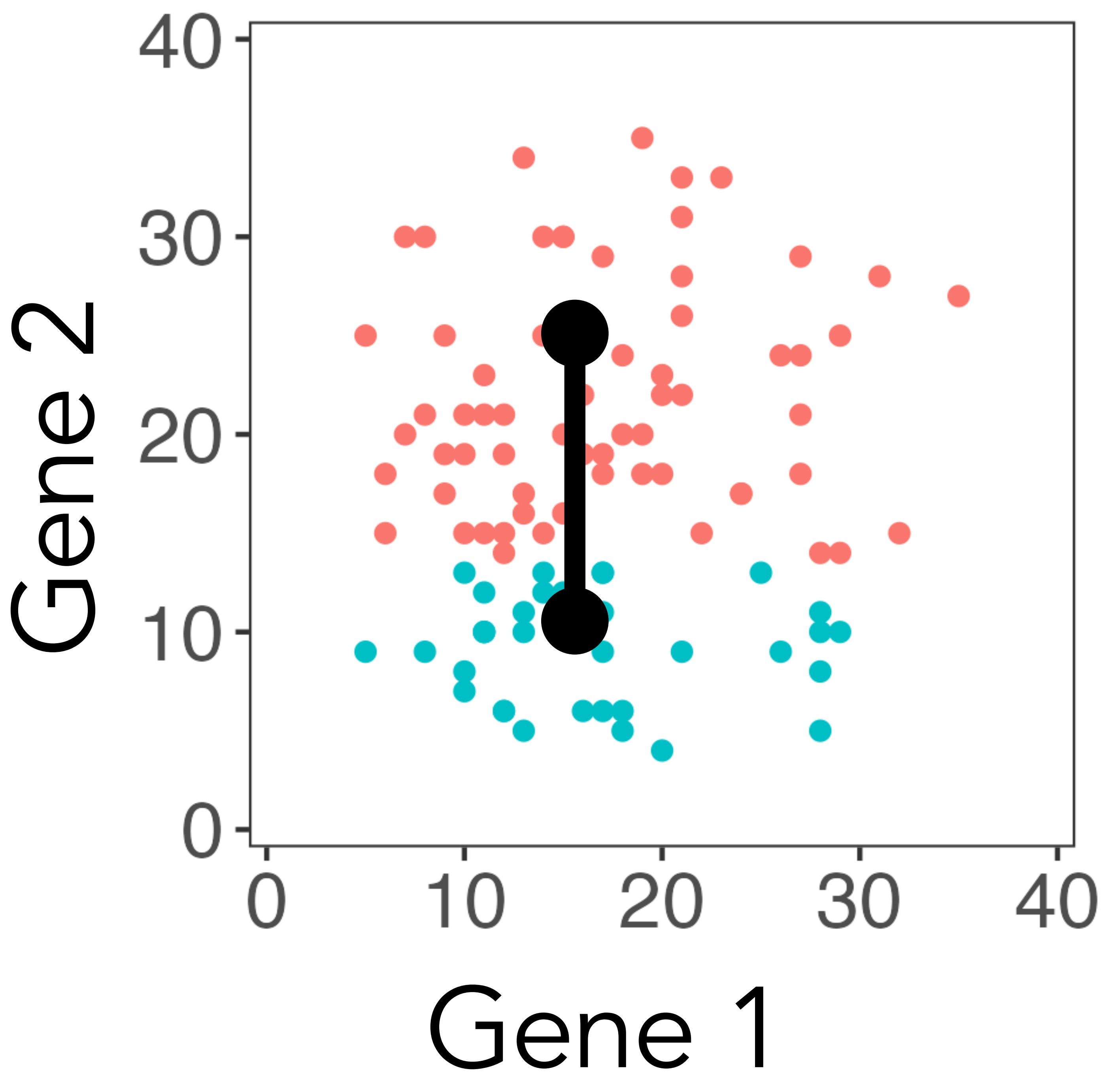
Example 2: which genes are differentially expressed across cell types?



Naive method:

Step 1: cluster the observations.

Example 2: which genes are differentially expressed across cell types?

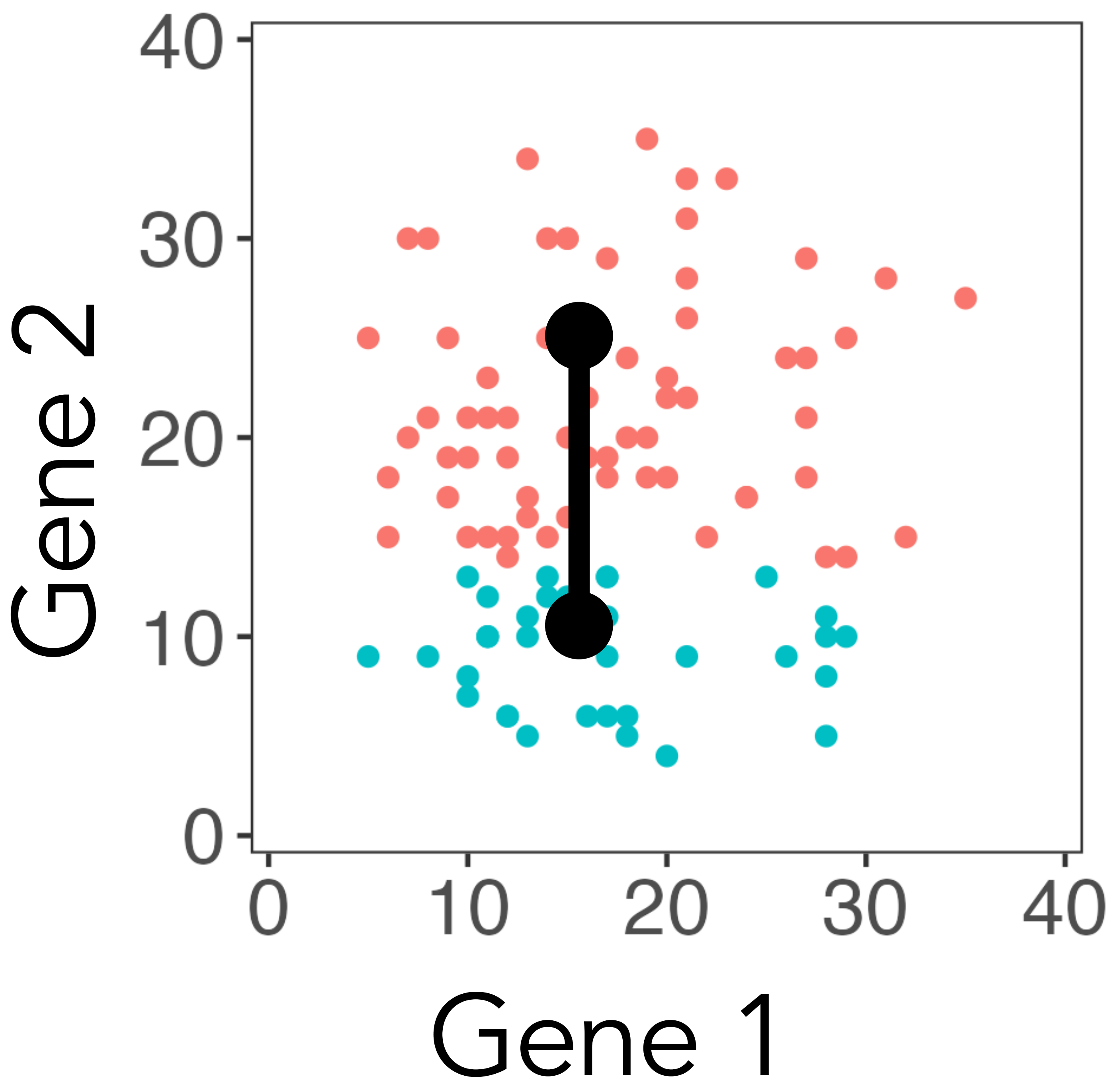


Naive method:

Step 1: cluster the observations.

Step 2: test for differential expression of Gene 2 across the two clusters using a t-test.

Example 2: which genes are differentially expressed across cell types?



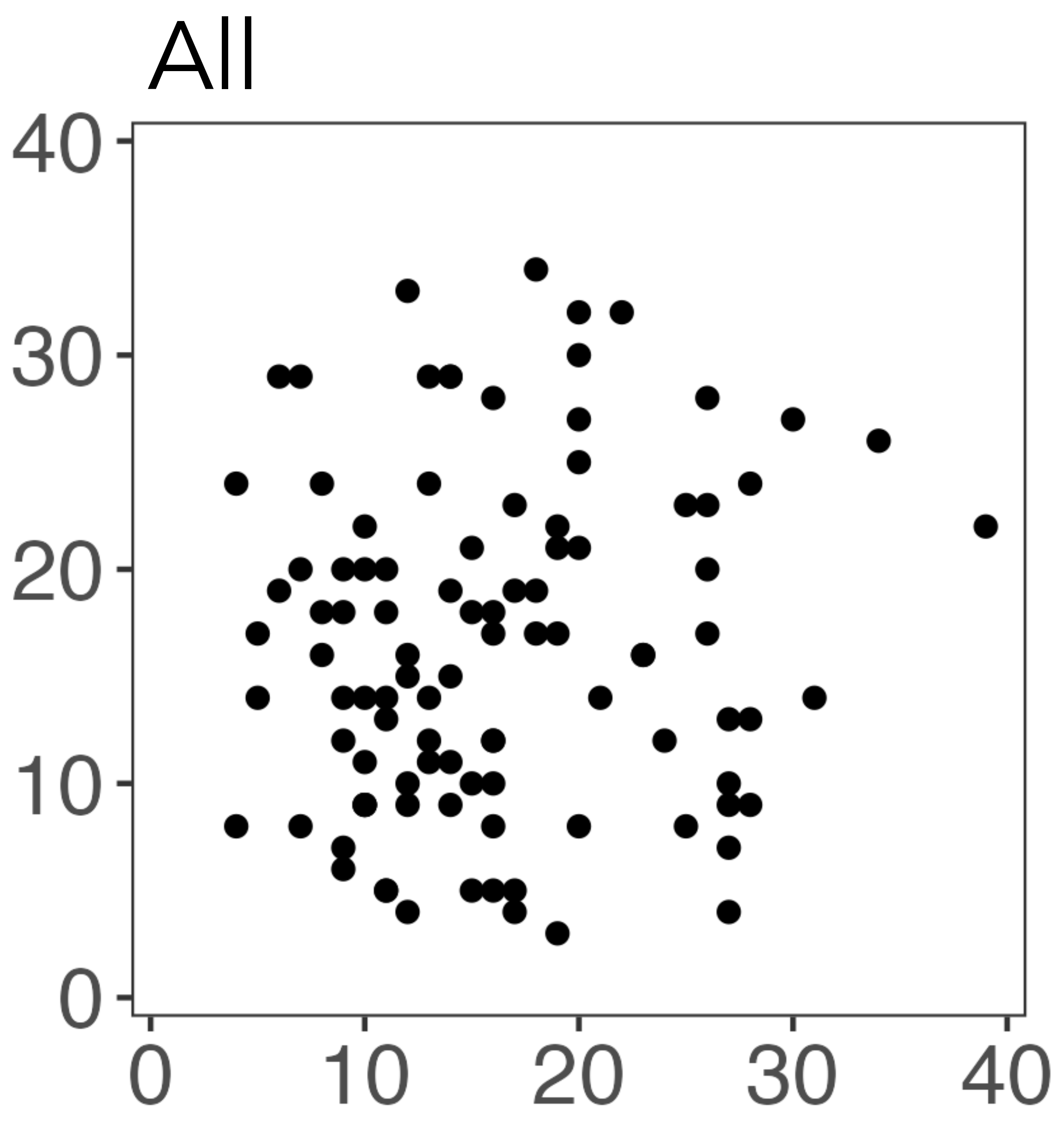
$p < 10^{-10}$ 😱

Naive method:

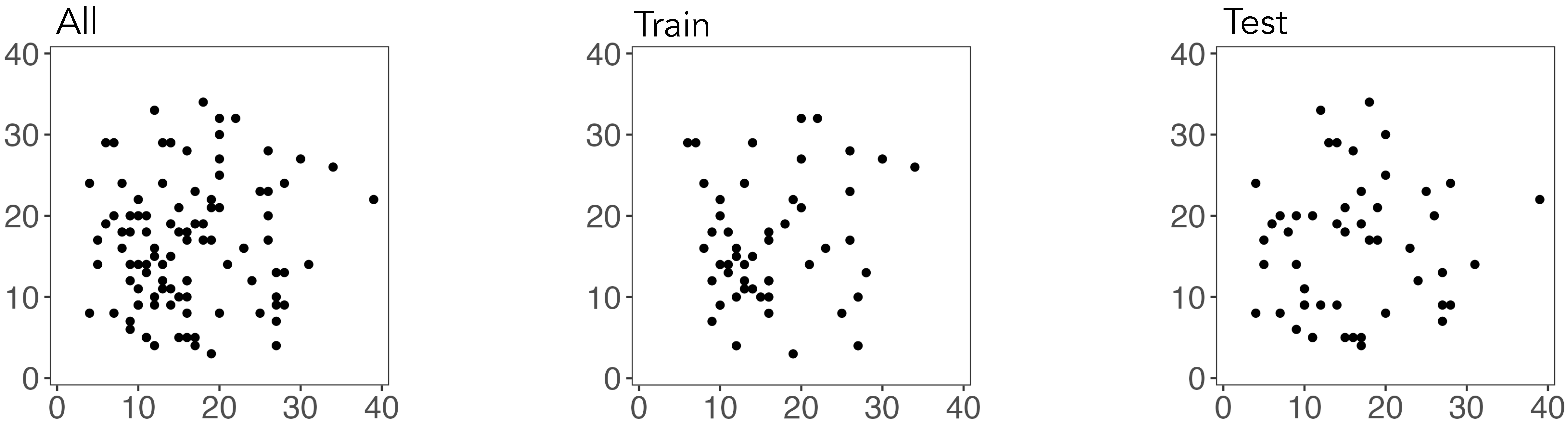
Step 1: cluster the observations.

Step 2: test for differential expression of Gene 2 across the two clusters using a t-test.

Sample splitting cannot be used for example 2

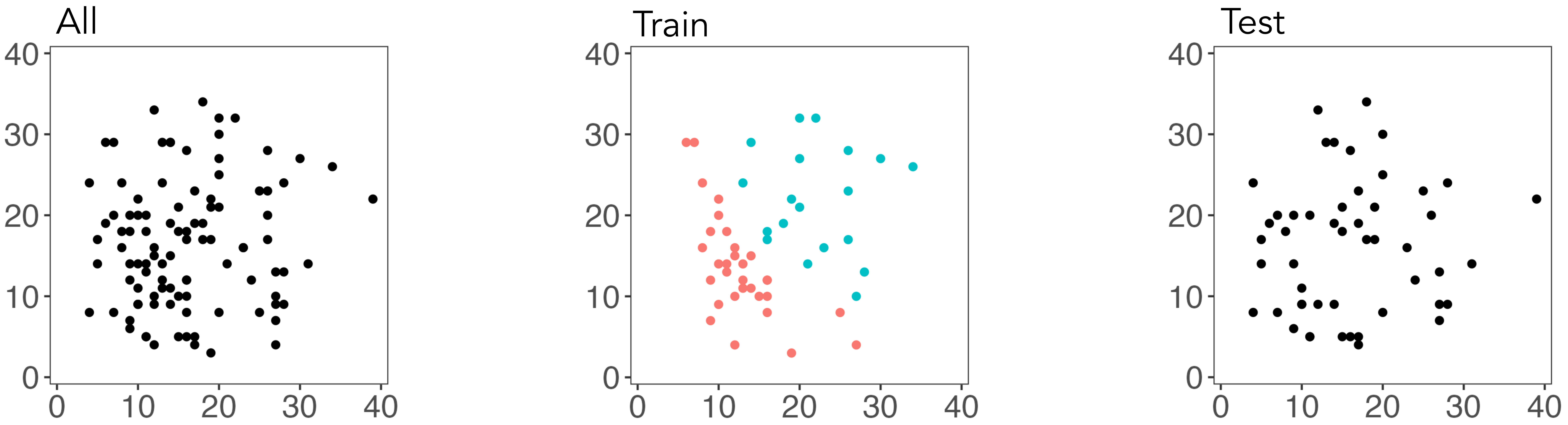


Sample splitting cannot be used for example 2



Step 1: split observations into train/test.

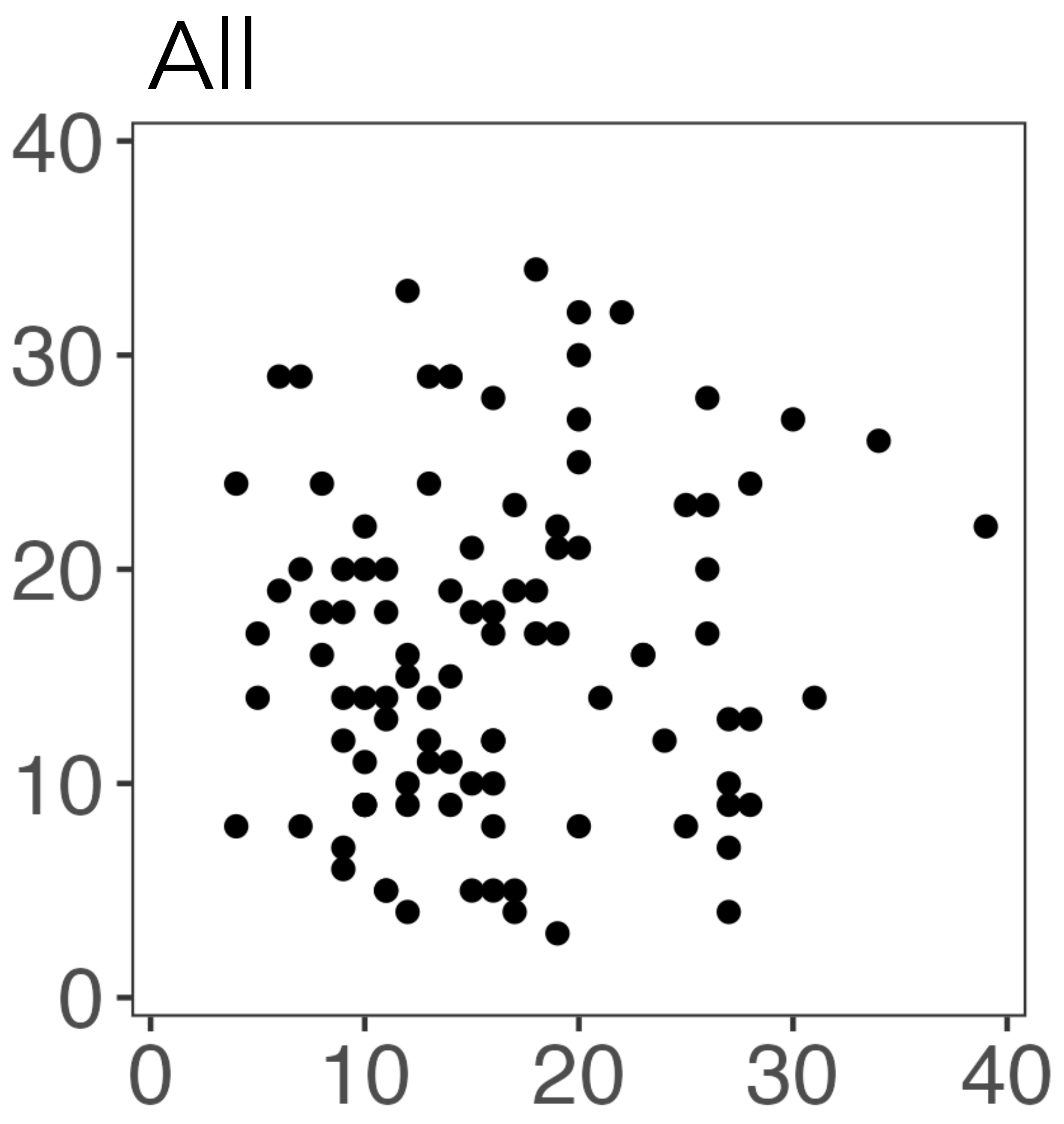
Sample splitting cannot be used for example 2

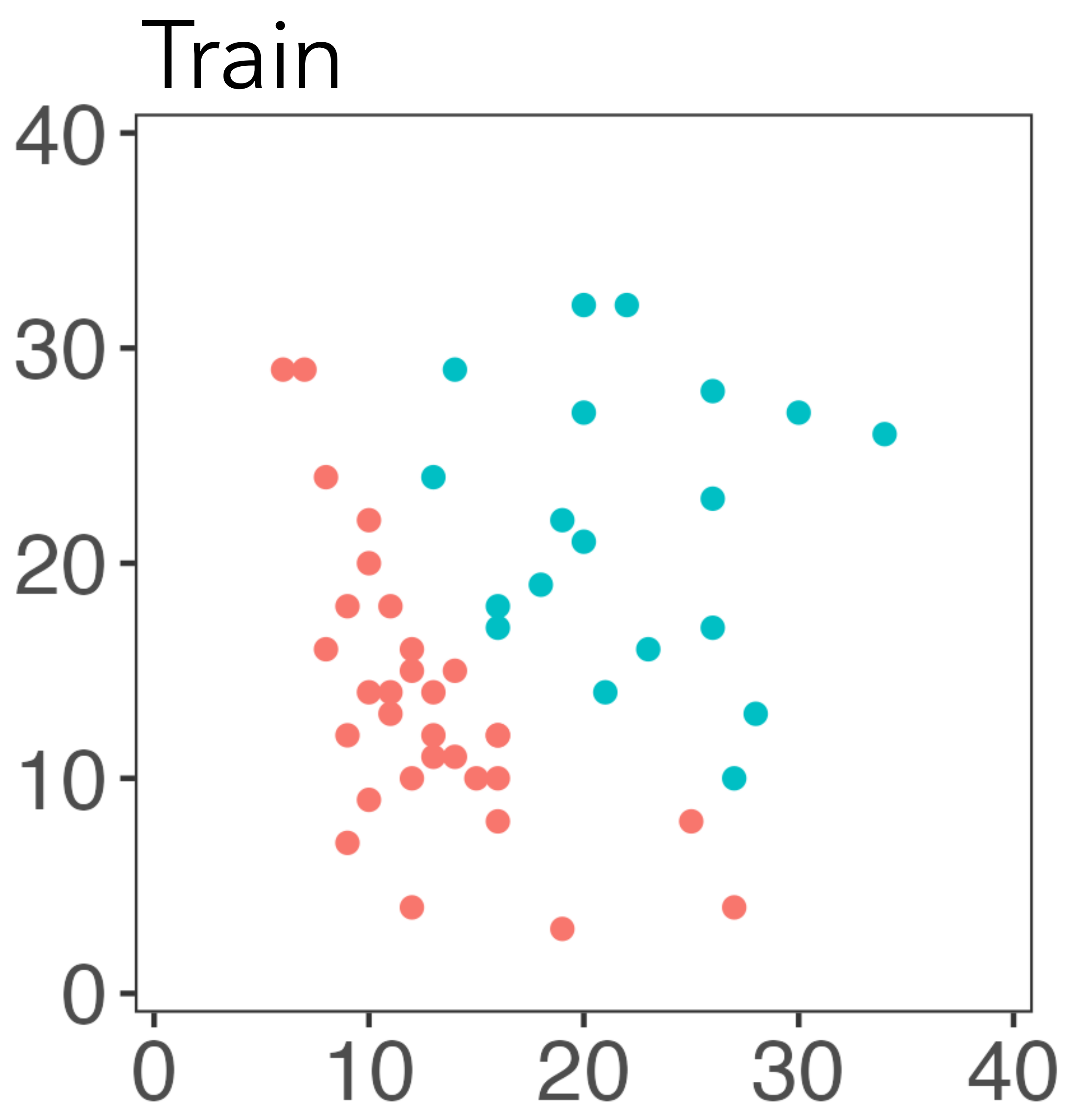


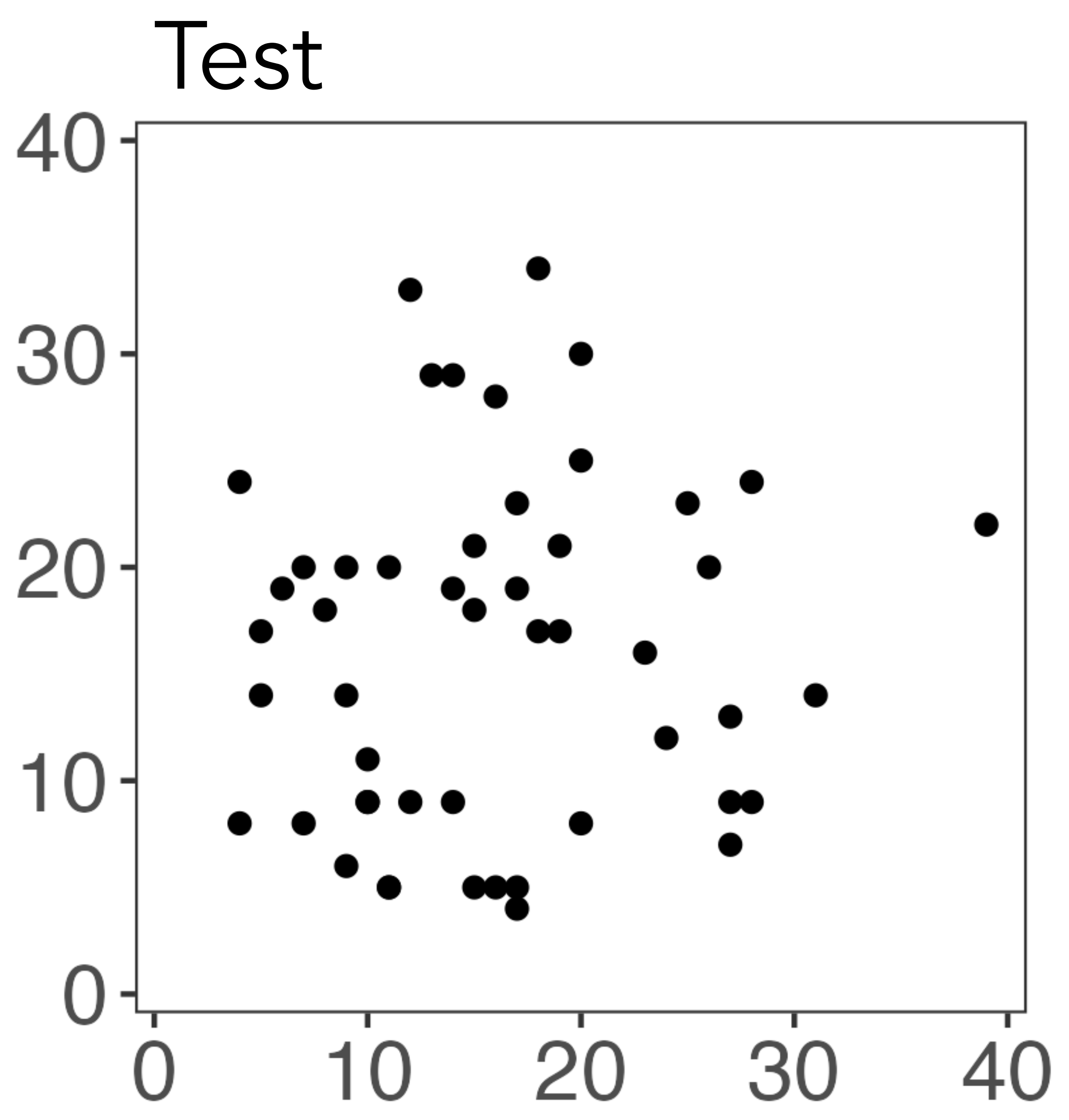
Step 1: split observations into train/test.

Step 2: cluster the training set.

Sample splitting cannot be used for example 2





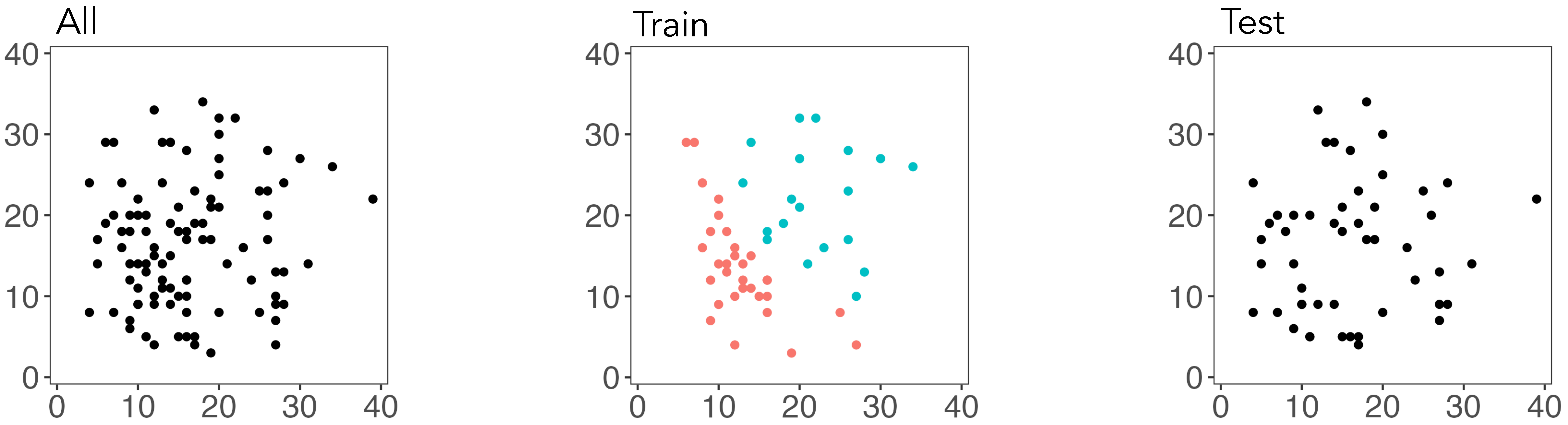


Step 1: split observations into train/test.

Step 2: cluster the training set.

Step 3: test for difference in means using test set.

Sample splitting cannot be used for example 2



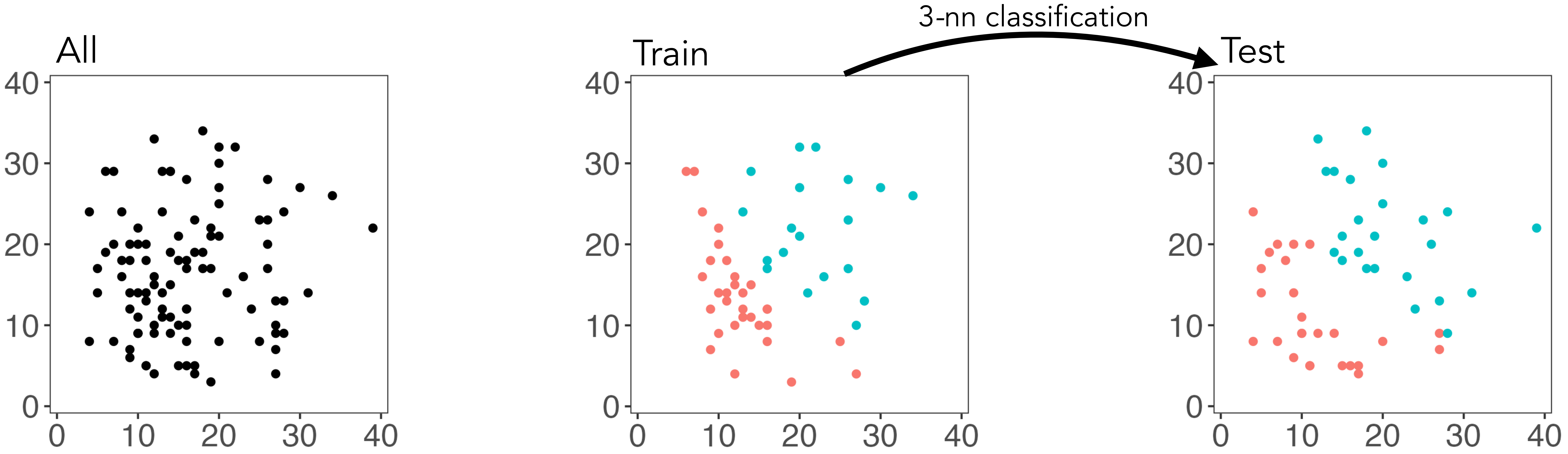
Step 1: split observations into train/test.

Step 2: cluster the training set.

Step 2.5: assign labels to observations in test set.

Step 3: test for difference in means using test set.

Sample splitting cannot be used for example 2



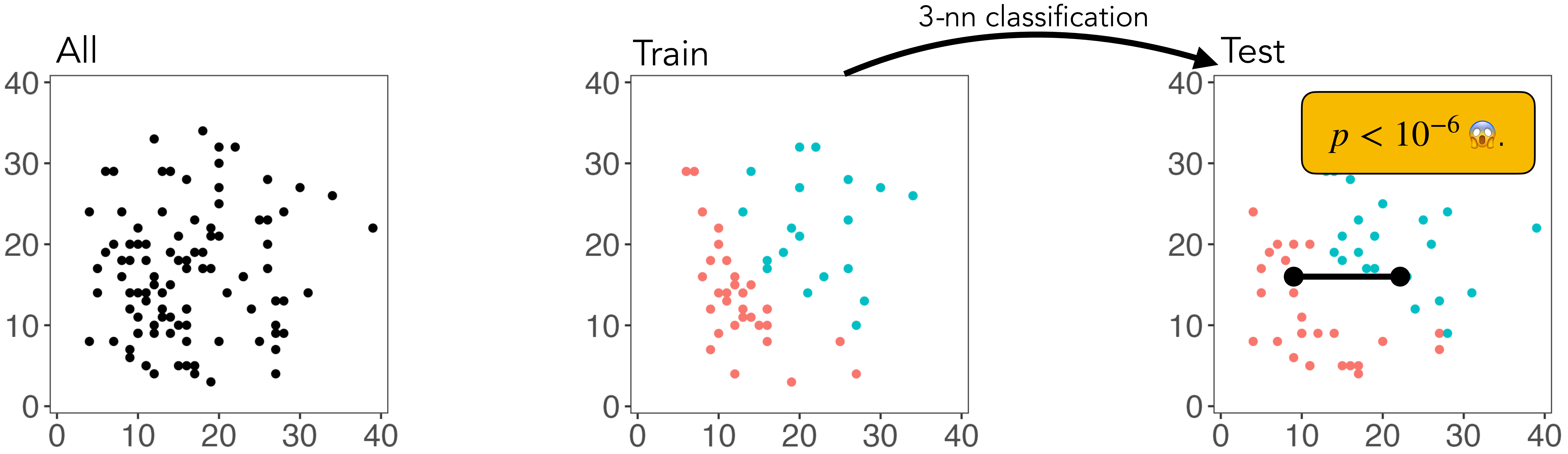
Step 1: split observations into train/test.

Step 2: cluster the training set.

Step 2.5: assign labels to observations in test set.

Step 3: test for difference in means using test set.

Sample splitting cannot be used for example 2



Step 1: split observations into train/test.

Step 2: cluster the training set.

Step 2.5: assign labels to observations in test set.

Step 3: test for difference in means using test set.

Example 2 remains a hard problem

Lähnemann *et al. Genome Biology* (2020) 21:31
https://doi.org/10.1186/s13059-020-1926-6

Genome Biology

REVIEW Open Access

Eleven grand challenges in single-cell data science

David Lähnemann[1,2,3], Johannes Köster[1,4], Ewa Szczurek[5], Davis J. McCarthy[6,7], Stephanie C. Hicks[8], Mark D. Robinson[9], Catalina A. Vallejos[10,11], Kieran R. Campbell[12,13,14], Niko Beerenwinkel[15,16], Ahmed Mahfouz[17,18], Luca Pinello[19,20,21], Pavel Skums[22], Alexandros Stamatakis[23,24], Camille Stephan-Otto Attolini[25], Samuel Aparicio[13,26], Jasmijn Baaijens[27], Marleen Balver
Buys de Barbanson[29,30,31], Antonio Cappuccio[32], Giacomo Corleone[33], Bas E. Dutilh[28,34],
Maria Florescu[29,30,31], Victor Guryev[35], Rens Holmer[36], Katharina Jahn[15,16], Thamar Jessur
Emma M. Keizer[37], Indu Khatri[38], Szymon M. Kielbasa[39], Jan O. Korbel[40], Alexey M. Kozlov
Tzu-Hao Kuo[3], Boudewijn P.F. Lelieveldt[41,42], Ion I. Mandoiu[43], John C. Marioni[44,45,46],
Tobias Marschall[47,48], Felix Mölder[1,49], Amir Niknejad[50,51], Lukasz Raczkowski[5], Marcel Rei
Jeroen de Ridder[29,30], Antoine-Emmanuel Saliba[52], Antonios Somarakis[42], Oliver Stegle[4
Fabian J. Theis[54], Huan Yang[55], Alex Zelikovsky[56,57], Alice C. McHardy[3], Benjamin J. Rapha
Sohrab P. Shah[59] and Alexander Schönhuth[27,28*]

Status

Currently, the vast majority of differential expression detection methods assume that the groups of cells to be compared are known in advance (e.g., experimental conditions or cell types). However, current analysis pipelines typically rely on clustering or cell type assignment to identify such groups, before downstream differential analysis is performed, without propagating the uncertainty in these assignments or accounting for the double use of data (clustering, differential testing between clusters).

Typical practice is to ignore this problem

Seurat 4.0.6 | Install | Get started | Vignettes | Extensions | FAQ | News | Reference | Archive

Gene expression markers of identity classes

Source: `R/generics.R`, `R/differential_expression.R`

Finds markers (differentially expressed genes) for identity classes

```
FindMarkers(object, ...)
```

Details

p-value adjustment is performed using bonferroni correction based on the total number of genes in the dataset. Other correction methods are not recommended, as Seurat pre-filters genes using the arguments above, reducing the number of tests performed. Lastly, as Aaron Lun has pointed out, p-values should be interpreted cautiously, as the genes used for clustering are the same genes tested for differential expression.

Outline

1. Motivation: settings where sample splitting doesn't work
2. **Poisson thinning**
3. Data thinning
4. Application to human fetal cell atlas data
5. Application to cardiomyocyte differentiation data
6. Ongoing work

Reminder: sample splitting does not help us with our motivating examples

scRNA-seq dataset

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

Train

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	28

Test

	Gene 1	Gene 2
Cell 3	11	5
Cell 4	22	21

Reminder: sample splitting does not help us with our motivating examples

scRNA-seq dataset

	Gene 1	**Gene 2**
Cell 1	18	6
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

Train

	Gene 1	**Gene 2**
Cell 1	18	6
Cell 2	31	28

Estimating clusters on training set

Test

	Gene 1	**Gene 2**
Cell 3	11	5
Cell 4	22	21

Reminder: sample splitting does not help us with our motivating examples

scRNA-seq dataset

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

Train

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	28

Estimating clusters on training set

Test

	Gene 1	Gene 2
Cell 3	11	5
Cell 4	22	21

does not yield cluster assignments for test set.

An alternative: Poisson thinning

X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

An alternative: Poisson thinning

X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

$X^{(1)}$

	Gene 1	Gene 2
Cell 1	14	1
Cell 2	10	6
Cell 3	5	17
Cell 4	6	25

$X^{(2)}$

	Gene 3	Gene 4
Cell 1	4	5
Cell 2	21	2
Cell 3	6	14
Cell 4	16	9

An alternative: Poisson thinning

X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	$\boxed{8}$
Cell 3	11	31
Cell 4	22	34

X_{ij}

$X^{(1)}$

	Gene 1	Gene 2
Cell 1	14	1
Cell 2	[illegible]	$\boxed{6}$
Cell 3	5	17
Cell 4	6	25

$X^{(2)}$

	Gene 3	Gene 4
Cell 1	[illegible]	5
Cell 2	21	$\boxed{2}$
Cell 3	6	14
Cell 4	16	9

An alternative: Poisson thinning

X

	Gene 1	Gene 2
Cell 1	18	[illegible]
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

X_{ij}

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, \epsilon)$$

$X^{(1)}$

	Gene 1	Gene 2
[illegible]	14	1
[illegible]	[illegible]	6
Cell 3	5	17
Cell 4	6	25

$$X_{ij}^{(2)} := X_{ij} - X_{ij}^{(1)}$$

$X^{(2)}$

	Gene 3	Gene 4
Cell 1	[illegible]	5
Cell 2	21	2
Cell 3	6	14
Cell 4	16	9

An alternative: Poisson thinning

X

	Gene 1	Gene 2
Cell 1	18	[illegible]
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

X_{ij}

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, \epsilon)$$

$X^{(1)}$

	Gene 1	Gene 2
Cell 1	14	1
Cell 2	[illegible]	6
Cell 3	5	17
Cell 4	6	25

$$X_{ij}^{(2)} := X_{ij} - X_{ij}^{(1)}$$

$X^{(2)}$

	Gene 3	Gene 4
Cell 1	[illegible]	5
Cell 2	21	2
Cell 3	6	14
Cell 4	16	9

If $X_{ij} \sim \text{Poisson}(\Lambda_{ij})$, then:

1. $X_{ij}^{(1)} \sim \text{Poisson}(\epsilon\Lambda_{ij})$
2. $X_{ij}^{(2)} \sim \text{Poisson}((1-\epsilon)\Lambda_{ij})$
3. $X_{ij}^{(1)} \perp\!\!\!\perp X_{ij}^{(2)}$

A very well-known result.

An alternative: Poisson thinning

X

	Gene 1	Ge
Cell 1	18	[illegible]
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

X_{ij}

$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, \epsilon)$

$X^{(1)}$

	ene 1	Gene 2
[illegible]	14	1
[illegible]	[illegible]	6
Cell 3	5	17
Cell 4	6	25

Estimate clusters.

$X^{(2)}$

$X_{ij}^{(2)} := X_{ij} - X_{ij}^{(1)}$

	Gene 3	Gene 4
ell 1	[illegible]	5
ell 2	21	2
Cell 3	6	14
Cell 4	16	9

If $X_{ij} \sim \text{Poisson}(\Lambda_{ij})$, then:

1. $X_{ij}^{(1)} \sim \text{Poisson}(\epsilon \Lambda_{ij})$
2. $X_{ij}^{(2)} \sim \text{Poisson}((1-\epsilon)\Lambda_{ij})$
3. $X_{ij}^{(1)} \perp\!\!\!\perp X_{ij}^{(2)}$

A very well-known result.

An alternative: Poisson thinning

X

	Gene 1	Gene 2
Cell 1	18	[illegible]
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

X_{ij}

$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, \epsilon)$

$X^{(1)}$

	Gene 1	Gene 2
[illegible]	14	1
[illegible]	[illegible]	6
Cell 3	5	17
Cell 4	6	25

Estimate clusters.

$X_{ij}^{(2)} := X_{ij} - X_{ij}^{(1)}$

$X^{(2)}$

	Gene 3	Gene 4
Cell 1	[illegible]	5
Cell 2	21	2
Cell 3	6	14
Cell 4	16	9

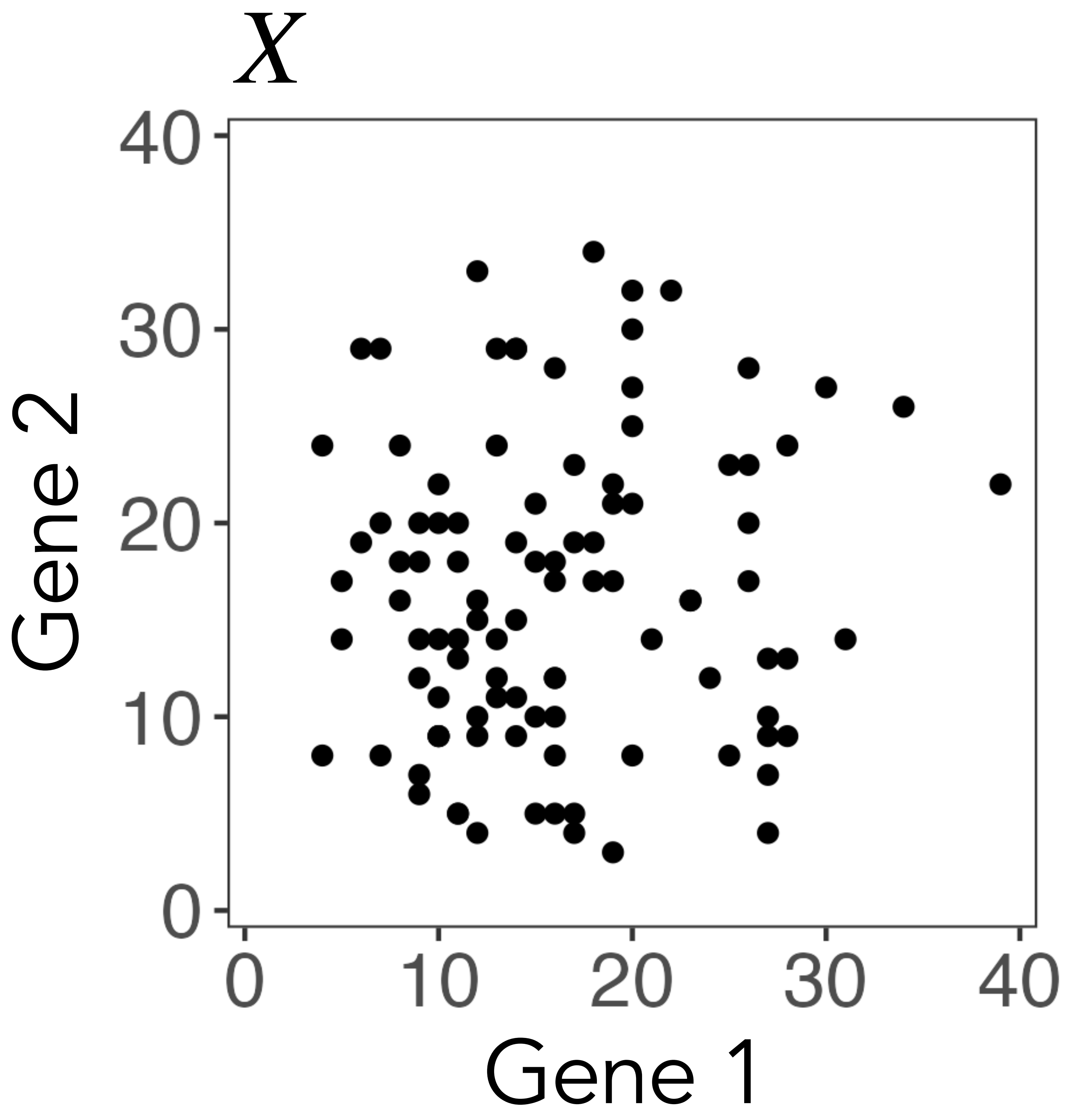
Evaluate clusters or test for differential expression.

If $X_{ij} \sim \text{Poisson}(\Lambda_{ij})$, then:

1. $X_{ij}^{(1)} \sim \text{Poisson}(\epsilon \Lambda_{ij})$
2. $X_{ij}^{(2)} \sim \text{Poisson}((1-\epsilon)\Lambda_{ij})$
3. $X_{ij}^{(1)} \perp\!\!\!\perp X_{ij}^{(2)}$

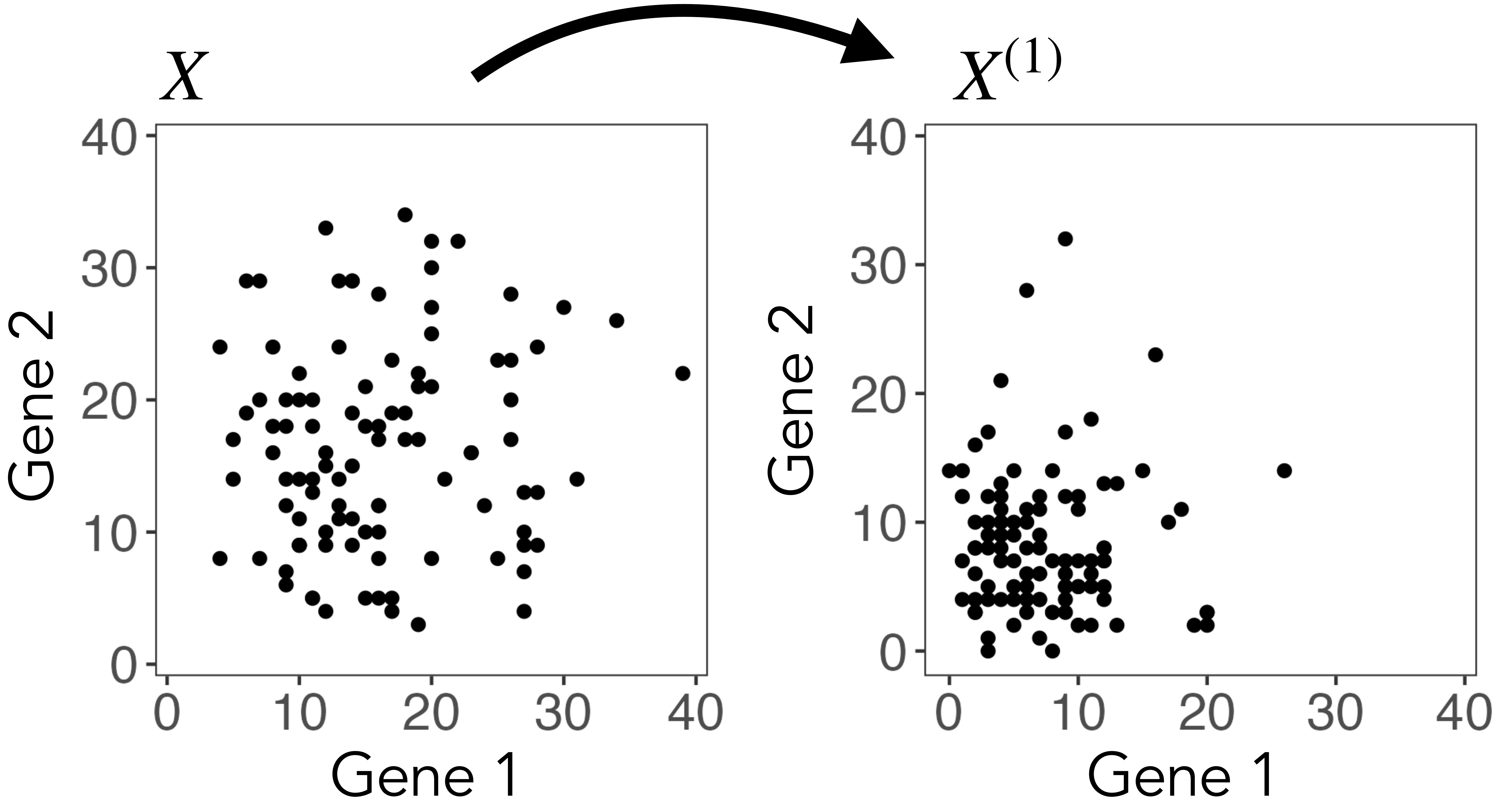
A very well-known result.

Visualizing thinning on a dataset with one true cluster



Visualizing thinning on a dataset with one true cluster

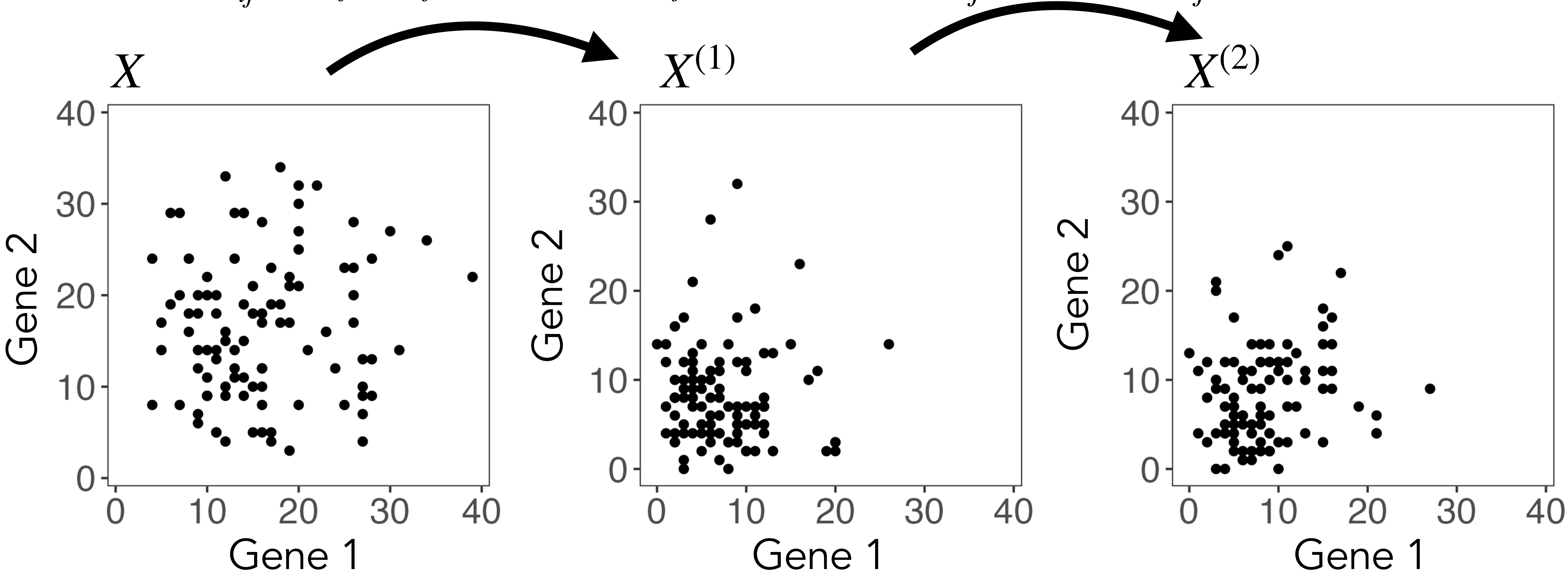
$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5)$$



Visualizing thinning on a dataset with one true cluster

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5)$$

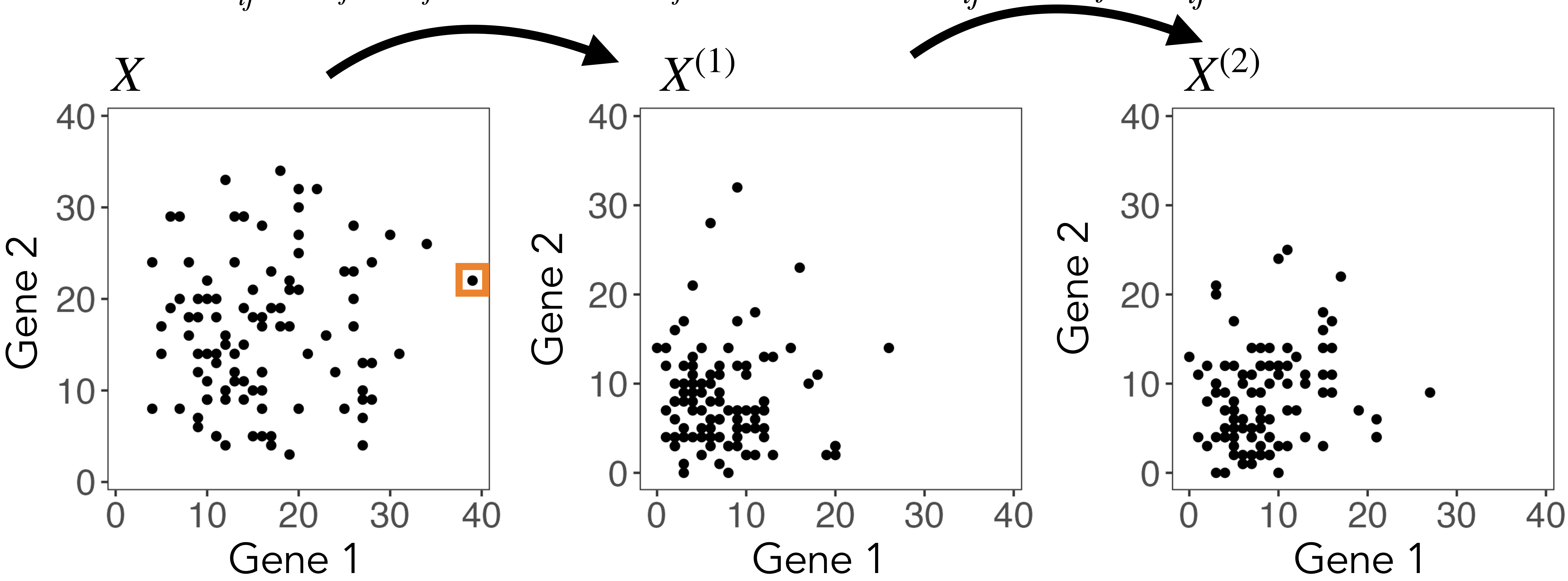
$$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



Visualizing thinning on a dataset with one true cluster

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5)$$

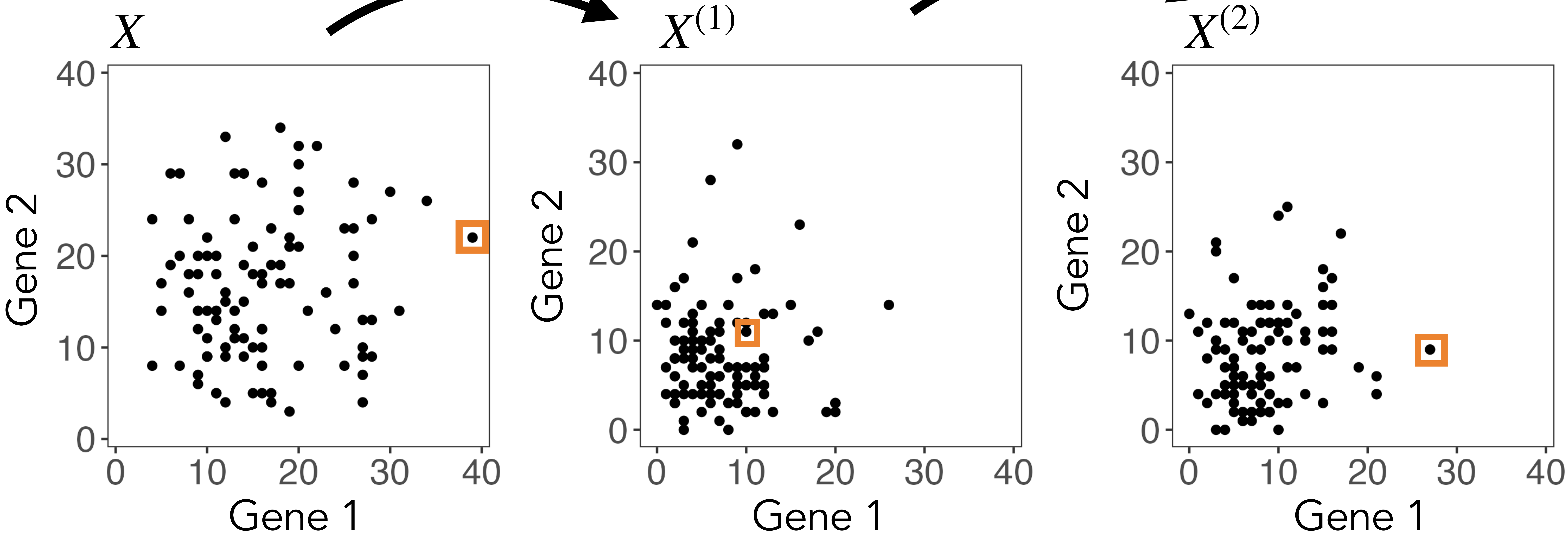
$$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



Visualizing thinning on a dataset with one true cluster

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5)$$

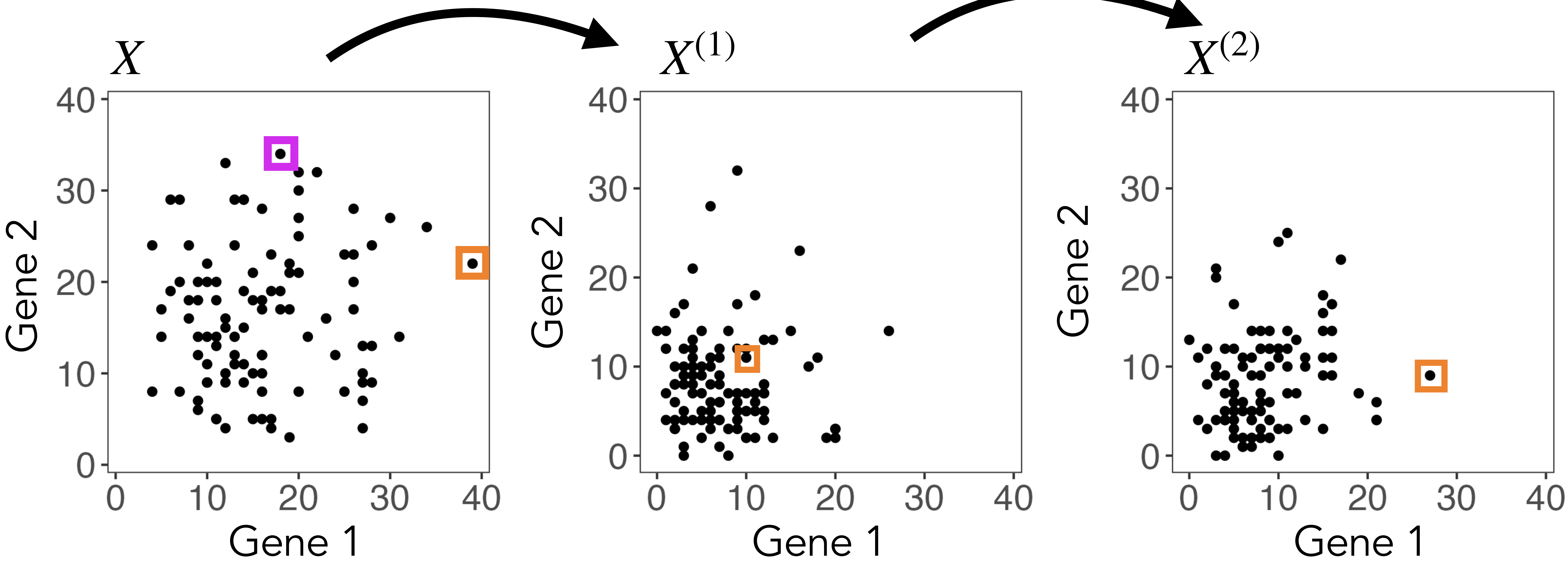
$$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



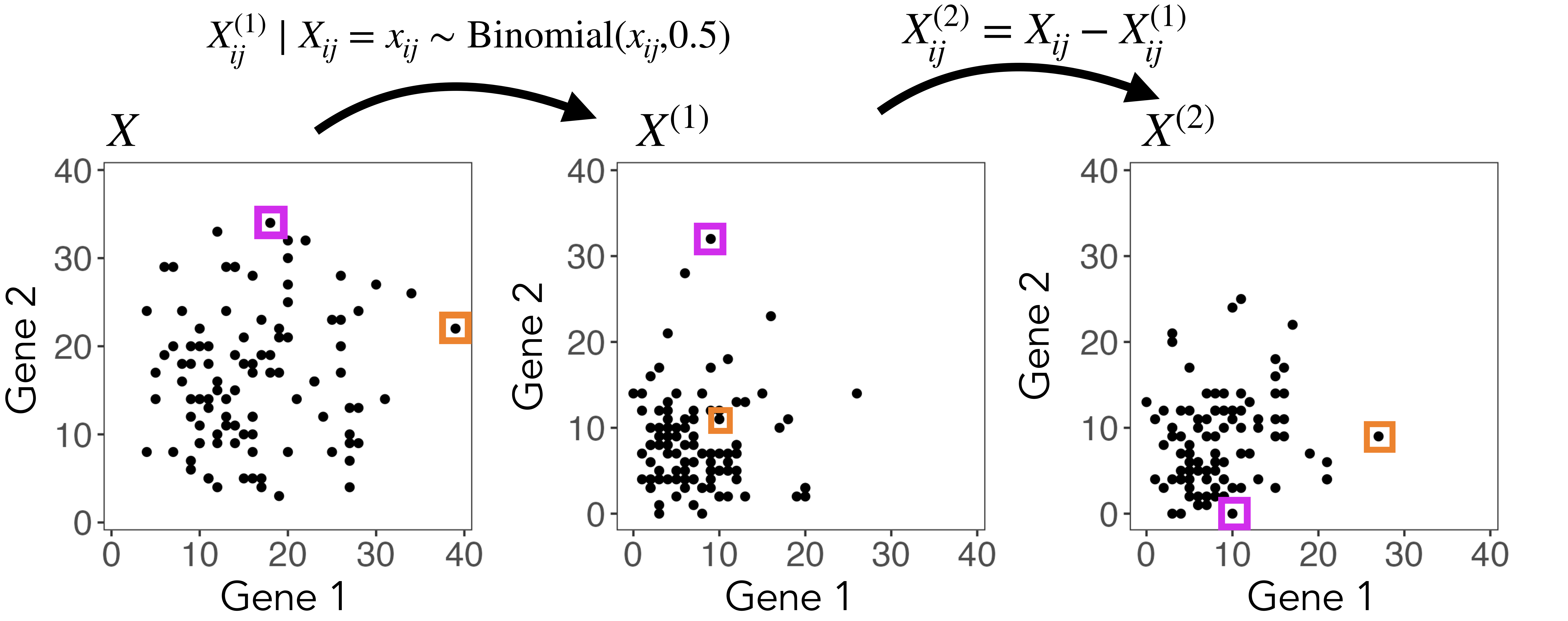
Visualizing thinning on a dataset with one true cluster

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5)$$

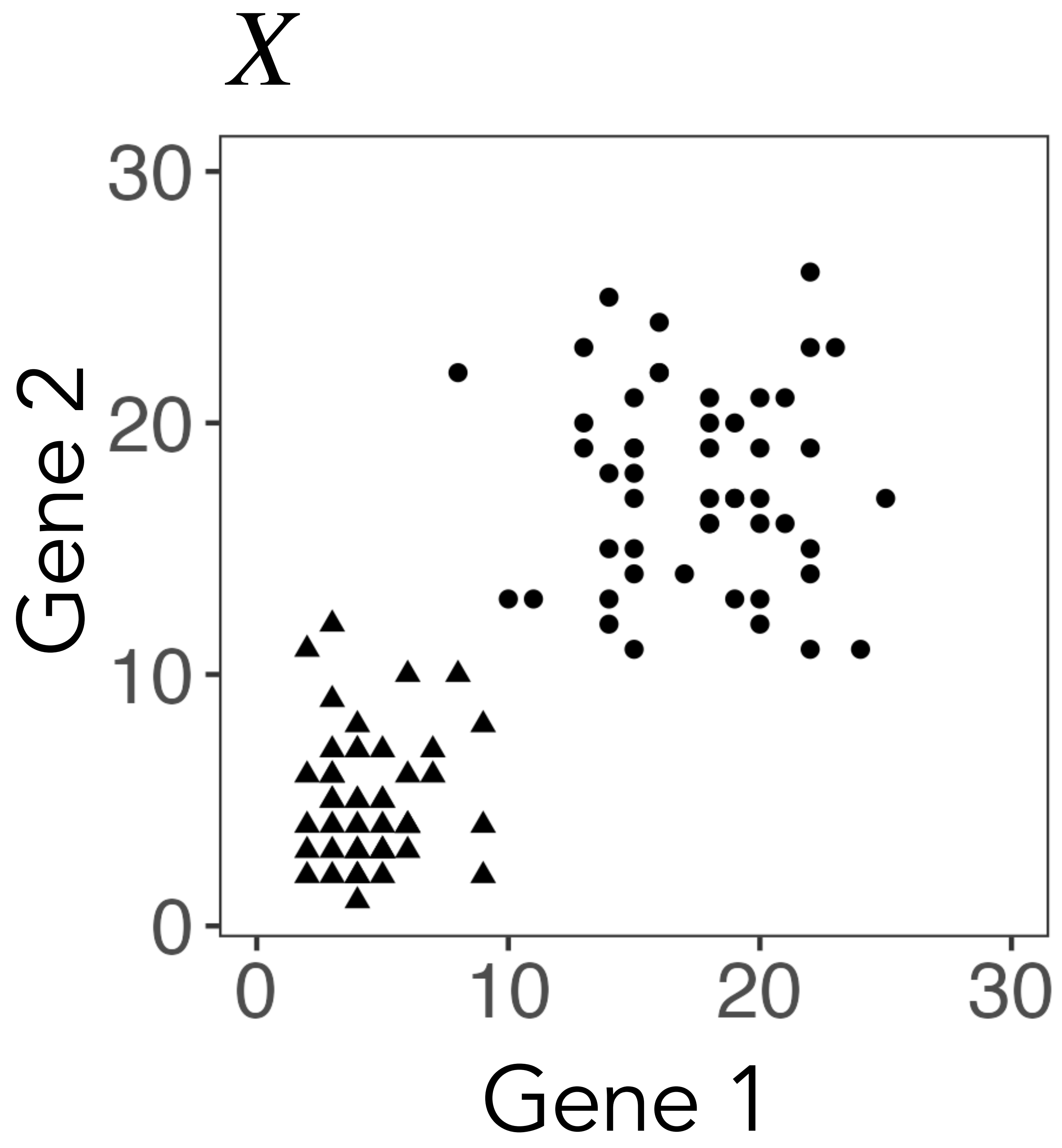
$$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



Visualizing thinning on a dataset with one true cluster

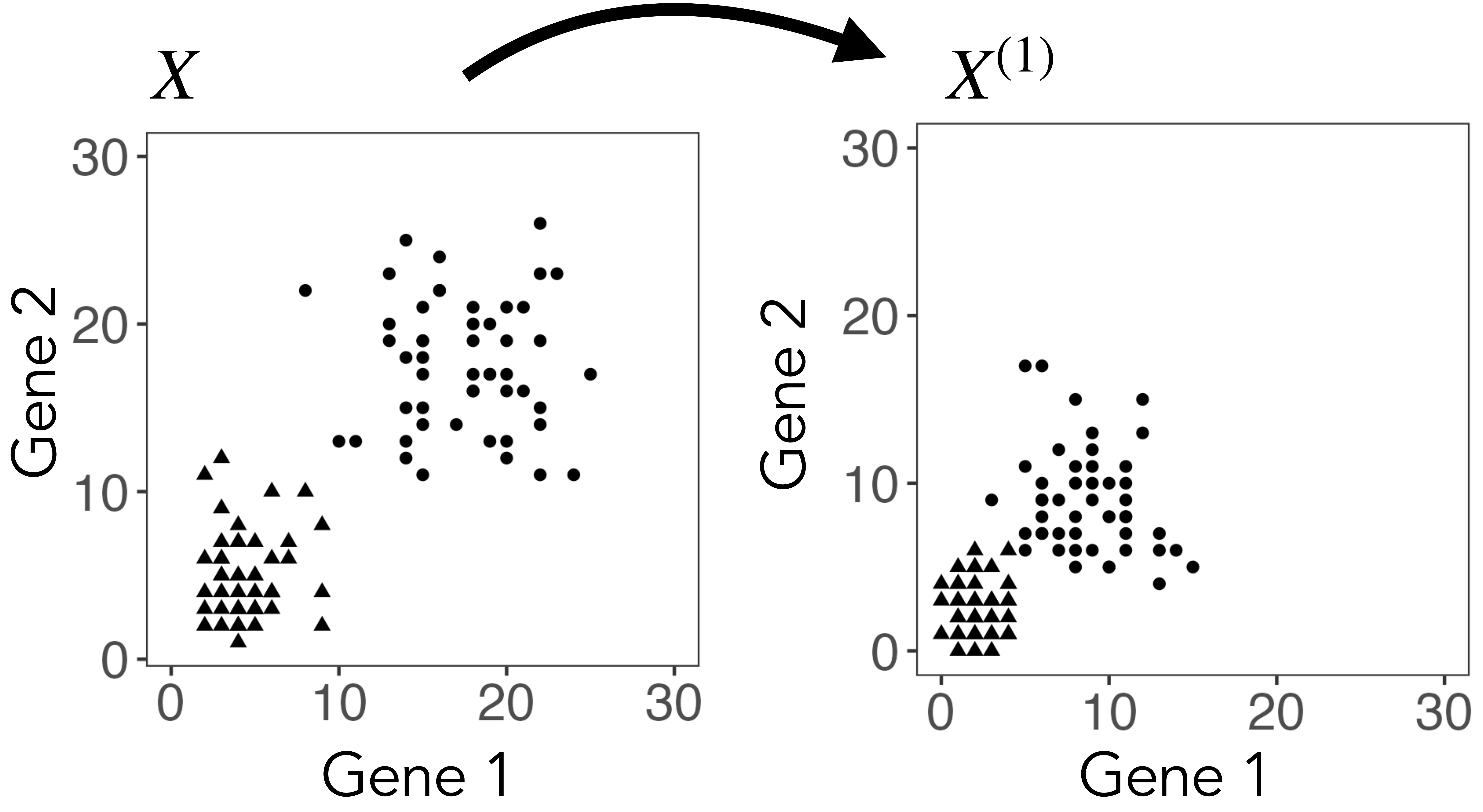


Visualizing thinning on a dataset with two true clusters



Visualizing thinning on a dataset with two true clusters

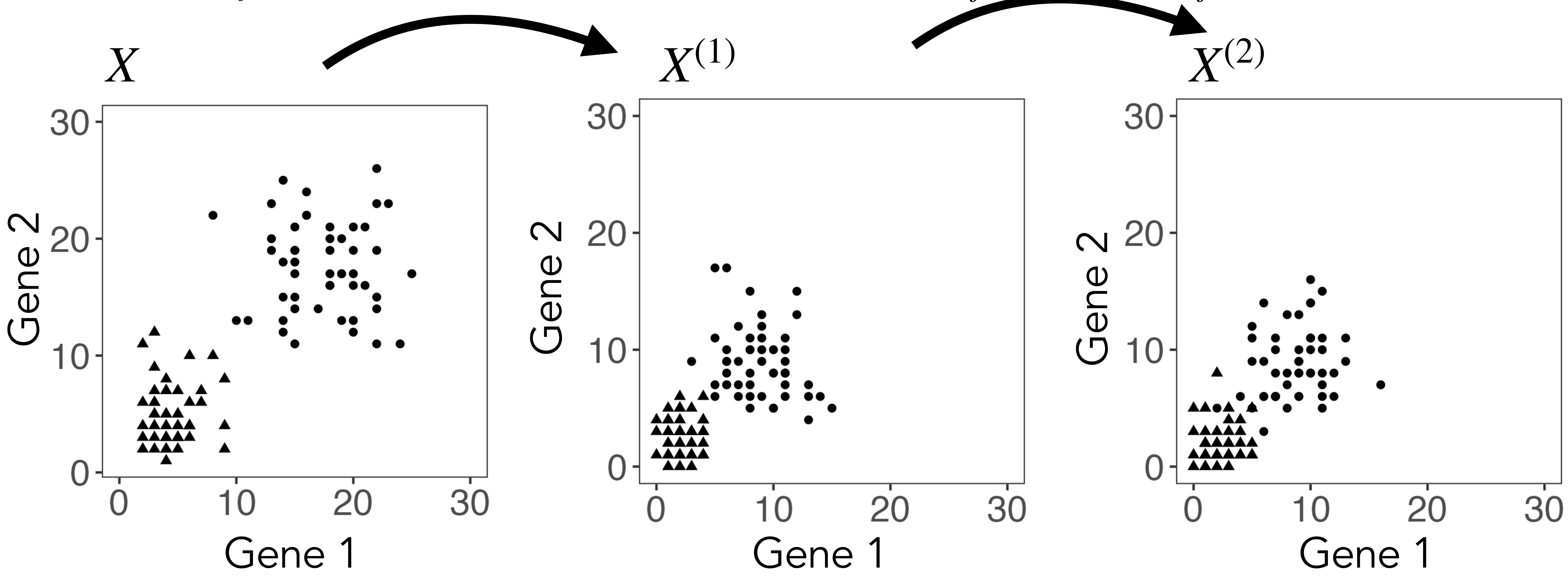
$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5)$$



Visualizing thinning on a dataset with two true clusters

$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5)$

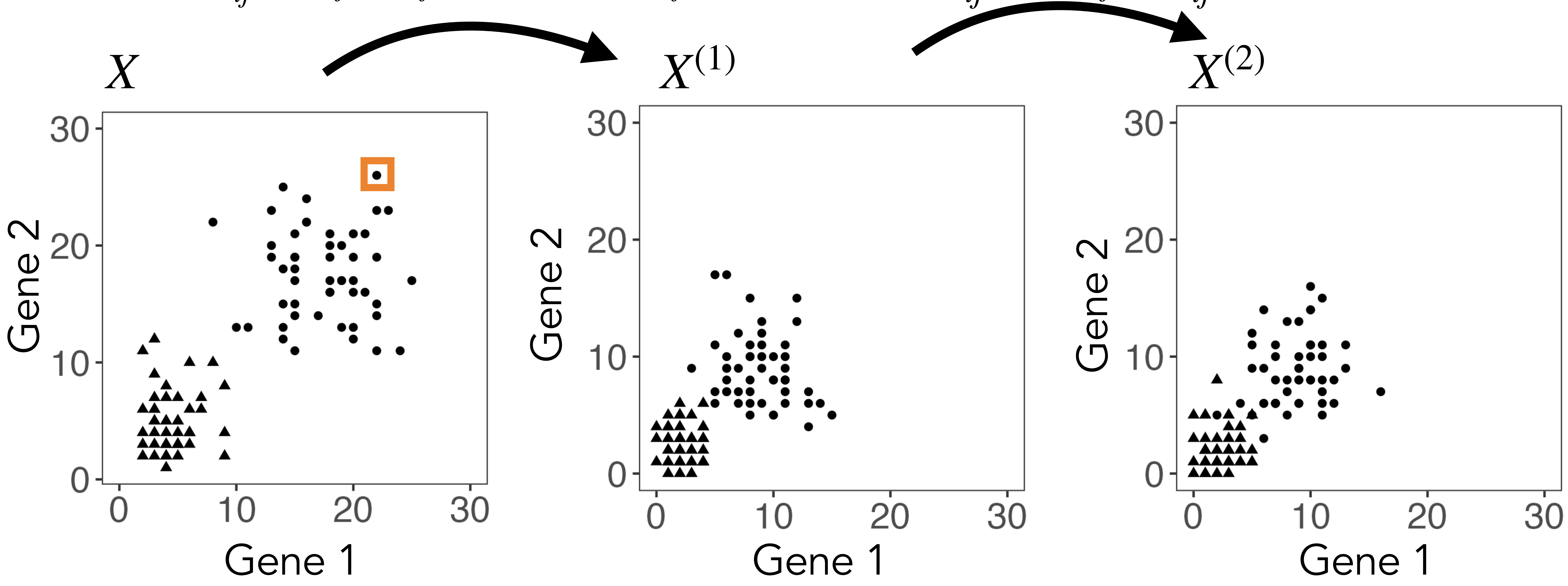
$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$



Visualizing thinning on a dataset with two true clusters

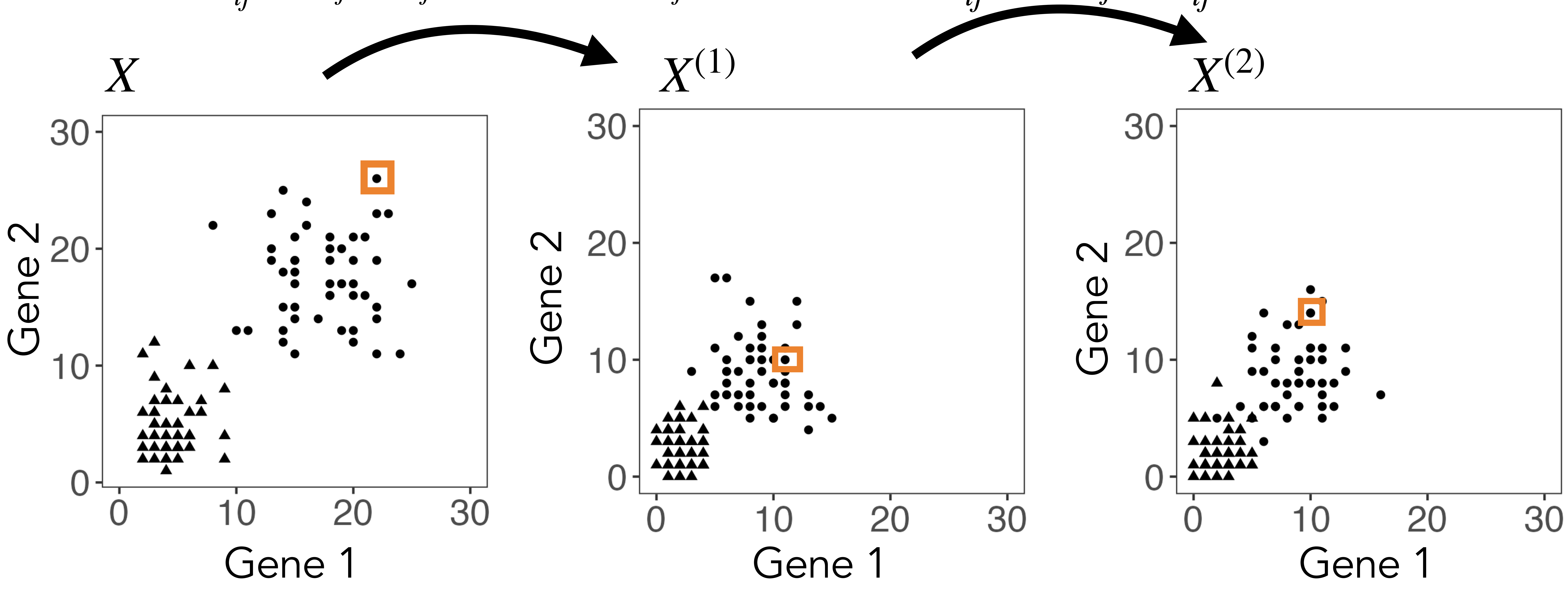
$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5)$$

$$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



Visualizing thinning on a dataset with two true clusters

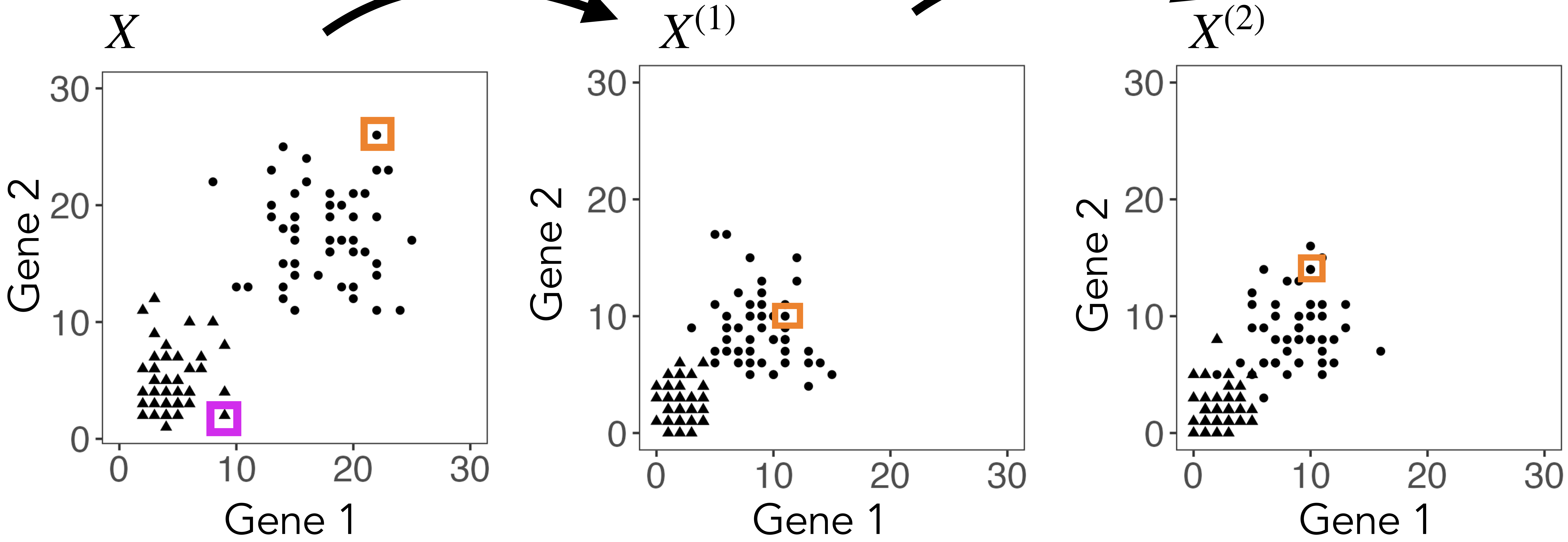
$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5) \qquad X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



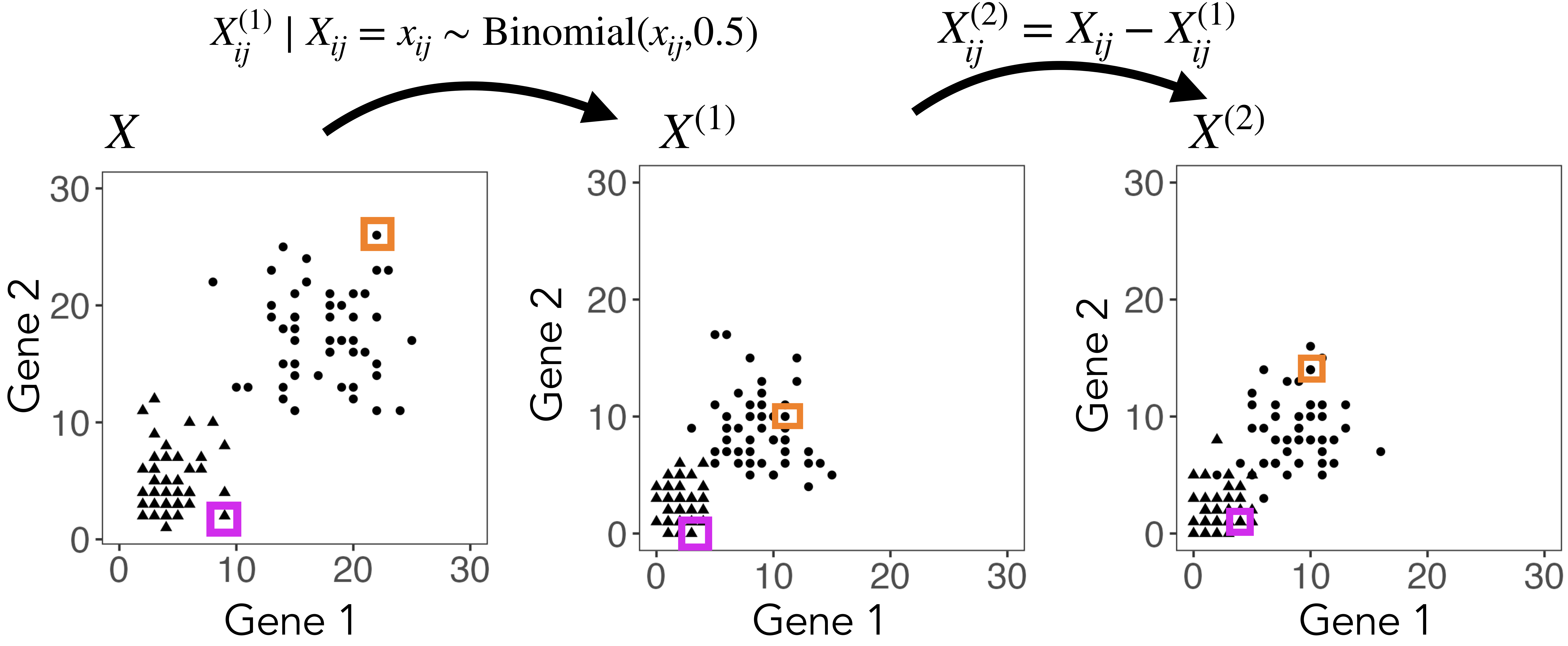
Visualizing thinning on a dataset with two true clusters

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5)$$

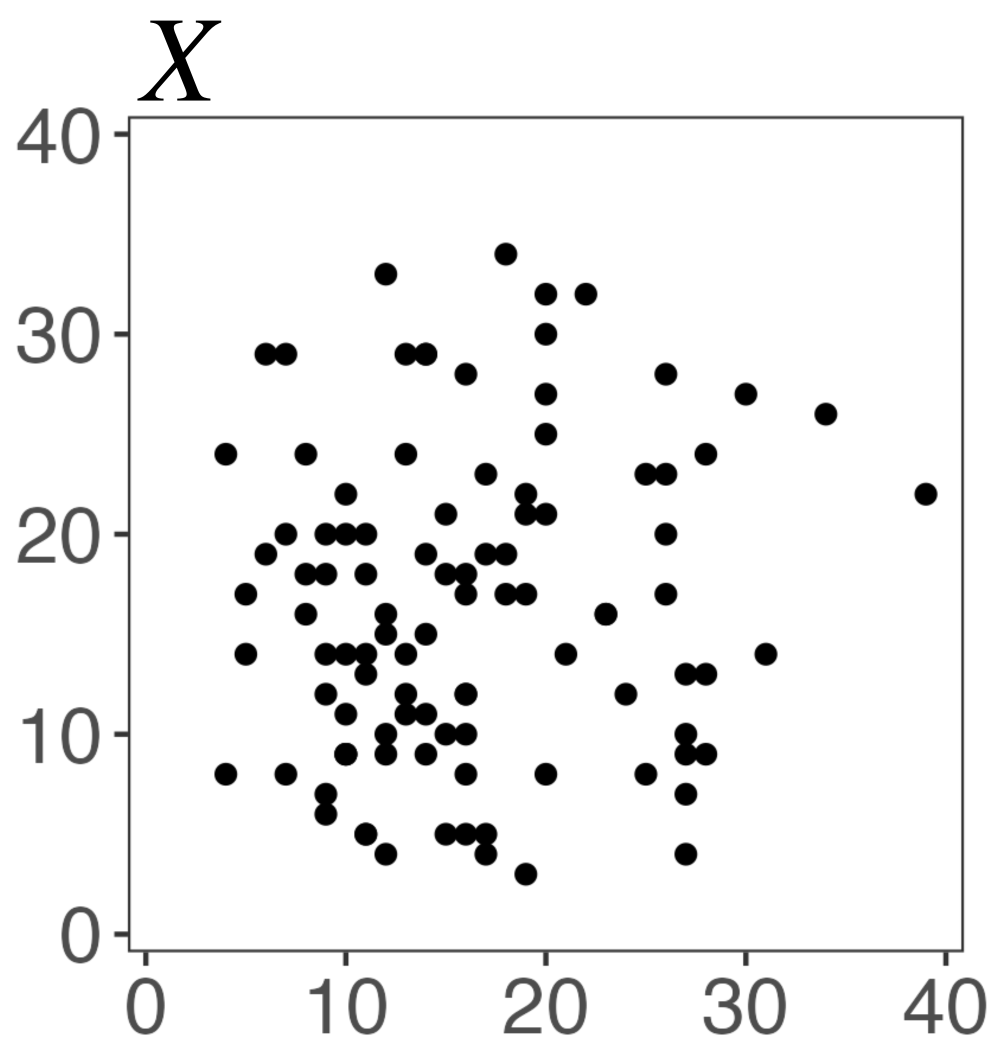
$$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



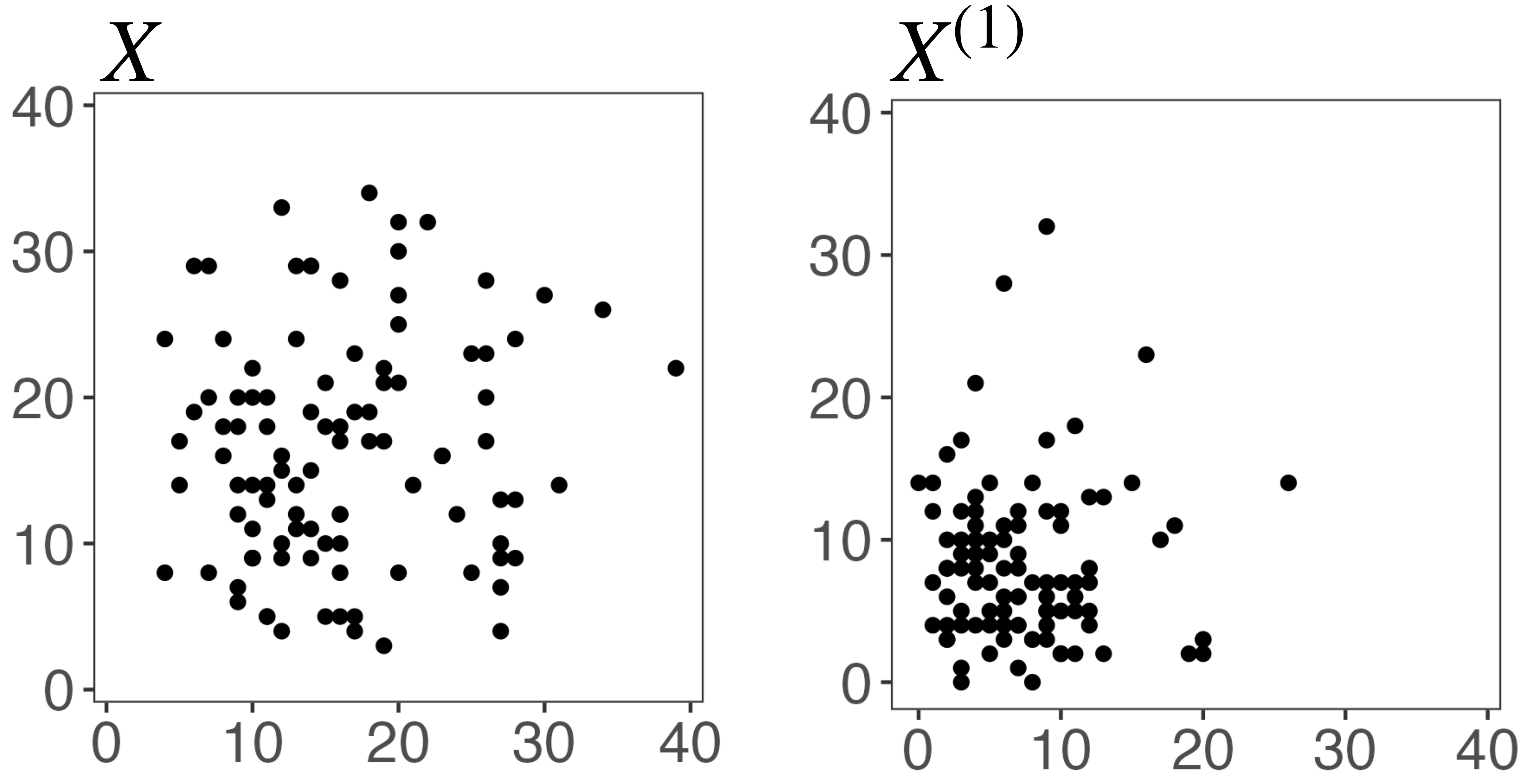
Visualizing thinning on a dataset with two true clusters



Thinning avoids the pitfall of sample splitting on our motivating examples

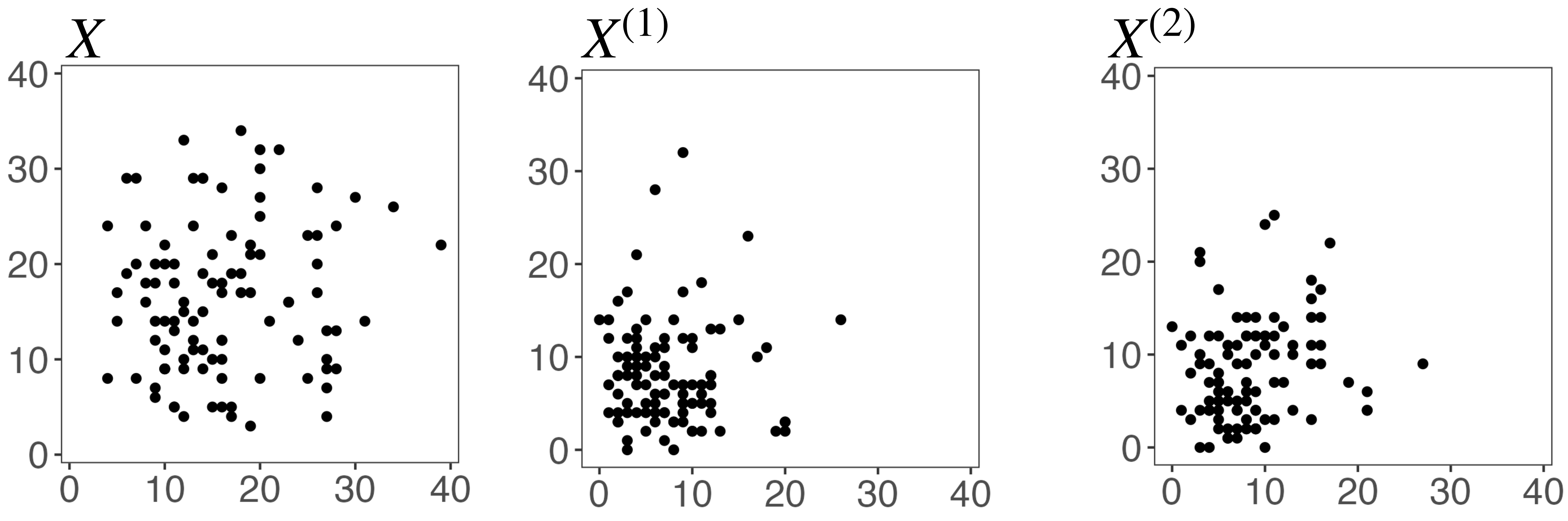


Thinning avoids the pitfall of sample splitting on our motivating examples



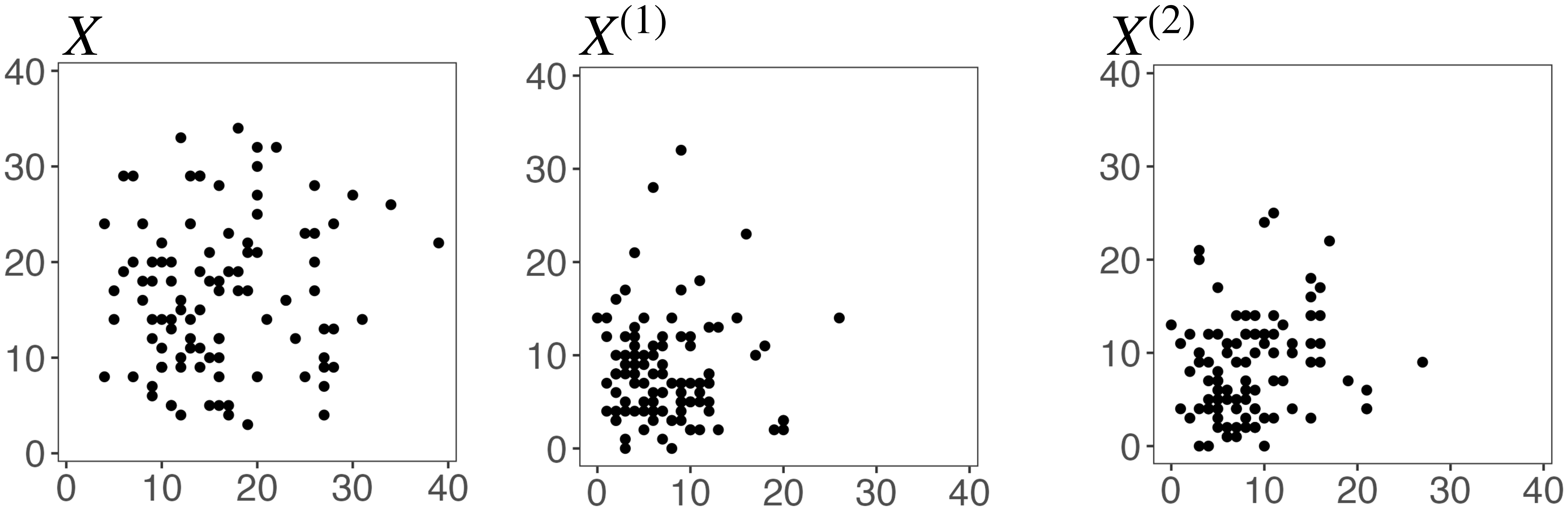
Step 1: thin observations into train/test.

Thinning avoids the pitfall of sample splitting on our motivating examples



Step 1: thin observations into train/test.

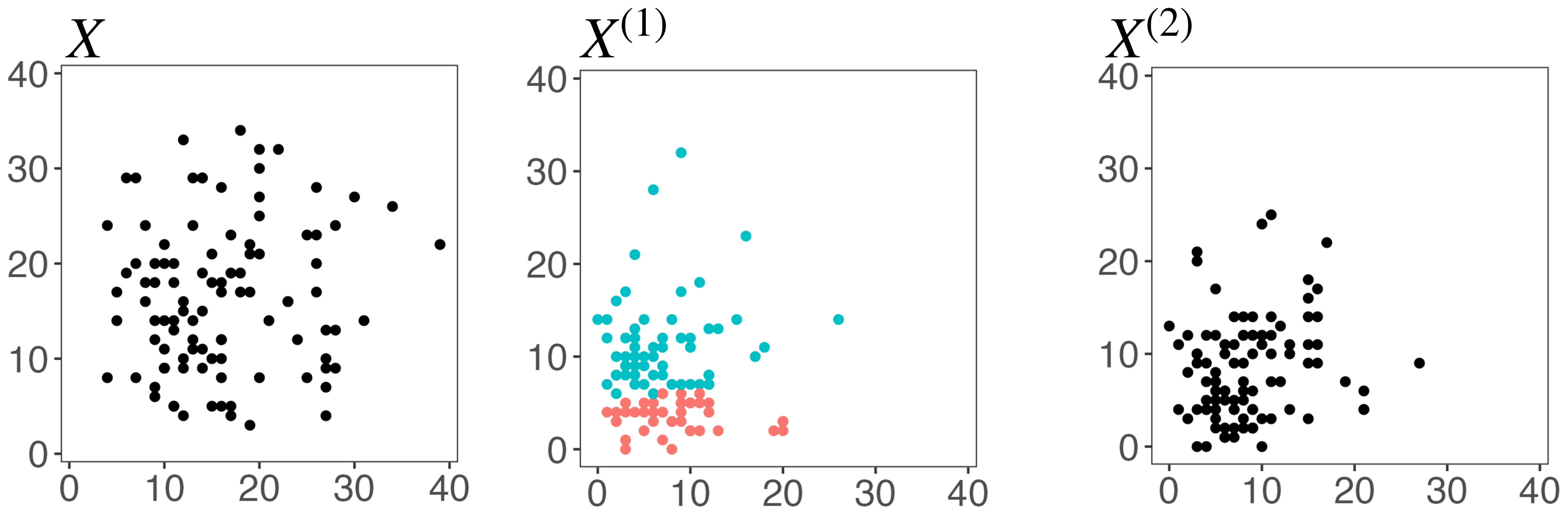
Thinning avoids the pitfall of sample splitting on our motivating examples



Step 1: thin observations into train/test.

Step 2: cluster the training set.

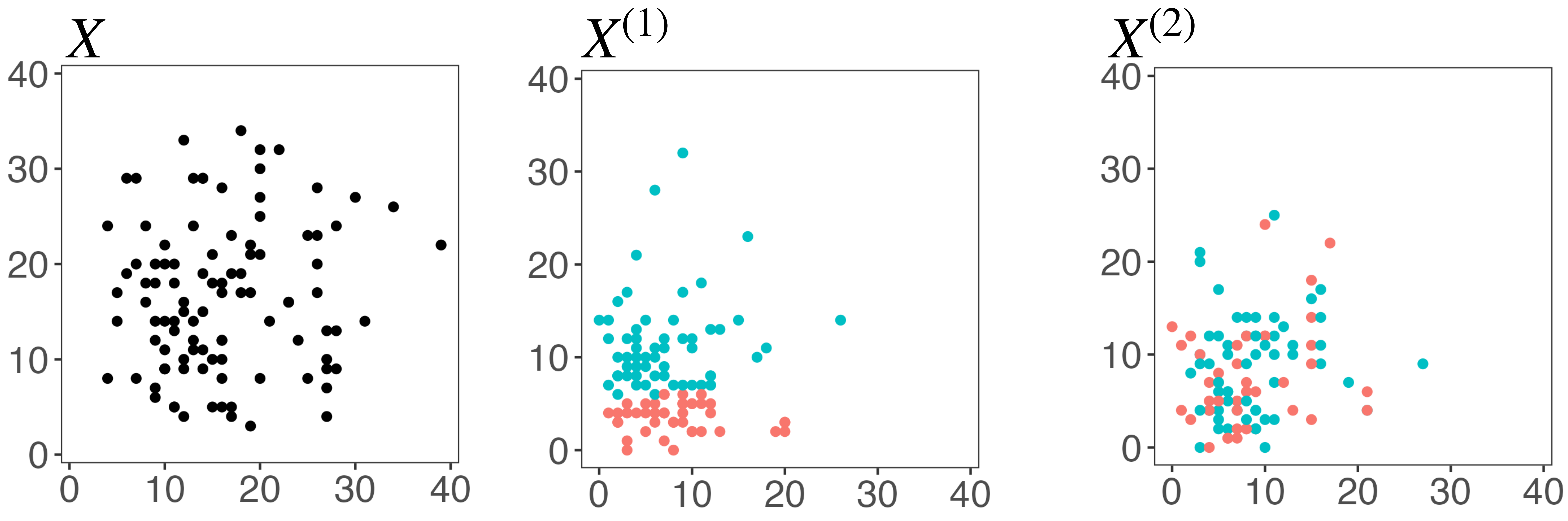
Thinning avoids the pitfall of sample splitting on our motivating examples



Step 1: thin observations into train/test.

Step 2: cluster the training set.

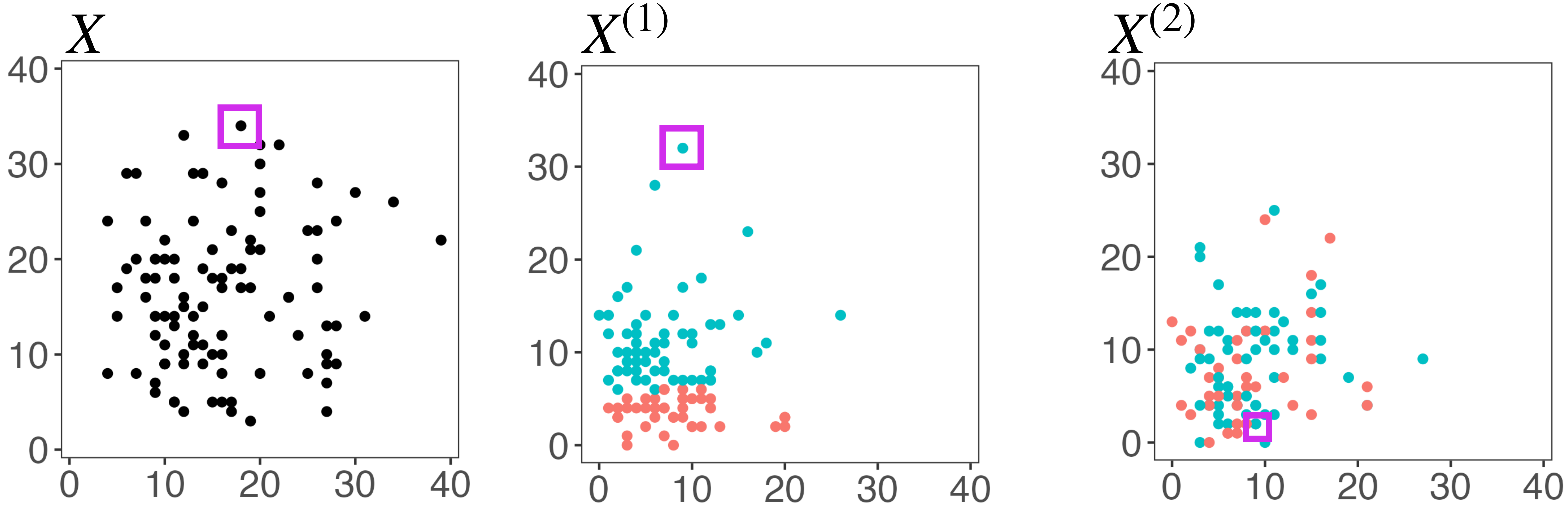
Thinning avoids the pitfall of sample splitting on our motivating examples



Step 1: thin observations into train/test.

Step 2: cluster the training set.

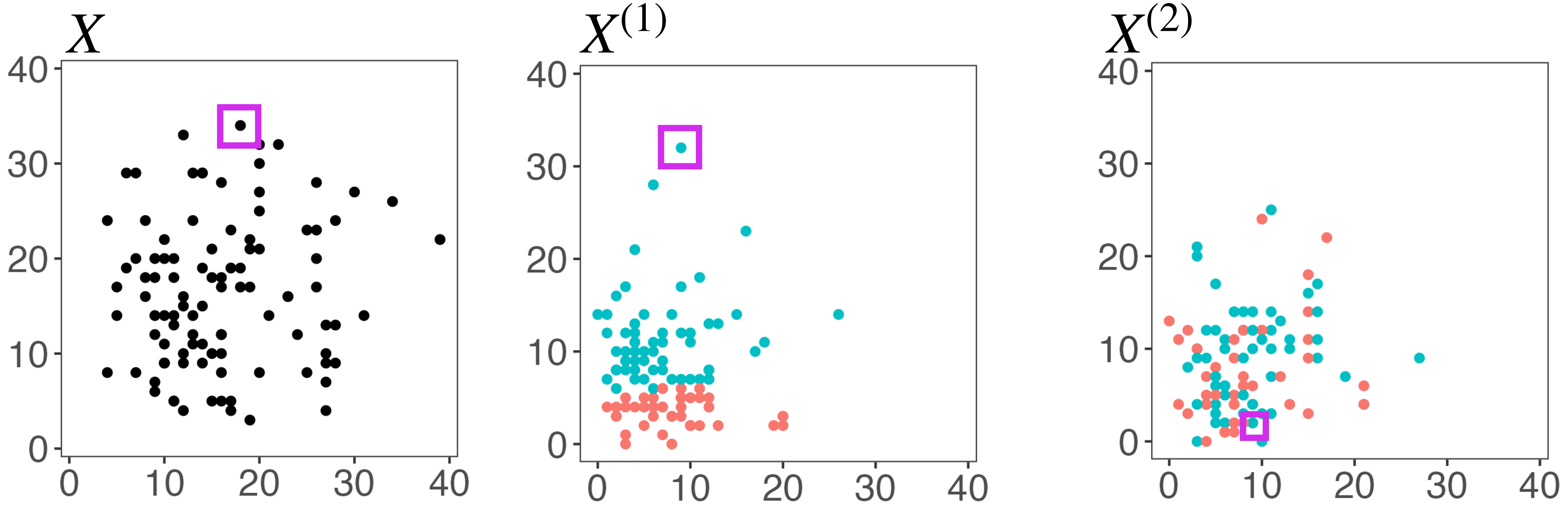
Thinning avoids the pitfall of sample splitting on our motivating examples



Step 1: thin observations into train/test.

Step 2: cluster the training set.

Thinning avoids the pitfall of sample splitting on our motivating examples

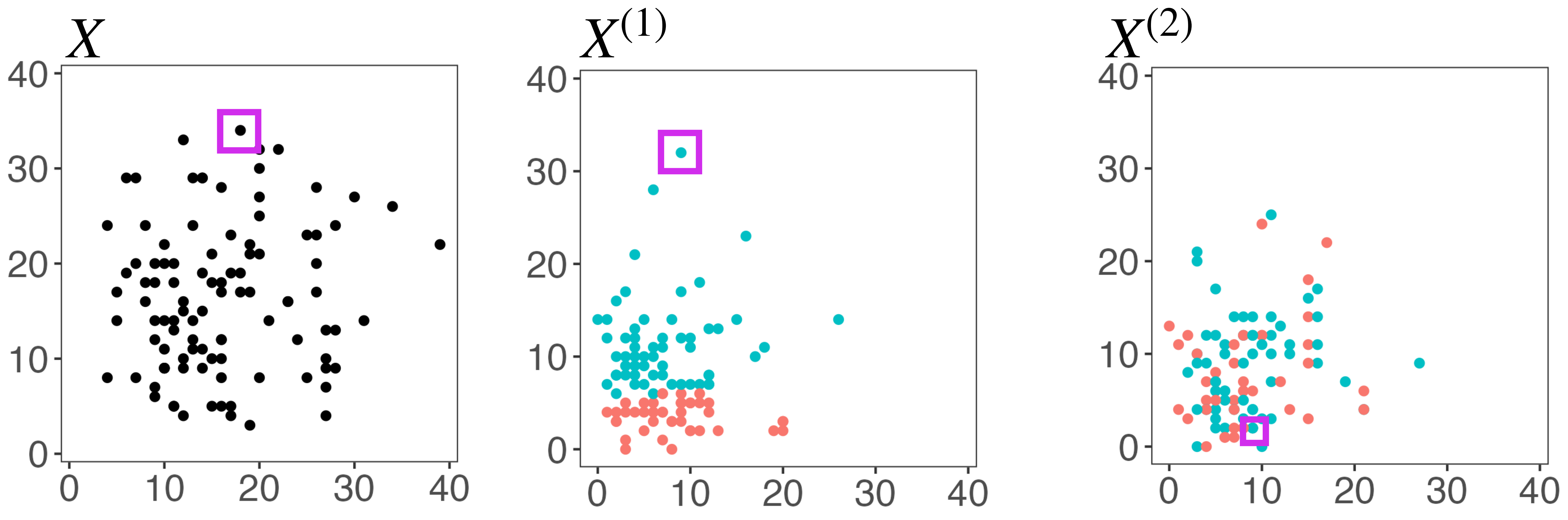


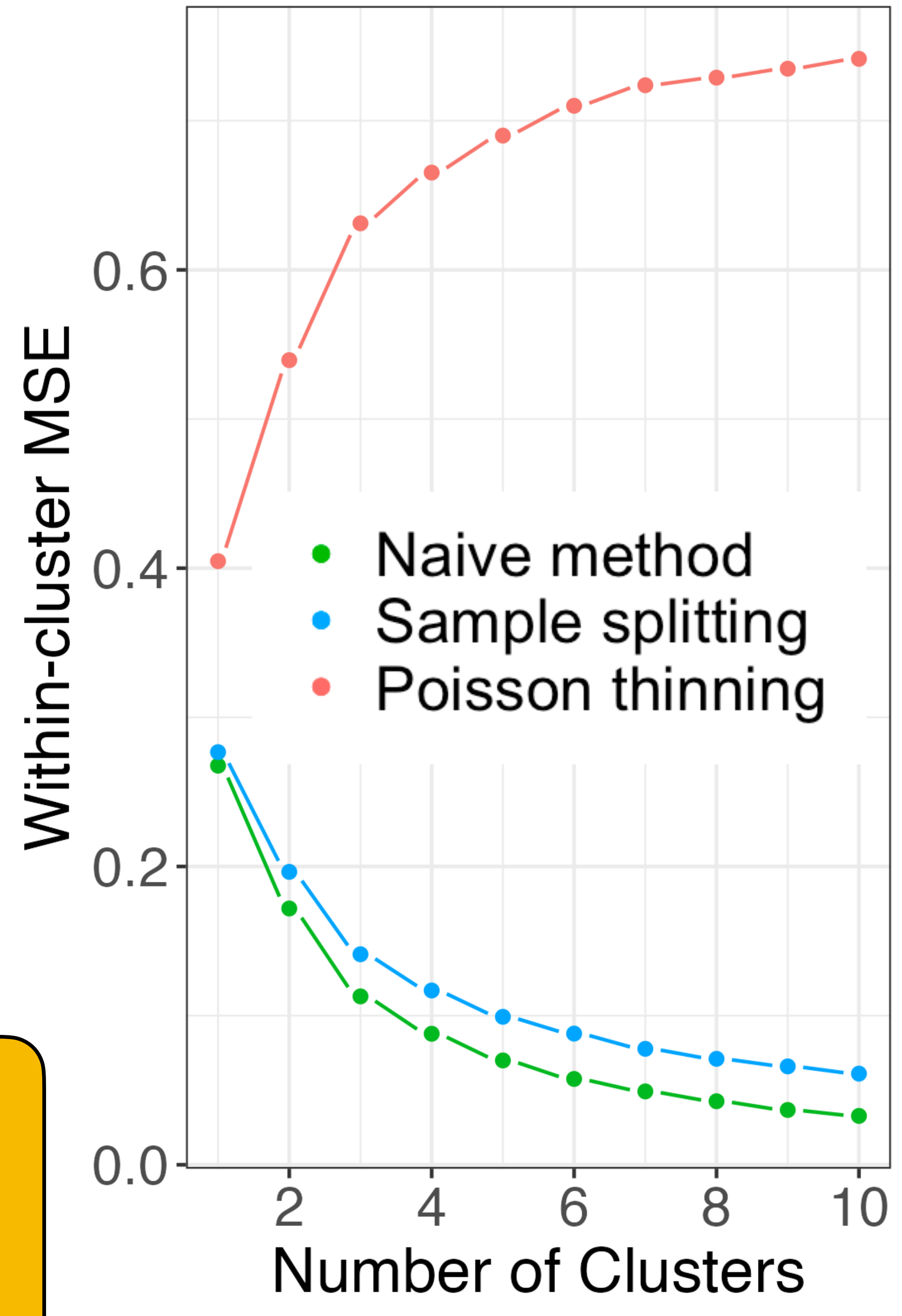
Step 1: thin observations into train/test.

Step 2: cluster the training set.

Step 3: evaluate clusters or test for difference in means on test set.

Thinning avoids the pitfall of sample splitting on our motivating examples



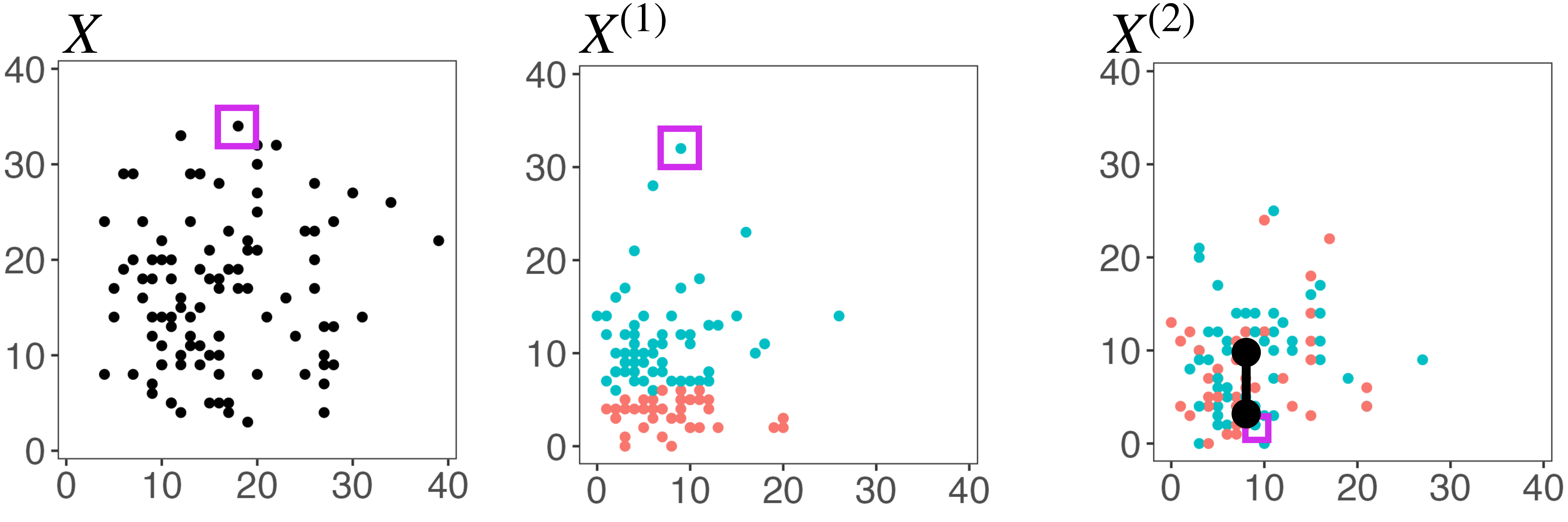


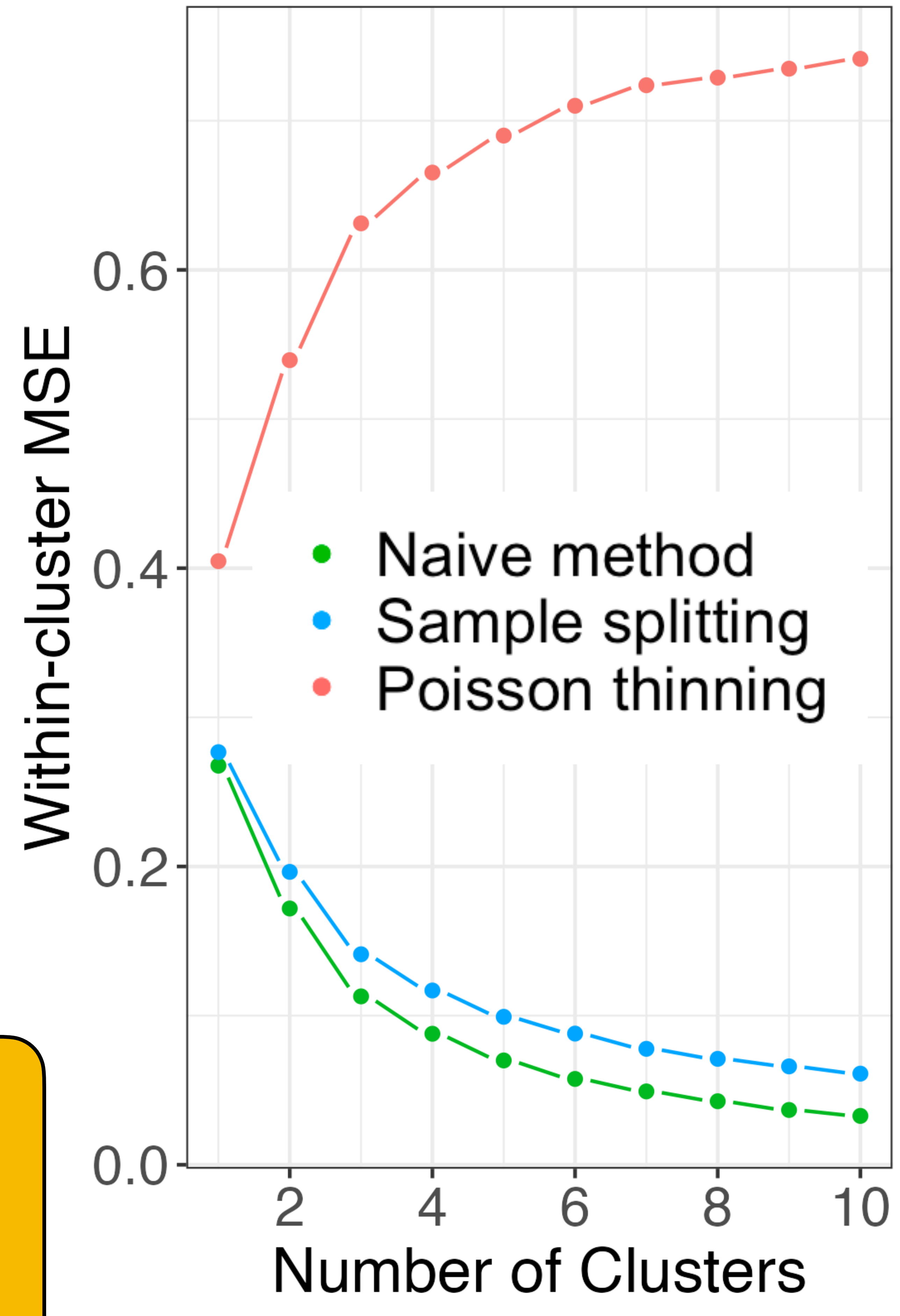
Step 1: thin observations into train/test.

Step 2: cluster the training set.

Step 3: evaluate clusters or test for difference in means on test set.

Thinning avoids the pitfall of sample splitting on our motivating examples



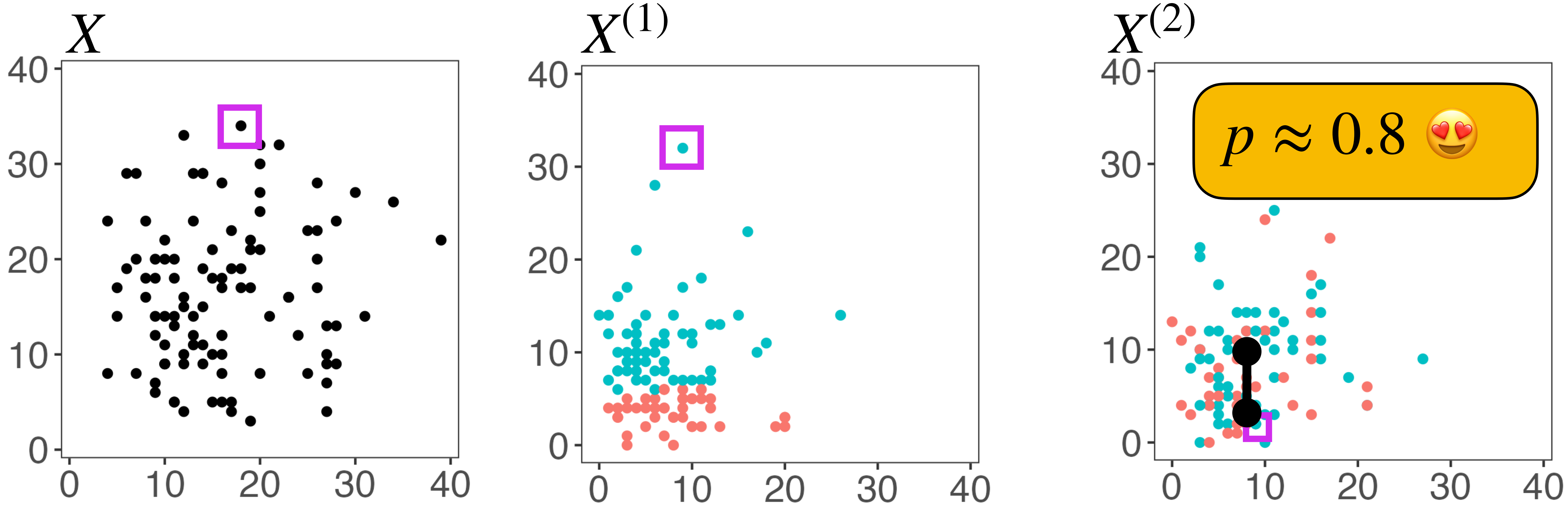


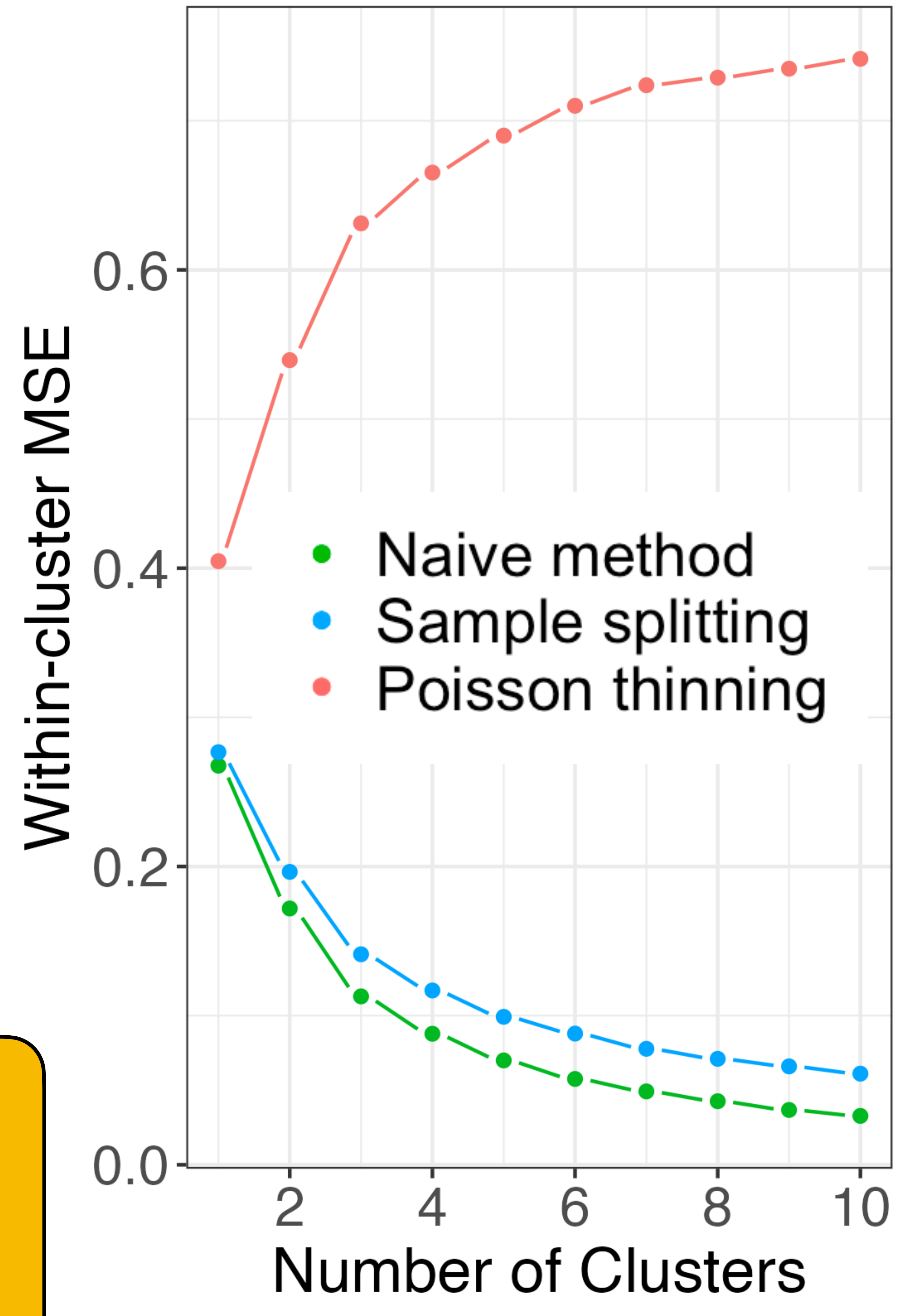
Step 1: thin observations into train/test.

Step 2: cluster the training set.

Step 3: evaluate clusters or test for difference in means on test set.

Thinning avoids the pitfall of sample splitting on our motivating examples



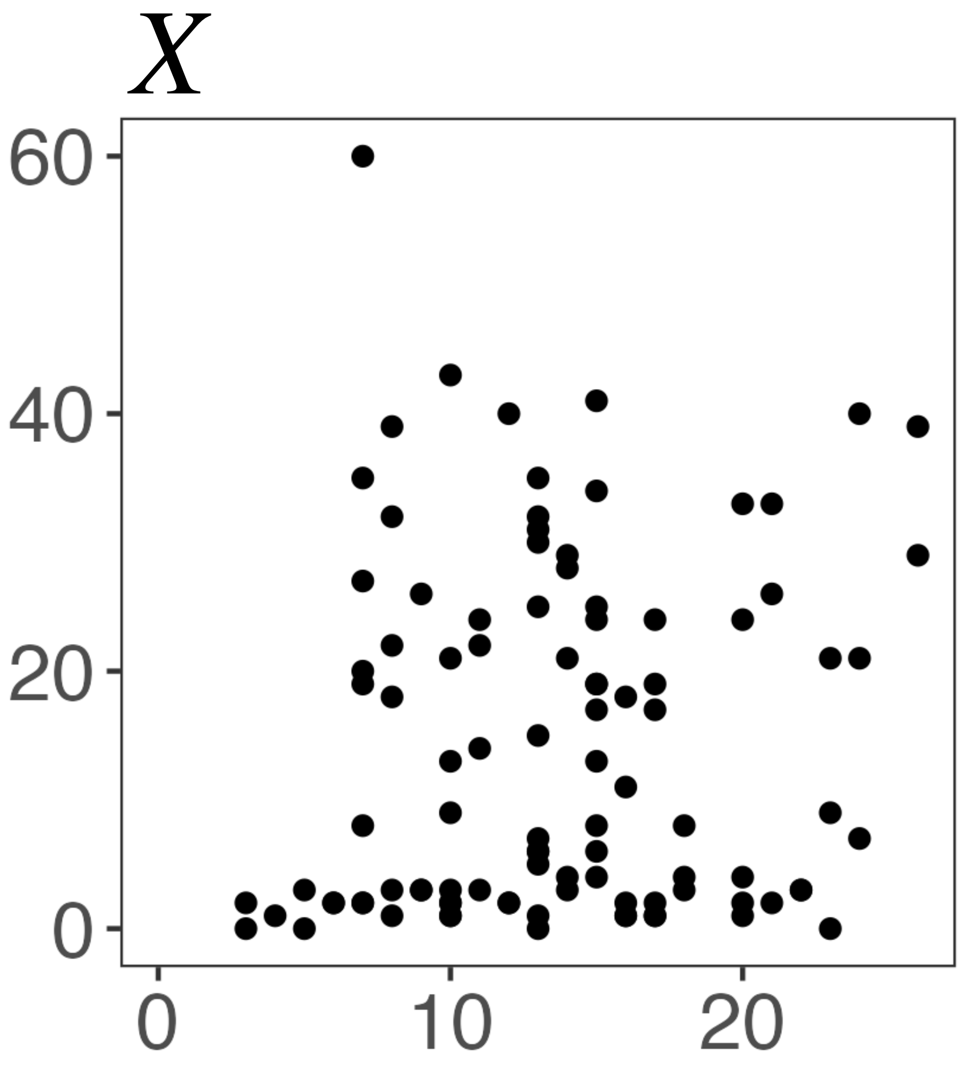


Step 1: thin observations into train/test.

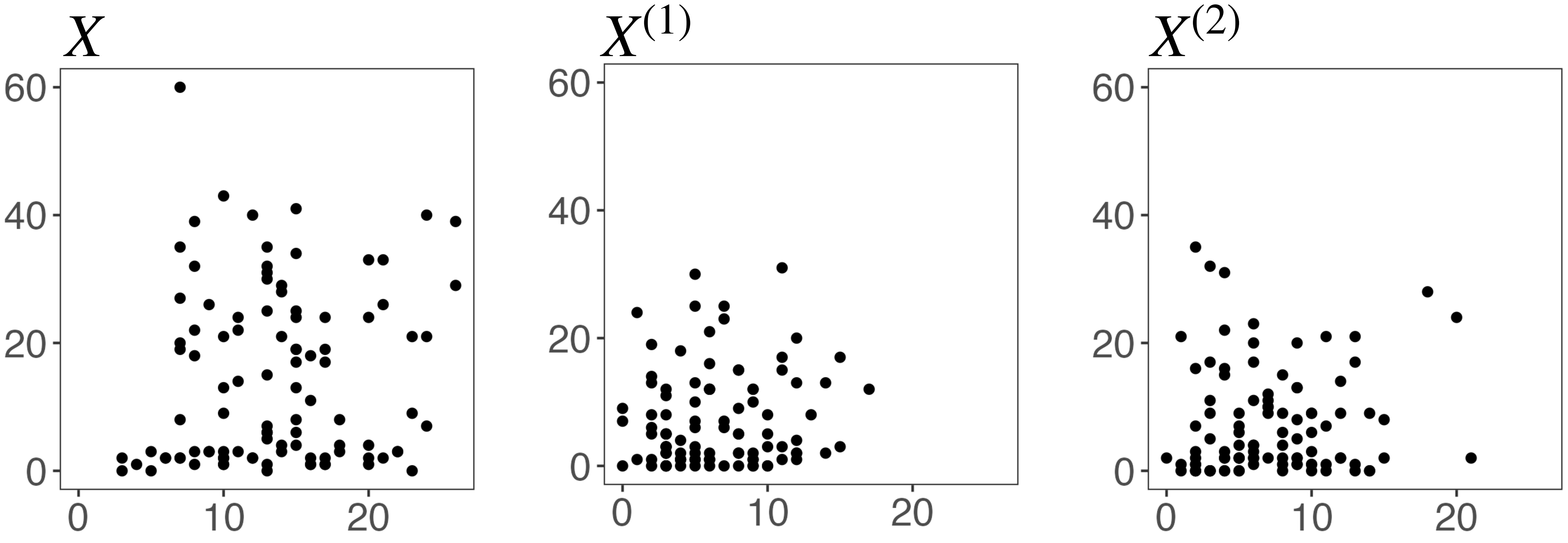
Step 2: cluster the training set.

Step 3: evaluate clusters or test for difference in means on test set.

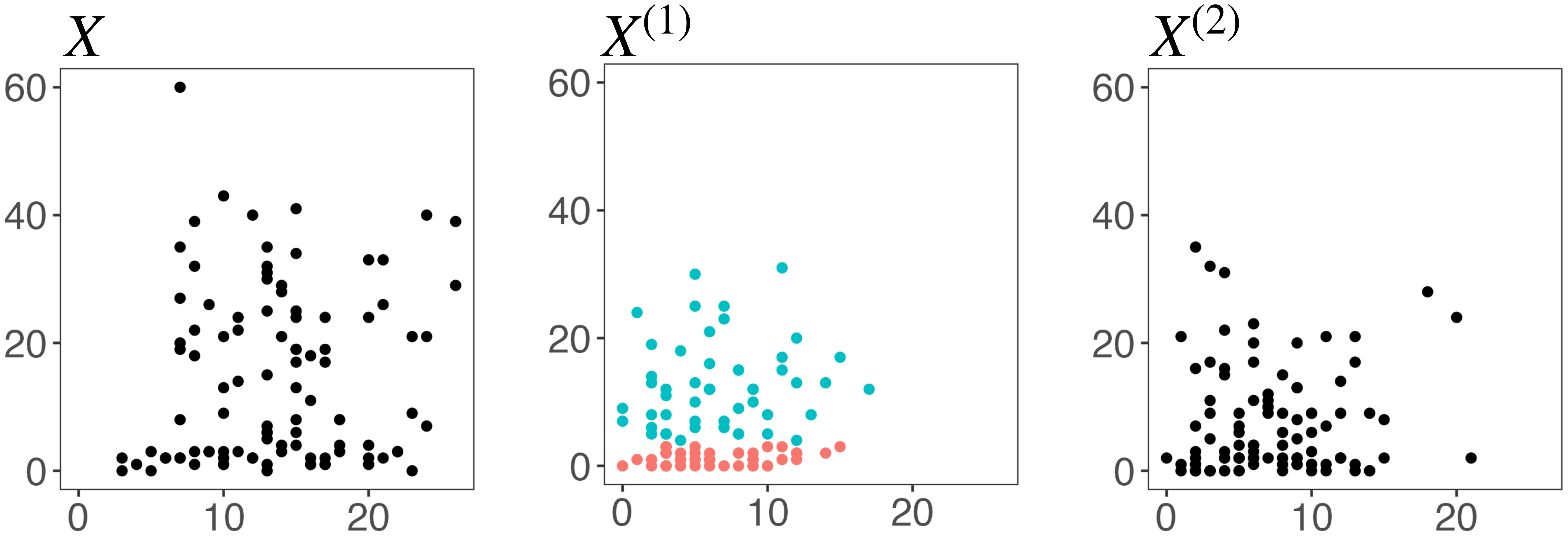
Thinning avoids the pitfall of sample splitting on our motivating examples



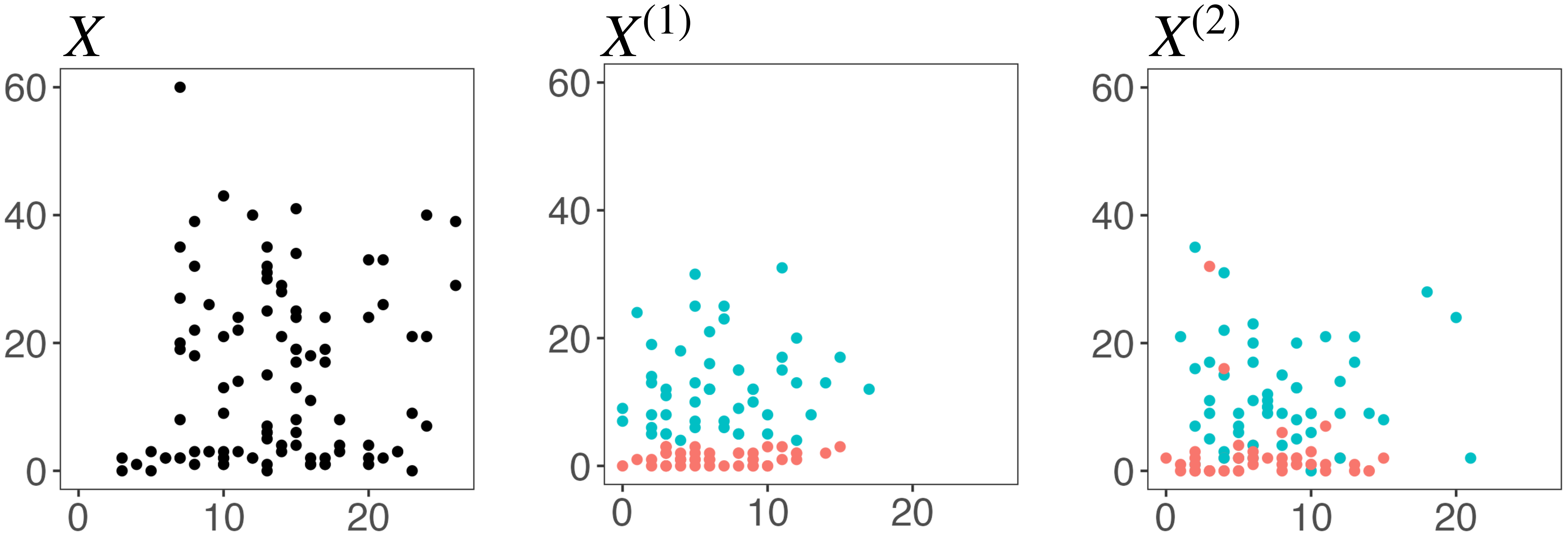
Thinning avoids the pitfall of sample splitting on our motivating examples



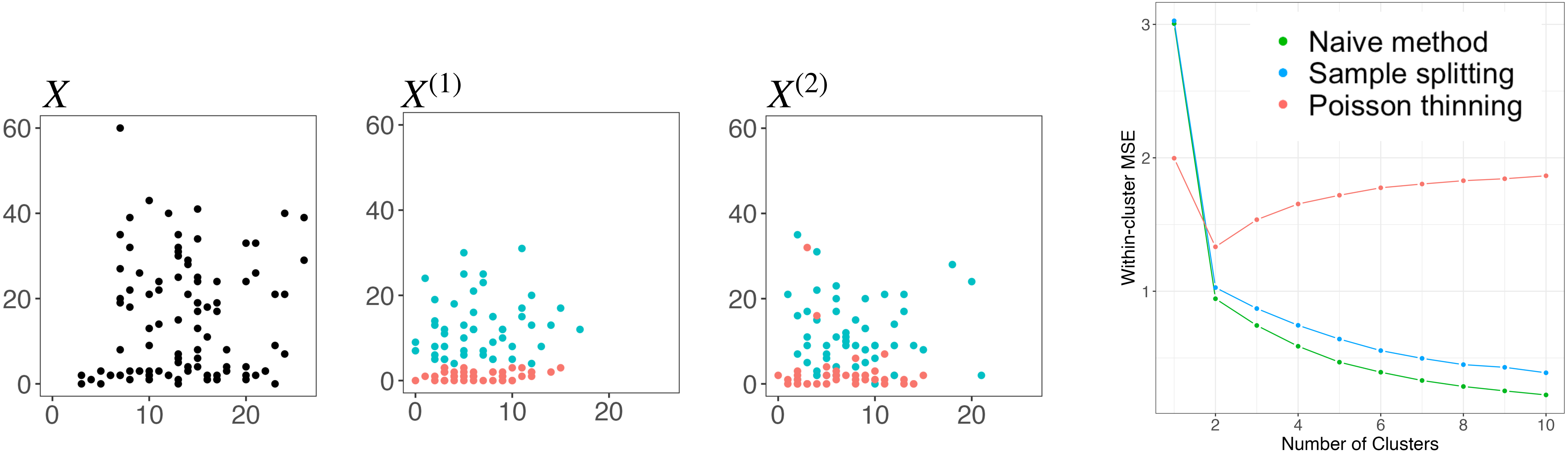
Thinning avoids the pitfall of sample splitting on our motivating examples



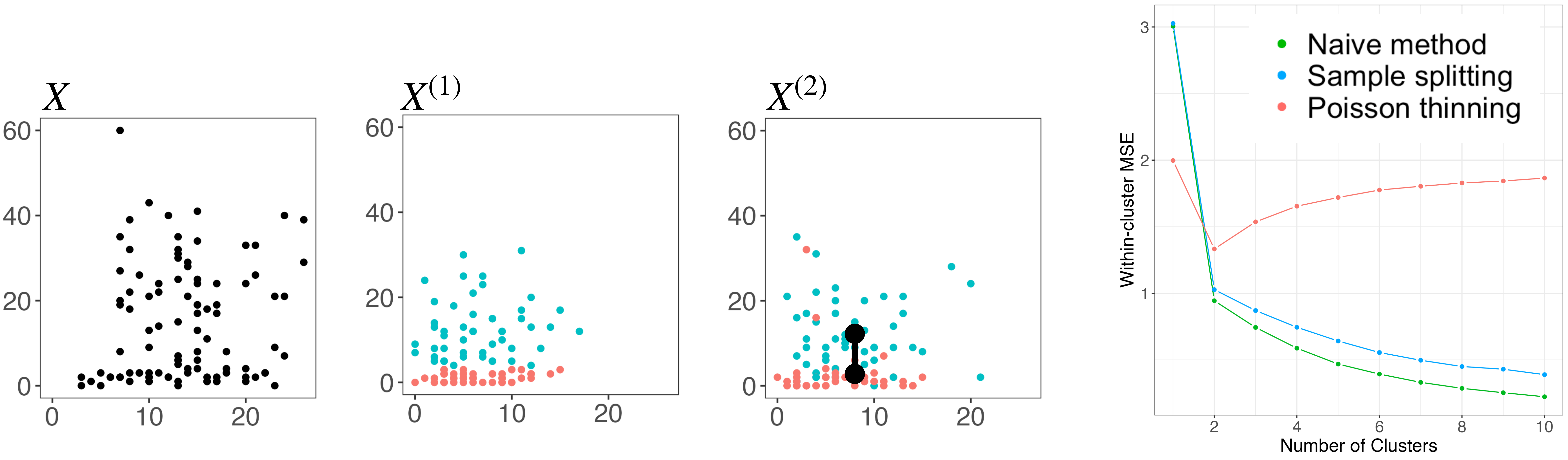
Thinning avoids the pitfall of sample splitting on our motivating examples



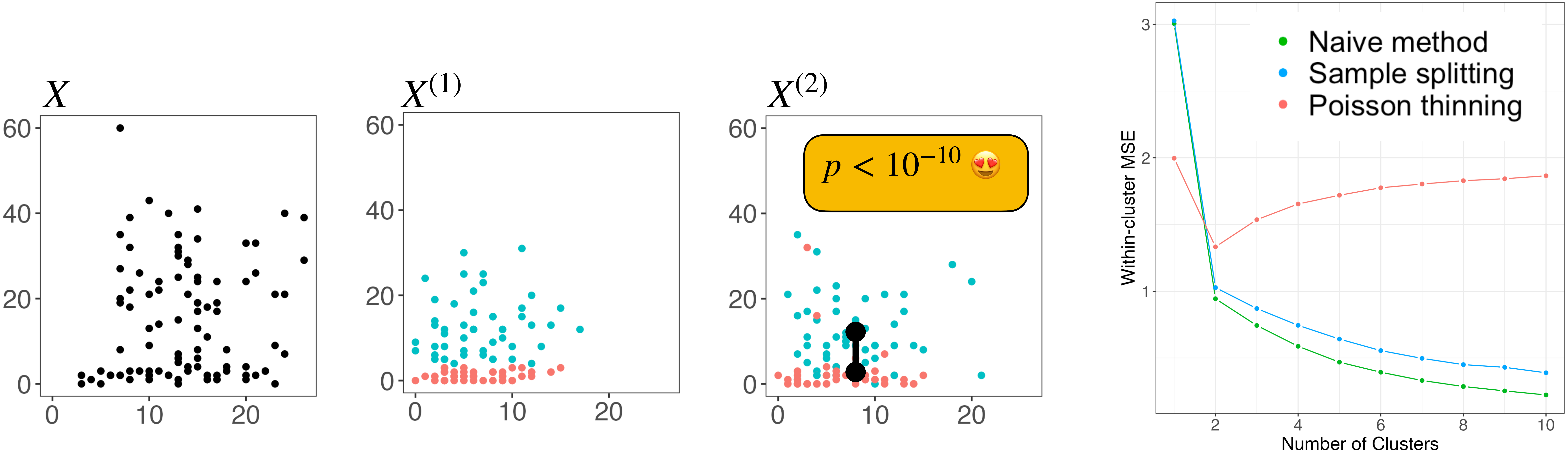
Thinning avoids the pitfall of sample splitting on our motivating examples



Thinning avoids the pitfall of sample splitting on our motivating examples



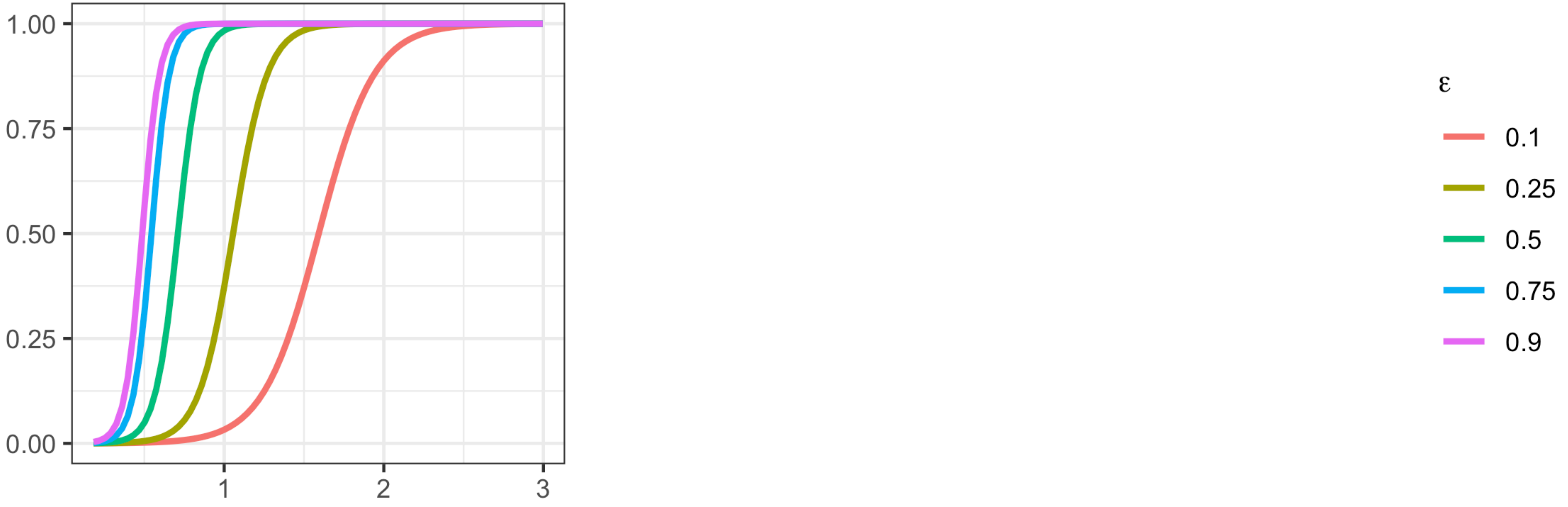
Thinning avoids the pitfall of sample splitting on our motivating examples



When letting $X_{ij}^{(1)} \sim \text{Binomial}(X_{ij}, \epsilon)$, how should we pick ϵ?

Large values of ϵ are helpful for estimating cell types

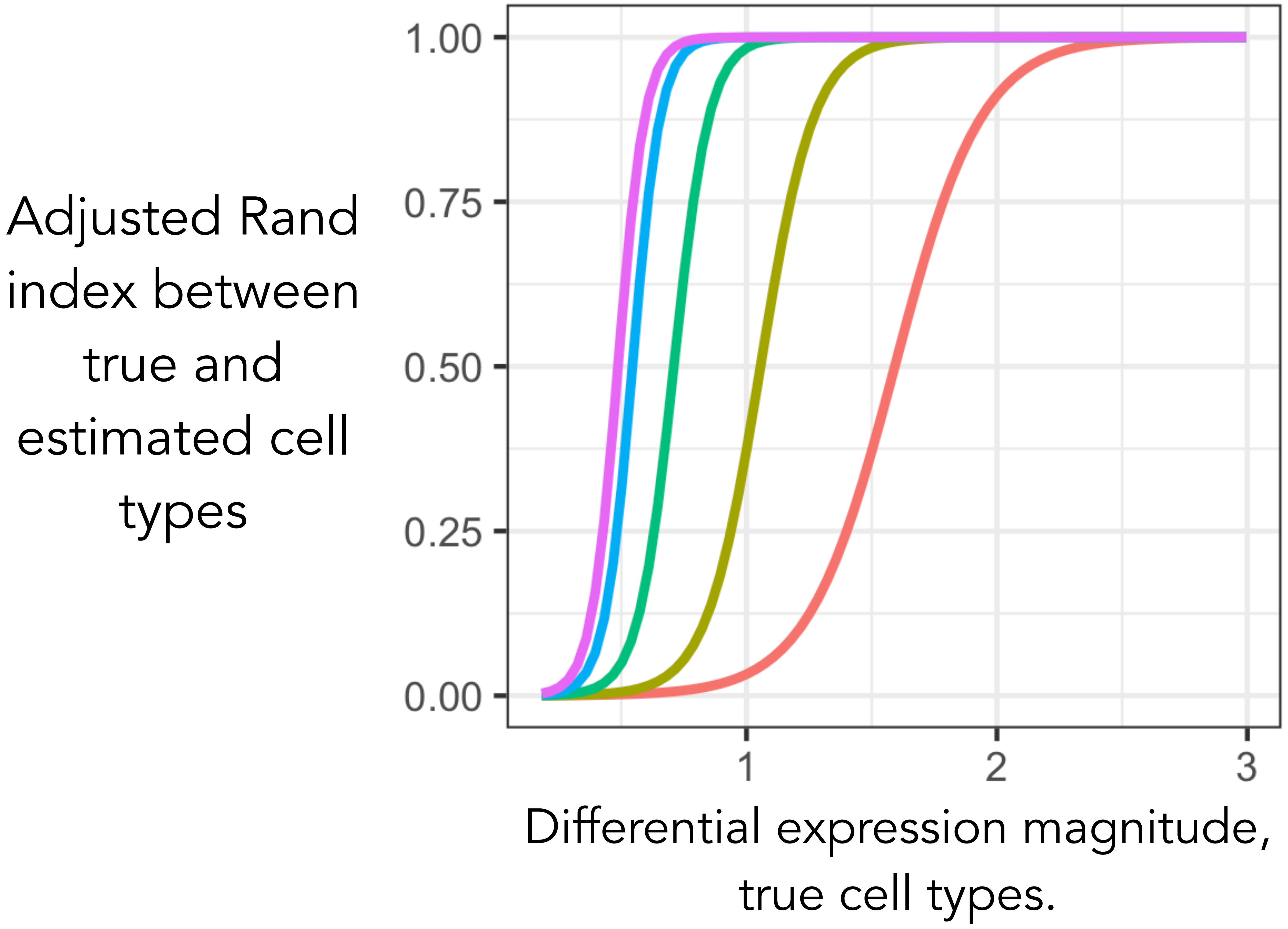
Adjusted Rand index between true and estimated cell types



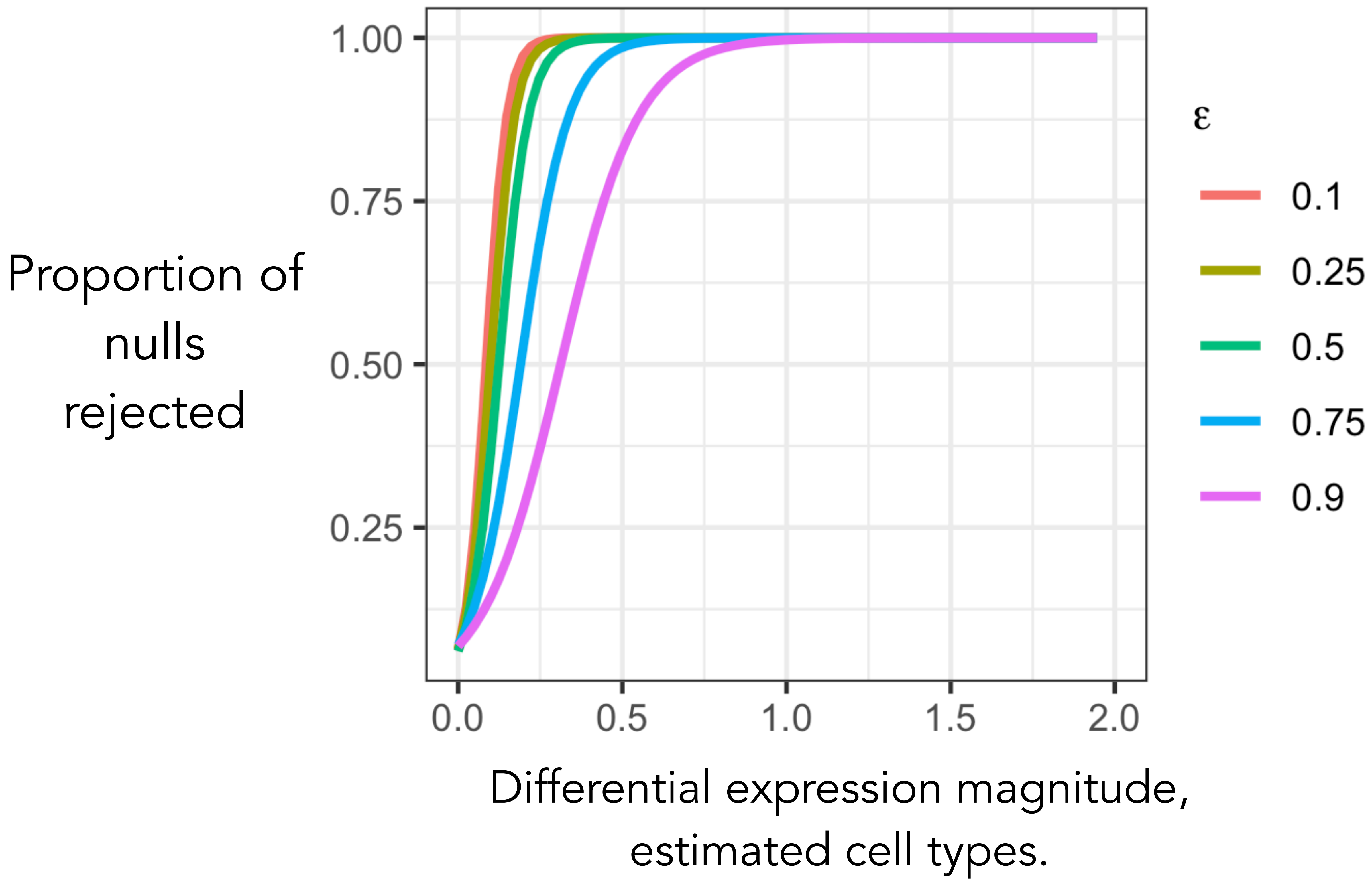
Differential expression magnitude, true cell types.

When letting $X_{ij}^{(1)} \sim \text{Binomial}(X_{ij}, \epsilon)$, how should we pick ϵ?

Large values of ϵ are helpful for estimating cell types,



but leave less power for differential expression testing.



Poisson thinning is useful in the analysis of single-cell RNA sequencing data

Biostatistics (2022) **00**, 00, *pp.* 1–18
https://doi.org/10.1093/biostatistics/kxac047

Inference after latent variable estimation for single-cell RNA sequencing data

ANNA NEUFELD*
Department of Statistics, University of Washington, Seattle, WA 98195, USA
aneufeld@uw.edu

LUCY L. GAO
Department of Statistics, University of British Columbia, BC V6T 1Z4, Canada

JOSHUA POPP
Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA

ALEXIS BATTLE
Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA and Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218, USA

DANIELA WITTEN
Department of Statistics, University of Washington, Seattle, WA 98195, USA and Department of Biostatistics, University of Washington, Seattle, WA 98195, USA

R package and tutorials:
https://anna-neufeld.github.io/countsplit/

Is the Poisson assumption reasonable?

Perspective | Published: 24 May 2021

Separating measurement and expression models clarifies confusion in single-cell RNA sequencing analysis

Abhishek Sarkar & Matthew Stephens

Nature Genetics **53**, 770–777 (2021) | Cite this article

Abstract

The high proportion of zeros in typical single-cell RNA sequencing datasets has led to widespread but inconsistent use of terminology such as dropout and missing data. Here, we argue that much of this terminology is unhelpful and confusing, and outline simple ideas to help to reduce confusion. These include: (1) observed single-cell RNA sequencing counts reflect both true gene expression levels and measurement error, and carefully distinguishing between these contributions helps to clarify thinking; and (2) method development should start with a Poisson measurement model, rather than more complex models, because it is simple and generally consistent with existing data. We outline how several existing methods can be viewed within this framework and highlight how these methods differ in their

Generalizations of Poisson thinning are needed

Choudhary and Satija *Genome Biology* (2022) 23:27
https://doi.org/10.1186/s13059-021-02584-9

Genome Biology

RESEARCH Open Access

Comparison and evaluation of statistical error models for scRNA-seq

Check for updates

Saket Choudhary[1] and Rahul Satija[1,2*]

Results: Here, we analyze 59 scRNA-seq datasets that span a wide range of technologies, systems, and sequencing depths in order to evaluate the performance of different error models. We find that while a Poisson error model appears appropriate for sparse datasets, we observe clear evidence of overdispersion for genes with sufficient sequencing depth in all biological systems, necessitating the use of a negative binomial model. Moreover, we find that the degree of overdispersion varies widely across datasets, systems, and gene abundances, and argues for a data-driven approach for parameter estimation.

Generalizations of Poisson thinning are needed

Choudhary and Satija *Genome Biology* (2022) 23:27
https://doi.org/10.1186/s13059-021-02584-9

Genome Biology

RESEARCH Open Access

Comparison and evaluation of statistical error models for scRNA-seq

Check for updates

Saket Choudhary[1] and Rahul Satija[1,2*]

When $X \sim \text{Poisson}(\Lambda)$:

- $\mathrm{E}[X] = \Lambda$,
- $\mathrm{Var}(X) = \Lambda$.

Results: Here, we analyze 59 scRNA-seq datasets that span a wide range of technologies, systems, and sequencing depths in order to evaluate the performance of different error models. We find that while a Poisson error model appears appropriate for sparse datasets, we observe clear evidence of overdispersion for genes with sufficient sequencing depth in all biological systems, necessitating the use of a negative binomial model. Moreover, we find that the degree of overdispersion varies widely across datasets, systems, and gene abundances, and argues for a data-driven approach for parameter estimation.

Generalizations of Poisson thinning are needed

Choudhary and Satija *Genome Biology* (2022) 23:27
https://doi.org/10.1186/s13059-021-02584-9

Genome Biology

RESEARCH Open Access

Comparison and evaluation of statistical error models for scRNA-seq

Saket Choudhary[1] and Rahul Satija[1,2*]

When $X \sim \text{Poisson}(\Lambda)$:

- $\mathrm{E}[X] = \Lambda$,
- $\mathrm{Var}(X) = \Lambda$.

When $X \sim \text{NB}(\Lambda, b)$:

- $\mathrm{E}[X] = \Lambda$,
- $\mathrm{Var}(X) = \Lambda + \dfrac{\Lambda^2}{b}$.

Results: Here, we analyze 59 scRNA-seq datasets that span a wide range of technologies, systems, and sequencing depths in order to evaluate the performance of different error models. We find that while a Poisson error model appears appropriate for sparse datasets, we observe clear evidence of overdispersion for genes with sufficient sequencing depth in all biological systems, necessitating the use of a negative binomial model. Moreover, we find that the degree of overdispersion varies widely across datasets, systems, and gene abundances, and argues for a data-driven approach for parameter estimation.

Poisson thinning fails when applied to negative binomial data

X

	Gene 1	[illegible]
Cell 1	18	[illegible]
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

X_{ij}

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, \epsilon)$$

$X^{(1)}$

	ene 1	Gene 2
[illegible]	14	1
[illegible]	[illegible]	6
Cell 3	5	17
Cell 4	6	25

$$X_{ij}^{(2)} := X_{ij} - X_{ij}^{(1)}$$

$X^{(2)}$

	Gene 3	Gene 4
ell 1	[illegible]	5
ell 2	21	2
Cell 3	6	14
Cell 4	16	9

Poisson thinning fails when applied to negative binomial data

X

	Gene 1	Ge
Cell 1	18	[illegible]
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

X_{ij}

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, \epsilon)$$

$$X_{ij}^{(2)} := X_{ij} - X_{ij}^{(1)}$$

$X^{(1)}$

	ene 1	Gene 2
	14	1
	[illegible]	6
Cell 3	5	17
Cell 4	6	25

$X^{(2)}$

	Gene 3	Gene 4
ell 1	[illegible]	5
ell 2	21	2
Cell 3	6	14
Cell 4	16	9

If $X_{ij} \sim \text{Poisson}(\Lambda_{ij})$, then:

1. $X_{ij}^{(1)} \sim \text{Poisson}(\epsilon\Lambda_{ij})$
2. $X_{ij}^{(2)} \sim \text{Poisson}((1-\epsilon)\Lambda_{ij})$
3. $X_{ij}^{(1)} \perp\!\!\!\perp X_{ij}^{(2)}$

Poisson thinning fails when applied to negative binomial data

X

	Gene 1	Gene 2
Cell 1	18	[illegible]
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

X_{ij}

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, \epsilon)$$

$X^{(1)}$

	Gene 1	Gene 2
Cell 1	14	1
Cell 2	[illegible]	6
Cell 3	5	17
Cell 4	6	25

$$X_{ij}^{(2)} := X_{ij} - X_{ij}^{(1)}$$

$X^{(2)}$

	Gene 3	Gene 4
Cell 1	[illegible]	5
Cell 2	21	2
Cell 3	6	14
Cell 4	16	9

If $X_{ij} \sim \text{NB}(\Lambda_{ij}, b_{ij})$, then:

1. $X_{ij}^{(1)} \sim \text{Poisson}(\epsilon \Lambda_{ij})$
2. $X_{ij}^{(2)} \sim \text{Poisson}((1-\epsilon)\Lambda_{ij})$
3. $X_{ij}^{(1)} \perp\!\!\!\perp X_{ij}^{(2)}$

Poisson thinning fails when applied to negative binomial data

X

	Gene 1	Ge
Cell 1	18	[illegible]
Cell 2	31	$\boxed{8}$
Cell 3	11	31
Cell 4	22	34

X_{ij}

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, \epsilon)$$

$X^{(1)}$

	ene 1	Gene 2
[illegible]	14	1
[illegible]	[illegible]	$\boxed{6}$
Cell 3	5	17
Cell 4	6	25

$$X_{ij}^{(2)} := X_{ij} - X_{ij}^{(1)}$$

$X^{(2)}$

	Gene 3	Gene 4
ell 1	[illegible]	5
ell 2	21	$\boxed{2}$
Cell 3	6	14
Cell 4	16	9

If $X_{ij} \sim \text{NB}(\Lambda_{ij}, b_{ij})$, then:

1. $\cancel{X_{ij}^{(1)} \sim \text{Poisson}(\epsilon\Lambda_{ij})}$
2. $\cancel{X_{ij}^{(2)} \sim \text{Poisson}((1-\epsilon)\Lambda_{ij})}$
3. $\cancel{X_{ij}^{(1)} \perp\!\!\!\perp X_{ij}^{(2)}}$

Poisson thinning fails when applied to negative binomial data

X

	Gene 1	Gene 2
Cell 1	18	[illegible]
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

X_{ij}

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, \epsilon)$$

$X^{(1)}$

	Gene 1	Gene 2
Cell 1	14	1
Cell 2	[illegible]	6
Cell 3	5	17
Cell 4	6	25

$$X_{ij}^{(2)} := X_{ij} - X_{ij}^{(1)}$$

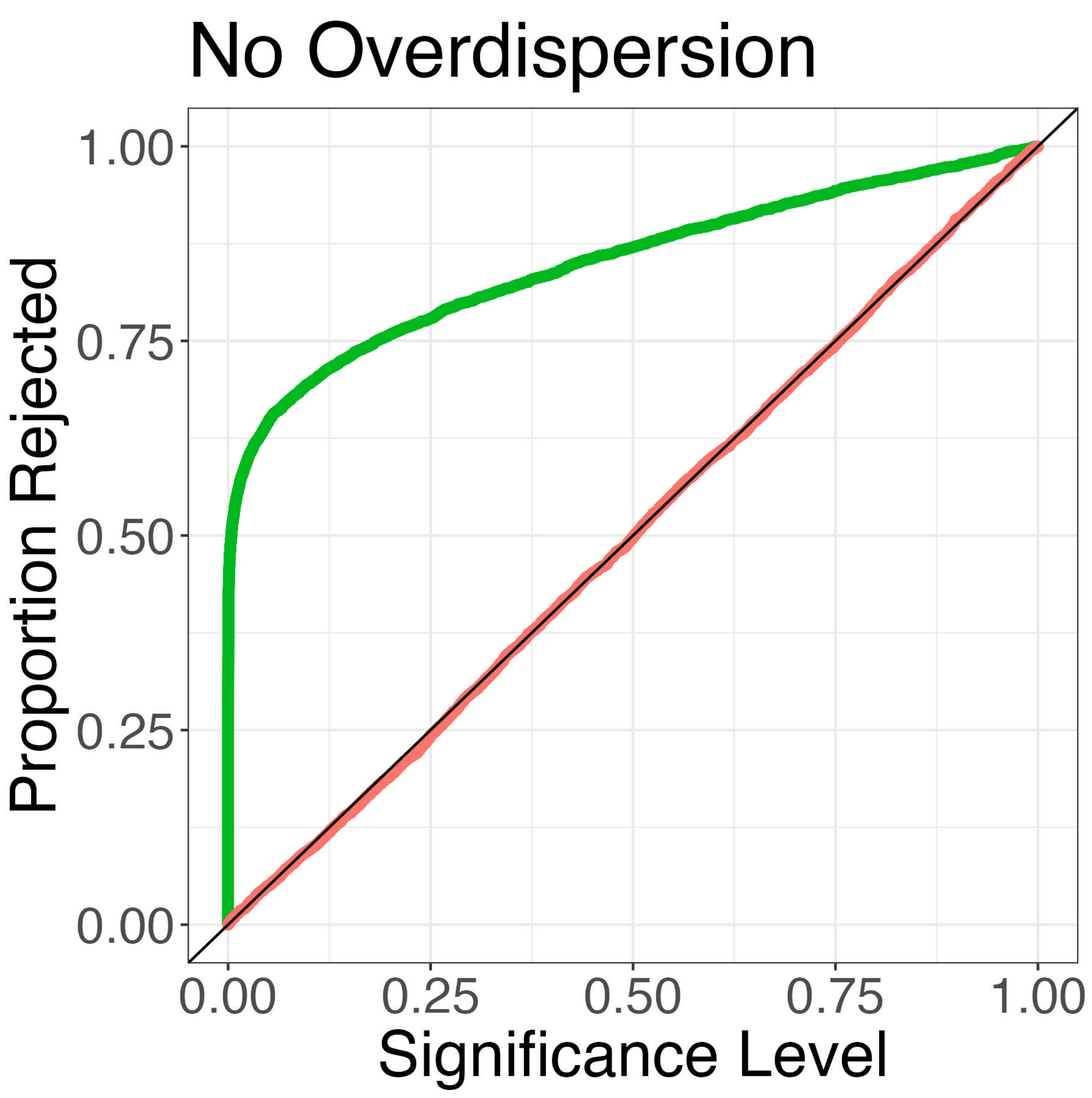
$X^{(2)}$

	Gene 3	Gene 4
Cell 1	[illegible]	5
Cell 2	21	2
Cell 3	6	14
Cell 4	16	9

If $X_{ij} \sim \text{NB}(\Lambda_{ij}, b_{ij})$, then:

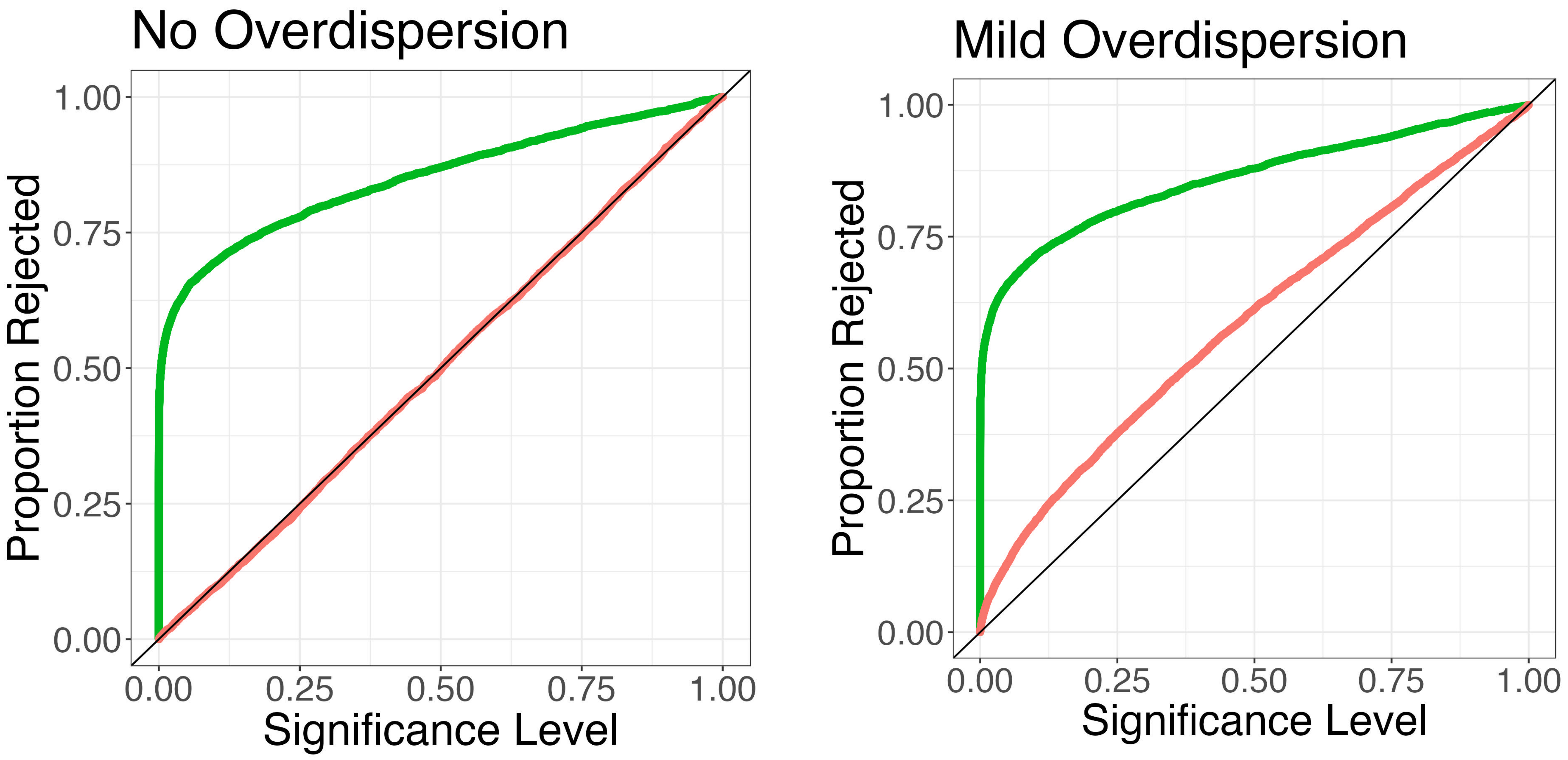
1. $\text{E}[X_{ij}^{(1)}] = \epsilon \Lambda_{ij}$,
2. $\text{E}[X_{ij}^{(2)}] = (1-\epsilon)\Lambda_{ij}$,
3. $\text{Cov}\left(X_{ij}^{(1)}, X_{ij}^{(2)}\right) > 0.$

Poisson thinning fails when applied to negative binomial data



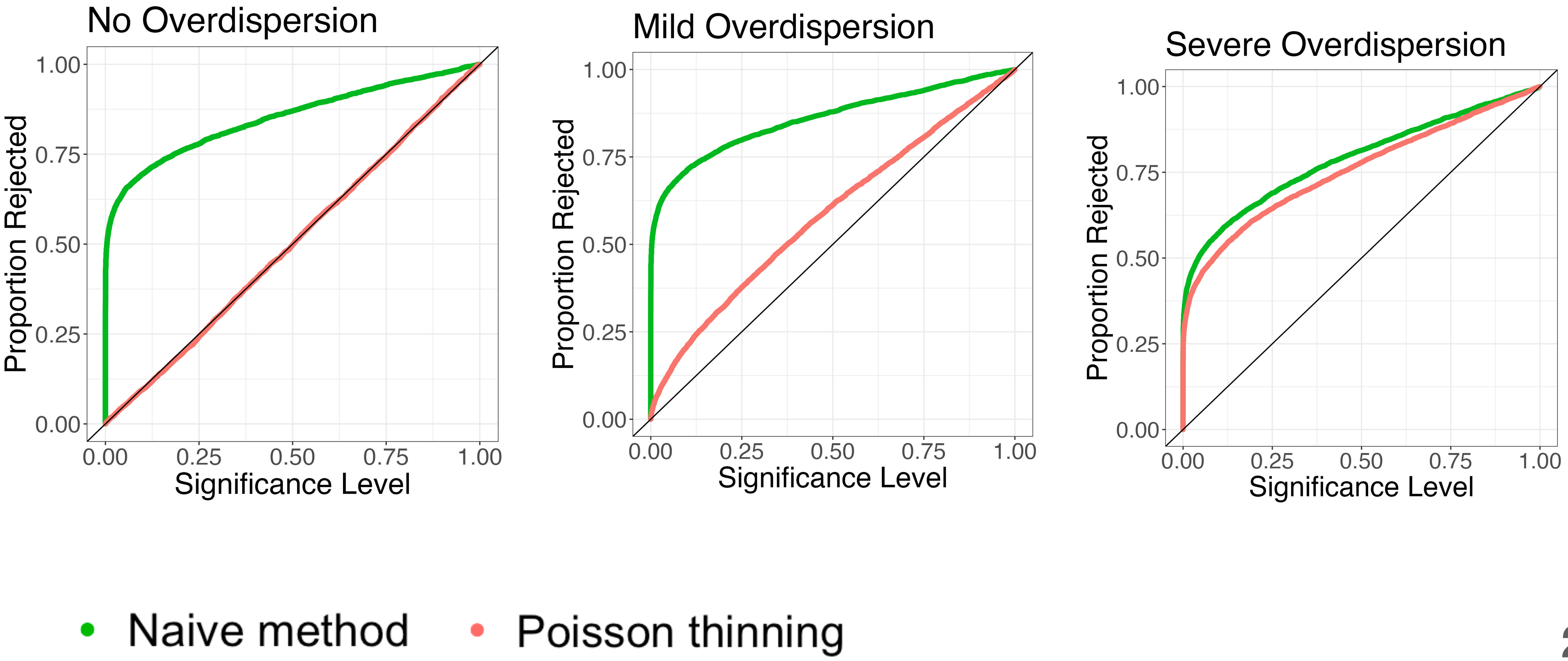
• Naive method • Poisson thinning

Poisson thinning fails when applied to negative binomial data



• Naive method • Poisson thinning

Poisson thinning fails when applied to negative binomial data



Outline

1. Motivation: settings where sample splitting doesn't work
2. Poisson thinning
3. **Data thinning**
4. Application to human fetal cell atlas data
5. Application to cardiomyocyte differentiation data
6. Ongoing work

What did we like about Poisson thinning?

We split a single observation X into $X^{(1)}$ and $X^{(2)}$ such that:

(1) $X^{(1)}$ and $X^{(2)}$ have the same distribution as X, up to a parameter scaling.

(2) $X^{(1)} \perp\!\!\!\perp X^{(2)}$.

What did we like about Poisson thinning?

We split a single observation X into $X^{(1)}$ and $X^{(2)}$ such that:

(1) $X^{(1)}$ and $X^{(2)}$ have the same distribution as X, up to a parameter scaling.

(2) $X^{(1)} \perp\!\!\!\perp X^{(2)}$.

Can we achieve these same properties when X is not Poisson?

The Poisson distribution is “convolution-closed”

The Poisson distribution is "convolution-closed"

If $X' \sim \mathrm{Poisson}\,(\epsilon\Lambda)$ and $X'' \sim \mathrm{Poisson}((1-\epsilon)\Lambda)$, with X' independent of X'', then $X' + X'' \sim \mathrm{Poisson}\,(\Lambda)$.

The Poisson distribution is "convolution-closed"

If $X' \sim \text{Poisson}\,(\epsilon\Lambda)$ and $X'' \sim \text{Poisson}((1-\epsilon)\Lambda)$, with X' independent of X'', then $X' + X'' \sim \text{Poisson}\,(\Lambda)$.

The well-known Poisson thinning operator "undoes" this sum, by noting that the conditional distribution of $X' \mid X' + X'' = x$ is $\text{Binomial}\,(x, \epsilon)$.

The Poisson distribution is "convolution-closed"

If $X' \sim \mathrm{Poisson}\,(\epsilon\Lambda)$ and $X'' \sim \mathrm{Poisson}((1-\epsilon)\Lambda)$, with X' independent of X'', then $X' + X'' \sim \mathrm{Poisson}\,(\Lambda)$.

The well-known Poisson thinning operator "undoes" this sum, by noting that the conditional distribution of $X' \mid X' + X'' = x$ is $\mathrm{Binomial}\,(x, \epsilon)$.

The Poisson distribution is "convolution-closed"

If $X' \sim \text{Poisson}\,(\epsilon\Lambda)$ and $X'' \sim \text{Poisson}((1-\epsilon)\Lambda)$, with X' independent of X'', then $X' + X'' \sim \text{Poisson}\,(\Lambda)$.

The well-known Poisson thinning operator "undoes" this sum, by noting that the conditional distribution of $X' \mid X' + X'' = x$ is $\text{Binomial}\,(x, \epsilon)$.

The negative binomial distribution is also convolution-closed.

The negative binomial distribution is “convolution-closed”

The negative binomial distribution is "convolution-closed"

If $X' \sim \mathrm{NB}\left(\epsilon\Lambda, \epsilon b\right)$ and $X'' \sim \mathrm{NB}((1-\epsilon)\Lambda, (1-\epsilon)b)$, with X' independent of X'', then $X' + X'' \sim \mathrm{NB}\left(\Lambda, b\right)$.

The negative binomial distribution is "convolution-closed"

If $X' \sim \mathrm{NB}\left(\epsilon\Lambda, \epsilon b\right)$ and $X'' \sim \mathrm{NB}((1-\epsilon)\Lambda, (1-\epsilon)b)$, with X' independent of X'', then $X' + X'' \sim \mathrm{NB}\left(\Lambda, b\right)$.

The conditional distribution of $X' \mid X' + X'' = x$ is $\mathrm{BetaBinomial}\left(x, \epsilon b, (1-\epsilon)b\right)$.

The negative binomial distribution is "convolution-closed"

If $X' \sim \mathrm{NB}\left(\epsilon\Lambda, \epsilon b\right)$ and $X'' \sim \mathrm{NB}((1-\epsilon)\Lambda, (1-\epsilon)b)$, with X' independent of X'', then $X' + X'' \sim \mathrm{NB}\left(\Lambda, b\right)$.

The conditional distribution of $X' \mid X' + X'' = x$ is $\mathrm{BetaBinomial}\left(x, \epsilon b, (1-\epsilon)b\right)$.

We can "undo" this sum!

Negative binomial data thinning

X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

Negative binomial data thinning

X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

$X^{(1)}$

	Cene 1	Gene 2
Cell 1	14	1
Cell 2	10	6
Cell 3	5	17
Cell 4	6	25

$X^{(2)}$

	Gene 3	Gene 4
Cell 1	4	5
Cell 2	21	2
Cell 3	6	14
Cell 4	16	9

Negative binomial data thinning

X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

X_{ij}

$X^{(1)}$

	Cene 1	Gene 2
Cell 1	14	1
Cell 2	[illegible]	6
Cell 3	5	17
Cell 4	6	25

$X^{(2)}$

	Gene 3	Gene 4
Cell 1	[illegible]	5
Cell 2	21	2
Cell 3	6	14
Cell 4	16	9

Negative binomial data thinning

X

[illegible]	[illegible]	[illegible]
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

X_{ij}

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{BetaBinomial}(x_{ij}, \epsilon b_{ij}, (1-\epsilon) b_{ij})$$

$X^{(1)}$

	[illegible]	**Gene 2**
[illegible]	14	1
[illegible]	[illegible]	6
Cell 3	5	17
Cell 4	6	25

$X^{(2)}$

	Gene 3	**Gene 4**
Cell 1	[illegible]	5
Cell 2	21	2
Cell 3	6	14
Cell 4	16	9

$$X_{ij}^{(2)} := X_{ij} - X_{ij}^{(1)}$$

Negative binomial data thinning

X

Cell 1	[illegible]	[illegible]
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

X_{ij}

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{BetaBinomial}(x_{ij}, \epsilon b_{ij}, (1-\epsilon) b_{ij})$$

$X^{(1)}$

	Gene 1	Gene 2
[illegible]	14	1
[illegible]	[illegible]	6
Cell 3	5	17
Cell 4	6	25

$X^{(2)}$

	Gene 3	Gene 4
Cell 1	[illegible]	5
Cell 2	21	2
Cell 3	6	14
Cell 4	16	9

$$X_{ij}^{(2)} := X_{ij} - X_{ij}^{(1)}$$

If $X_{ij} \sim \text{NB}\left(\Lambda_{ij}, b_{ij}\right)$, then:

1. $X_{ij}^{(1)} \sim \text{NB}(\epsilon \Lambda_{ij}, \epsilon b_{ij})$
2. $X_{ij}^{(2)} \sim \text{NB}((1-\epsilon)\Lambda_{ij}, (1-\epsilon) b_{ij})$
3. $X_{ij}^{(1)} \perp\!\!\!\perp X_{ij}^{(2)}$.

A new result.

Negative binomial data thinning

X

Cell 1	[illegible]	[illegible]
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

X_{ij}

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \mathrm{BetaBinomial}(x_{ij}, \epsilon b_{ij}, (1-\epsilon) b_{ij})$$

$X^{(1)}$

	Gene 1	Gene 2
Cell 1	14	1
Cell 2	[illegible]	6
Cell 3	5	17
Cell 4	6	25

Estimate clusters.

$X^{(2)}$

	Gene 3	Gene 4
Cell 1	[illegible]	5
Cell 2	21	2
Cell 3	6	14
Cell 4	16	9

$$X_{ij}^{(2)} := X_{ij} - X_{ij}^{(1)}$$

If $X_{ij} \sim \mathrm{NB}\left(\Lambda_{ij}, b_{ij}\right)$, then:

1. $X_{ij}^{(1)} \sim \mathrm{NB}(\epsilon \Lambda_{ij}, \epsilon b_{ij})$
2. $X_{ij}^{(2)} \sim \mathrm{NB}((1-\epsilon)\Lambda_{ij}, (1-\epsilon) b_{ij})$
3. $X_{ij}^{(1)} \perp\!\!\!\perp X_{ij}^{(2)}$.

A new result.

Negative binomial data thinning

X

	[illegible]	[illegible]
Cell 1	[illegible]	[illegible]
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34

X_{ij}

$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{BetaBinomial}(x_{ij}, \epsilon b_{ij}, (1-\epsilon) b_{ij})$$

$X^{(1)}$

	Gene 1	Gene 2
[illegible]	14	1
[illegible]	[illegible]	6
Cell 3	5	17
Cell 4	6	25

Estimate clusters.

$X^{(2)}$

$$X_{ij}^{(2)} := X_{ij} - X_{ij}^{(1)}$$

	Gene 3	Gene 4
Cell 1	[illegible]	5
Cell 2	21	2
Cell 3	6	14
Cell 4	16	9

Evaluate clusters or test for differential expression.

If $X_{ij} \sim \text{NB}\left(\Lambda_{ij}, b_{ij}\right)$, then:

1. $X_{ij}^{(1)} \sim \text{NB}(\epsilon \Lambda_{ij}, \epsilon b_{ij})$
2. $X_{ij}^{(2)} \sim \text{NB}((1-\epsilon)\Lambda_{ij}, (1-\epsilon) b_{ij})$
3. $X_{ij}^{(1)} \perp\!\!\!\perp X_{ij}^{(2)}$.

A new result.

What if we do not know the value of the overdispersion parameter?

Negative binomial thinning algorithm

Suppose $X \sim \mathrm{NB}\left(\Lambda, b\right)$.

Draw

$X^{(1)} \sim \mathrm{BetaBinomial}(x, \epsilon b, (1-\epsilon)b)$,

$X^{(2)} = X - X^{(1)}$, then:

1) $X^{(1)} \sim \mathrm{NB}\left(\epsilon\Lambda, \epsilon b\right)$.
2) $X^{(2)} \sim \mathrm{NB}\left((1-\epsilon)\Lambda, (1-\epsilon)b\right)$
3) $X^{(1)} \perp\!\!\!\perp X^{(2)}$.

What if we do not know the value of the overdispersion parameter?

Negative binomial thinning algorithm

Suppose $X \sim \mathrm{NB}\left(\Lambda, b\right)$.

Draw

$X^{(1)} \sim \mathrm{BetaBinomial}(x, \epsilon b, (1-\epsilon)b)$,

$X^{(2)} = X - X^{(1)}$, then:

1) $X^{(1)} \sim \mathrm{NB}\left(\epsilon\Lambda, \epsilon b\right)$.
2) $X^{(2)} \sim \mathrm{NB}\left((1-\epsilon)\Lambda, (1-\epsilon)b\right)$
3) $X^{(1)} \perp\!\!\!\perp X^{(2)}$.

What if we do not know the value of the overdispersion parameter?

Negative binomial thinning algorithm

Suppose $X \sim \mathrm{NB}\left(\Lambda, b\right)$.

Draw

$X^{(1)} \sim \mathrm{BetaBinomial}(x, \epsilon b, (1-\epsilon)b)$,

$X^{(2)} = X - X^{(1)}$, then:

1) ~~$X^{(1)} \sim \mathrm{NB}\left(\epsilon\Lambda, \epsilon b\right)$.~~
2) ~~$X^{(2)} \sim \mathrm{NB}\left((1-\epsilon)\Lambda, (1-\epsilon)b\right)$~~
3) ~~$X^{(1)} \perp\!\!\!\perp X^{(2)}$.~~

What if we do not know the value of the overdispersion parameter?

Negative binomial thinning algorithm

Suppose $X \sim \mathrm{NB}\left(\Lambda, b\right)$.

Draw

$X^{(1)} \sim \mathrm{BetaBinomial}(x, \epsilon\tilde{b}, (1-\epsilon)\tilde{b}\ )$,

$X^{(2)} = X - X^{(1)}$, then:

1) $\mathrm{E}[X^{(1)}] = \epsilon\Lambda.$
2) $\mathrm{E}[X^{(2)}] = (1-\epsilon)\Lambda$
3) $\mathrm{Cov}\left(X^{(1)}, X^{(2)}\right) = \epsilon(1-\epsilon)\frac{\Lambda^2}{b}\left(1 - \frac{b+1}{\tilde{b}+1}\right).$

Negative binomial thinning is useful for scRNA-seq data

arXiv > stat > arXiv:2307.12985

Search...

Help | Advanced

Statistics > Methodology

[Submitted on 24 Jul 2023]

Negative binomial count splitting for single-cell RNA sequencing data

Anna Neufeld, Joshua Popp, Lucy L. Gao, Alexis Battle, Daniela Witten

R package and tutorials:
https://anna-neufeld.github.io/countsplit/

We can follow the same recipe for any convolution-closed distribution

Distribution of X:	Draw $X^{(1)} \mid X = x$ from $G_{\epsilon,x}$, where $G_{\epsilon,x}$ is:	Distribution of $X^{(1)}$:	Distribution of $X^{(2)}$, where $X^{(2)} = X - X^{(1)}$:
$\text{Poisson}(\lambda)$	$\text{Binomial}(x, \epsilon)$	$\text{Poisson}(\epsilon\lambda)$	$\text{Poisson}((1-\epsilon)\lambda)$
$\text{N}(\mu, \sigma^2)$	$\text{N}(\epsilon x, \epsilon(1-\epsilon)\sigma^2)$	$\text{N}(\epsilon\mu, \epsilon\sigma^2)$	$\text{N}((1-\epsilon)\mu, (1-\epsilon)\sigma^2)$
$\text{NegativeBinomial}(\mu, b)$	$\text{BetaBinomial}(x, \epsilon b, (1-\epsilon)b)$.	$\text{NegativeBinomial}(\epsilon\mu, \epsilon b)$	$\text{NegativeBinomial}((1-\epsilon)\mu, (1-\epsilon)b)$
$\text{Binomial}(r, p)$	$\text{Hypergeometric}(\epsilon r, (1-\epsilon)r, x)$.	$\text{Binomial}(\epsilon r, p)$	$\text{Binomial}((1-\epsilon)r, p)$
$\text{Gamma}(\alpha, \beta)$	$x \cdot \text{Beta}\left(\epsilon\alpha, (1-\epsilon)\alpha\right)$.	$\text{Gamma}(\epsilon\alpha, \beta)$	$\text{Gamma}((1-\epsilon)\alpha, \beta)$
$\text{Exponential}(\lambda)$	$x \cdot \text{Beta}(\epsilon, (1-\epsilon))$.	$\text{Gamma}(\epsilon, \lambda)$	$\text{Gamma}(1-\epsilon, \lambda)$
$\text{N}_k(\mu, \Sigma)$	$\text{N}(\epsilon x, \epsilon(1-\epsilon)\Sigma)$.	$\text{N}_k(\epsilon\mu, \epsilon\Sigma)$	$\text{N}_k((1-\epsilon)\mu, (1-\epsilon)\Sigma)$
$\text{Multinomial}_k(r, p)$	$\text{MultivarHypergeom}(x_1, \ldots, x_K, \epsilon r)$	$\text{Multinom}_k(\epsilon r, p)$	$\text{Multinomial}_k((1-\epsilon)r, p)$
$\text{Wishart}_p(n, \Sigma)$.	$x^{1/2} Z x^{1/2}$, where . $Z \sim \text{MatrixBeta}_p(\epsilon n/2, (1-\epsilon)n/2)$	$\text{Wishart}_p(\epsilon n, \Sigma)$	$\text{Wishart}_p((1-\epsilon)n, \Sigma)$

Data thinning is a simple alternative to sample splitting that can be used in a variety of settings

arXiv > stat > arXiv:2301.07276

Search...

Help | Advanced

Statistics > Methodology

[Submitted on 18 Jan 2023]

Data thinning for convolution-closed distributions

Anna Neufeld, Ameer Dharamshi, Lucy L. Gao, Daniela Witten

We propose data thinning, a new approach for splitting an observation into two or more independent parts that sum to the original observation, and that follow the same distribution as the original observation, up to a (known) scaling of a parameter. This proposal is very general, and can be applied to any observation drawn from a "convolution closed" distribution, a class that includes the Gaussian, Poisson, negative binomial, Gamma, and binomial distributions, among others. It is similar in spirit to -- but distinct from, and more easily applicable than -- a recent proposal known as data fission. Data thinning has a number of applications to model selection, evaluation, and inference. For instance, cross-validation via data thinning provides an attractive alternative to the "usual" approach of cross-validation via sample splitting, especially in unsupervised settings in which the latter is not applicable. In simulations and in an application to single-cell RNA-sequencing data, we show that data thinning can be used to validate the results of unsupervised learning approaches, such as k-means clustering and principal components analysis.

R package and tutorials: https://anna-neufeld.github.io/datathin/

Outline

1. Motivation: settings where sample splitting doesn’t work
2. Poisson thinning (count splitting)
3. Data thinning
4. **Application to human fetal cell atlas data**
5. Application to cardiomyocyte differentiation data
6. Ongoing work

How can we validate the results of a clustering?

RESEARCH ARTICLE

Cao *et al.*, *Science* **370**, 808 (2020)

HUMAN GENOMICS

A human cell atlas of fetal gene expression

Junyue Cao[1*], Diana R. O'Day[2], Hannah A. Pliner[3], Paul D. Kingsley[4], Mei Deng[2], Riza M. Daza[1], Michael A. Zager[3,5], Kimberly A. Aldinger[2,6], Ronnie Blecher-Gonen[1], Fan Zhang[7], Malte Spielmann[8,9], James Palis[4], Dan Doherty[2,3,6], Frank J. Steemers[7], Ian A. Glass[2,3,6], Cole Trapnell[1,3,10†], Jay Shendure[1,3,10,11†]

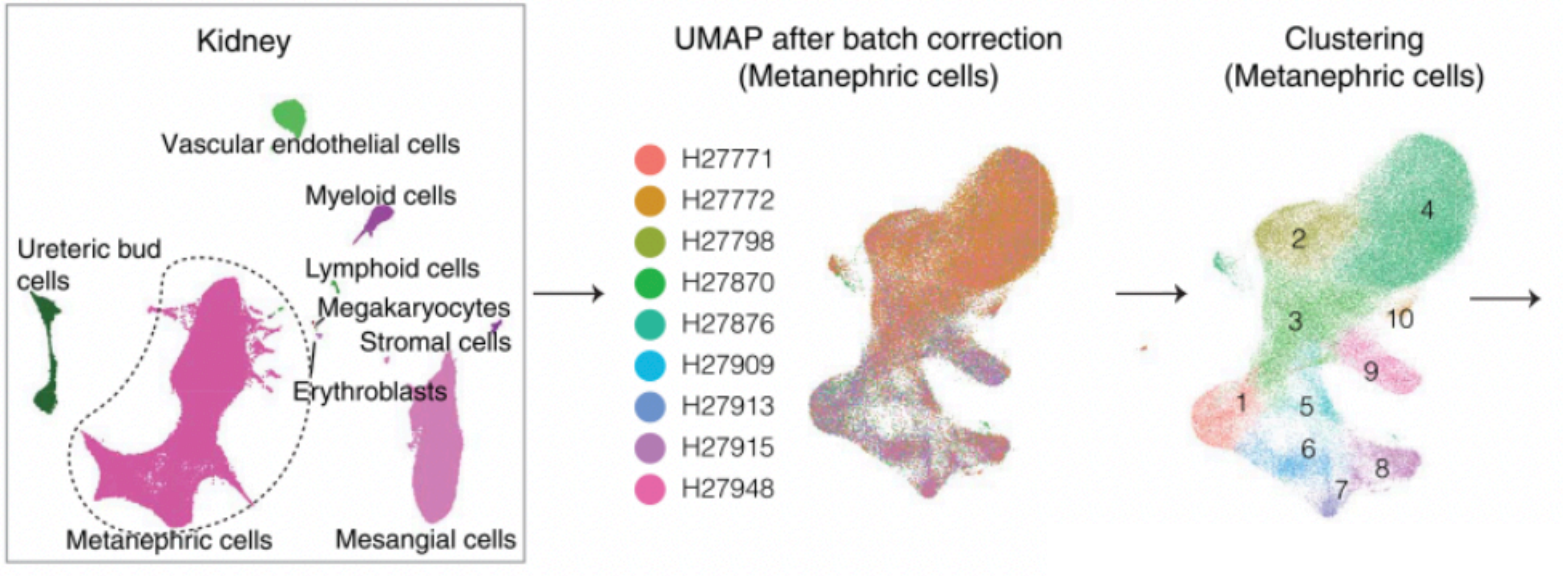
How can we validate the results of a clustering?

RESEARCH ARTICLE Cao *et al.*, *Science* **370**, 808 (2020)

HUMAN GENOMICS

A human cell atlas of fetal gene expression

Junyue Cao[1]*, Diana R. O'Day[2], Hannah A. Pliner[3], Paul D. Kingsley[4], Mei Deng[2], Riza M. Daza[1], Michael A. Zager[3,5], Kimberly A. Aldinger[2,6], Ronnie Blecher-Gonen[1], Fan Zhang[7], Malte Spielmann[8,9], James Palis[4], Dan Doherty[2,3,6], Frank J. Steemers[7], Ian A. Glass[2,3,6], Cole Trapnell[1,3,10]†, Jay Shendure[1,3,10,11]†



How can we validate the results of a clustering?

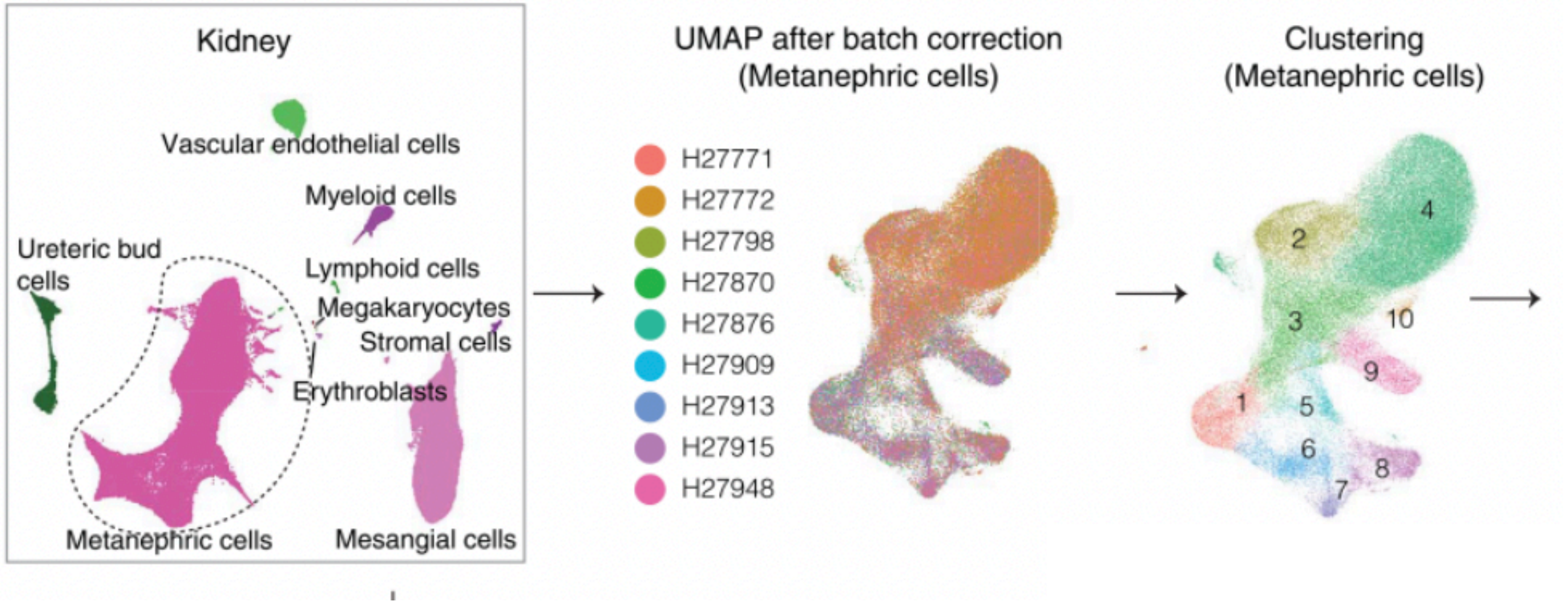

RESEARCH ARTICLE

Cao *et al.*, *Science* **370**, 808 (2020)

HUMAN GENOMICS

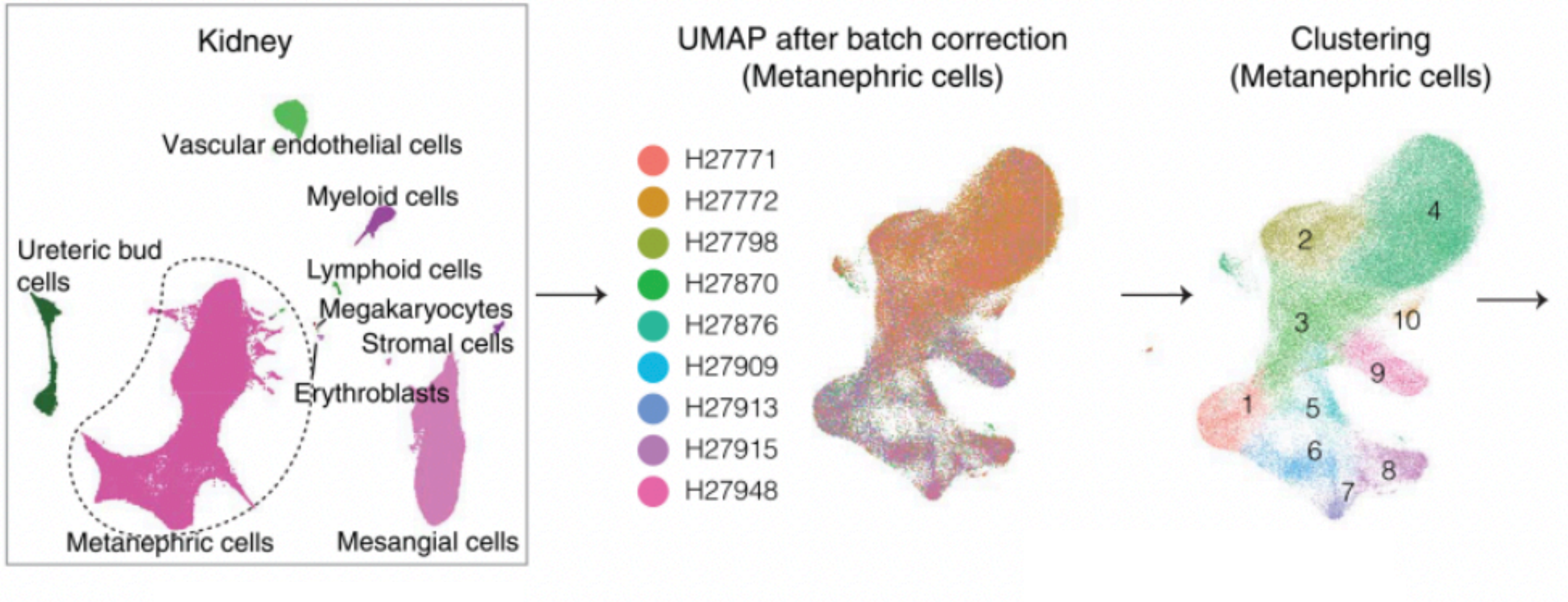
A human cell atlas of fetal gene expression

Junyue Cao[1]*, Diana R. O'Day[2], Hannah A. Pliner[3], Paul D. Kingsley[4], Mei Deng[2], Riza M. Daza[1], Michael A. Zager[3,5], Kimberly A. Aldinger[2,6], Ronnie Blecher-Gonen[1], Fan Zhang[7], Malte Spielmann[8,9], James Palis[4], Dan Doherty[2,3,6], Frank J. Steemers[7], Ian A. Glass[2,3,6], Cole Trapnell[1,3,10]†, Jay Shendure[1,3,10,11]†

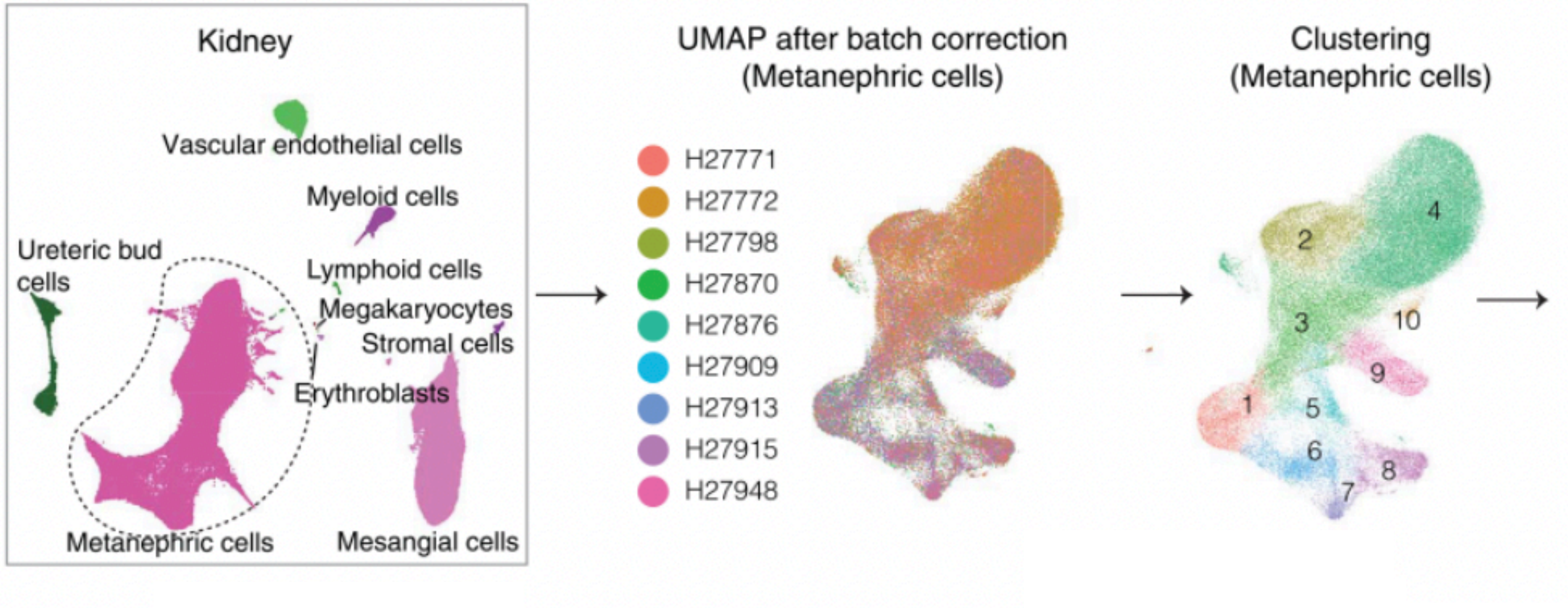


Are these clusters reproducible?

Can the cluster labels be reliably reproduced?



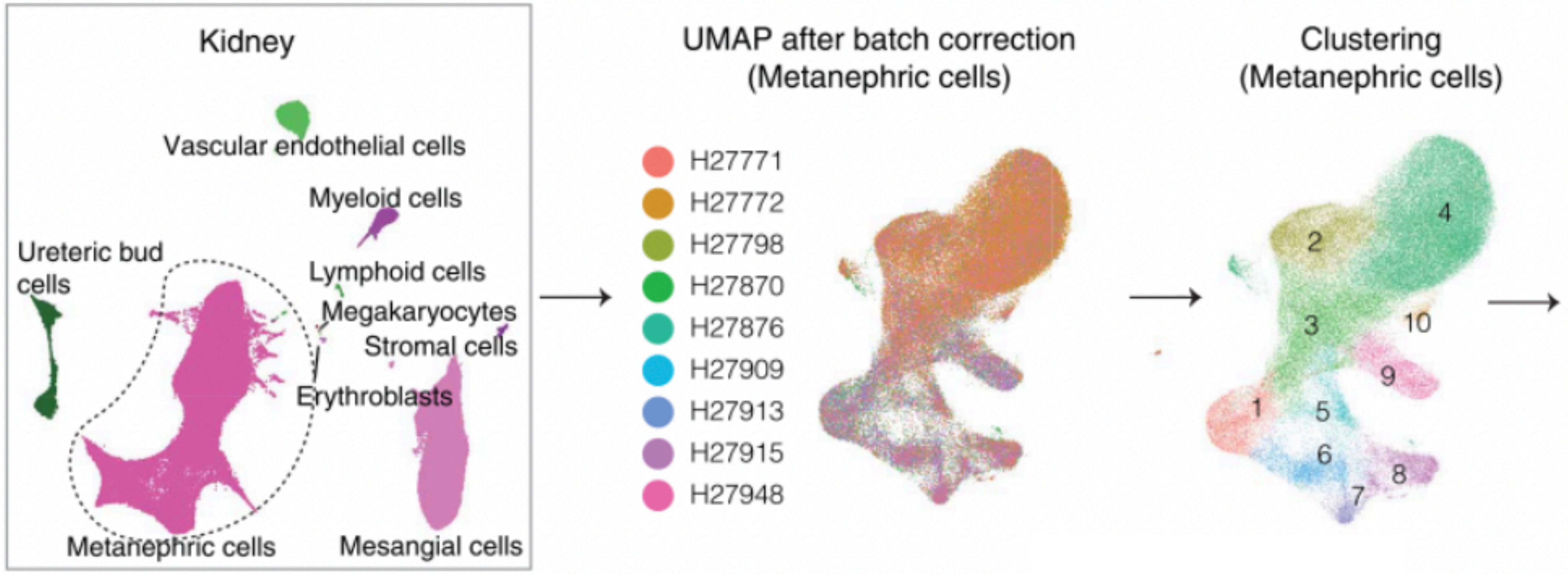
Can the cluster labels be reliably reproduced?



Intradataset cross validation (Cao et al.)

- **Step 1:** Cluster the cells.

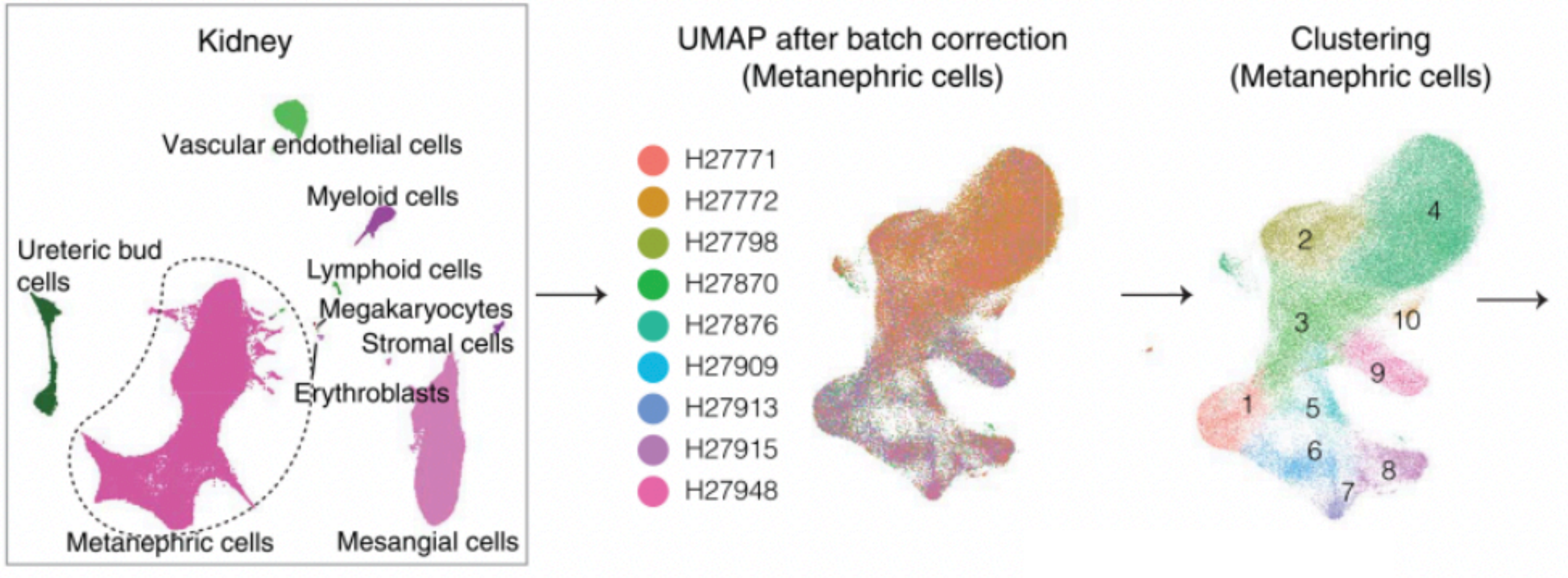
Can the cluster labels be reliably reproduced?



Intradataset cross validation (Cao et al.)

- **Step 1:** Cluster the cells.
- **Step 2:** Treat the cluster labels as the true responses. Train a classifier to predict these labels.

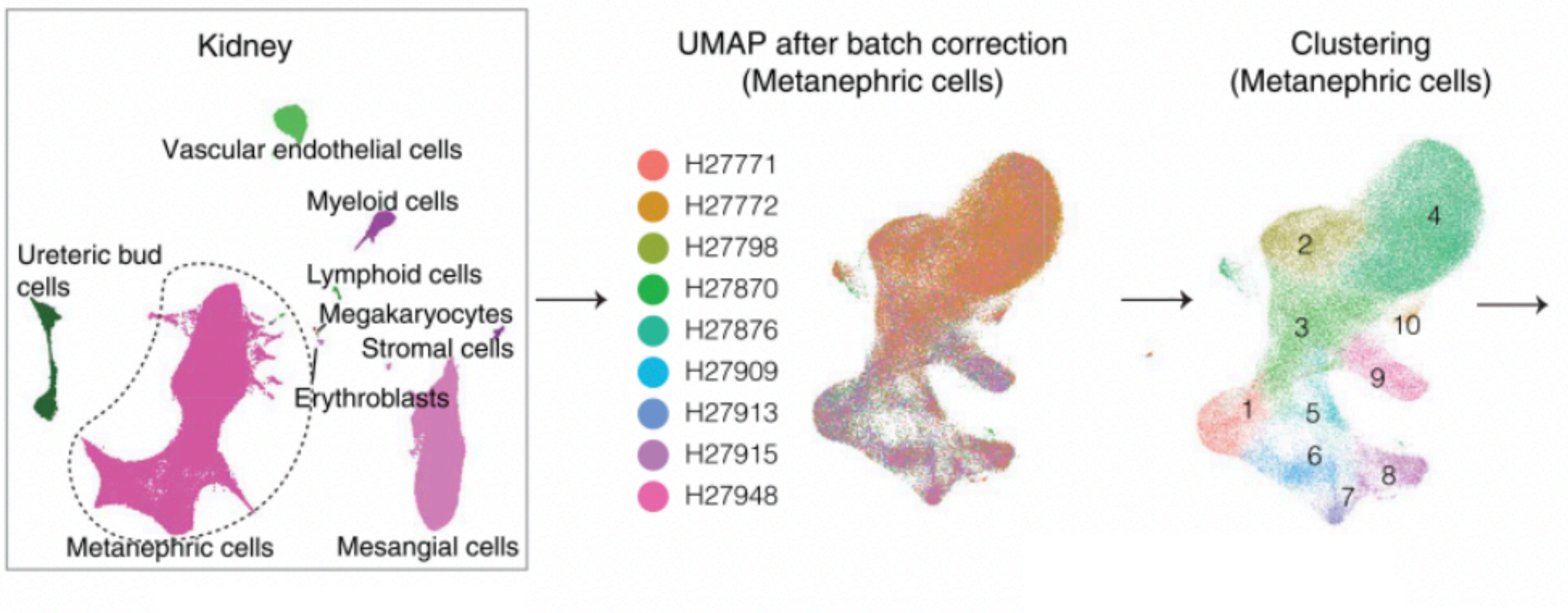
Can the cluster labels be reliably reproduced?



Intradataset cross validation (Cao et al.)

- **Step 1:** Cluster the cells.
- **Step 2:** Treat the cluster labels as the true responses. Train a classifier to predict these labels.
- **Step 3:** Compare original clustering labels to labels predicted by classifier.

Can the cluster labels be reliably reproduced?

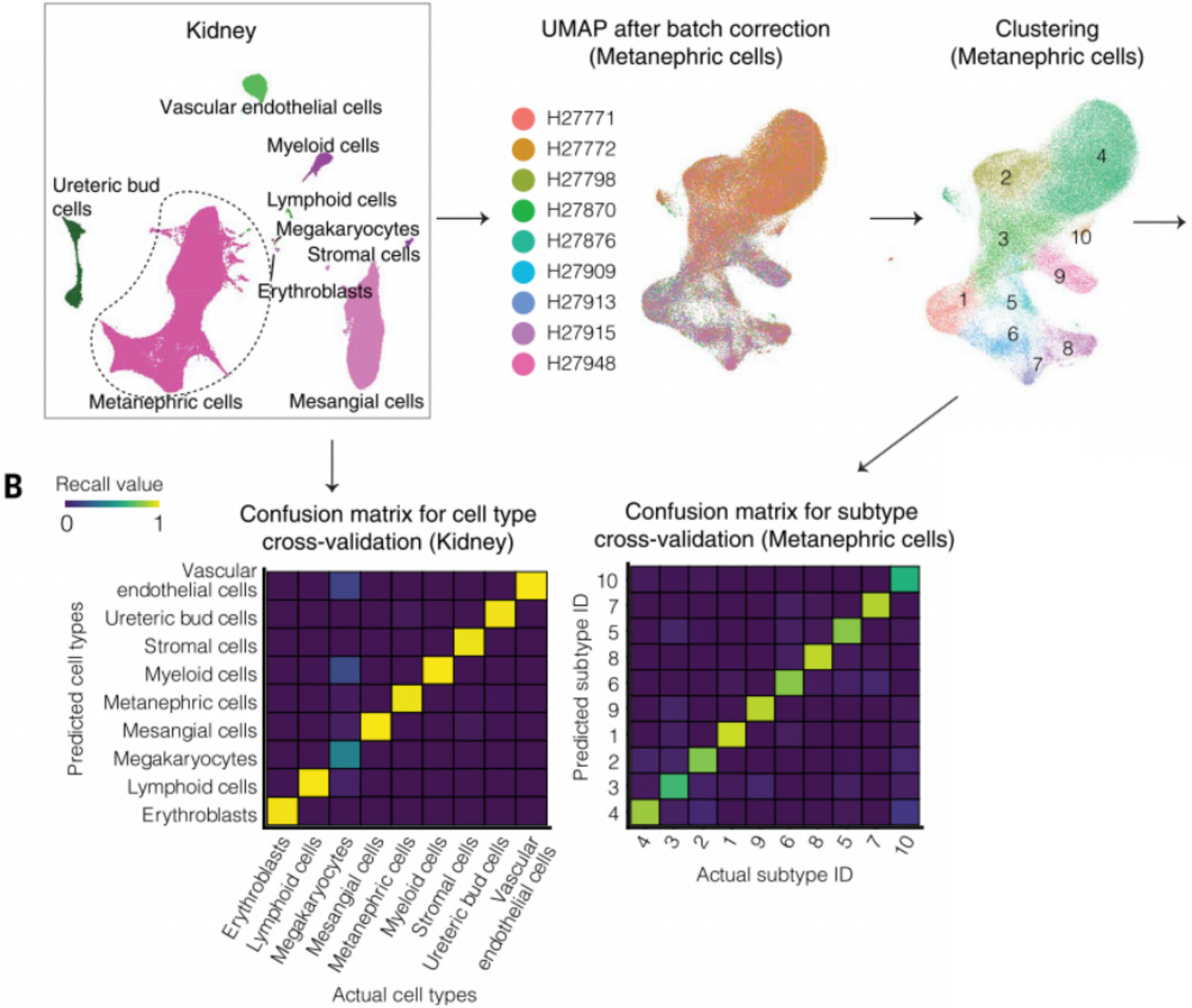


Intradataset cross validation (Cao et al.)

- **Step 1:** Cluster the cells.
- **Step 2:** Treat the cluster labels as the true responses. Train a classifier to predict these labels.

  Use cross validation to avoid double dipping between fitting and evaluating the classifier.
- **Step 3:** Compare original clustering labels to labels predicted by classifier.

Can the cluster labels be reliably reproduced?

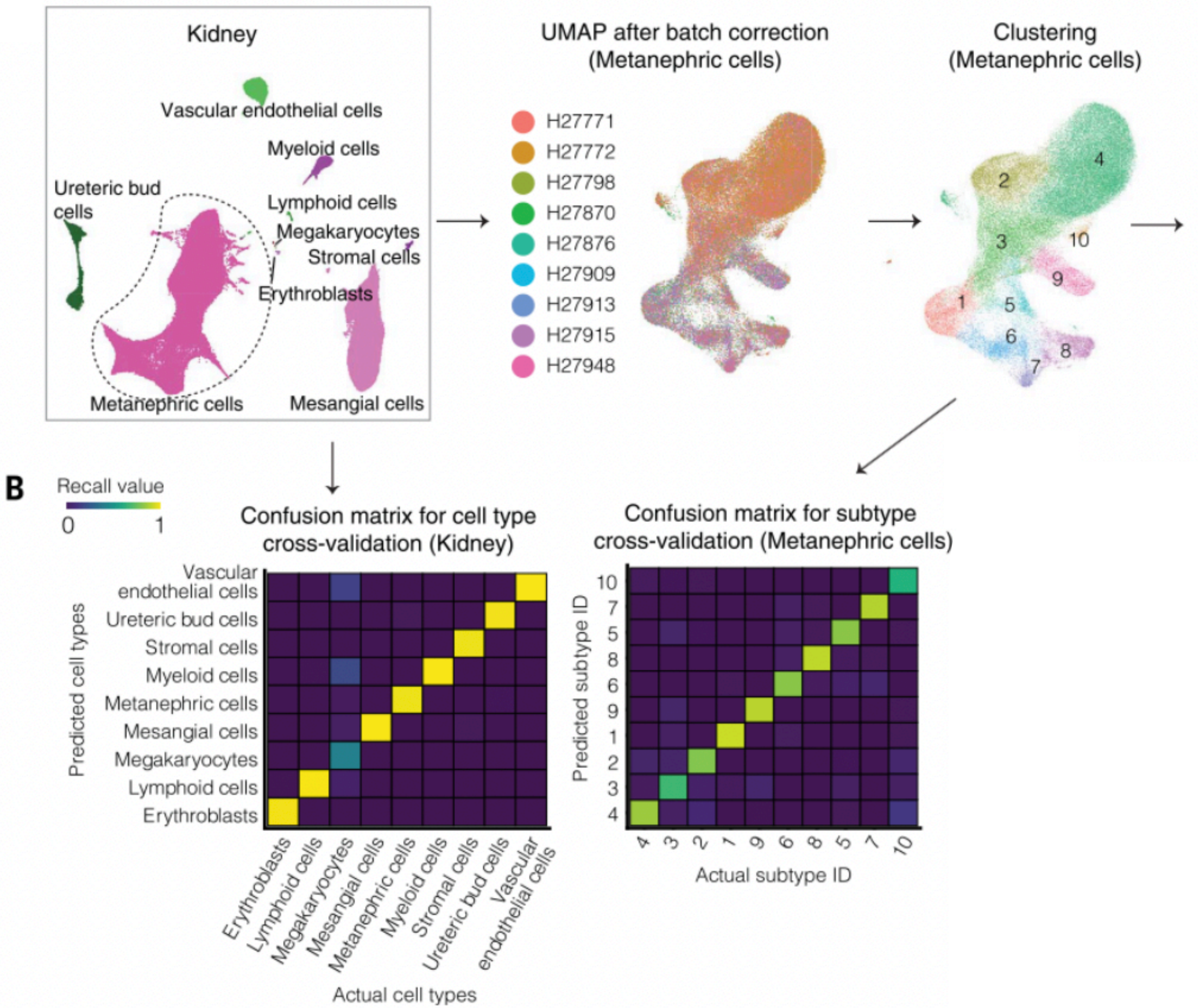


Intradataset cross validation (Cao et al.)

- **Step 1:** Cluster the cells.
- **Step 2:** Treat the cluster labels as the true responses. Train a classifier to predict these labels.

  Use cross validation to avoid double dipping between fitting and evaluating the classifier.
- **Step 3:** Compare original clustering labels to labels predicted by classifier.

Can the cluster labels be reliably reproduced?



Intradataset cross validation (Cao et al.)

- **Step 1:** Cluster the cells.

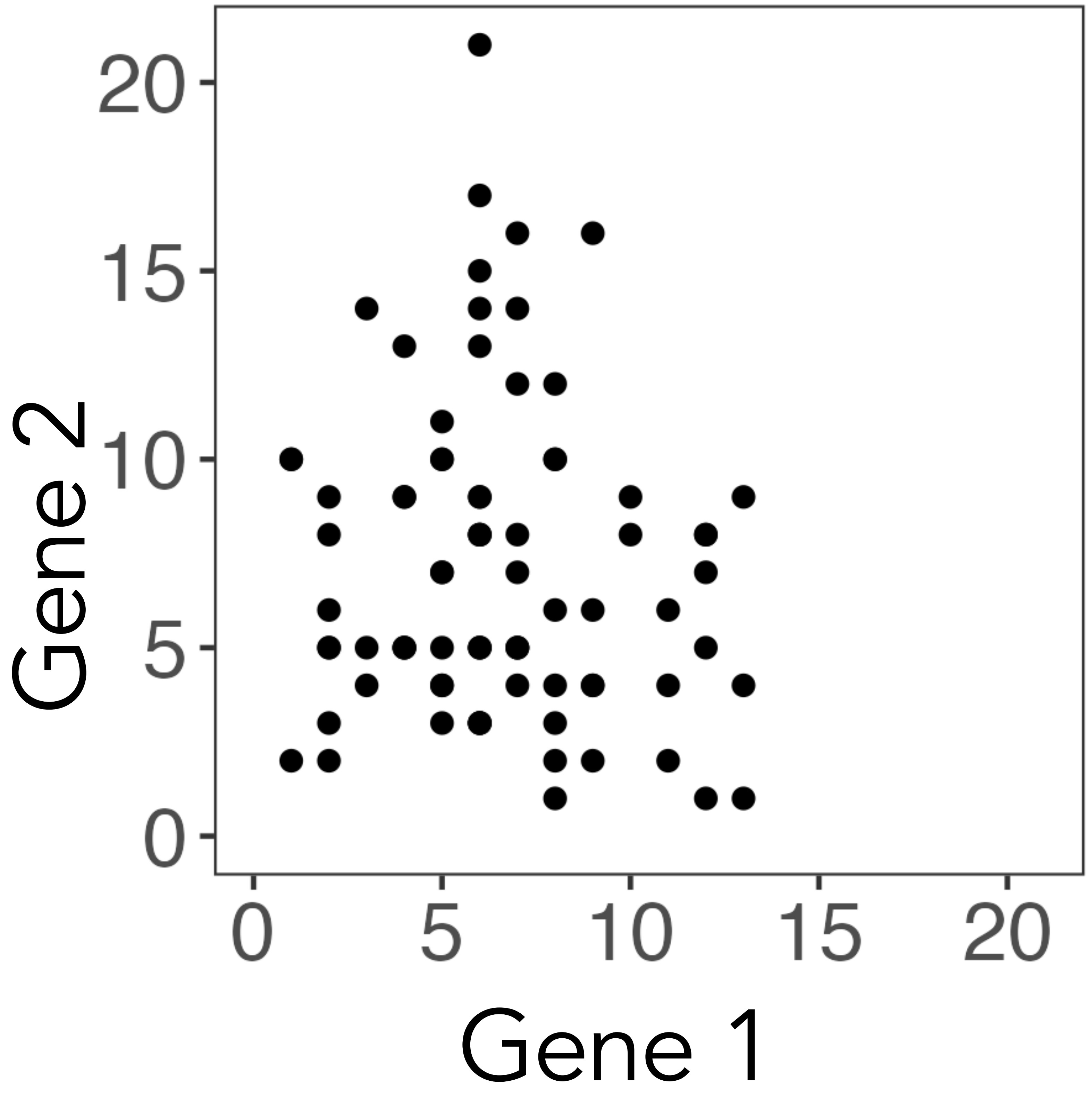
  But we already dipped in the data here!

- **Step 2:** Treat the cluster labels as the true responses. Train a classifier to predict these labels.

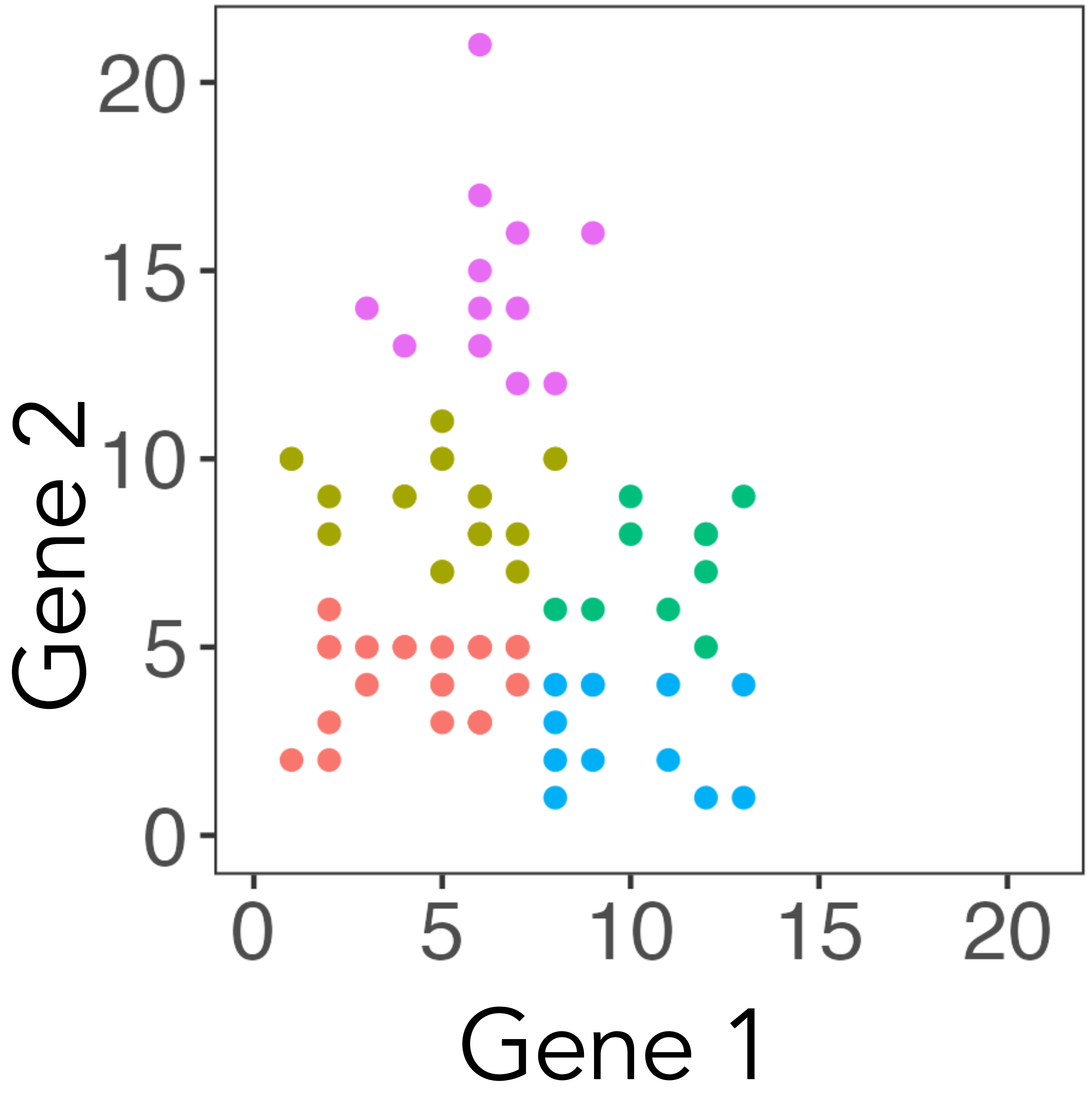
  Use cross validation to avoid double dipping between fitting and evaluating the classifier.

- **Step 3:** Compare original clustering labels to labels predicted by classifier.

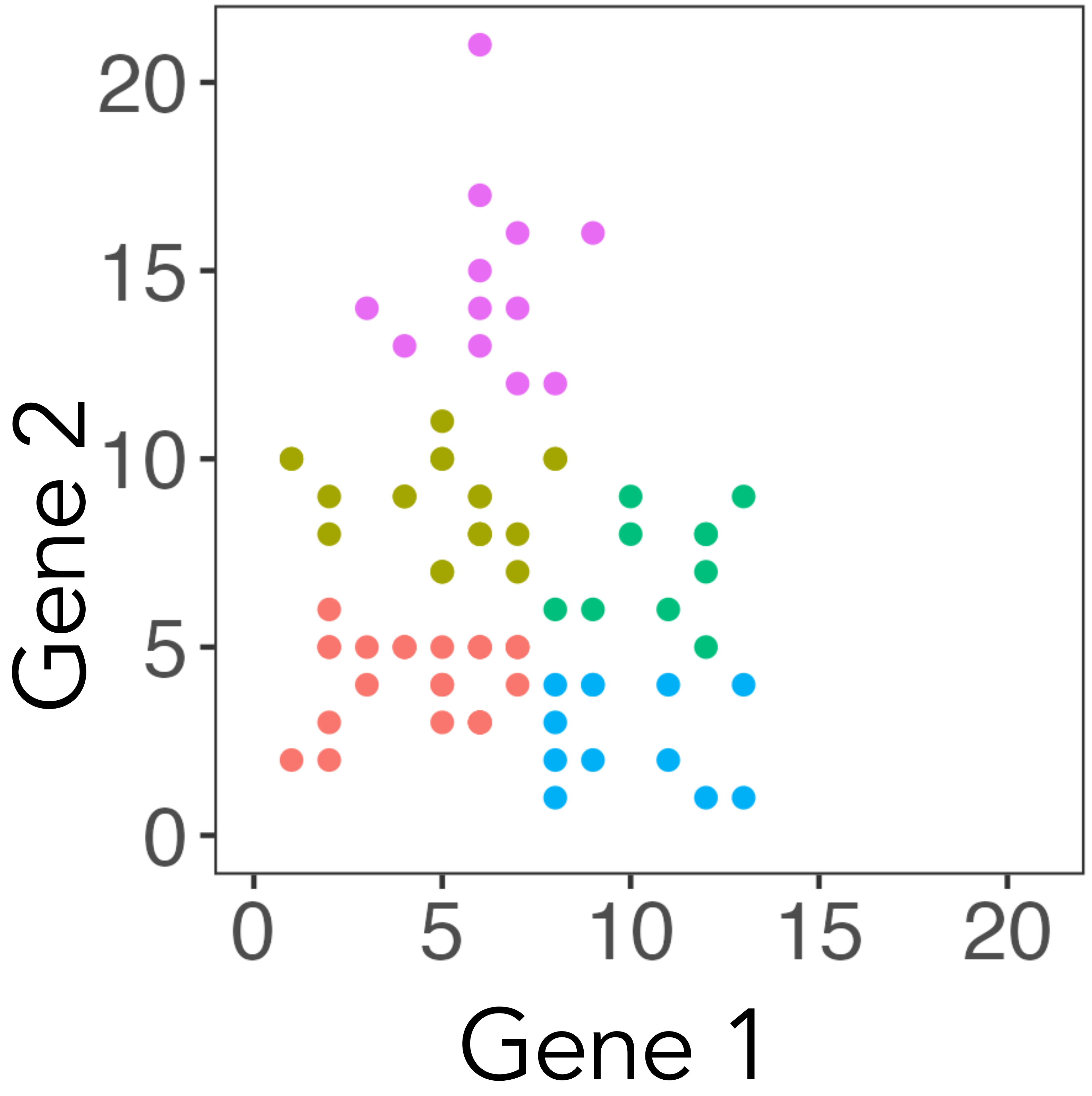
This cross validation procedure double dips

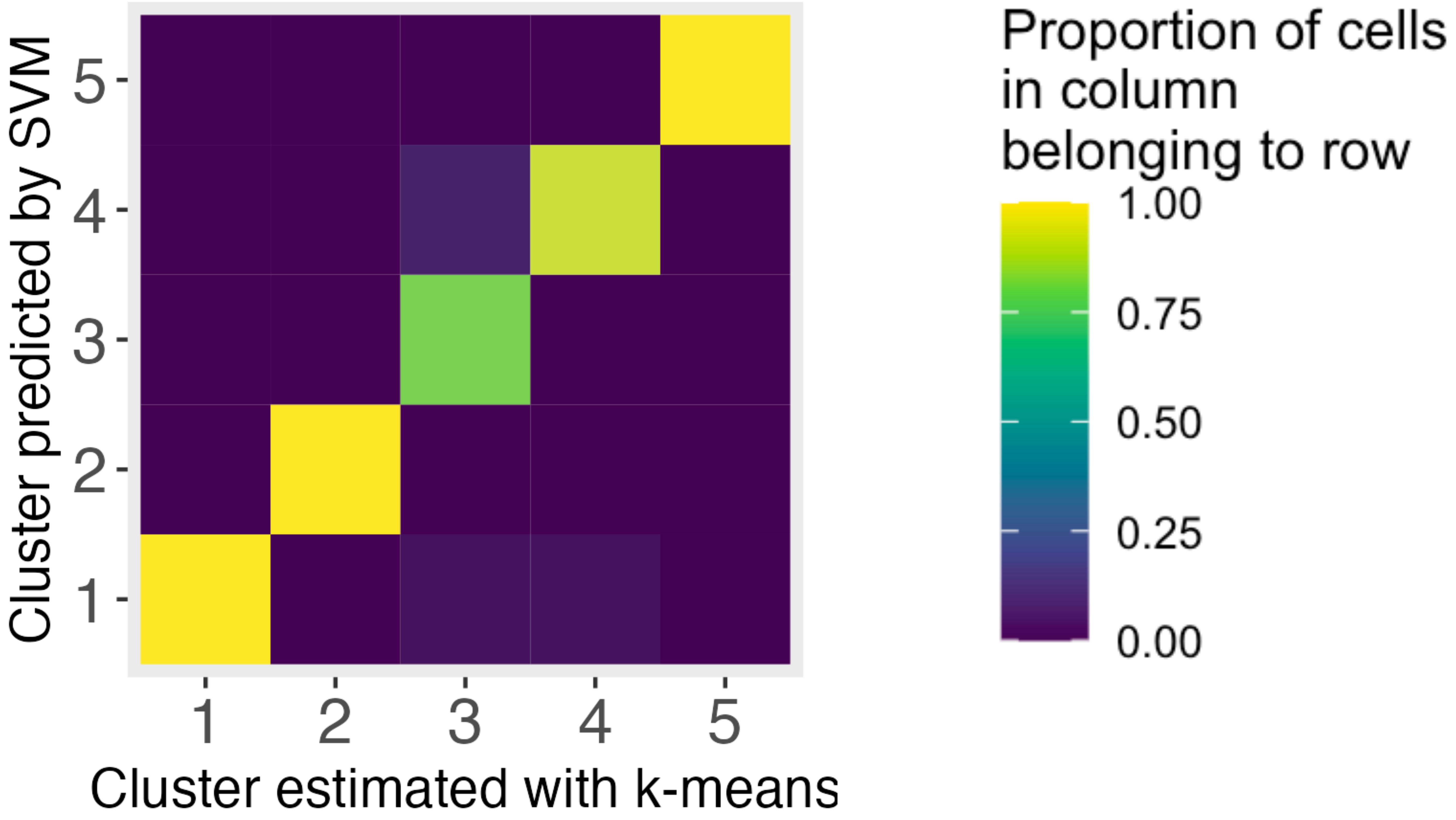


This cross validation procedure double dips



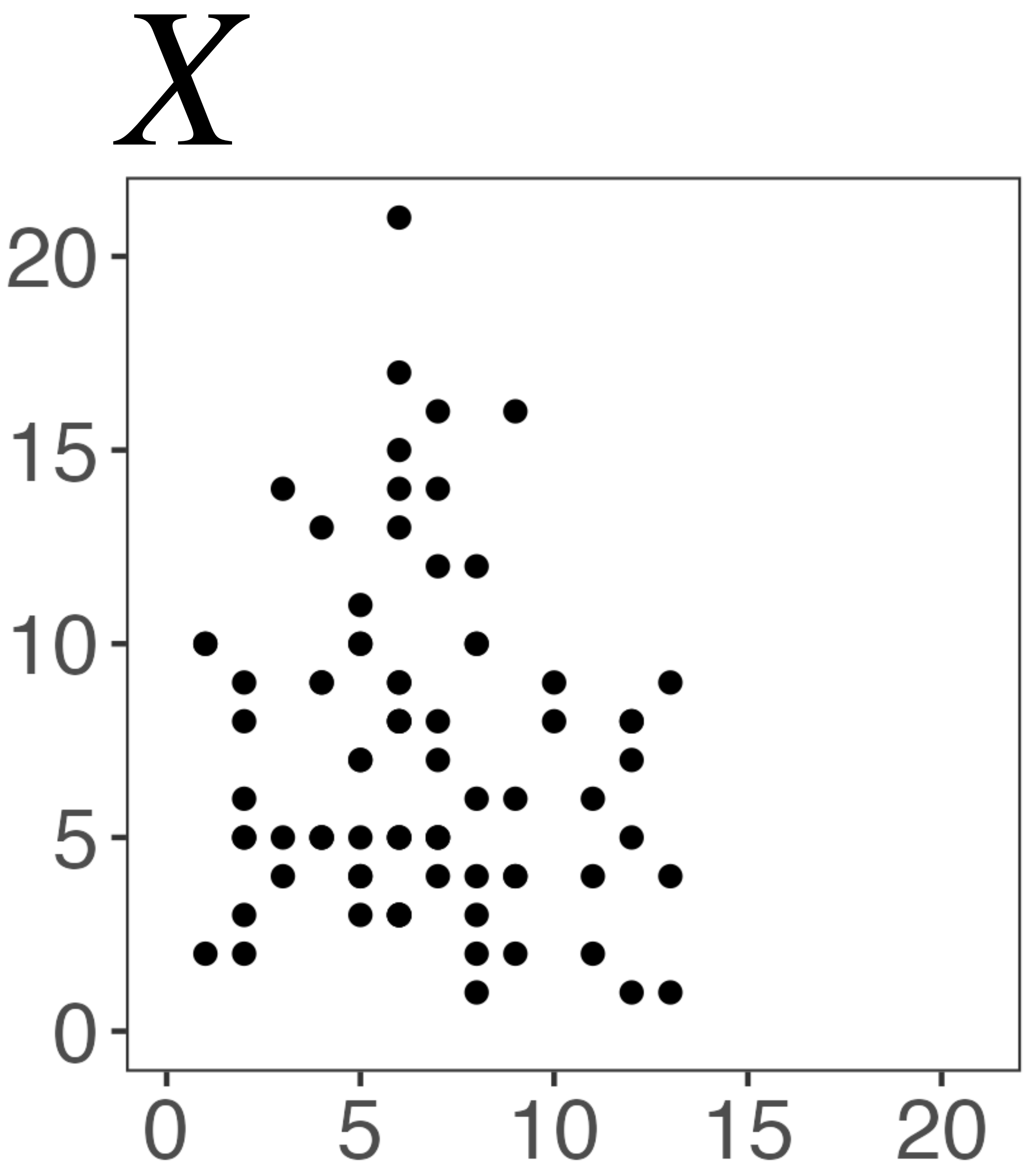
This cross validation procedure double dips



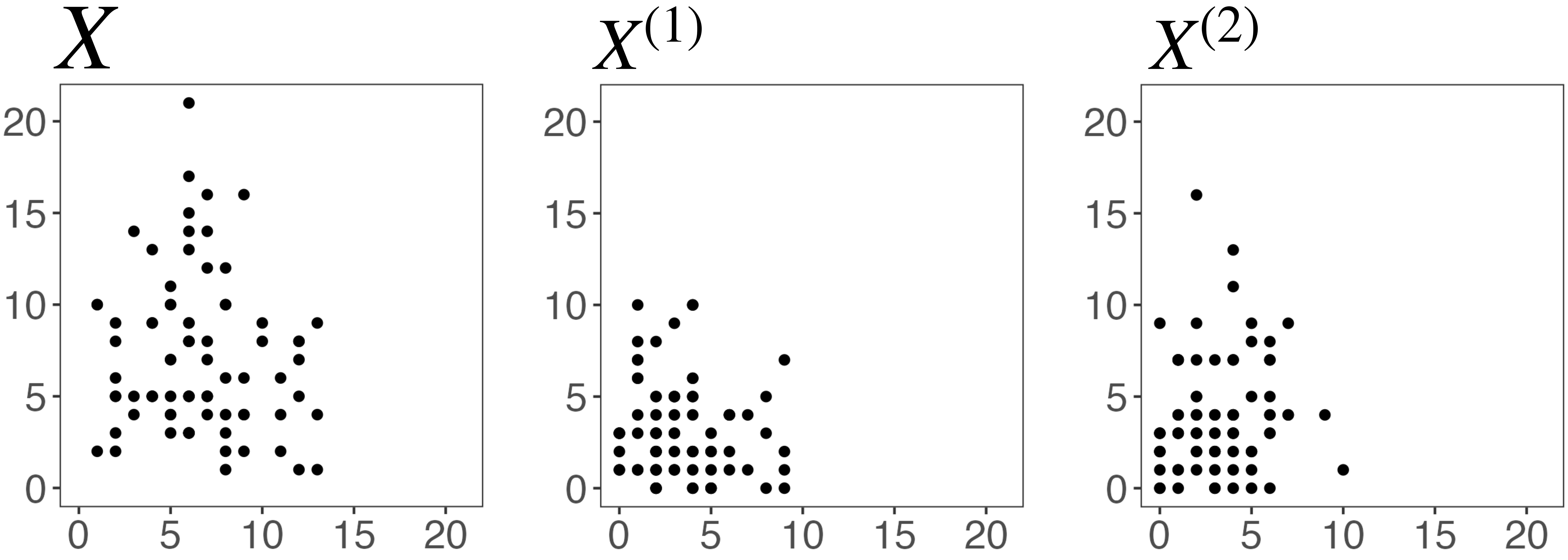


Classifier gets 96% accuracy to predict the five clusters, despite the fact that the five clusters are just random noise.

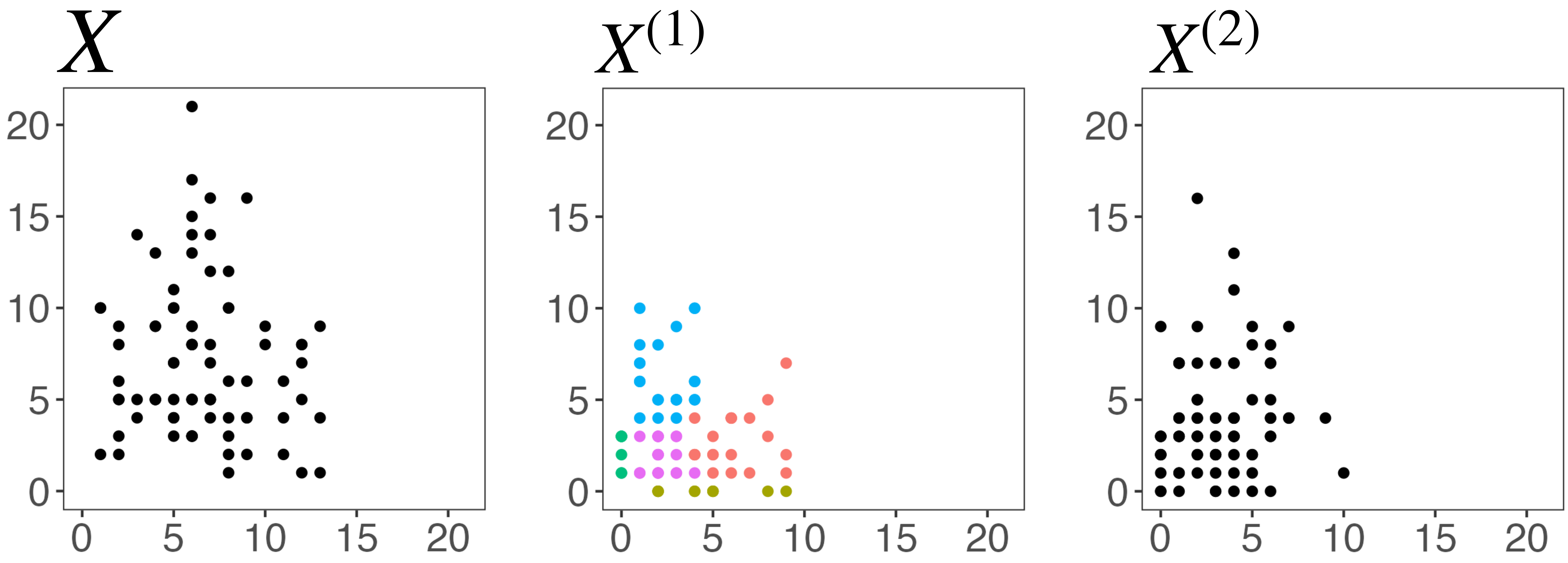
Data thinning provides a simple alternative



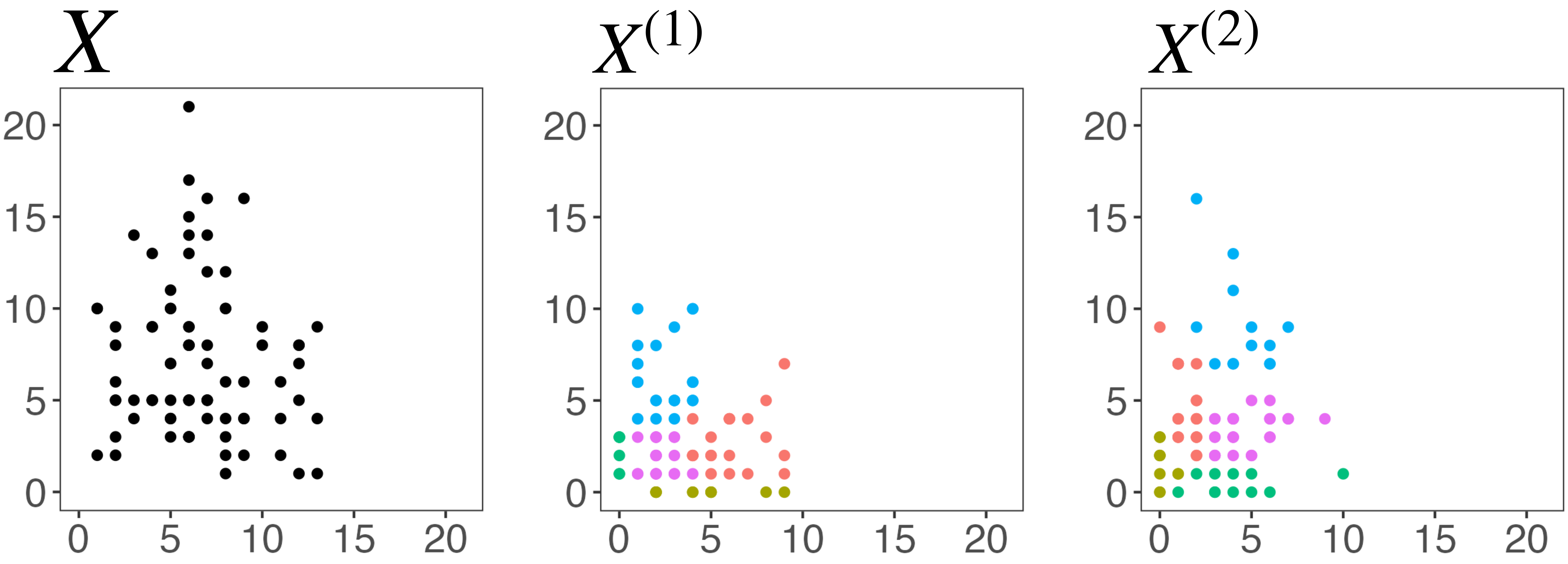
Data thinning provides a simple alternative



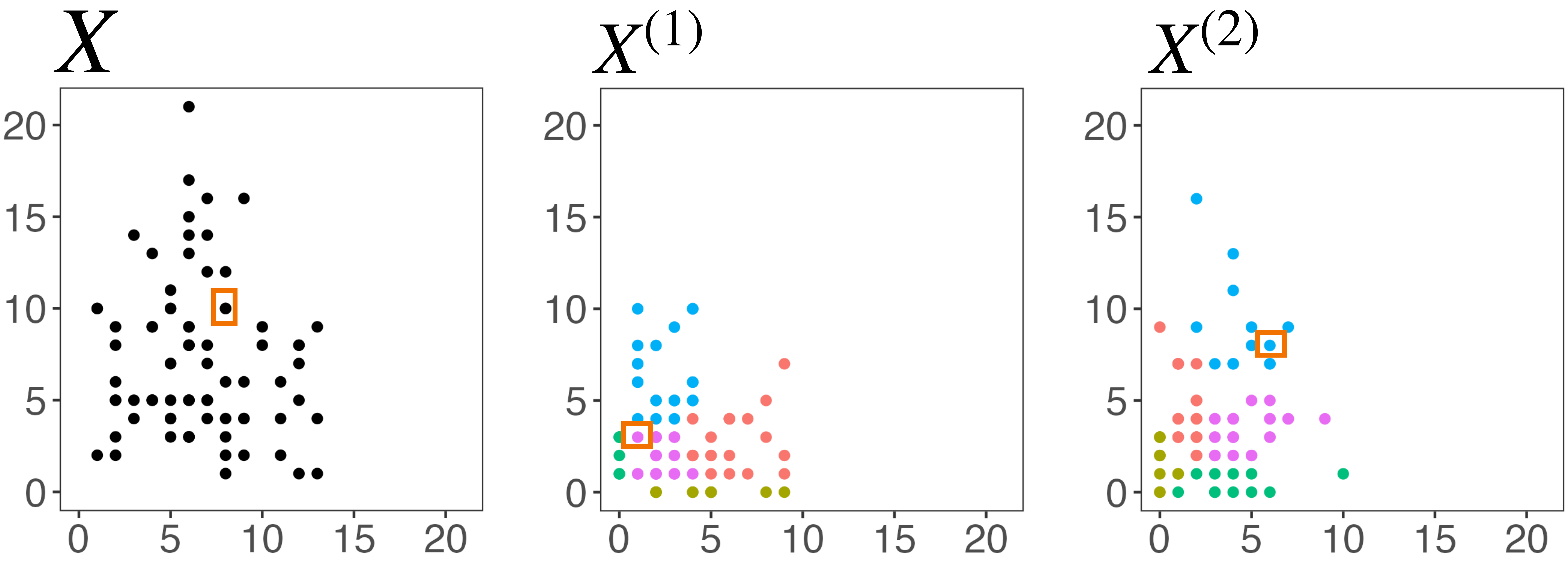
Data thinning provides a simple alternative



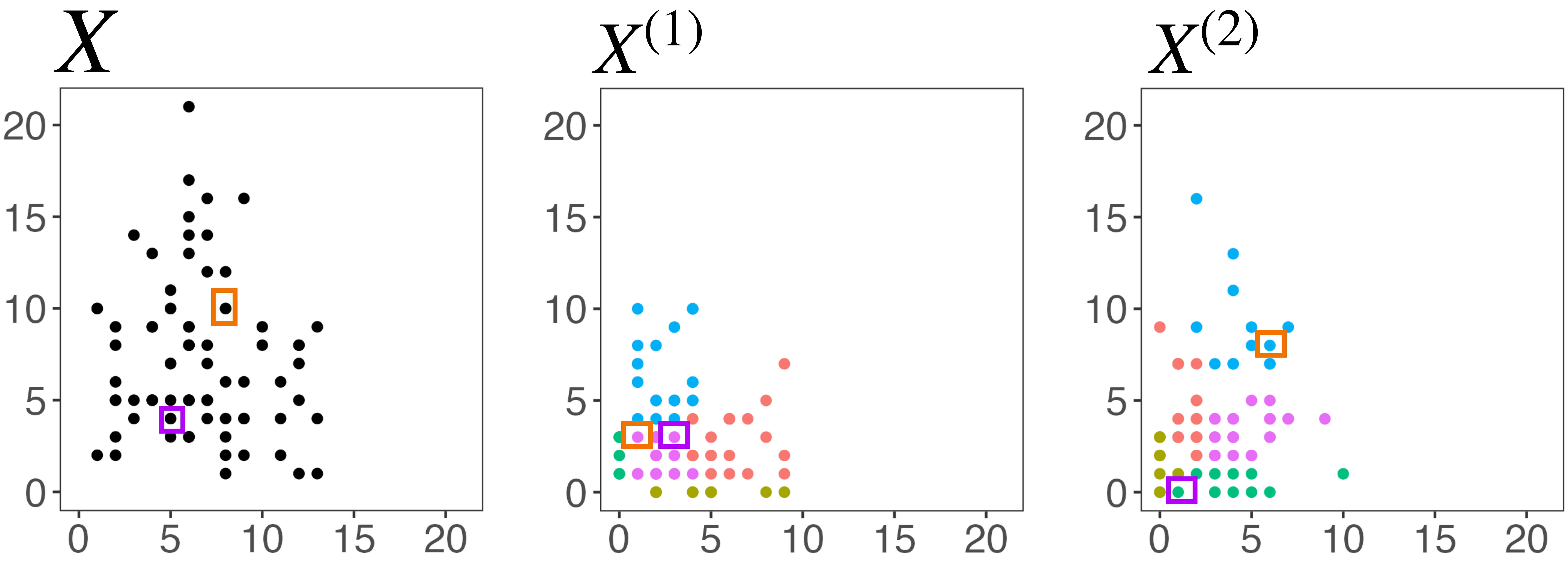
Data thinning provides a simple alternative



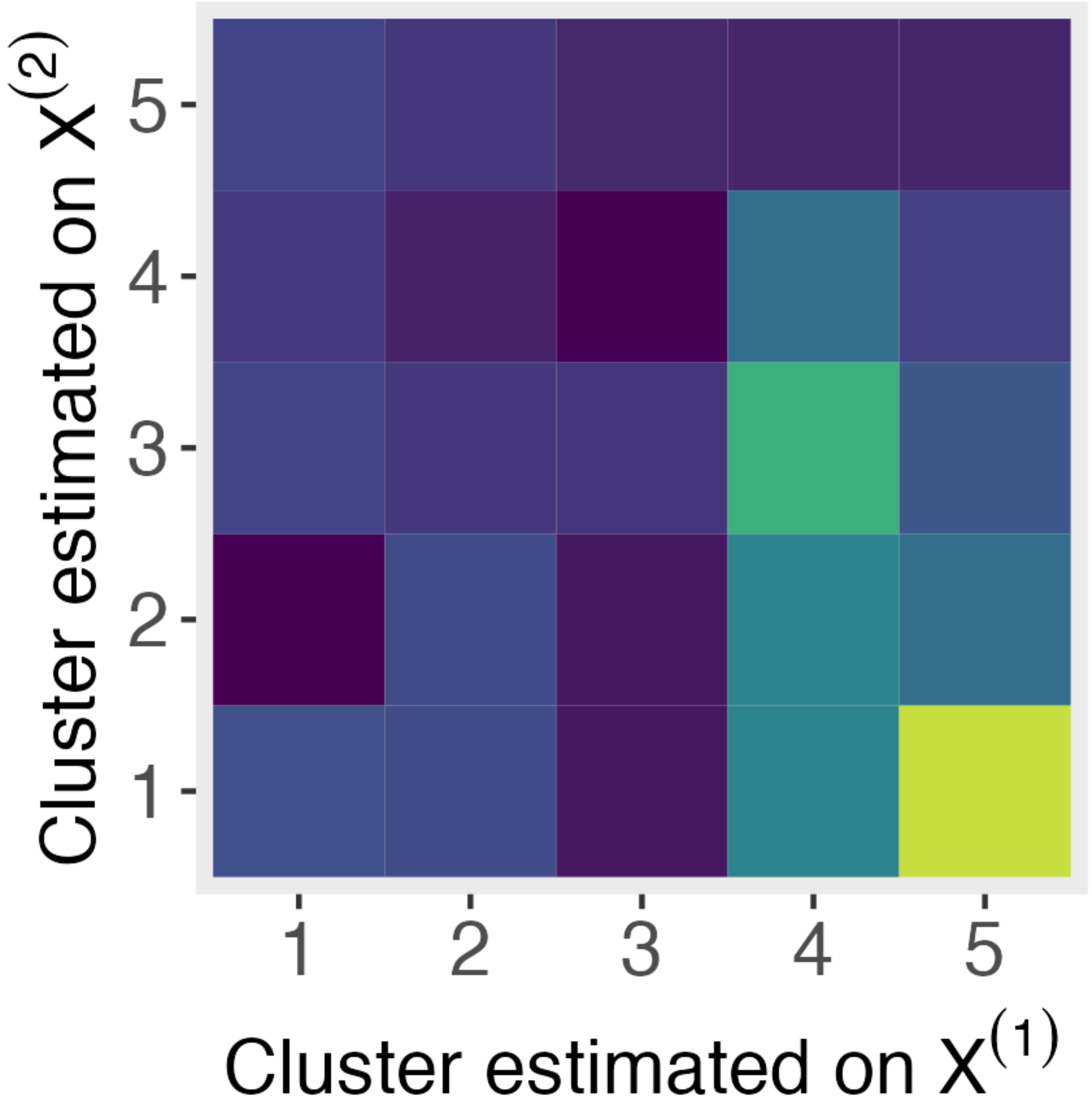
Data thinning provides a simple alternative

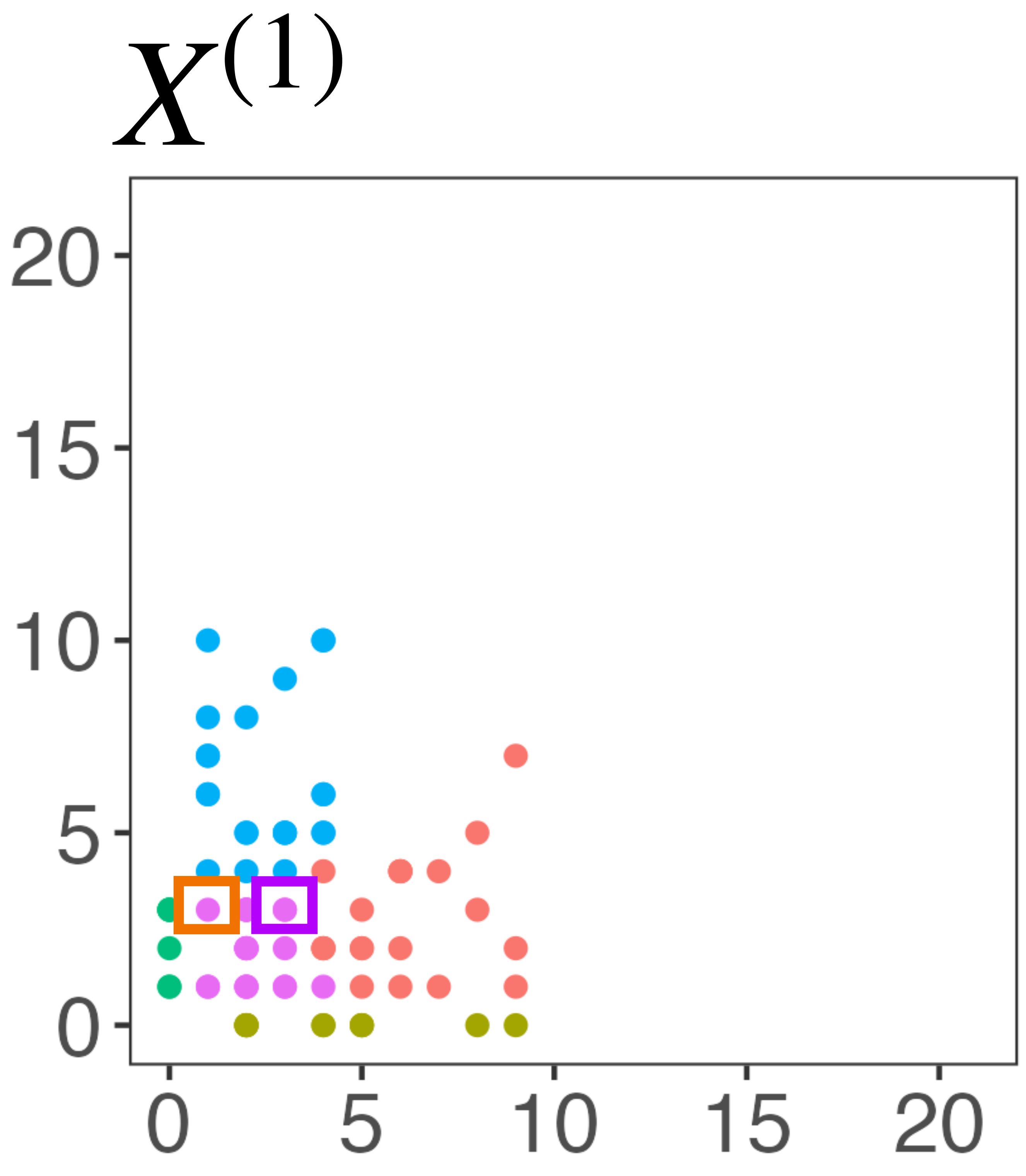


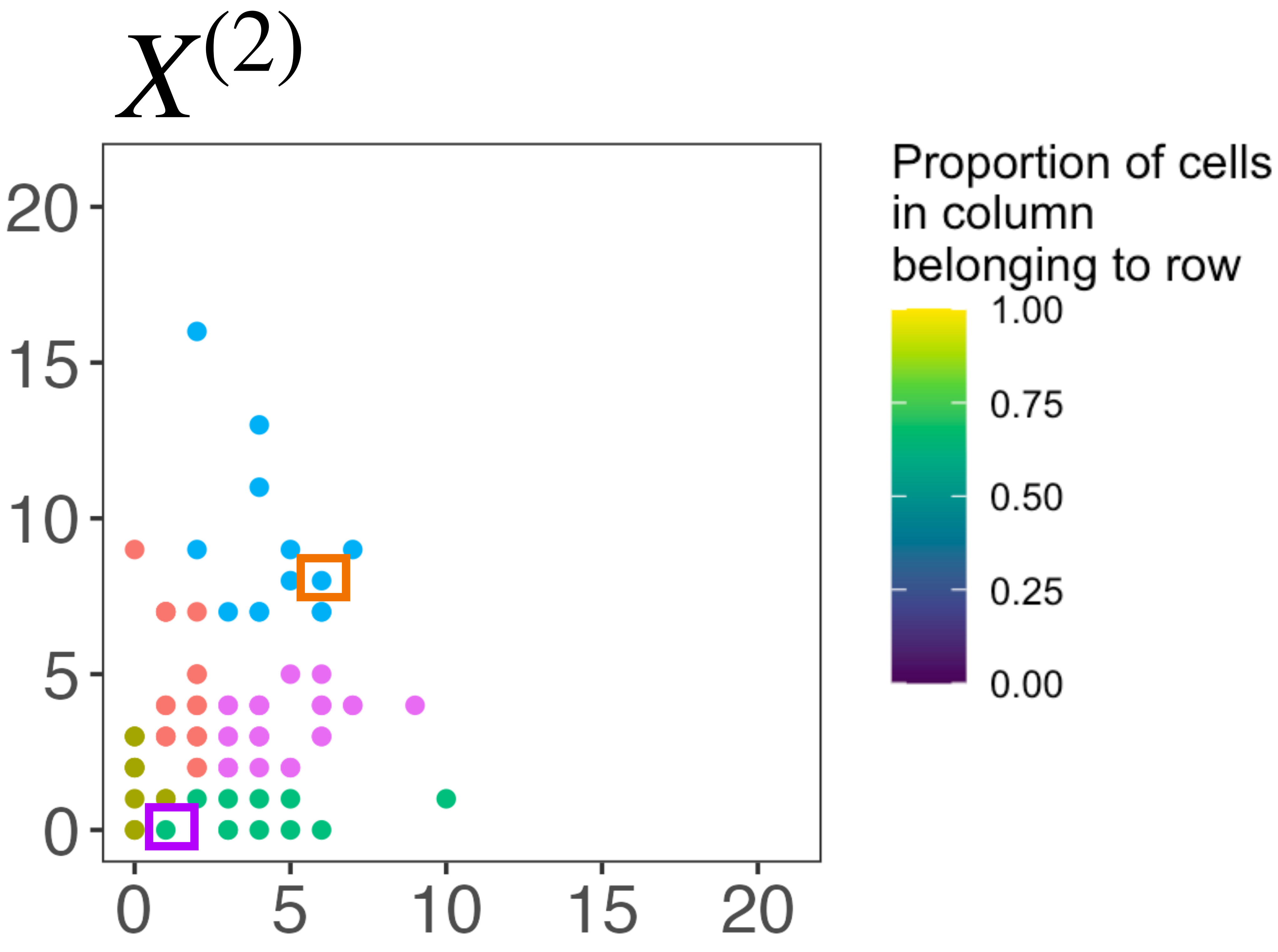
Data thinning provides a simple alternative



Data thinning provides a simple alternative

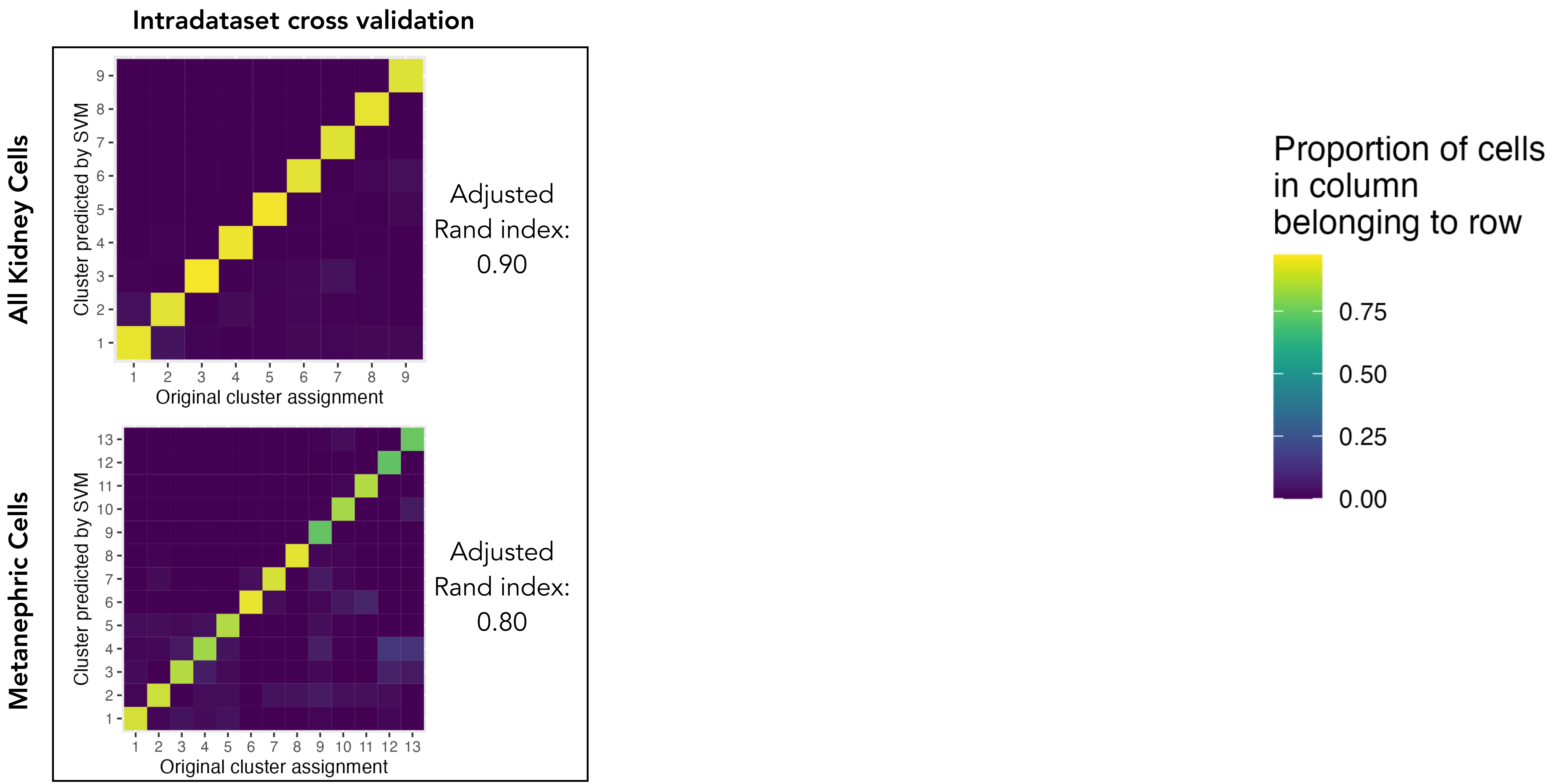






Adjusted Rand Index $= 0.01$

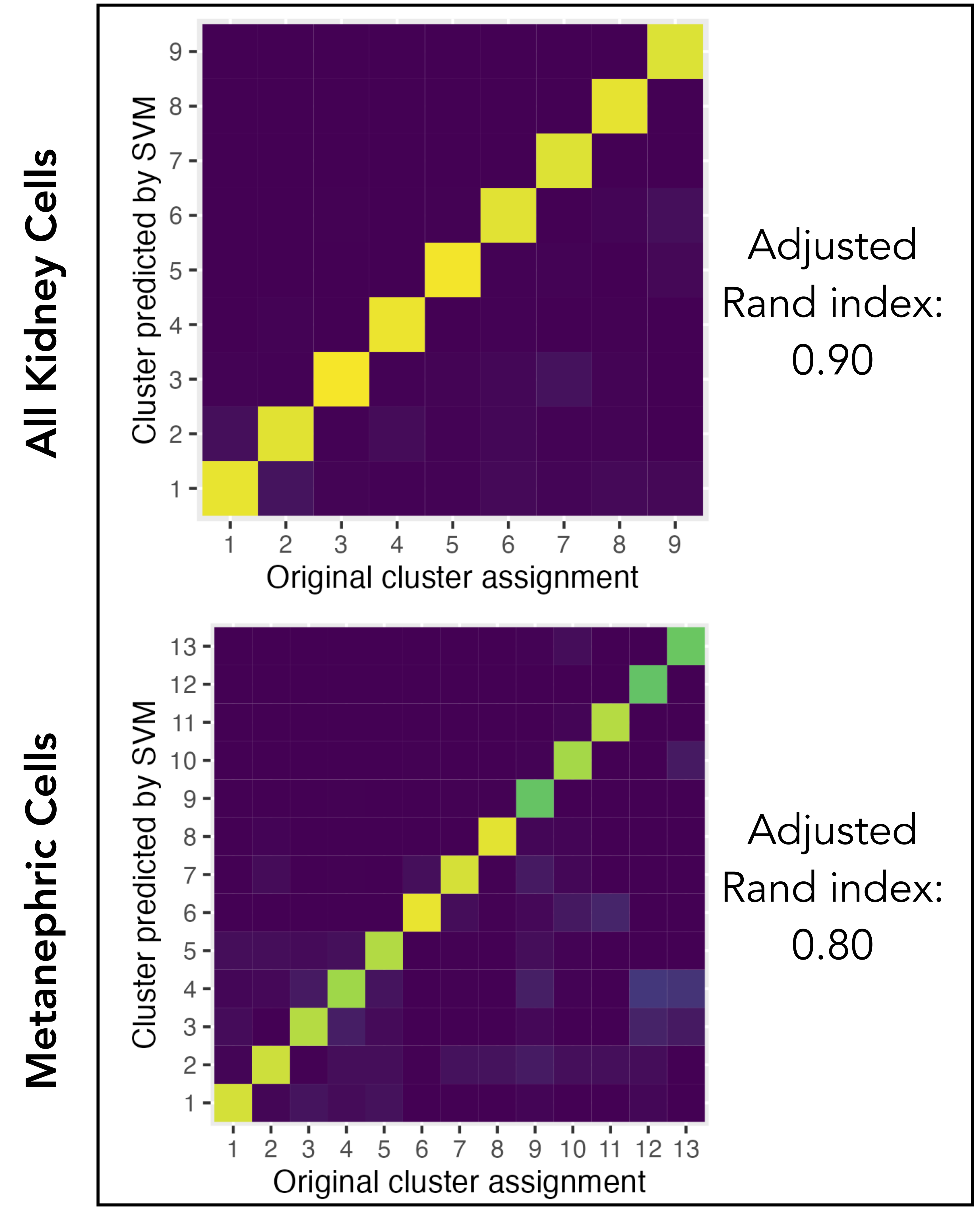
Re-analysis of Kidney cell data from fetal cell atlas

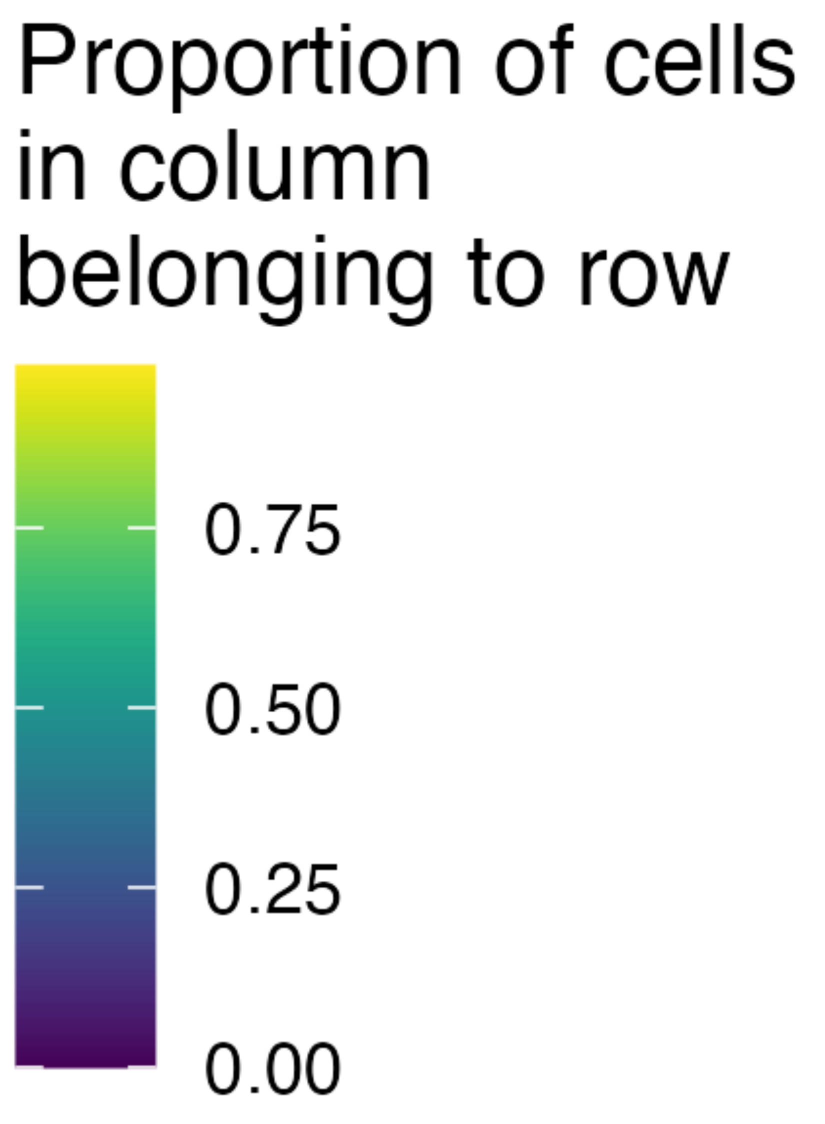


Re-analysis of Kidney cell data from fetal cell atlas

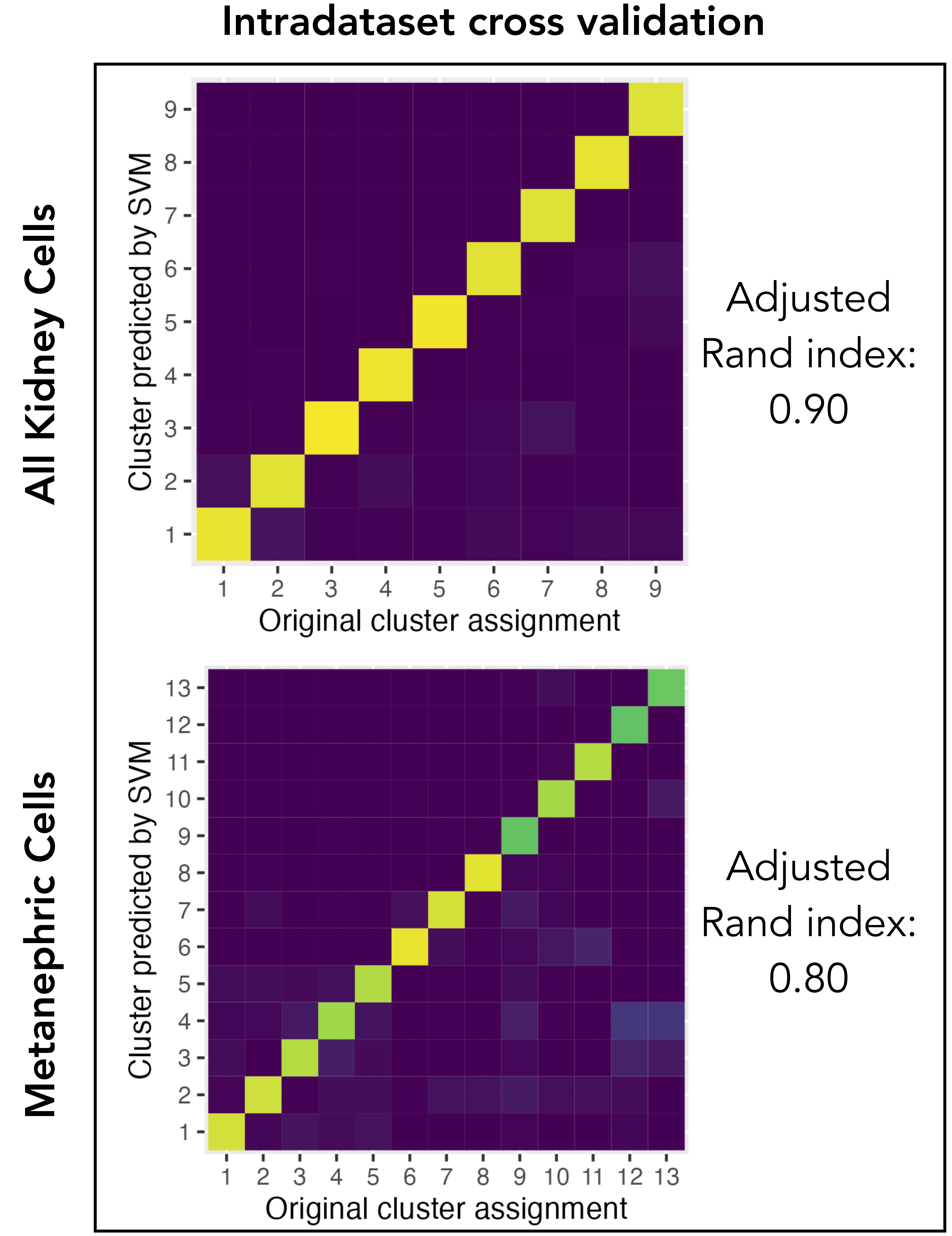
Intradataset cross validation

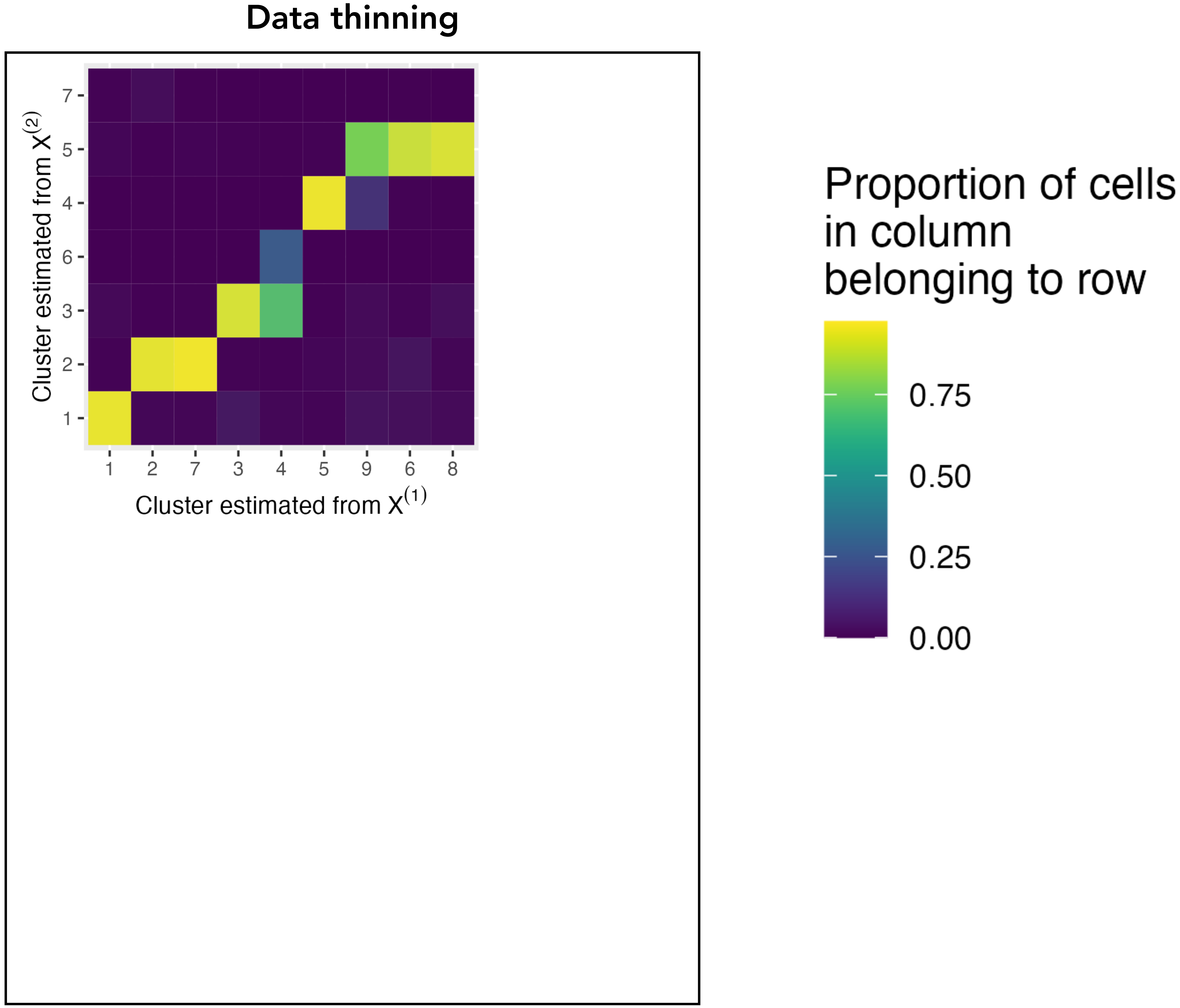
Data thinning



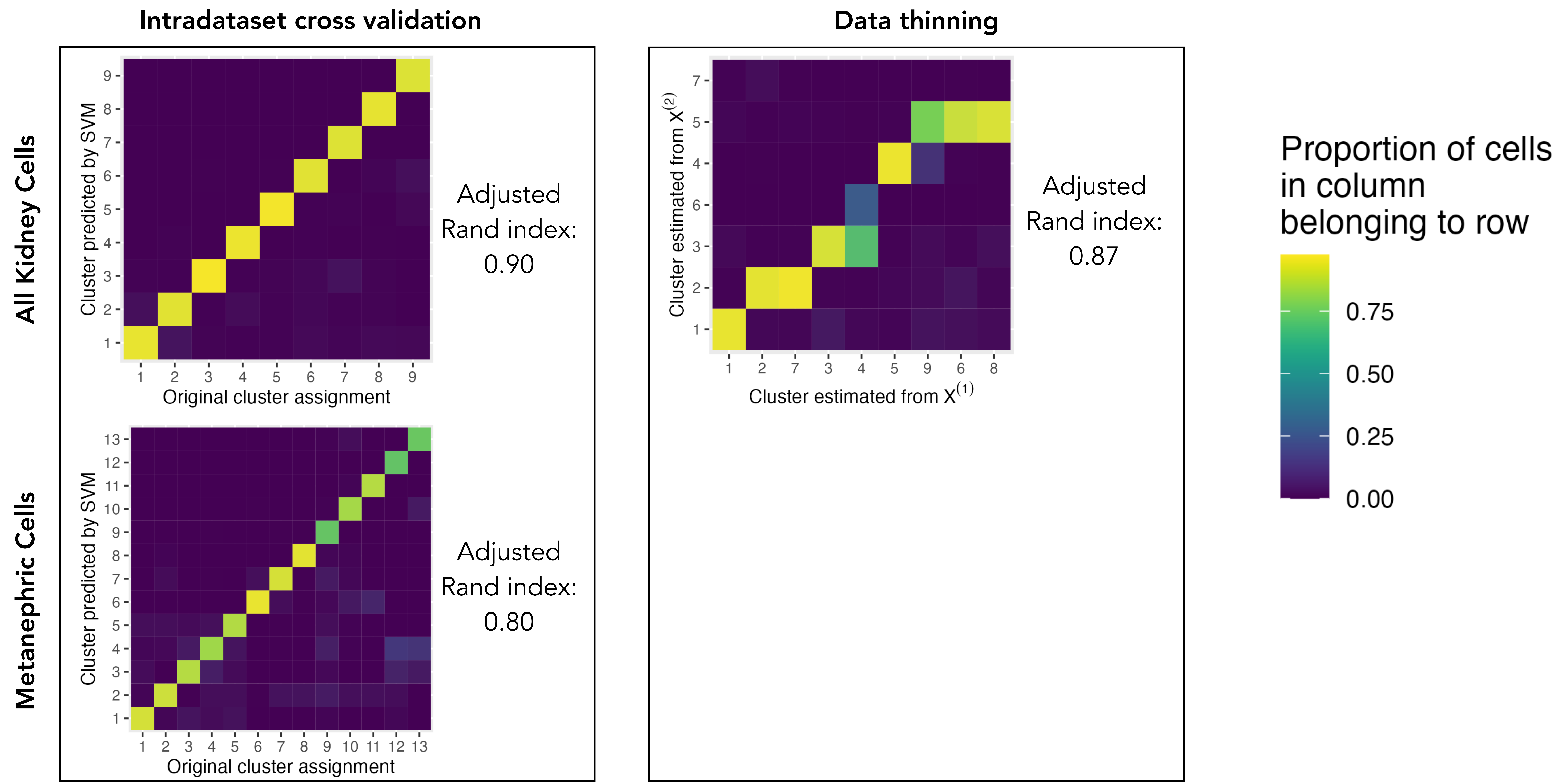


Re-analysis of Kidney cell data from fetal cell atlas

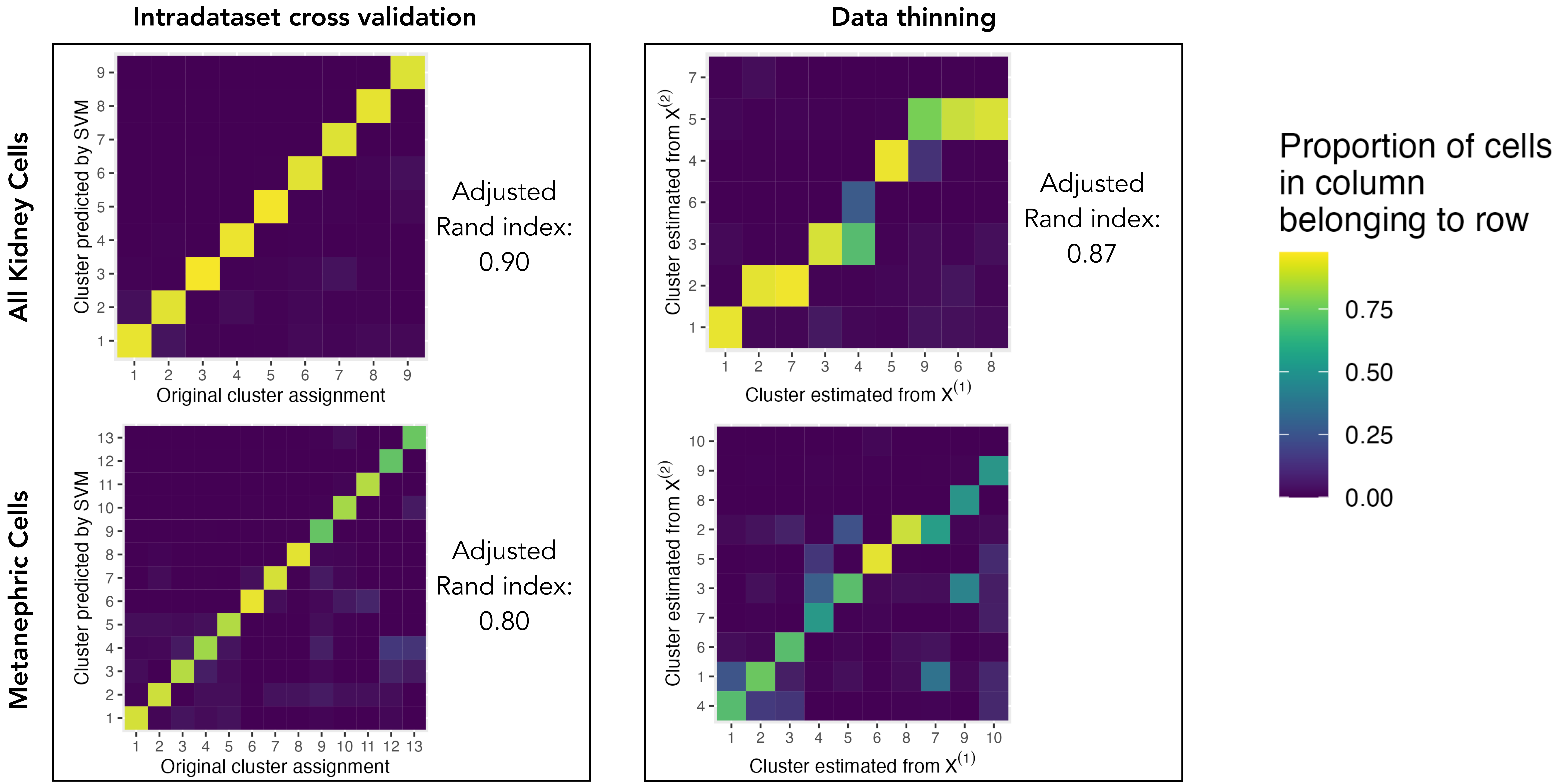




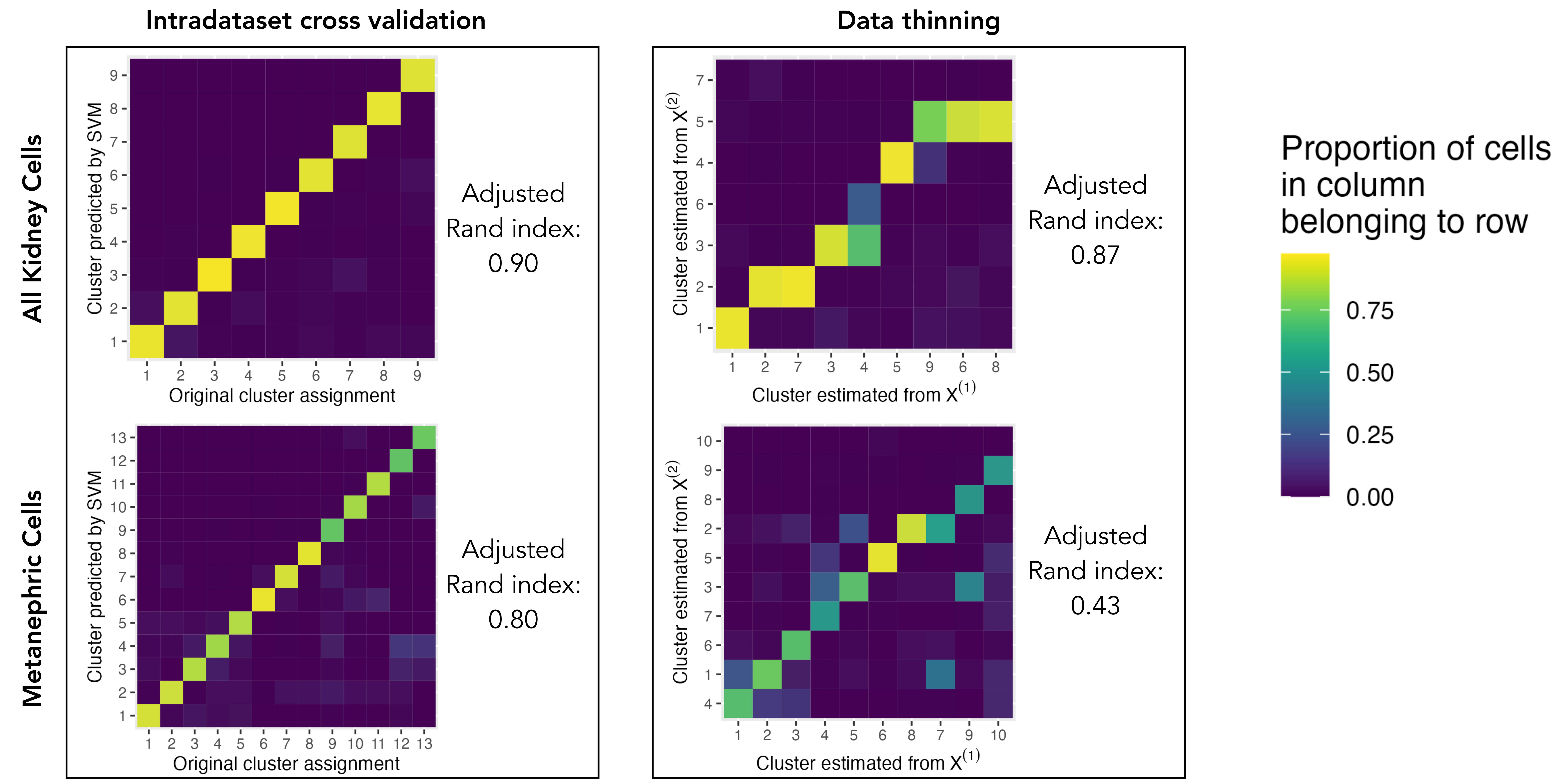
Re-analysis of Kidney cell data from fetal cell atlas



Re-analysis of Kidney cell data from fetal cell atlas



Re-analysis of Kidney cell data from fetal cell atlas



Outline

1. Motivation: settings where sample splitting doesn't work
2. Poisson thinning
3. Data thinning
4. Application to human fetal cell atlas data
5. **Application to cardiomyocyte differentiation data**
6. Ongoing work

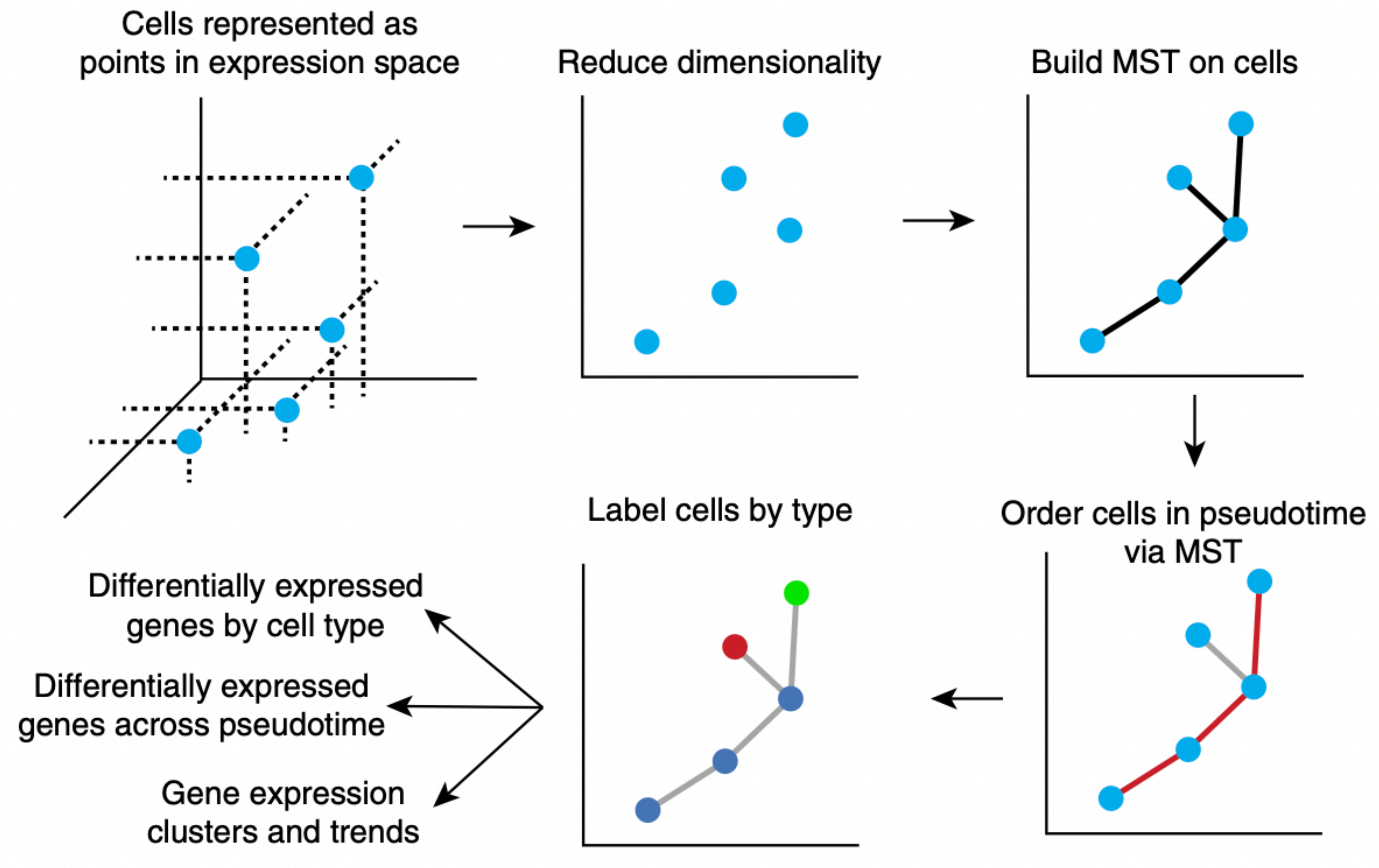
Which genes are differentially expressed along a developmental trajectory?

Published: 23 March 2014

The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells

Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J Lennon, Kenneth J Livak, Tarjei S Mikkelsen & John L Rinn

Nature Biotechnology **32**, 381–386 (2014) | Cite this article



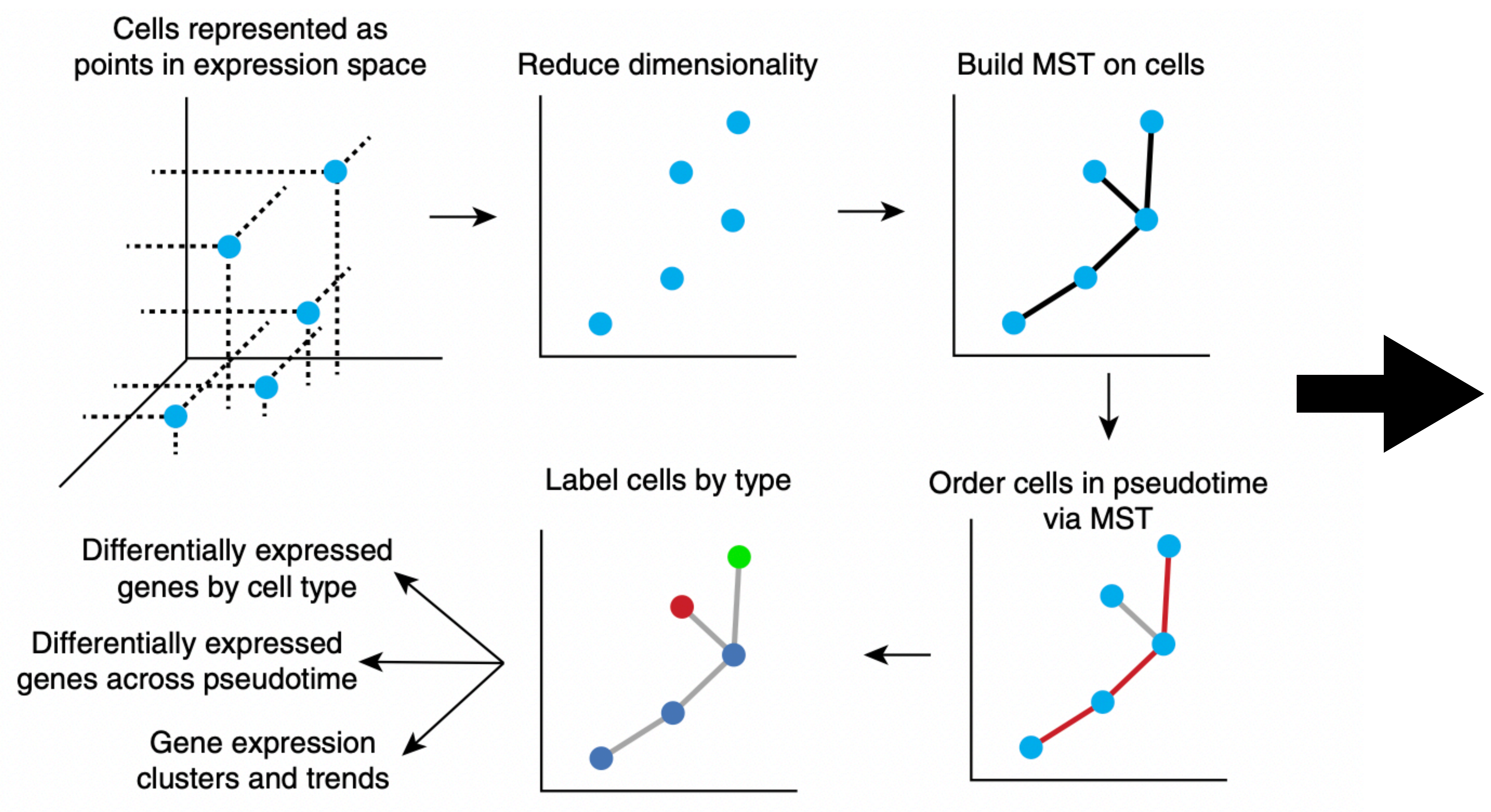
Which genes are differentially expressed along a developmental trajectory?

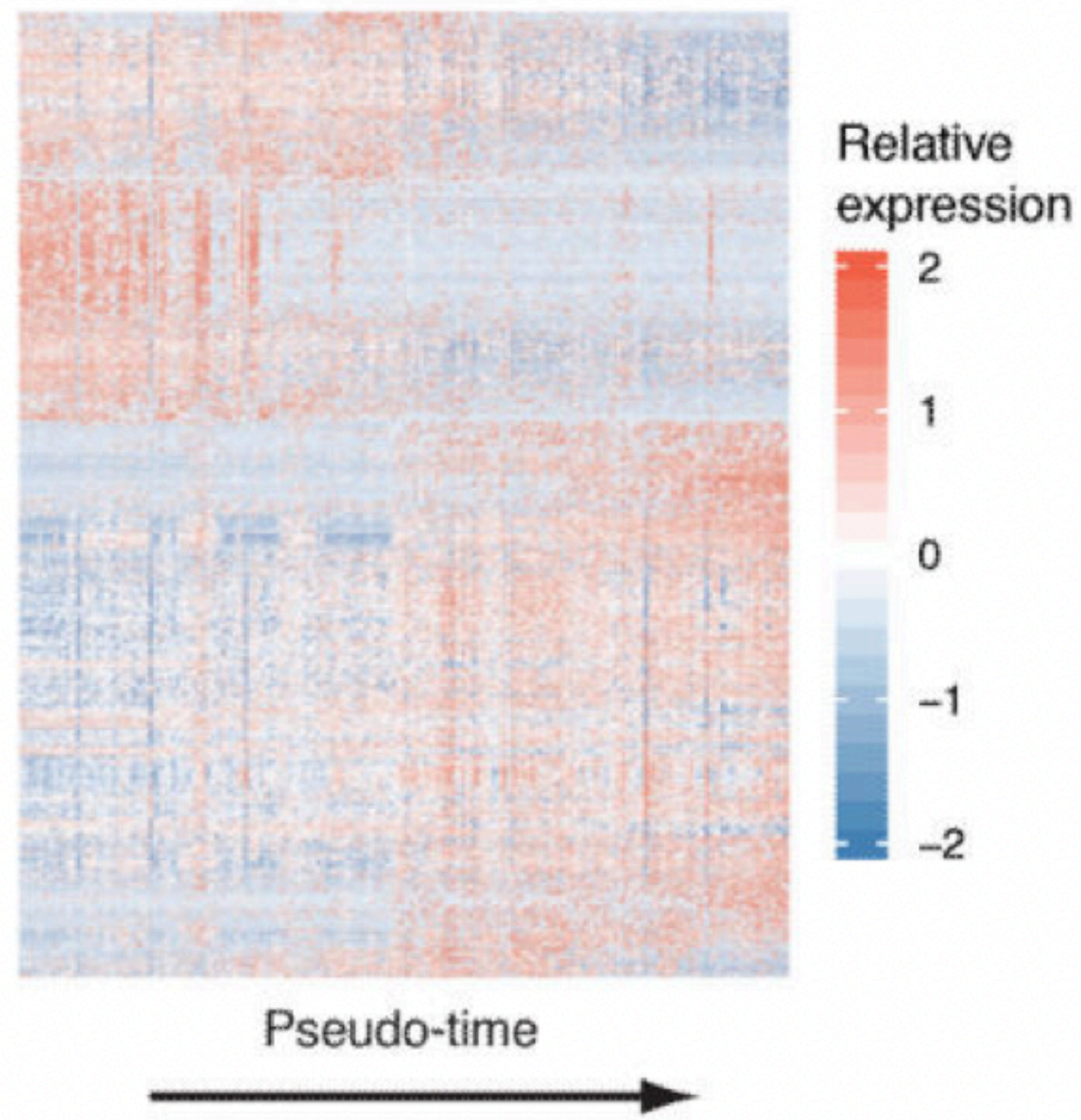
Published: 23 March 2014

The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells

Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J Lennon, Kenneth J Livak, Tarjei S Mikkelsen & John L Rinn

Nature Biotechnology **32**, 381–386 (2014) | Cite this article





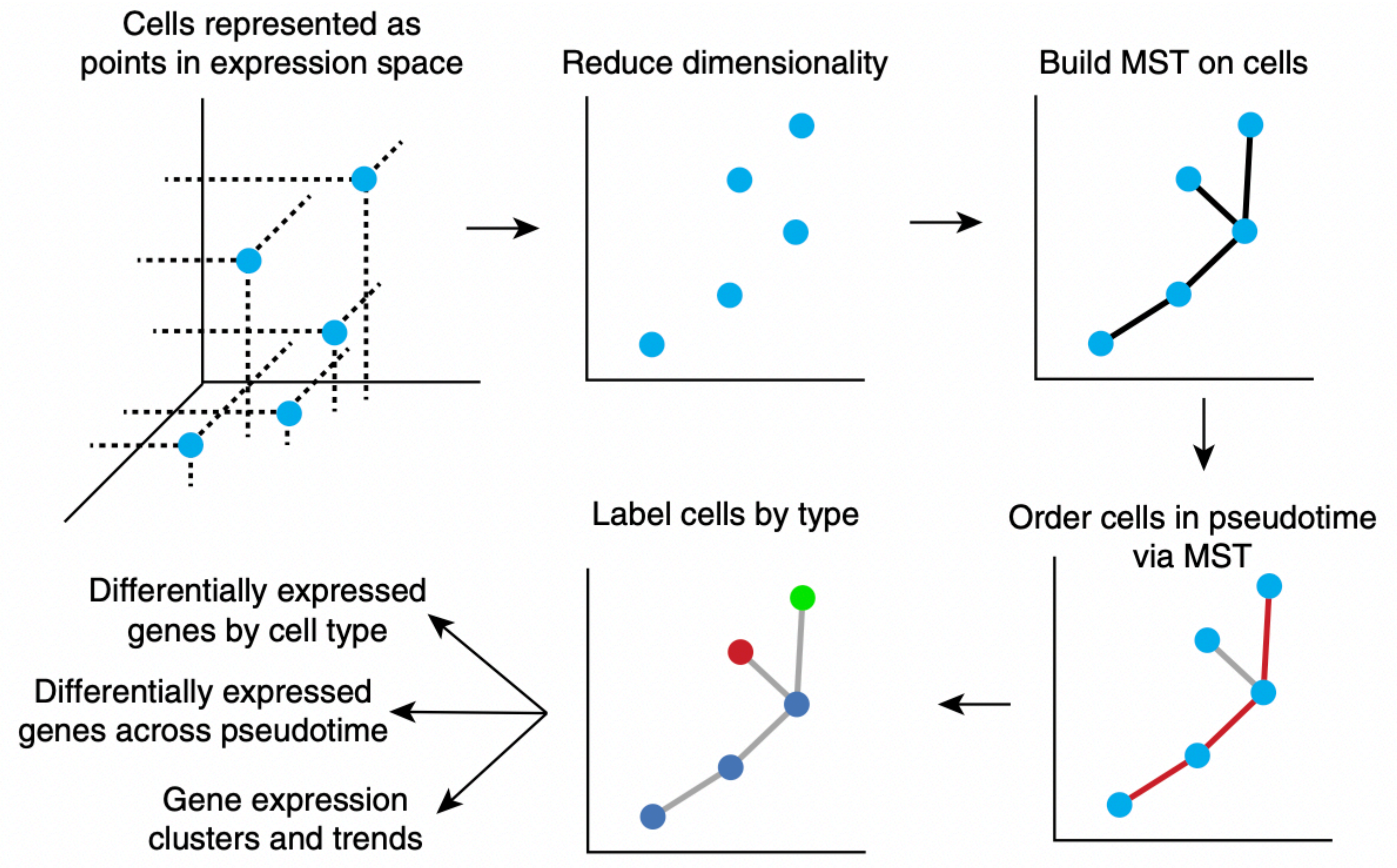
Which genes are differentially expressed along a developmental trajectory?

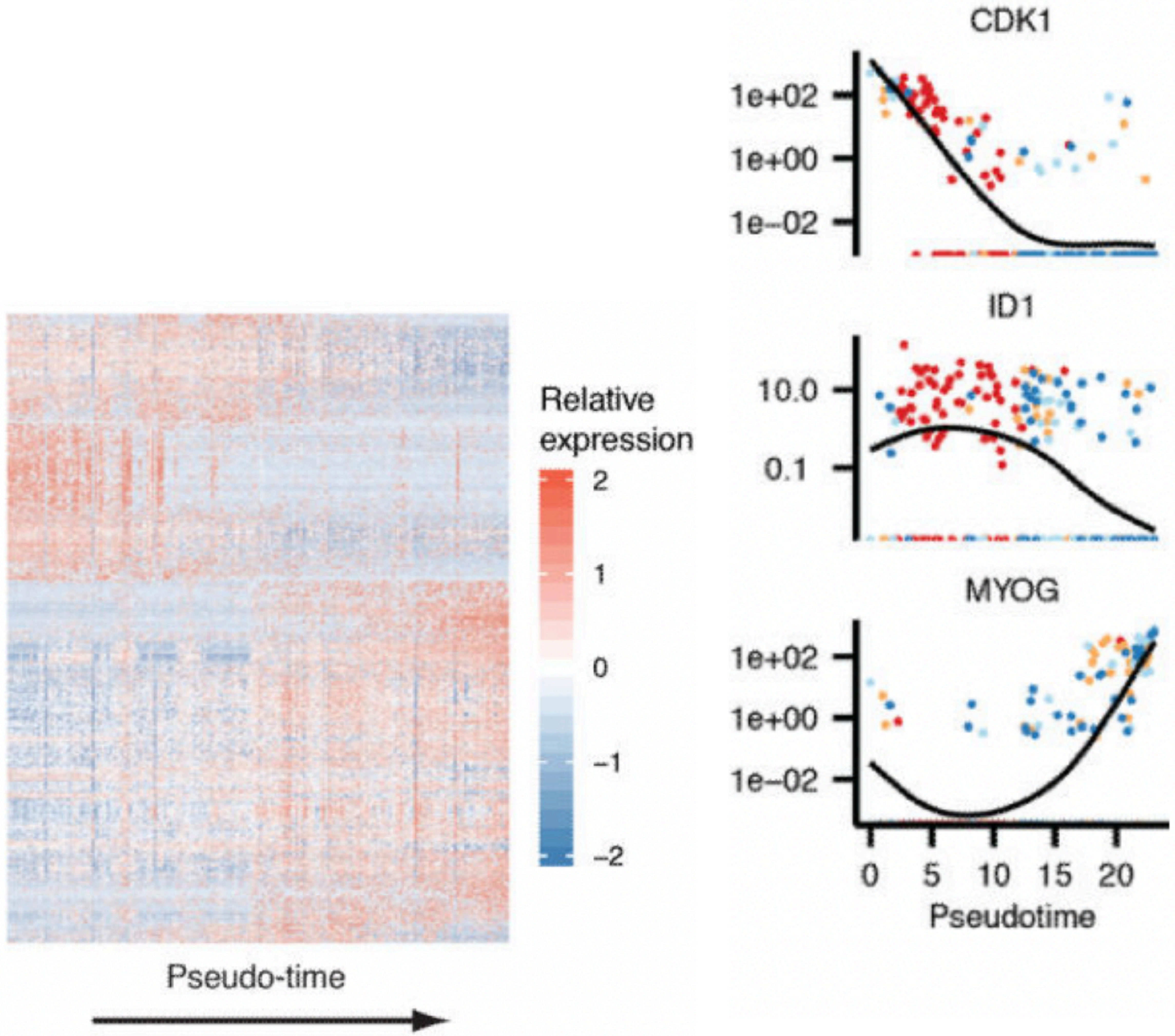
Published: 23 March 2014

The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells

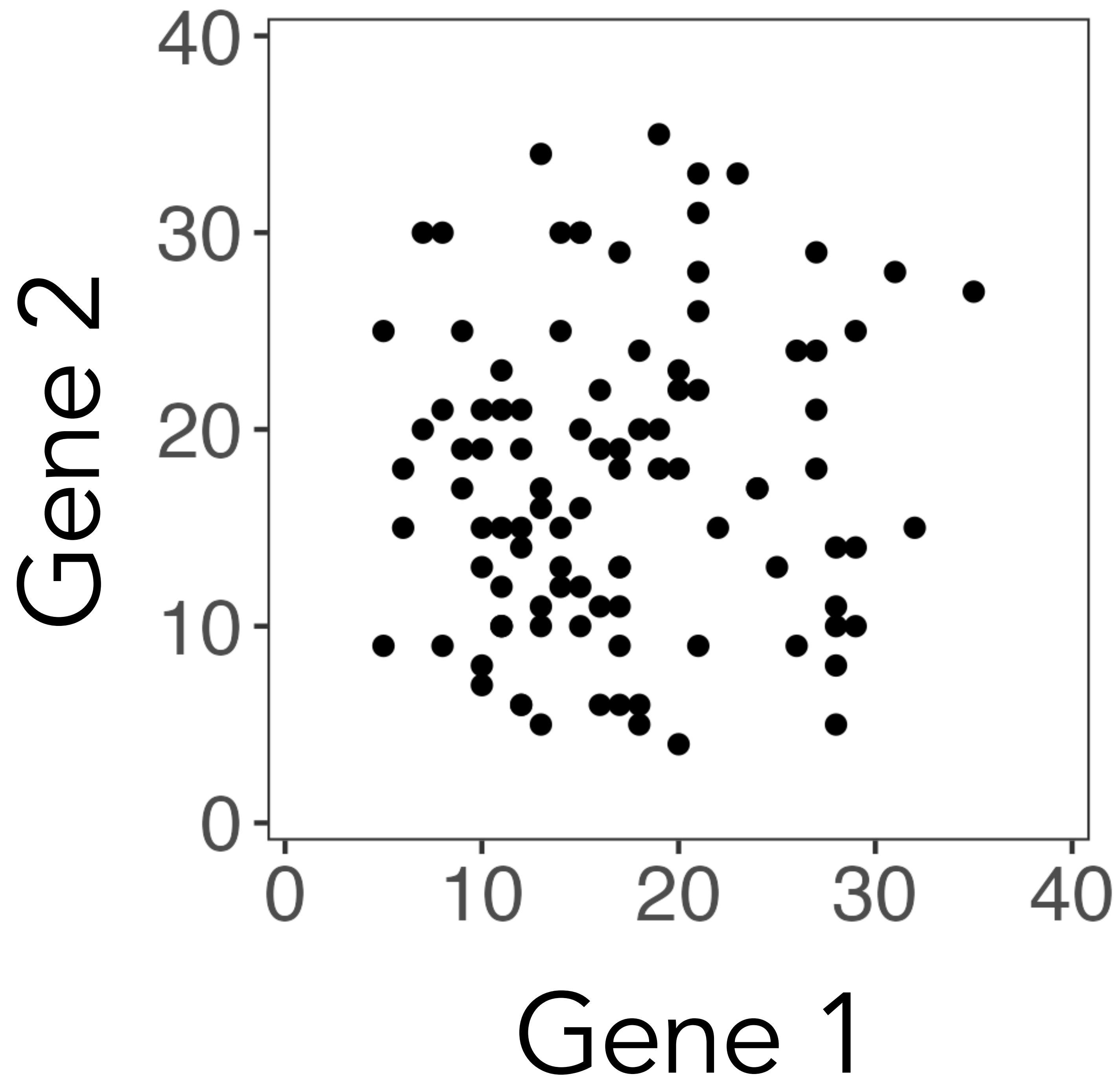
Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J Lennon, Kenneth J Livak, Tarjei S Mikkelsen & John L Rinn

Nature Biotechnology **32**, 381–386 (2014) | Cite this article

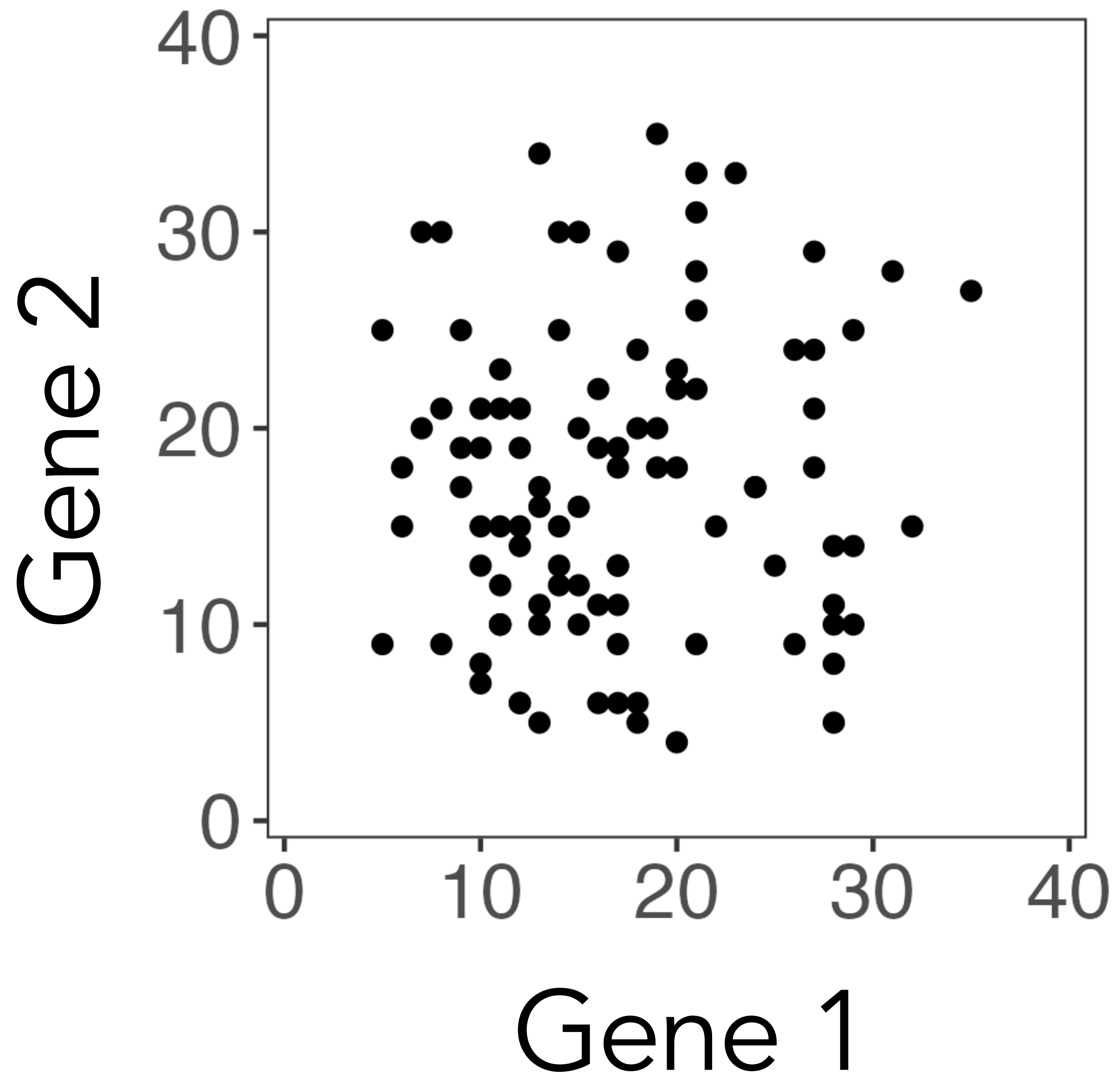




Testing for differential expression along an estimated trajectory is an example of double dipping.

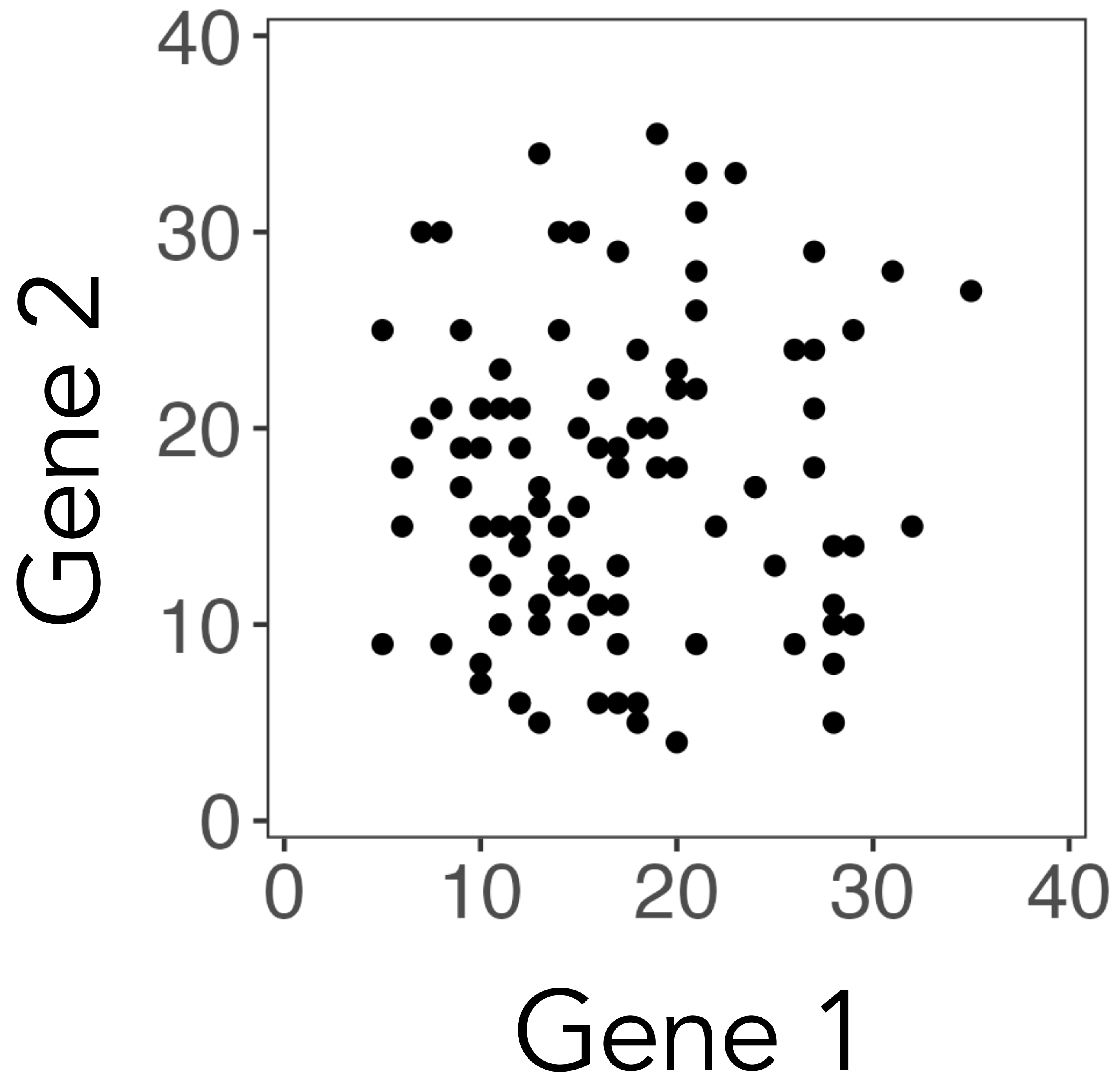


Testing for differential expression along an estimated trajectory is an example of double dipping.



Naive method:

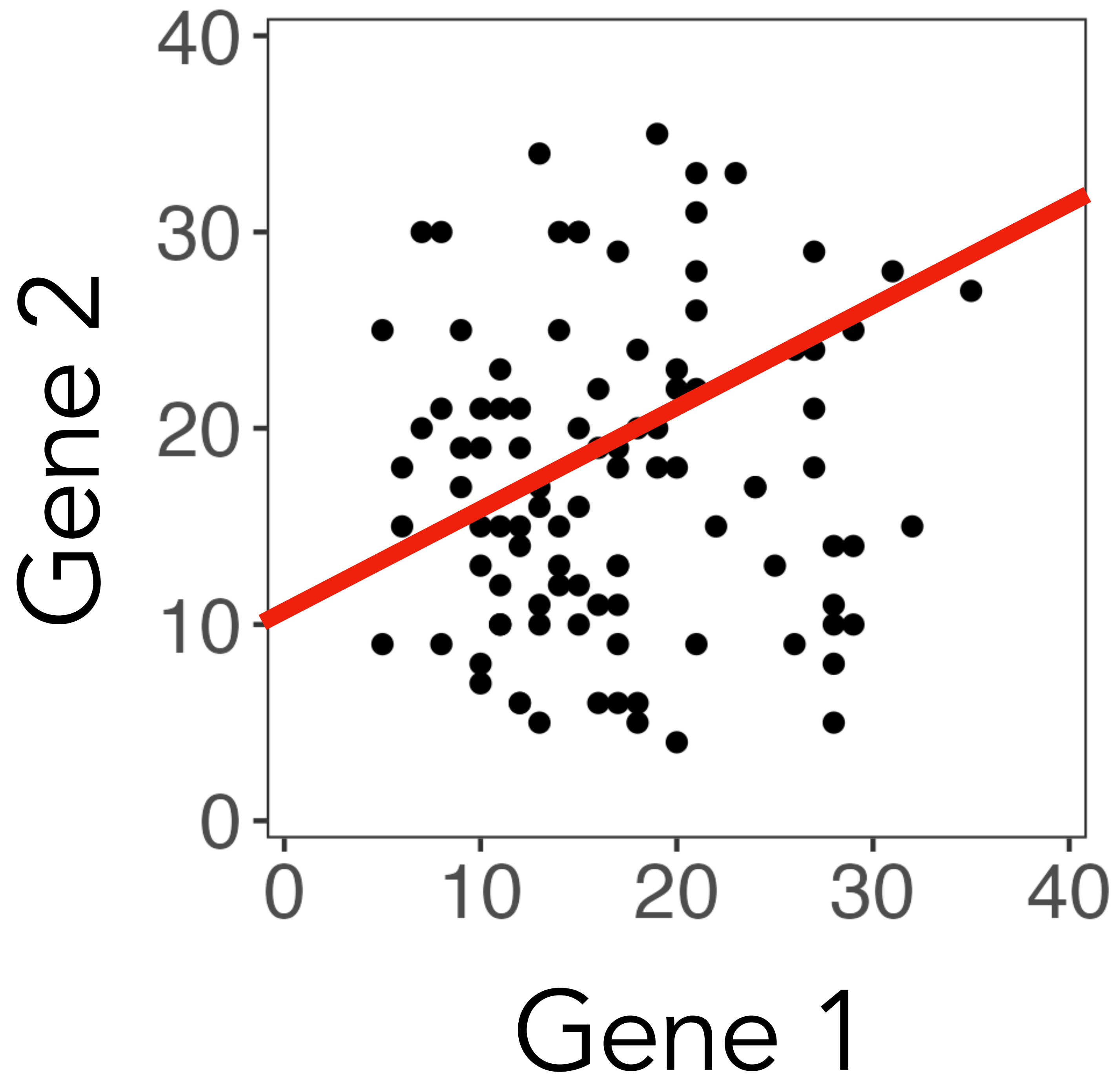
Testing for differential expression along an estimated trajectory is an example of double dipping.



Naive method:

Step 1: Estimate trajectory using the data. Denote this estimate with $\hat{L}(X)$.

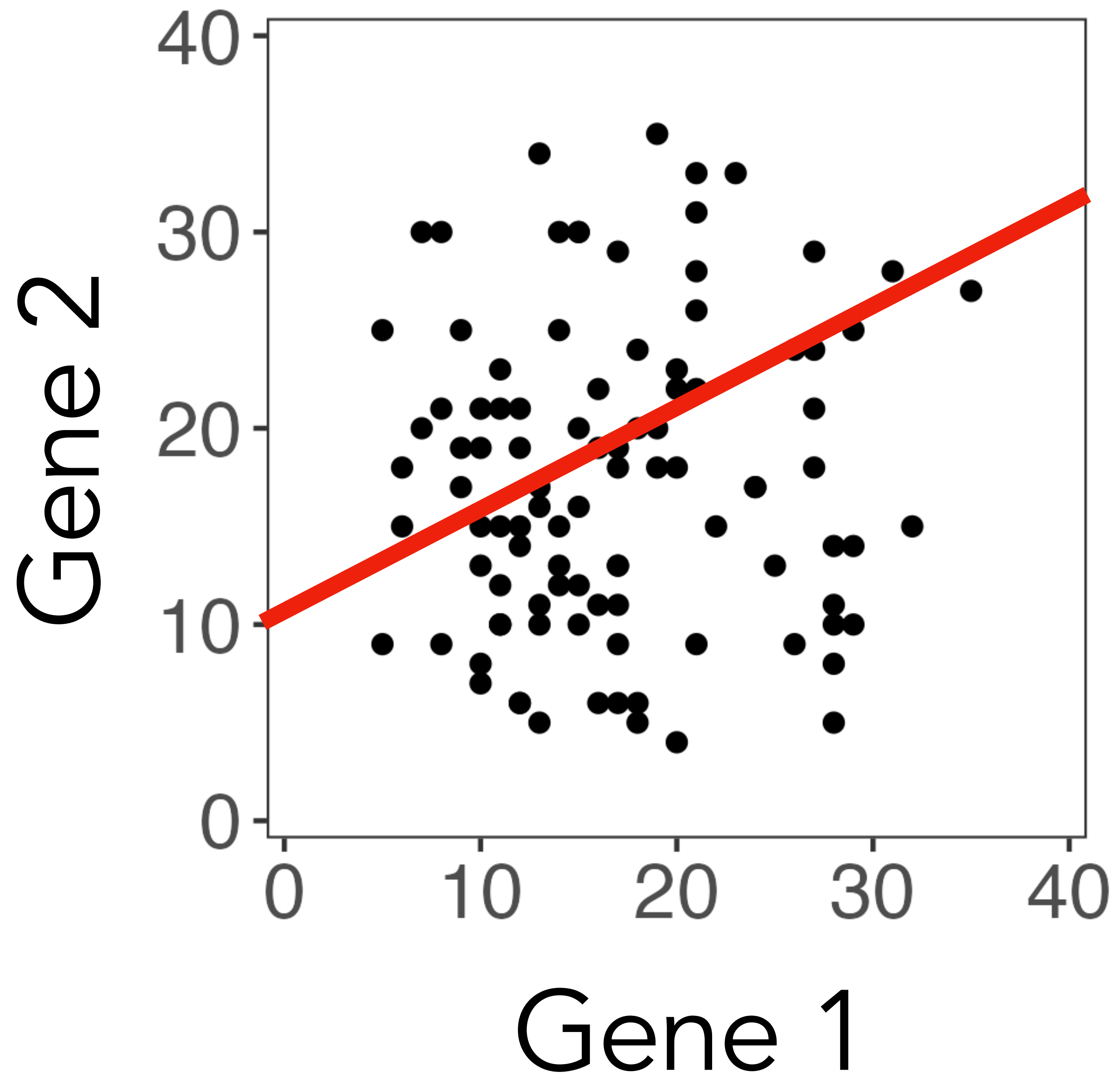
Testing for differential expression along an estimated trajectory is an example of double dipping.



Naive method:

Step 1: Estimate trajectory using the data. Denote this estimate with $\hat{L}(X)$.

Testing for differential expression along an estimated trajectory is an example of double dipping.

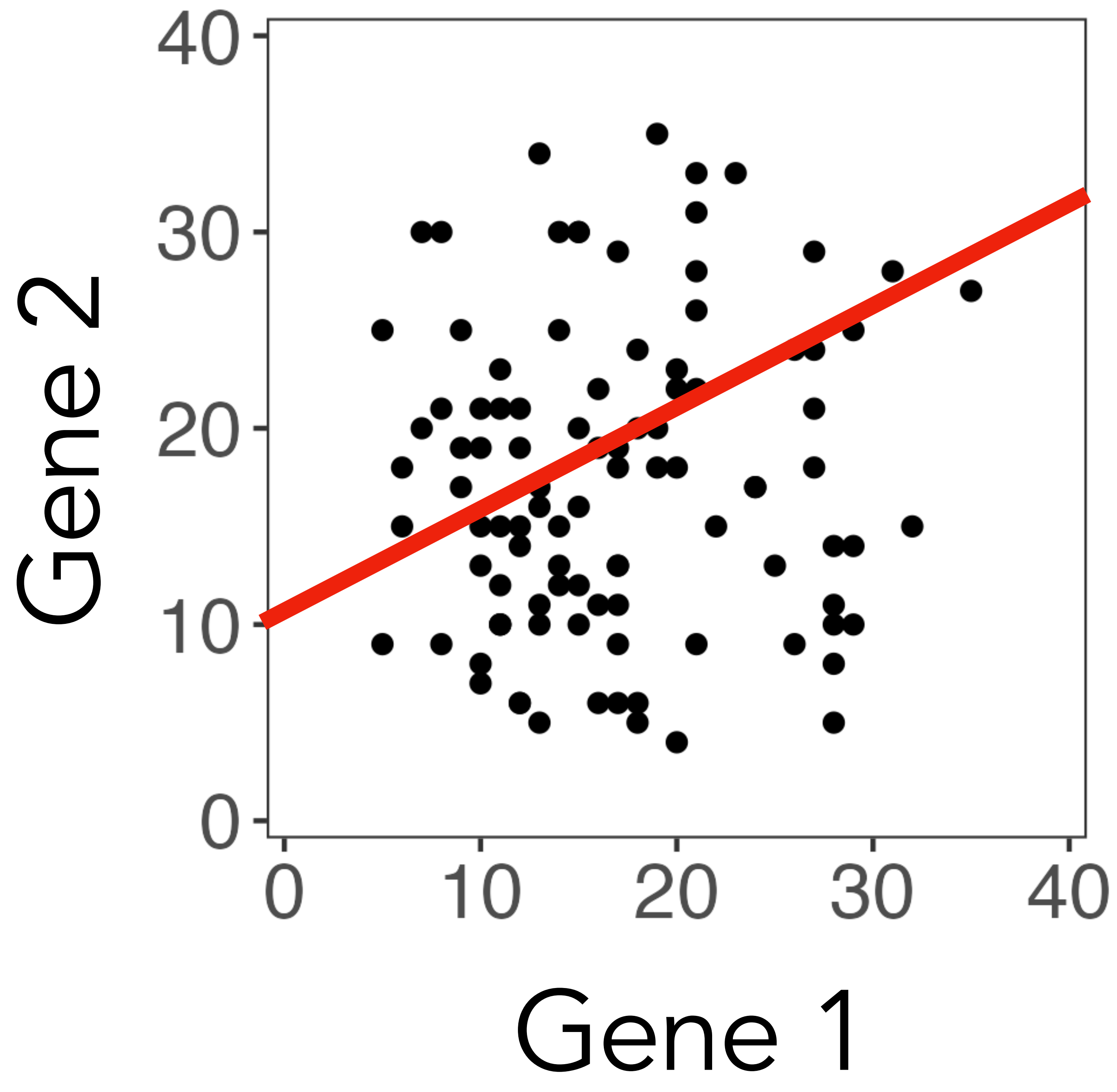


Naive method:

Step 1: Estimate trajectory using the data. Denote this estimate with $\hat{L}(X)$.

Step 2: Fit a GLM of X_j on $\hat{L}(X)$. Report p-value for the slope coefficient.

Testing for differential expression along an estimated trajectory is an example of double dipping.



$p < 10^{-10}$ 😱

Naive method:

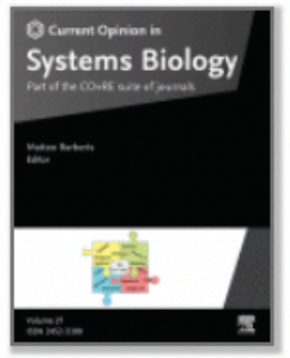
Step 1: Estimate trajectory using the data. Denote this estimate with $\hat{L}(X)$.

Step 2: Fit a GLM of X_j on $\hat{L}(X)$. Report p-value for the slope coefficient.

As in the cell type example, this problem has been pointed out

ELSEVIER

Current Opinion in Systems Biology
Volume 27, September 2021, 100344



Recent advances in trajectory inference from single-cell omics data

Louise Deconinck [1, 2], Robrecht Cannoodt [1, 2, 3], Wouter Saelens [4, 5], Bart Deplancke [4, 5], Yvan Saeys [1, 2]

However, a concern with this kind of analysis is circularity, as the same data points and features are used to perform the TI and the differential expression analysis. The TI step enforces a certain optimized ordering upon the cells, potentially enhancing expression differences along trajectories, leading to artificially low p-values and an inflated number of false positives. This is an

Common practice is to ignore the double dipping issue

nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-020-14766-3 OPEN

Check for updates

Trajectory-based differential expression analysis for single-cell sequencing data

Koen Van den Berge[1,2,3], Hector Roux de Bézieux[4,5], Kelly Street[6,7], Wouter Saelens[1,8], Robrecht Cannoodt[8,9,10], Yvan Saeys[1,8], Sandrine Dudoit[3,4,5,11]✉ & Lieven Clement[1,2,11]✉

at level α_I. It should be noted that, while the stage-wise testing paradigm theoretically controls the OFDR (given underlying assumptions are satisfied), the resulting p-values might still be too liberal since the same data are used for trajectory inference and differential expression. As mentioned before, we use p-values simply as numerical summaries for ranking the genes for further inspection.

Data with a true trajectory

PLOS GENETICS

OPEN ACCESS PEER-REVIEWED

RESEARCH ARTICLE

Single-cell sequencing reveals lineage-specific dynamic genetic regulation of gene expression during human cardiomyocyte differentiation

Reem Elorbany, Joshua M. Popp, Katherine Rhodes, Benjamin J. Strober, Kenneth Barr, Guanghao Qi, Yoav Gilad, Alexis Battle

Data with a true trajectory

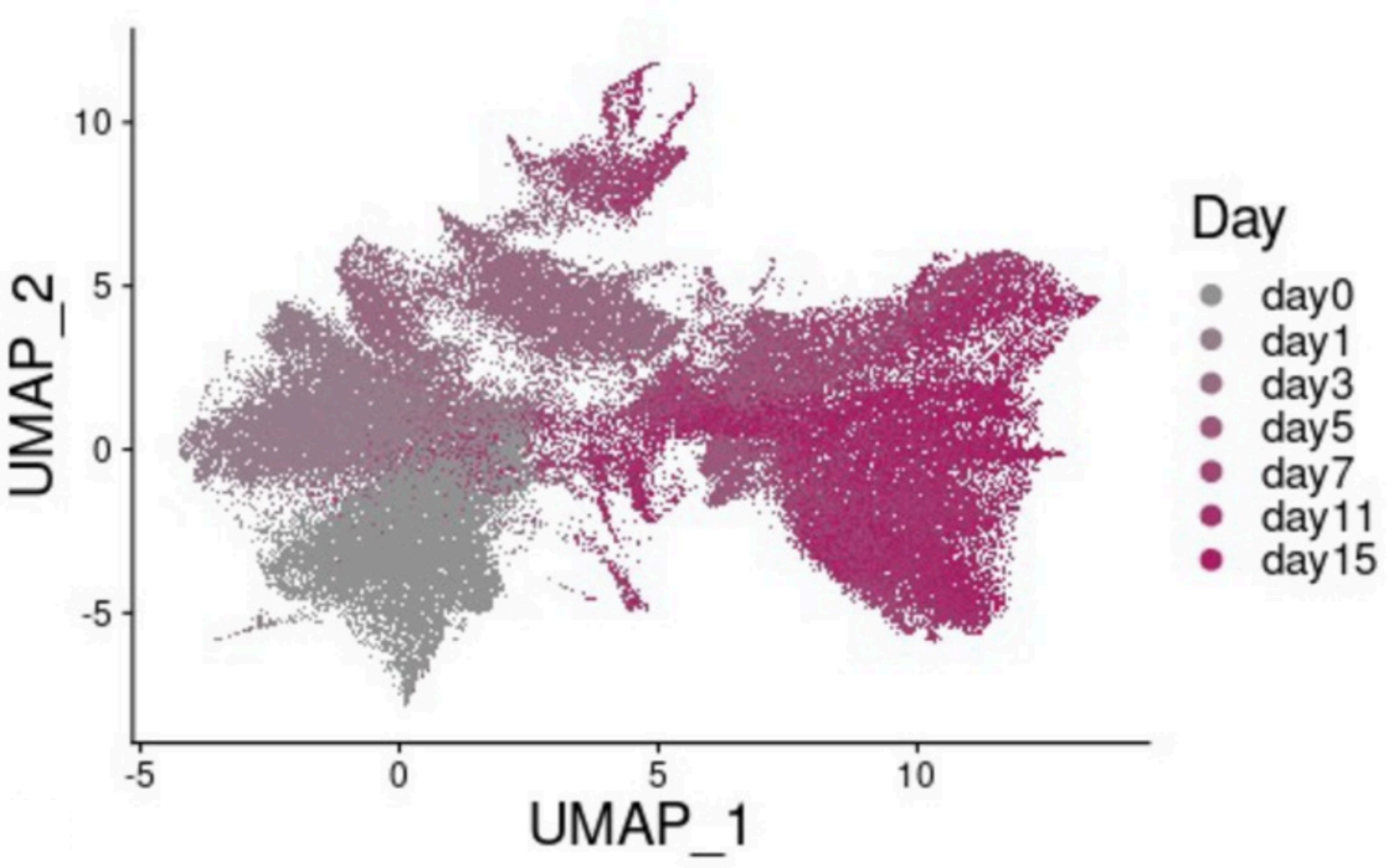
PLOS GENETICS

OPEN ACCESS PEER-REVIEWED

RESEARCH ARTICLE

Single-cell sequencing reveals lineage-specific dynamic genetic regulation of gene expression during human cardiomyocyte differentiation

Reem Elorbany, Joshua M. Popp, Katherine Rhodes, Benjamin J. Strober, Kenneth Barr, Guanghao Qi, Yoav Gilad, Alexis Battle



Data with a true trajectory

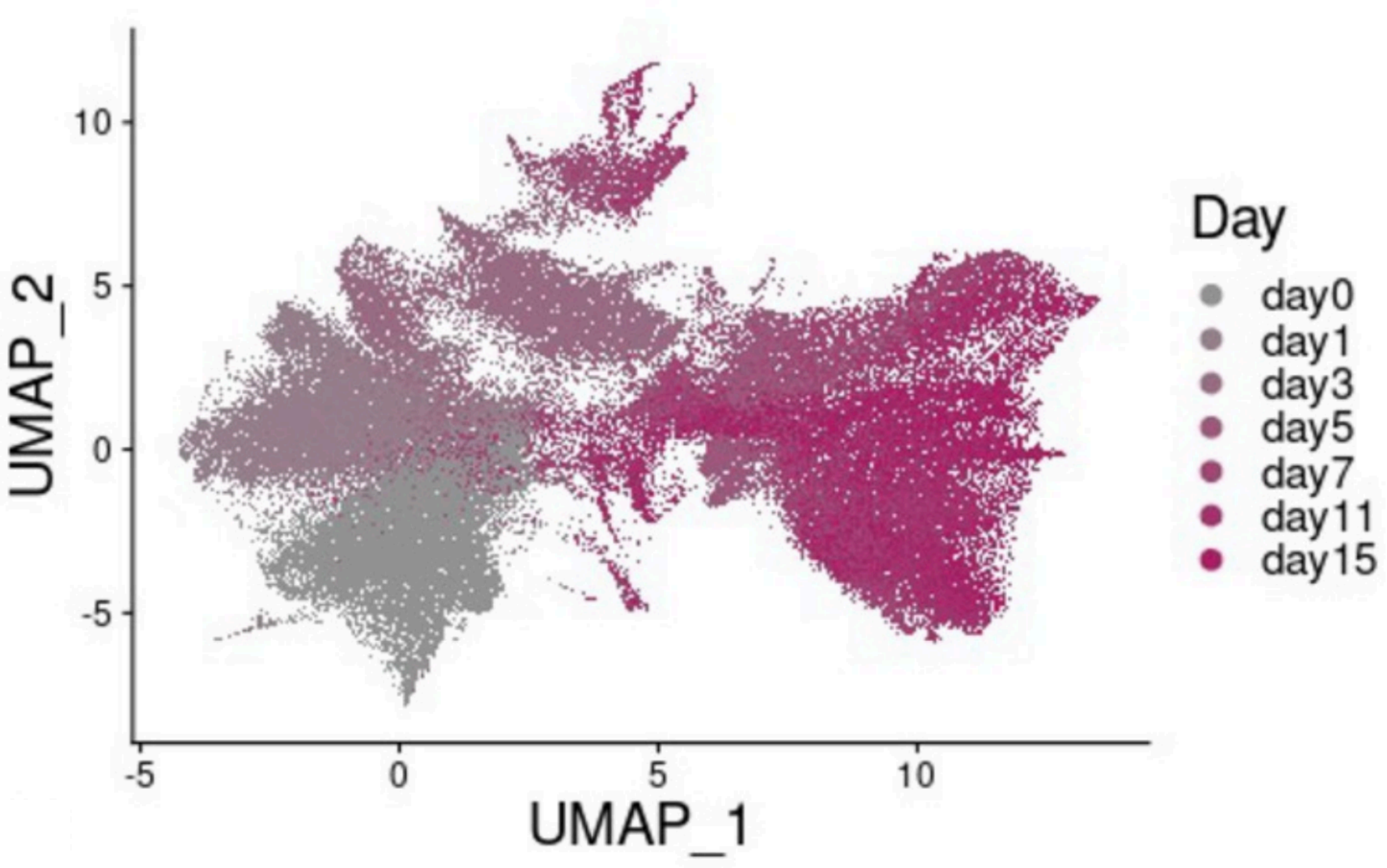
PLOS GENETICS

OPEN ACCESS PEER-REVIEWED

RESEARCH ARTICLE

Single-cell sequencing reveals lineage-specific dynamic genetic regulation of gene expression during human cardiomyocyte differentiation

Reem Elorbany, Joshua M. Popp, Katherine Rhodes, Benjamin J. Strober, Kenneth Barr, Guanghao Qi, Yoav Gilad, Alexis Battle



In this case, some true temporal information is observed (day).

We will ignore this, and construct a continuous trajectory (pseudotime) from the data.

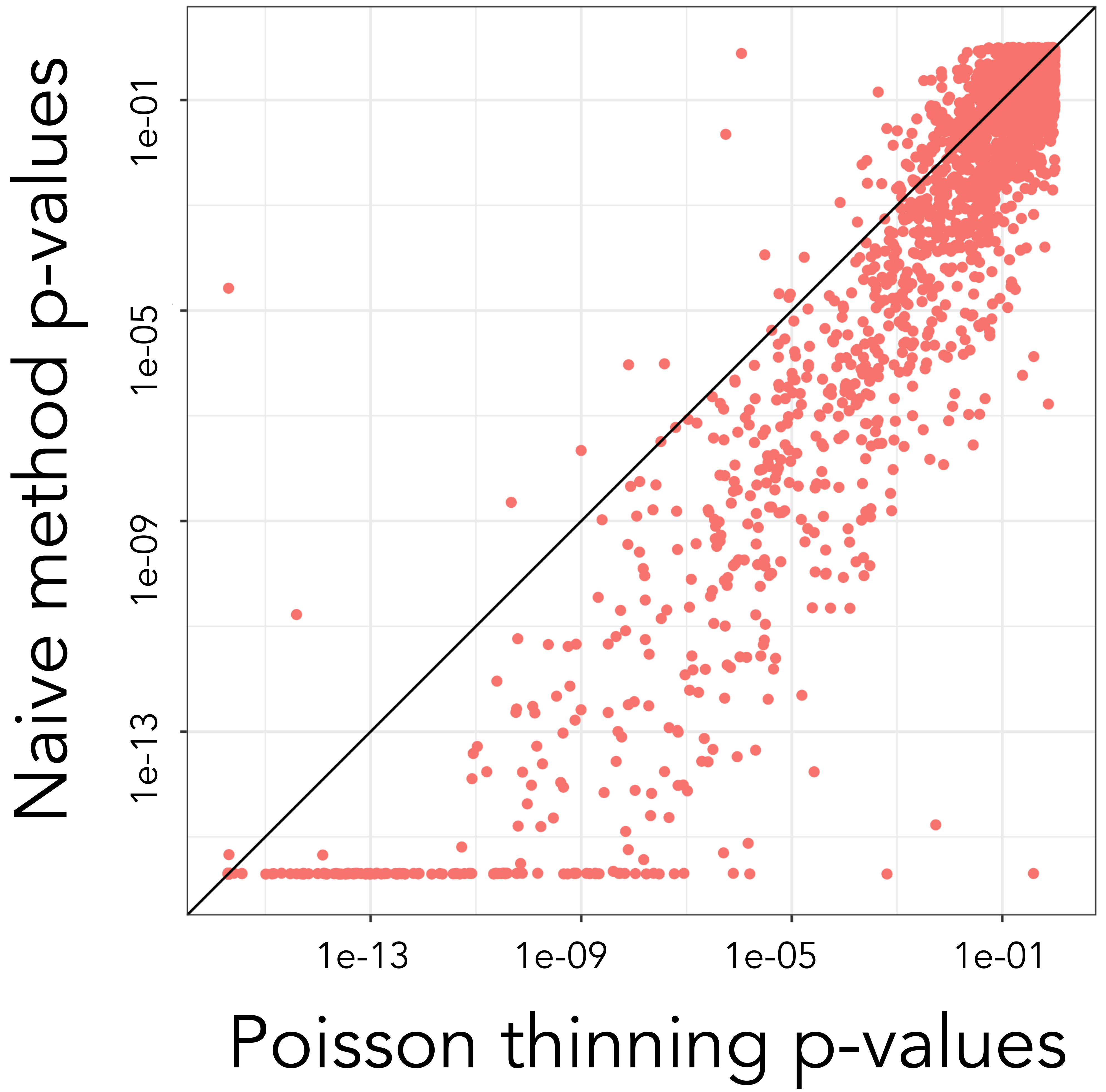
Comparing thinning to the naive method on data with a true trajectory

$\hat{L}(\cdot)$ function is pipeline from the Monocle3 R package (preprocessing + pseudotime).

Naive method: For each gene, fit a Poisson GLM of X_j on $\hat{L}(X)$ and report p-value.

Thinning: Apply Poisson thinning with $\epsilon = 0.5$ to get $X^{(1)}$ and $X^{(2)}$. For each gene, fit a Poisson GLM of $X_j^{(2)}$ on $\hat{L}(X^{(1)})$ and report p-value.

Comparing thinning to the naive method on data with a true trajectory

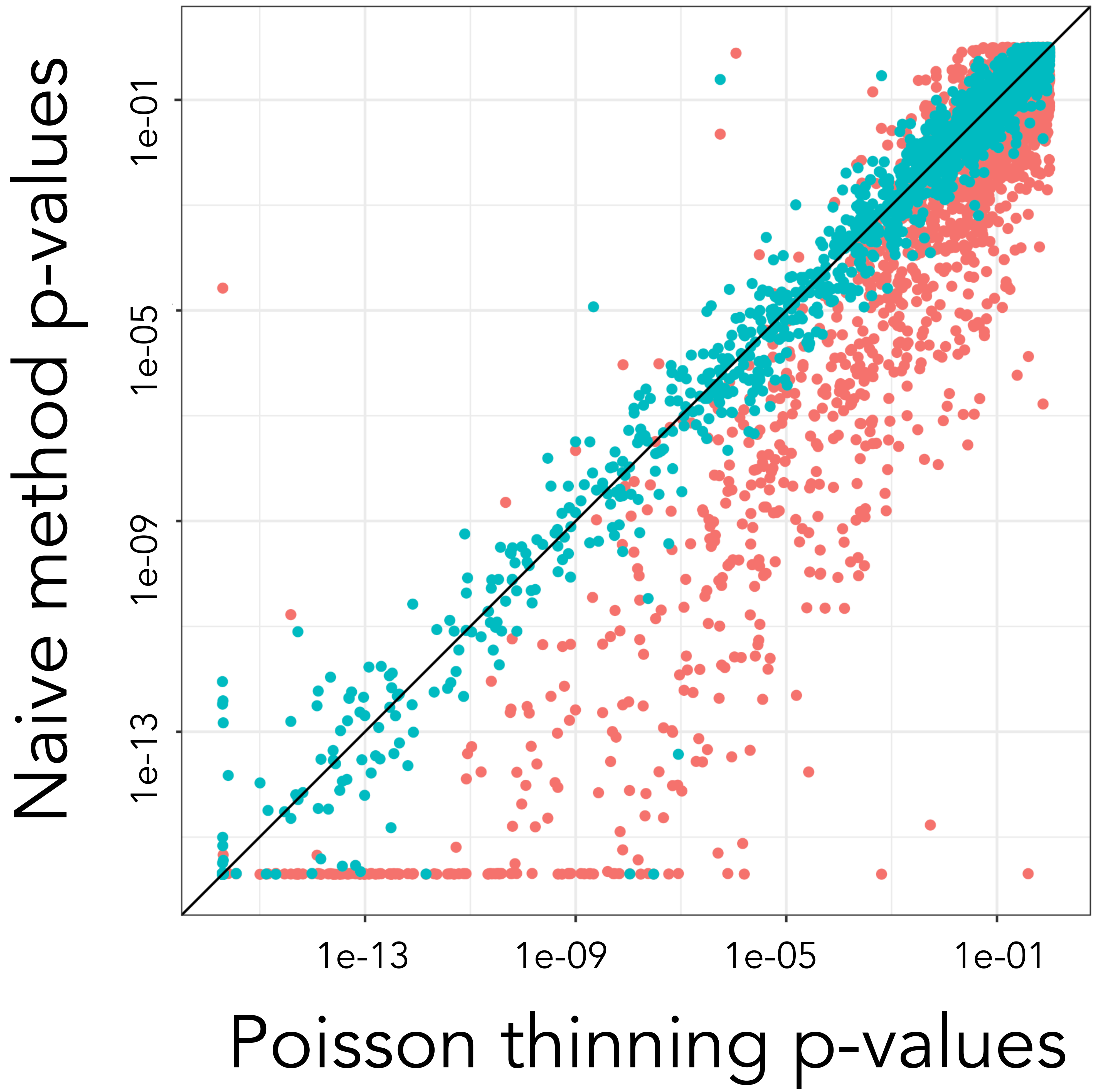


$\hat{L}(\cdot)$ function is pipeline from the Monocle3 R package (preprocessing + pseudotime).

Naive method: For each gene, fit a Poisson GLM of X_j on $\hat{L}(X)$ and report p-value.

Thinning: Apply Poisson thinning with $\epsilon = 0.5$ to get $X^{(1)}$ and $X^{(2)}$. For each gene, fit a Poisson GLM of $X_j^{(2)}$ on $\hat{L}(X^{(1)})$ and report p-value.

Comparing thinning to the naive method on data with a true trajectory



$\hat{L}(\cdot)$ function is pipeline from the Monocle3 R package (preprocessing + pseudotime).

Naive method: For each gene, fit a Poisson GLM of X_j on $\hat{L}(X)$ and report p-value.

Thinning: Apply Poisson thinning with $\epsilon = 0.5$ to get $X^{(1)}$ and $X^{(2)}$. For each gene, fit a Poisson GLM of $X_j^{(2)}$ on $\hat{L}(X^{(1)})$ and report p-value.

Data with no true trajectory

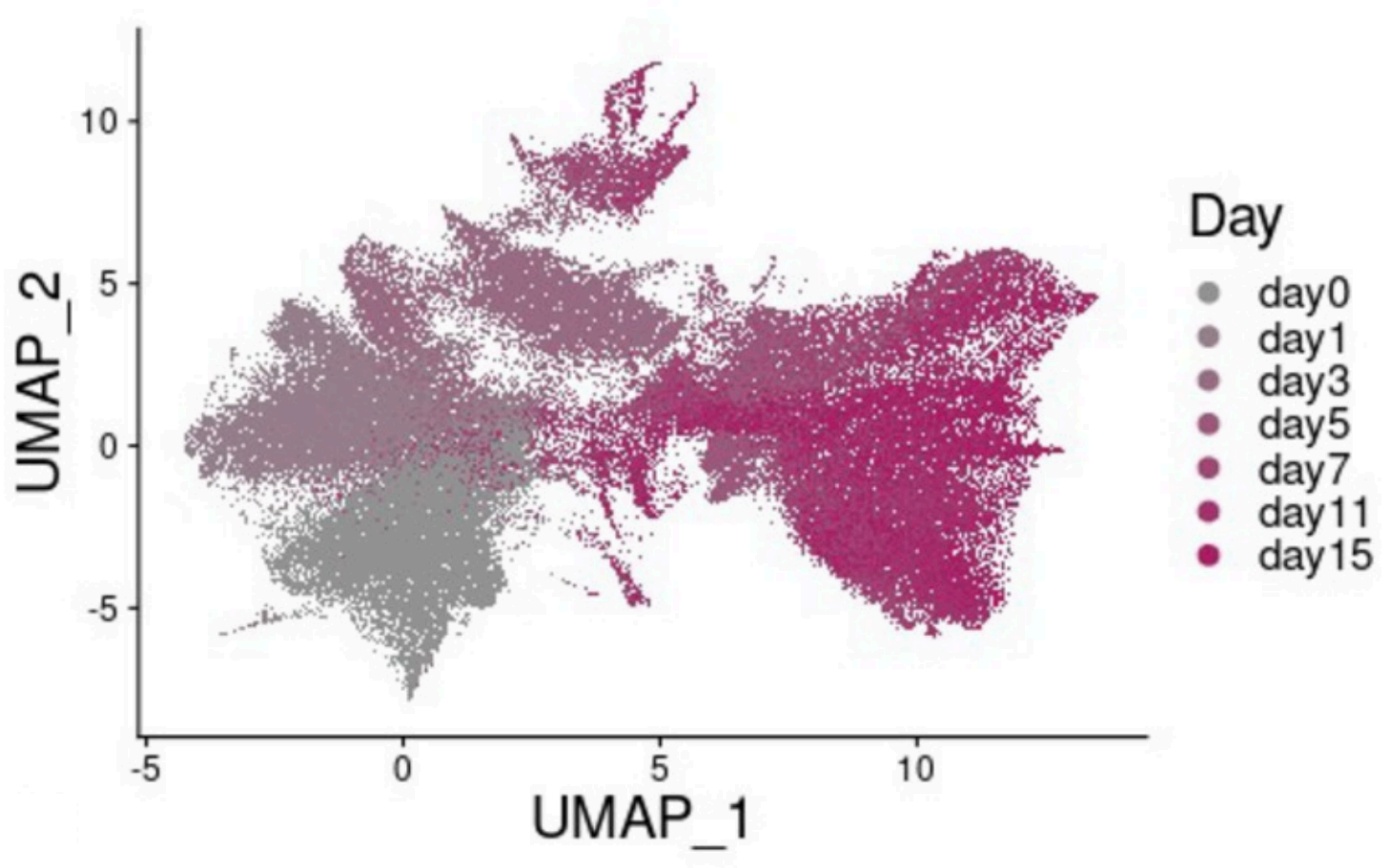
PLOS GENETICS

OPEN ACCESS PEER-REVIEWED

RESEARCH ARTICLE

Single-cell sequencing reveals lineage-specific dynamic genetic regulation of gene expression during human cardiomyocyte differentiation

Reem Elorbany, Joshua M. Popp, Katherine Rhodes, Benjamin J. Strober, Kenneth Barr, Guanghao Qi, Yoav Gilad, Alexis Battle



Data with no true trajectory

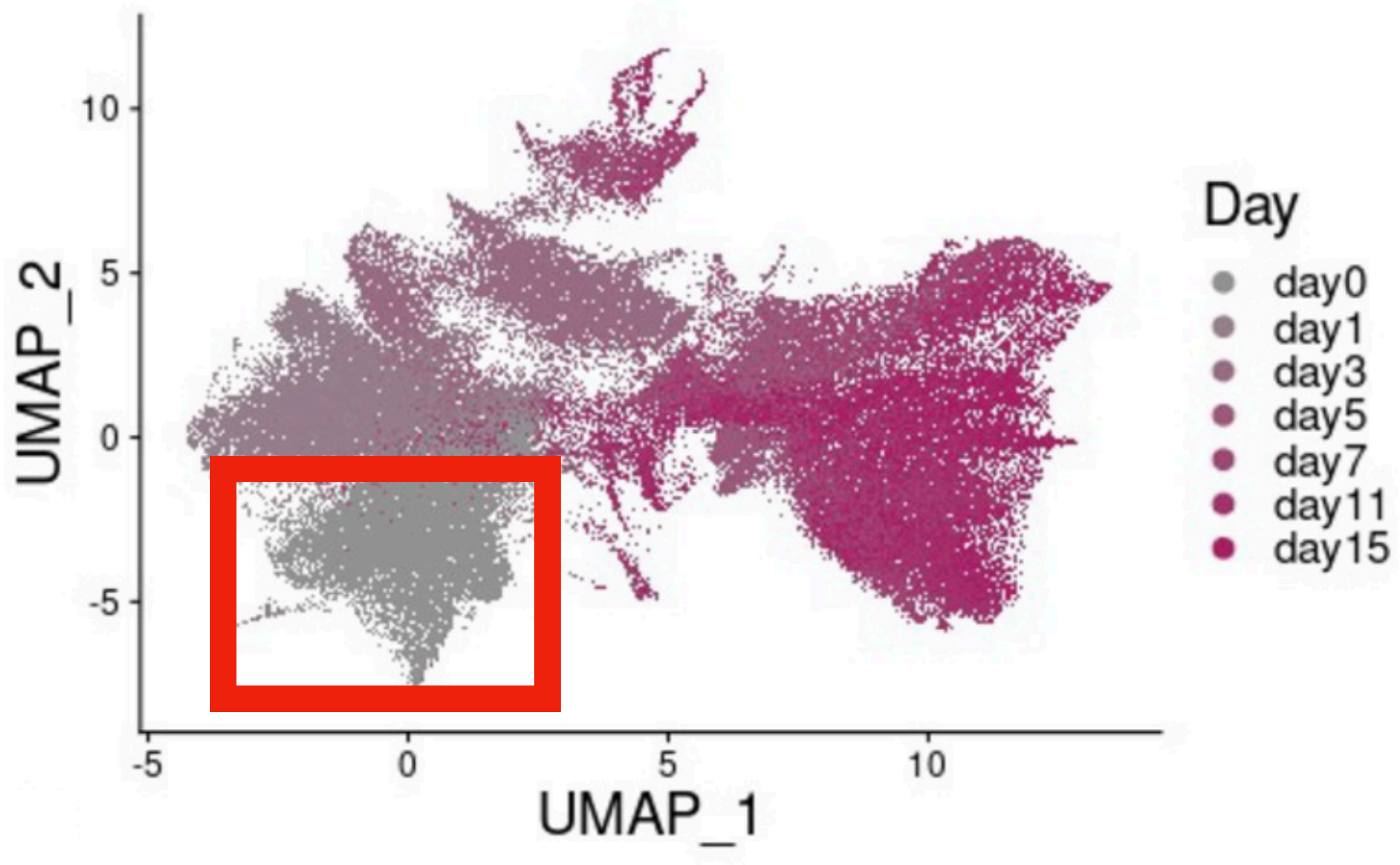
PLOS GENETICS

OPEN ACCESS PEER-REVIEWED

RESEARCH ARTICLE

Single-cell sequencing reveals lineage-specific dynamic genetic regulation of gene expression during human cardiomyocyte differentiation

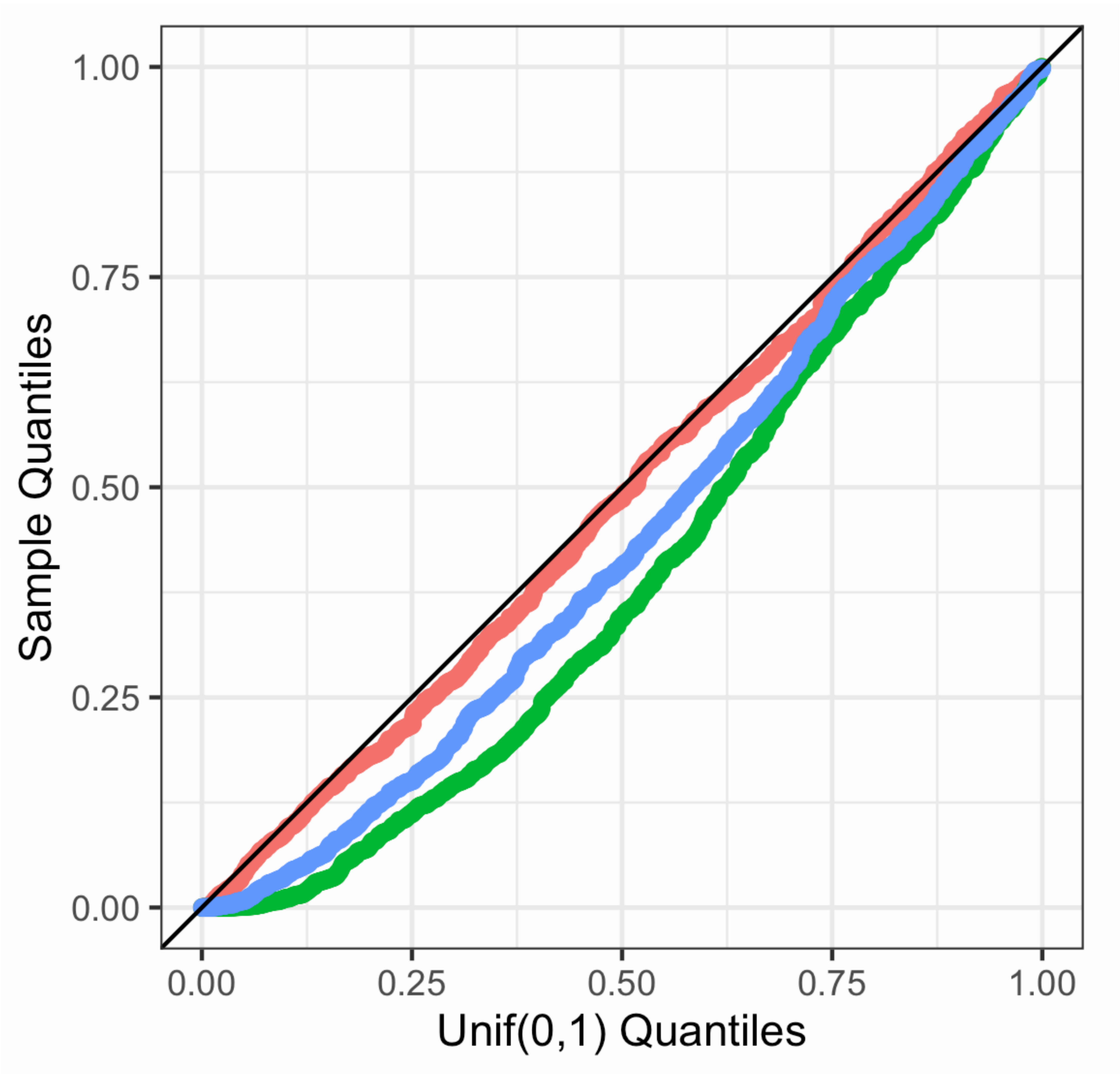
Reem Elorbany, Joshua M. Popp, Katherine Rhodes, Benjamin J. Strober, Kenneth Barr, Guanghao Qi, Yoav Gilad, Alexis Battle



Subset the data to day0 cells only.
Regress out metadata.

Comparing thinning to the naive method on data with no true trajectory

Day 0 Only



Data thinning, $\epsilon = 0.5$
Naive on full data
Naive on test data

Outline

1. Motivation: settings where sample splitting doesn’t work
2. Poisson thinning
3. Data thinning
4. Application to human fetal cell atlas data
5. Application to cardiomyocyte differentiation data
6. **Ongoing work**

Multifold data thinning can be used to carry out a full analysis pipeline without double dipping.

X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8

Multifold data thinning can be used to carry out a full analysis pipeline without double dipping.

X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8

$X^{(1)}$

	Gene 1	Gene 2
Cell 1	3	0
Cell 2	8	1

$X^{(2)}$

	Gene 1	Gene 2
Cell 1	2	3
Cell 2	5	3

⋮

$X^{(K)}$

	Gene 3	Gene 4
Cell 1	4	1
Cell 2	5	0

Multifold data thinning can be used to carry out a full analysis pipeline without double dipping.

X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8

X_{ij}

$X^{(1)}$

	Gene 1	Gene 2
Cell 1	[illegible]	0
Cell 2	[illegible]	1

$X_{ij}^{(1)}$

$X^{(2)}$

	Gene 1	Gene 2
Cell 1	2	3
Cell 2	5	3

$X_{ij}^{(2)}$

⋮

$X^{(K)}$

	[illegible]	Gene 4
Cell 1	4	1
Cell 2	5	0

$X_{ij}^{(K)}$

Multifold data thinning can be used to carry out a full analysis pipeline without double dipping.

X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8

X_{ij}

$X^{(1)}$

	Gene 1	Gene 2
Cell 1	[illegible]	0
Cell 2	[illegible]	1

$X_{ij}^{(1)}$

$X^{(2)}$

	Gene 1	Gene 2
Cell 1	2	3
Cell 2	5	3

$X_{ij}^{(2)}$

$\vdots$

$$\left(X_{ij}^{(1)}, \ldots, X_{ij}^{(K)}\right) \mid X_{ij} = x_{ij} \sim \text{Multinomial}\left(x_{ij}, \frac{1}{K}, \ldots, \frac{1}{K}\right)$$

$X^{(K)}$

	Ge[illegible]	Gene 4
Cell 1	4	1
Cell 2	5	0

$X_{ij}^{(K)}$

Multifold data thinning can be used to carry out a full analysis pipeline without double dipping.

X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8

X_{ij}

$X^{(1)}$

	Gene 1	Gene 2
Cell 1	[illegible]	0
Cell 2	[illegible]	1

$X^{(1)}_{ij}$

$X^{(2)}$

	Gene 1	Gene 2
Cell 1	2	3
Cell 2	5	3

$X^{(2)}_{ij}$

⋮

$$\left(X^{(1)}_{ij}, \ldots, X^{(K)}_{ij}\right) \mid X_{ij} = x_{ij} \sim \text{Multinomial}\left(x_{ij}, \frac{1}{K}, \ldots, \frac{1}{K}\right)$$

$X^{(K)}$

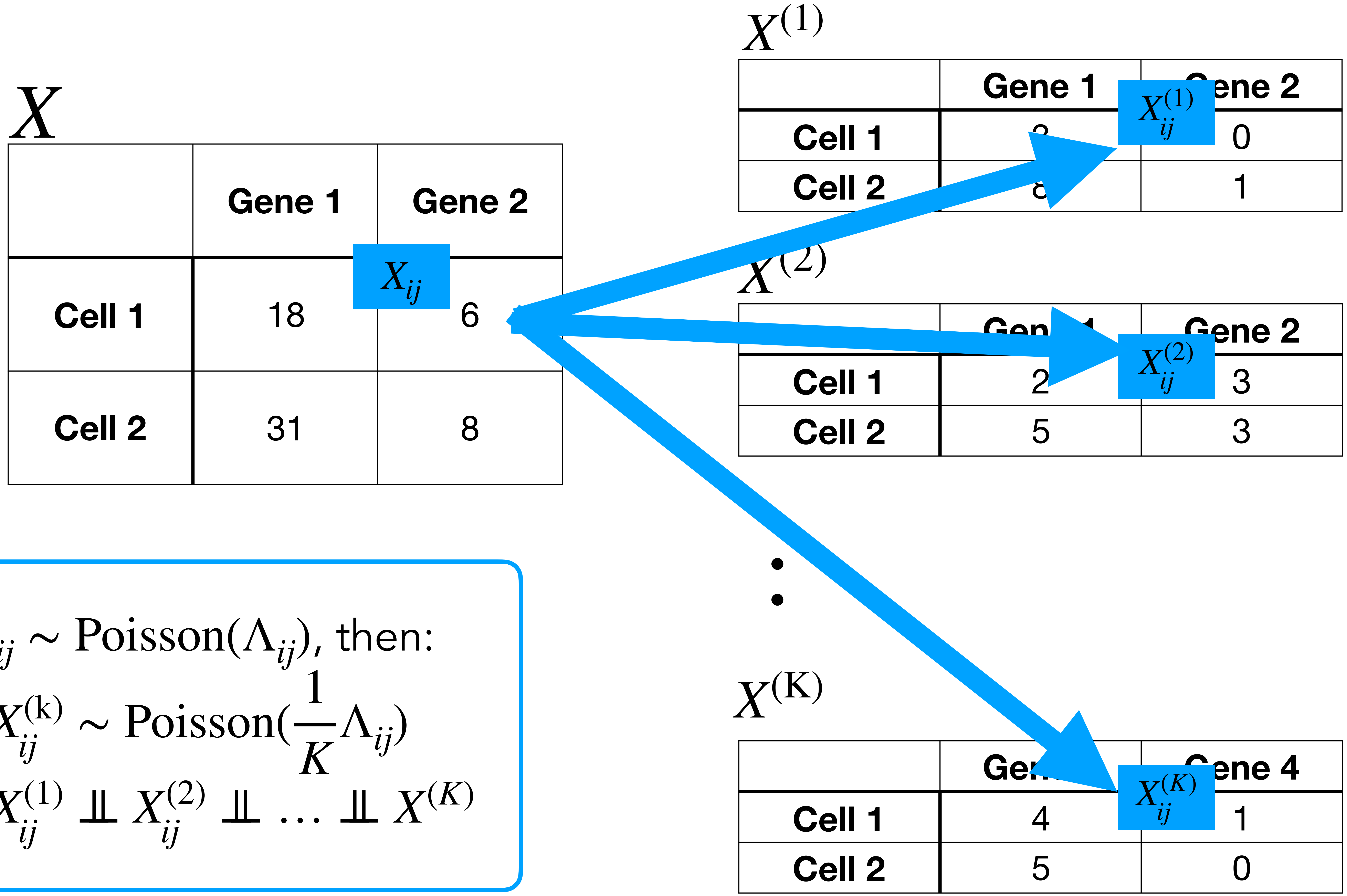
	Gen[illegible]	Gene 4
Cell 1	4	1
Cell 2	5	0

$X^{(K)}_{ij}$

If $X_{ij} \sim \text{Poisson}(\Lambda_{ij})$, then:

1. $X^{(k)}_{ij} \sim \text{Poisson}(\frac{1}{K}\Lambda_{ij})$
2. $X^{(1)}_{ij} \perp\!\!\!\perp X^{(2)}_{ij} \perp\!\!\!\perp \ldots \perp\!\!\!\perp X^{(K)}$

Multifold data thinning can be used to carry out a full analysis pipeline without double dipping.



If $X_{ij} \sim \text{Poisson}(\Lambda_{ij})$, then:

1. $X_{ij}^{(k)} \sim \text{Poisson}(\frac{1}{K}\Lambda_{ij})$
2. $X_{ij}^{(1)} \perp\!\!\!\perp X_{ij}^{(2)} \perp\!\!\!\perp \ldots \perp\!\!\!\perp X^{(K)}$

Multifold data thinning can be used to carry out a full analysis pipeline without double dipping.

X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8

X_{ij}

$X^{(1)}$

	Gene 1	Gene 2
Cell 1	[illegible]	0
Cell 2	[illegible]	1

$X_{ij}^{(1)}$

Estimate clusters.

$X^{(2)}$

	Gene 1	Gene 2
Cell 1	2	3
Cell 2	5	3

$X_{ij}^{(2)}$

⋮

$X^{(\mathrm{K})}$

	Gen[illegible]	Gene 4
Cell 1	4	1
Cell 2	5	0

$X_{ij}^{(K)}$

If $X_{ij} \sim \mathrm{Poisson}(\Lambda_{ij})$, then:

1. $X_{ij}^{(\mathrm{k})} \sim \mathrm{Poisson}(\frac{1}{K}\Lambda_{ij})$
2. $X_{ij}^{(1)} \perp\!\!\!\perp X_{ij}^{(2)} \perp\!\!\!\perp \ldots \perp\!\!\!\perp X^{(K)}$

Multifold data thinning can be used to carry out a full analysis pipeline without double dipping.

X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8

$X^{(1)}$

	Gene 1	Gene 2
Cell 1	[illegible]	0
Cell 2	[illegible]	1

Estimate clusters.

$X^{(2)}$

	[illegible]	Gene 2
Cell 1	2	3
Cell 2	5	3

Evaluate/ select number of clusters.

⋮

$X^{(K)}$

	[illegible]	Gene 4
Cell 1	4	1
Cell 2	5	0

If $X_{ij} \sim \text{Poisson}(\Lambda_{ij})$, then:

1. $X^{(k)}_{ij} \sim \text{Poisson}(\frac{1}{K}\Lambda_{ij})$
2. $X^{(1)}_{ij} \perp\!\!\!\perp X^{(2)}_{ij} \perp\!\!\!\perp \ldots \perp\!\!\!\perp X^{(K)}$

Multifold data thinning can be used to carry out a full analysis pipeline without double dipping.

X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8

X_{ij}

$X^{(1)}$

	Gene 1	Gene 2
Cell 1	[illegible]	0
Cell 2	8	1

$X^{(1)}_{ij}$

Estimate clusters.

$X^{(2)}$

	Gene 1	Gene 2
Cell 1	2	3
Cell 2	5	3

$X^{(2)}_{ij}$

Evaluate/ select number of clusters.

Cross-validate for stability

$\vdots$

$X^{(K)}$

	Ge[illegible]	Gene 4
Cell 1	4	1
Cell 2	5	0

$X^{(K)}_{ij}$

If $X_{ij} \sim \text{Poisson}(\Lambda_{ij})$, then:

1. $X^{(k)}_{ij} \sim \text{Poisson}(\frac{1}{K}\Lambda_{ij})$
2. $X^{(1)}_{ij} \perp\!\!\!\perp X^{(2)}_{ij} \perp\!\!\!\perp \ldots \perp\!\!\!\perp X^{(K)}$

Multifold data thinning can be used to carry out a full analysis pipeline without double dipping.

X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8

X_{ij}

$X^{(1)}$

	Gene 1	Gene 2
Cell 1	[illegible]	0
Cell 2	[illegible]	1

$X_{ij}^{(1)}$

Estimate clusters.

$X^{(2)}$

	Gene 1	Gene 2
Cell 1	2	3
Cell 2	5	3

$X_{ij}^{(2)}$

Evaluate/ select number of clusters.

Cross-validate for stability

$\vdots$

$X^{(K)}$

	[illegible]	Gene 4
Cell 1	4	1
Cell 2	5	0

$X_{ij}^{(K)}$

Differential expression testing on final, selected clusters.

If $X_{ij} \sim \text{Poisson}(\Lambda_{ij})$, then:

1. $X_{ij}^{(k)} \sim \text{Poisson}(\frac{1}{K}\Lambda_{ij})$
2. $X_{ij}^{(1)} \perp\!\!\!\perp X_{ij}^{(2)} \perp\!\!\!\perp \ldots \perp\!\!\!\perp X^{(K)}$

Additional future work

- **Inference after latent variable estimation:**
  - Propagating uncertainty in cell type or trajectory estimate.
  - Aggregating p-values across multiple random splits to improve power and stability.
- **Model selection for latent variable models:**
  - Integrating several steps of analysis, e.g. selecting number of PCs, number of highly variable genes, and number of clusters.
- **Additional applications of data thinning to scRNA-seq data, or other types of biological data.**
  - Please reach out if you have ideas!

Acknowledgements

Daniela Witten
University of Washington

Lucy Gao
University of British Columbia

Ameer Dharamshi
University of Washington

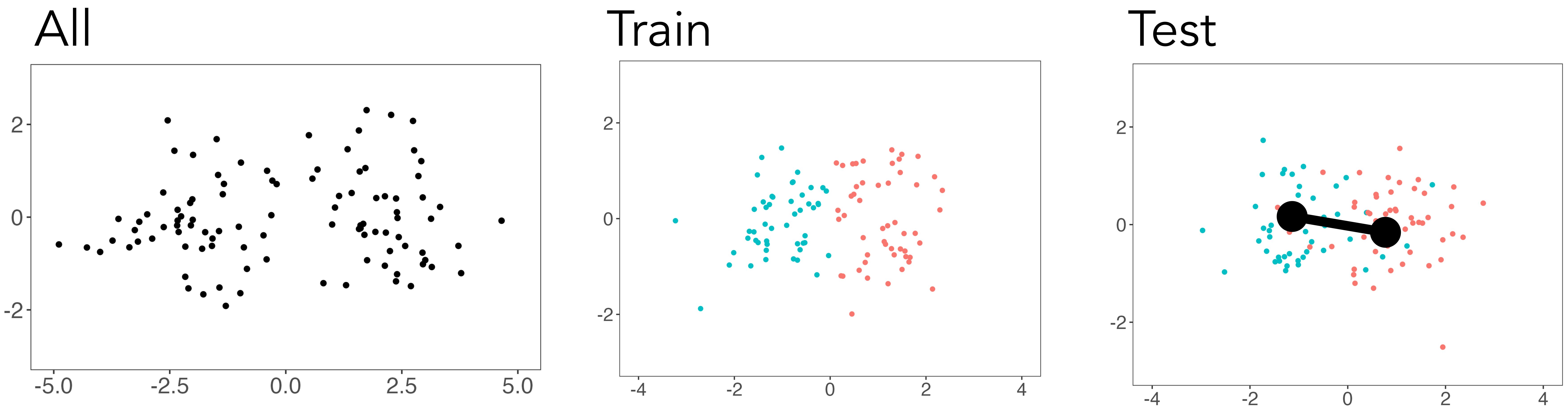
Alexis Battle
Johns Hopkins

Joshua Popp
Johns Hopkins

Questions?

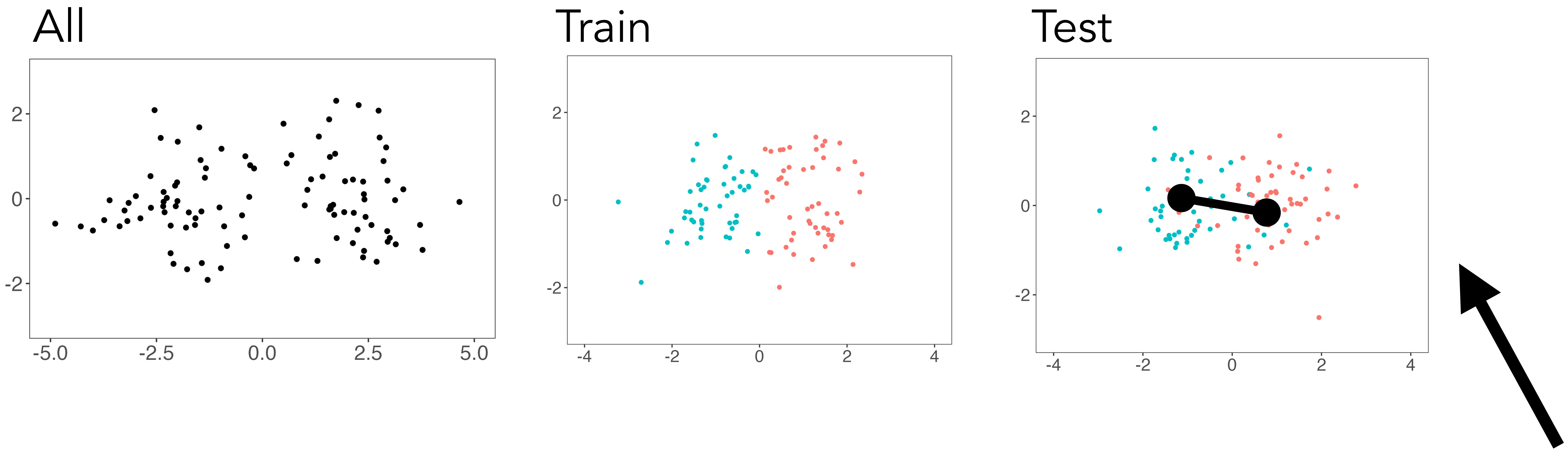
Comparison to selective inference for overall difference in cluster means

Data thinning:



Comparison to selective inference for overall difference in cluster means

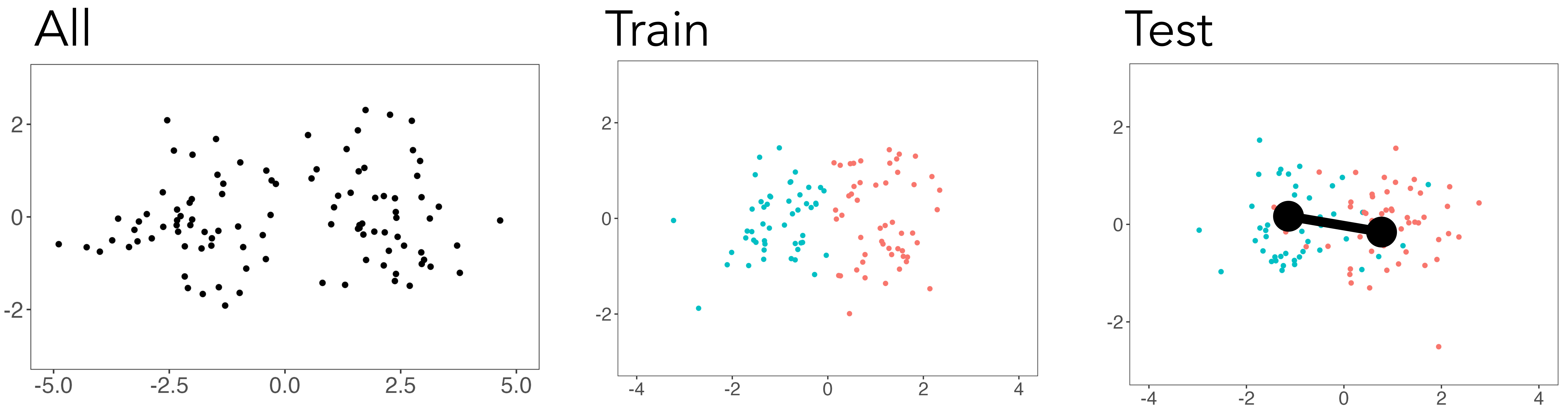
Data thinning:



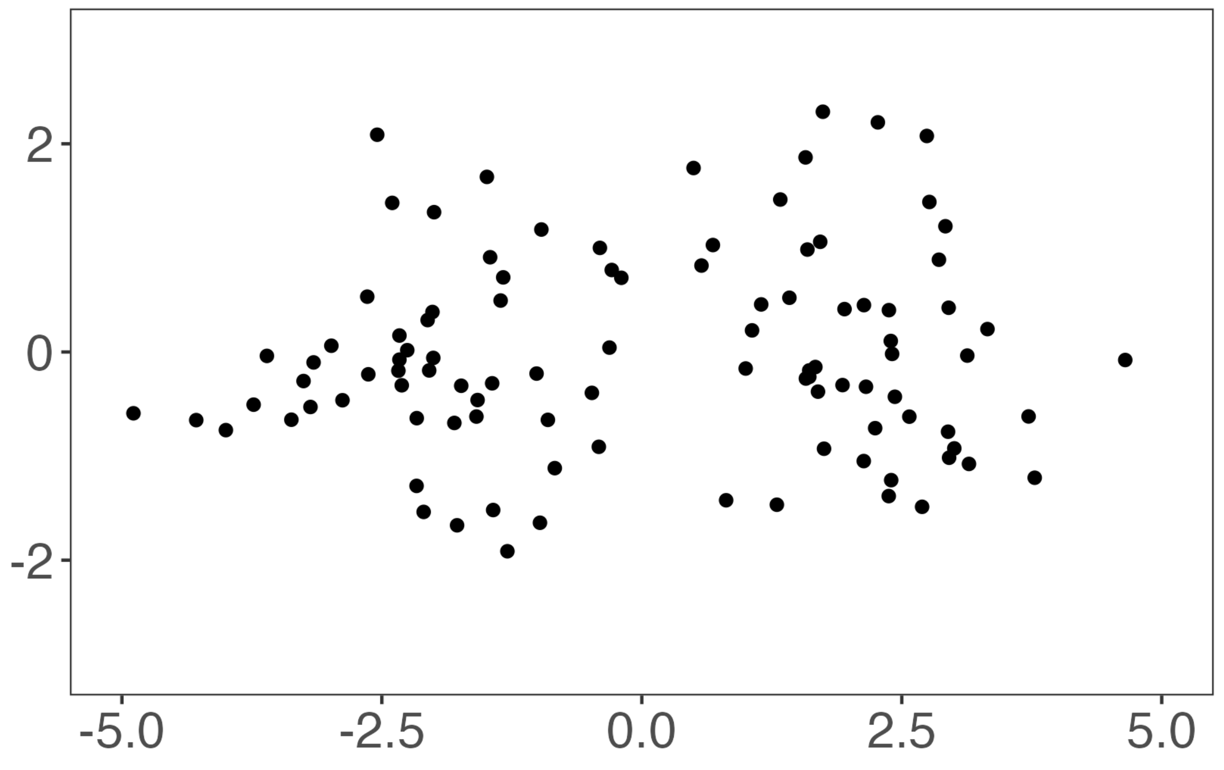
$$Pr_{H_0}\left(\left|\bar{\mathbf{X}}^{\text{test}}_{\hat{A}_{\text{train}}} - \bar{\mathbf{X}}^{\text{test}}_{\hat{B}_{\text{train}}}\right| \geq \left|\bar{X}^{\text{test}}_{\hat{A}_{\text{train}}} - \bar{X}^{\text{test}}_{\hat{B}_{train}}\right|\right)$$

Comparison to selective inference for overall difference in cluster means

Data thinning:



Selective Inference:

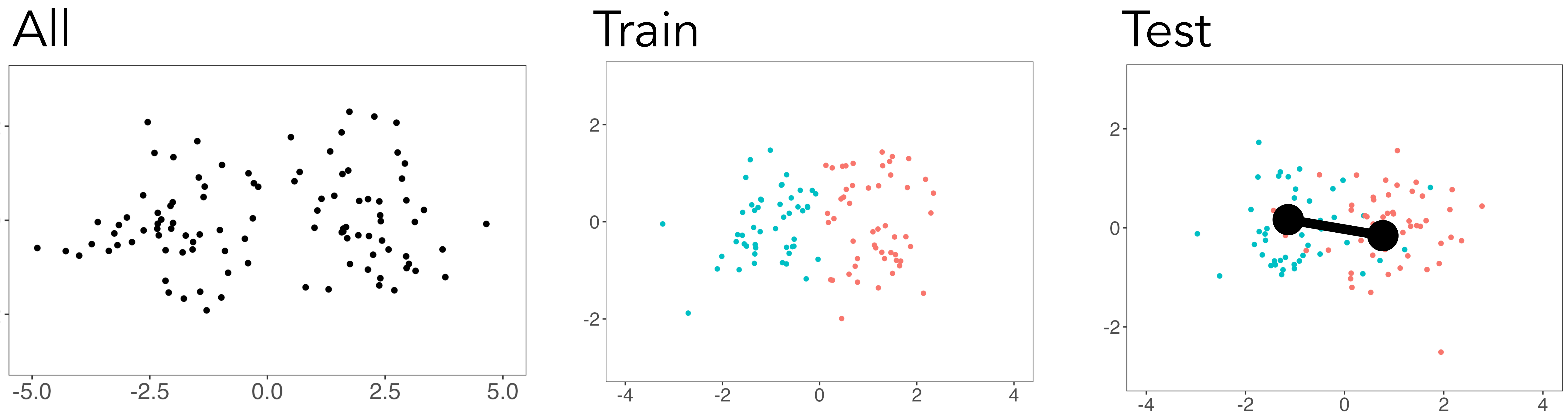


$$Pr_{H_0}\left(\left| \bar{\mathbf{X}}^{\text{test}}_{\hat{A}_{\text{train}}} - \bar{\mathbf{X}}^{\text{test}}_{\hat{B}_{\text{train}}} \right| \geq \left| \bar{X}^{\text{test}}_{\hat{A}_{\text{train}}} - \bar{X}^{\text{test}}_{\hat{B}_{train}} \right| \right)$$

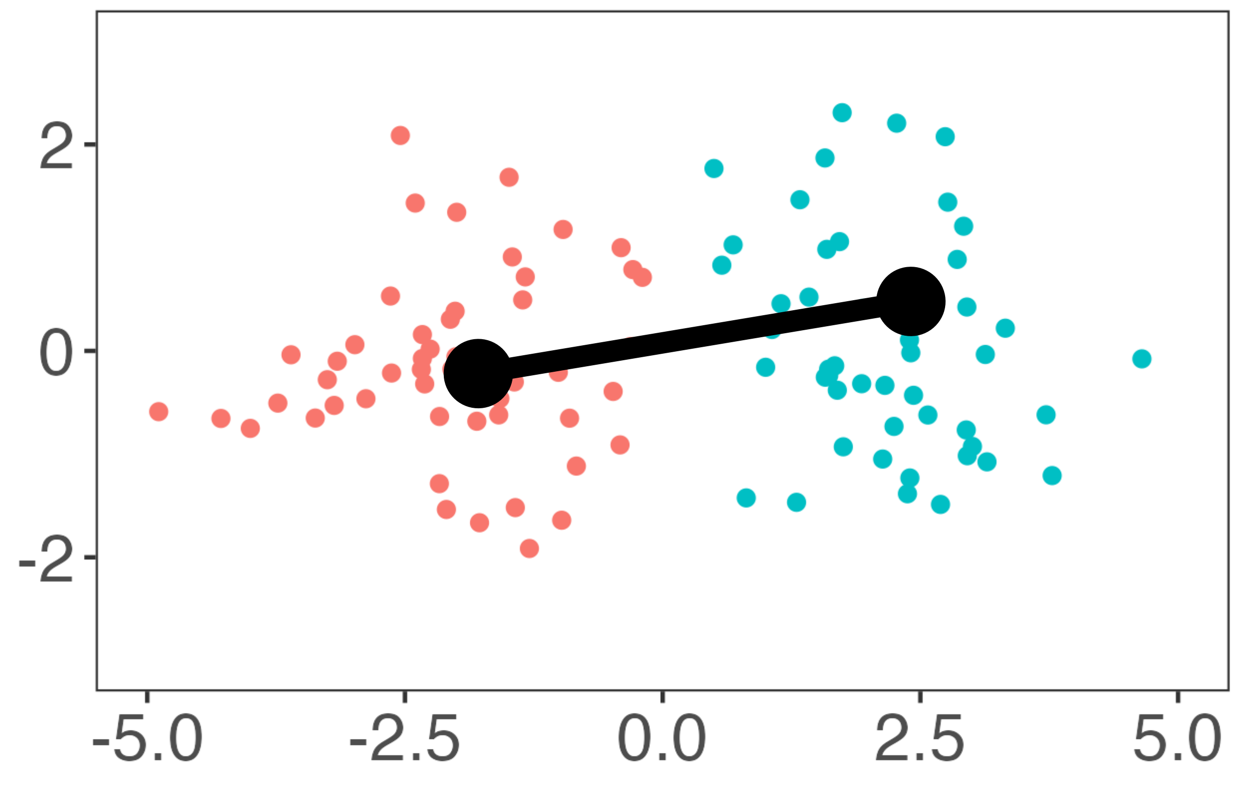
Chen and Witten, 2023, JMLR

Comparison to selective inference for overall difference in cluster means

Data thinning:



Selective Inference:

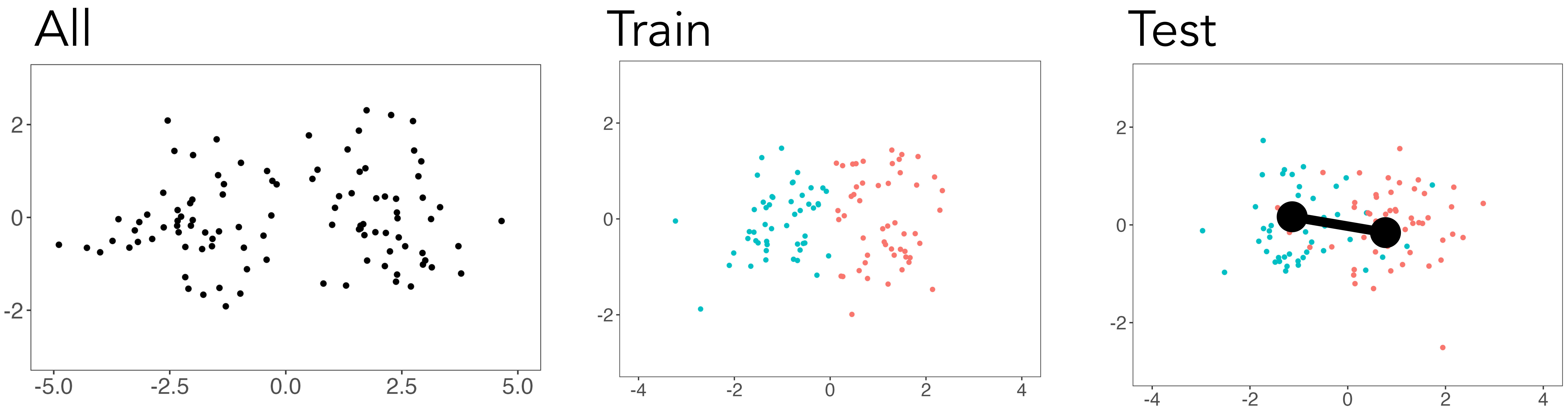


$$Pr_{H_0}\left(\left|\bar{\mathbf{X}}^{\text{test}}_{\hat{A}_{\text{train}}} - \bar{\mathbf{X}}^{\text{test}}_{\hat{B}_{\text{train}}}\right| \geq \left|\bar{X}^{\text{test}}_{\hat{A}_{\text{train}}} - \bar{X}^{\text{test}}_{\hat{B}_{train}}\right|\right)$$

Chen and Witten, 2023, JMLR

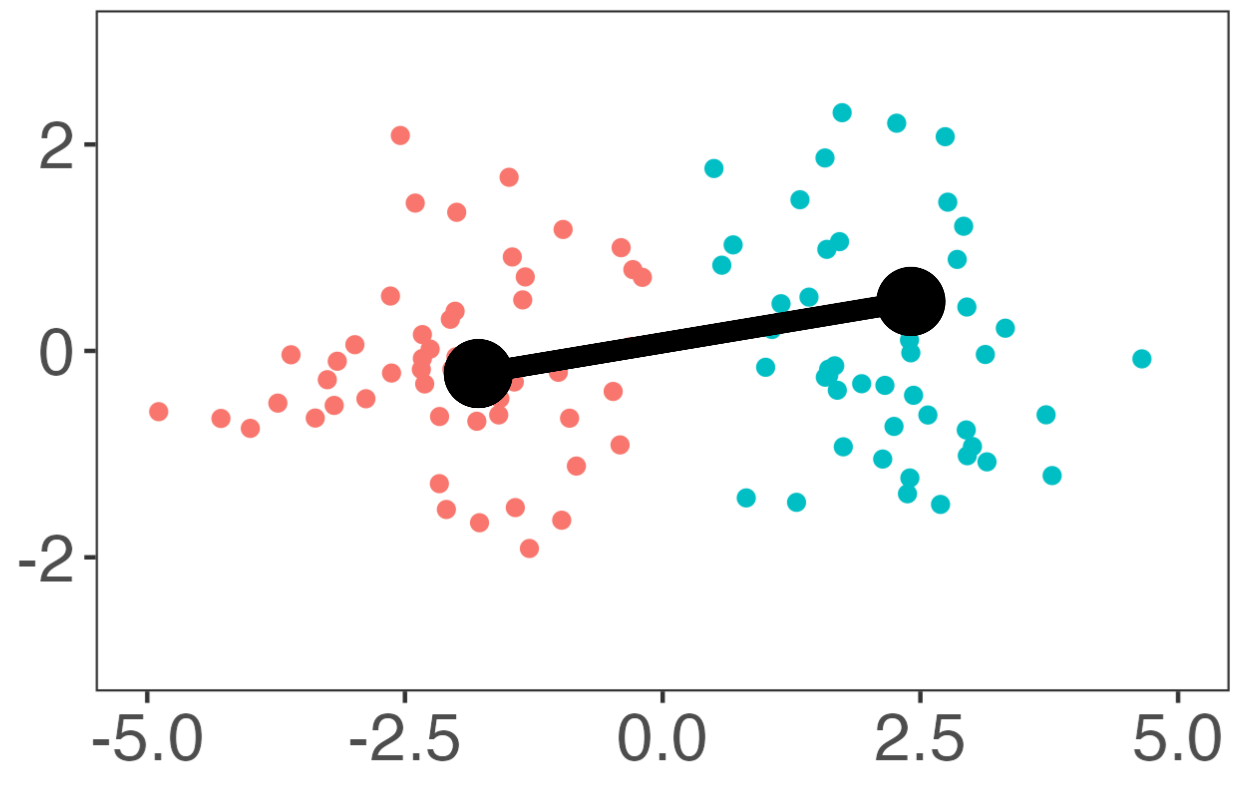
Comparison to selective inference for overall difference in cluster means

Data thinning:



$$Pr_{H_0}\left(\left|\bar{\mathbf{X}}^{\text{test}}_{\hat{A}_{\text{train}}} - \bar{\mathbf{X}}^{\text{test}}_{\hat{B}_{\text{train}}}\right| \geq \left|\bar{X}^{\text{test}}_{\hat{A}_{\text{train}}} - \bar{X}^{\text{test}}_{\hat{B}_{train}}\right|\right)$$

Selective Inference:

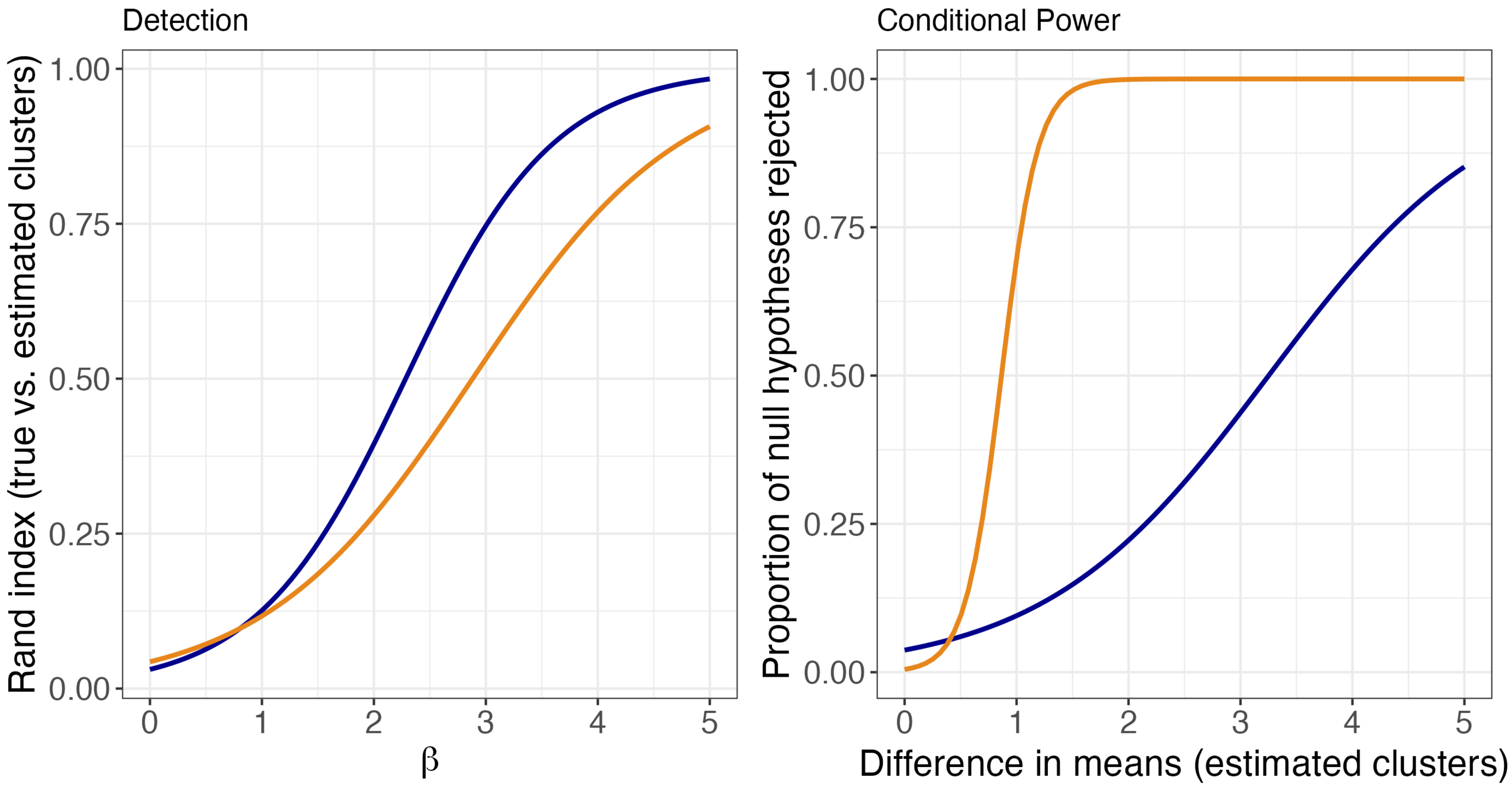


$$Pr_{H_0}\left(\left|\bar{\mathbf{X}}_{\hat{A}} - \bar{\mathbf{X}}_{\hat{B}}\right| \geq \left|\bar{X}_{\hat{A}} - \bar{X}_{\hat{B}}\right| \;\middle|\; \text{Clustering } \mathbf{X} \text{ results in clusters A and B}\right)$$

Chen and Witten, 2023, JMLR

Comparison to selective inference for overall difference in cluster means

$$X_{ij} \sim \begin{cases} N(0,1) & \text{if } j = 1,\ i \le 50 \\ N(\beta,1) & \text{if } j = 1,\ i > 50 \\ N(0,1) & \text{if } j = 2 \end{cases}$$



Chen and Witten, 2023, JMLR

Convolution-closed distributions

A family of distributions F_λ is "convolution-closed" in parameter λ if

- $X' \sim F_{\lambda_1}$
- $X'' \sim F_{\lambda_2}$
- $X' \perp\!\!\!\perp X''$

together imply that $X' + X'' \sim F_{\lambda_1+\lambda_2}$.

Joe, 1996, Journal of Applied Probability

Convolution-closed distributions

A family of distributions F_λ is "convolution-closed" in parameter λ if

- $X' \sim F_{\lambda_1}$
- $X'' \sim F_{\lambda_2}$
- $X' \perp\!\!\!\perp X''$

together imply that $X' + X'' \sim F_{\lambda_1+\lambda_2}$.

Distribution	Convolution-closed in:
$X \sim \mathrm{Poisson}(\lambda)$	λ
$X \sim \mathrm{N}(\mu, \sigma^2)$	(μ, σ^2)
$X \sim \mathrm{NegativeBinomial}(\mu, b)$	(μ, b)
$X \sim \mathrm{Gamma}(\alpha, \beta)$	α, if β is fixed
$X \sim \mathrm{Binomial}(r, p)$	r, if p is fixed
$X \sim \mathrm{N}_k\,(\mu, \Sigma)$.	(μ, Σ).
$X \sim \mathrm{Multinomial}_k\,(r, p)$	r, if p is fixed
$X \sim \mathrm{Wishart}_p\,(n, \Sigma)$	n, if p and Σ are fixed.

Joe, 1996, Journal of Applied Probability

Data thinning for convolution-closed distributions

Data thinning for convolution-closed distributions

$\boxed{\text{We observe realization } x \text{ from } X \sim F_\lambda.}$

Data thinning for convolution-closed distributions

We know x could have arisen as $x' + x''$, where
$$X' \sim F_{\epsilon\lambda}, \;\; X'' \sim F_{(1-\epsilon)\lambda}, \;\; X' \perp\!\!\!\perp X''.$$

$\Big\Downarrow$

We observe realization x from $X \sim F_\lambda$.

Data thinning for convolution-closed distributions

We know x could have arisen as $x' + x''$, where $X' \sim F_{\epsilon\lambda},\ X'' \sim F_{(1-\epsilon)\lambda},\ X' \perp\!\!\!\perp X''$.

$\Downarrow$

We observe realization x from $X \sim F_\lambda$.

If we had observed x' and x'', we would have satisfied our goal of data thinning!

Data thinning for convolution-closed distributions

We know x could have arisen as $x' + x''$, where $X' \sim F_{\epsilon\lambda}, \;\; X'' \sim F_{(1-\epsilon)\lambda}, \;\; X' \perp\!\!\!\perp X''$.

We observe realization x from $X \sim F_{\lambda}$.

If we had observed x' and x'', we would have satisfied our goal of data thinning!

Can we work backwards to recover x' and x''?

Data thinning for convolution-closed distributions

We know x could have arisen as $x' + x''$, where $X' \sim F_{\epsilon\lambda}$, $X'' \sim F_{(1-\epsilon)\lambda}$, $X' \perp\!\!\!\perp X''$.

$\Downarrow$

We observe realization x from $X \sim F_\lambda$.

If we had observed x' and x'', we would have satisfied our goal of data thinning!

Can we work backwards to recover x' and x''?

Let $G_{\epsilon,x}$ be the conditional distribution of $X' \mid X = x$.

Data thinning for convolution-closed distributions

We know x could have arisen as $x' + x''$, where $X' \sim F_{\epsilon\lambda}, \ X'' \sim F_{(1-\epsilon)\lambda}, \ X' \perp\!\!\!\perp X''$.

We observe realization x from $X \sim F_\lambda$.

Draw $X^{(1)}$ from $G_{\epsilon,x}$. Let $X^{(2)} := X - X^{(1)}$.

If we had observed x' and x'', we would have satisfied our goal of data thinning!

Can we work backwards to recover x' and x''?

Let $G_{\epsilon,x}$ be the conditional distribution of $X' \mid X = x$.

Data thinning for convolution-closed distributions

We know x could have arisen as $x' + x''$, where
$X' \sim F_{\epsilon\lambda}, \;\; X'' \sim F_{(1-\epsilon)\lambda}, \;\; X' \perp\!\!\!\perp X''$.

We observe realization x from $X \sim F_\lambda$.

Draw $X^{(1)}$ from $G_{\epsilon,x}$. Let $X^{(2)} := X - X^{(1)}$.

Theorem:
$X^{(1)} \sim F_{\epsilon\lambda}, X^{(2)} \sim F_{(1-\epsilon)\lambda}, \;\; X^{(1)} \perp\!\!\!\perp X^{(2)}$.

If we had observed x' and x'', we would have satisfied our goal of data thinning!

Can we work backwards to recover x' and x''?

Let $G_{\epsilon,x}$ be the conditional distribution of $X' \mid X = x$.