Data thinning to avoid double dipping

Anna Neufeld

BIRS Workshop

February, 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:

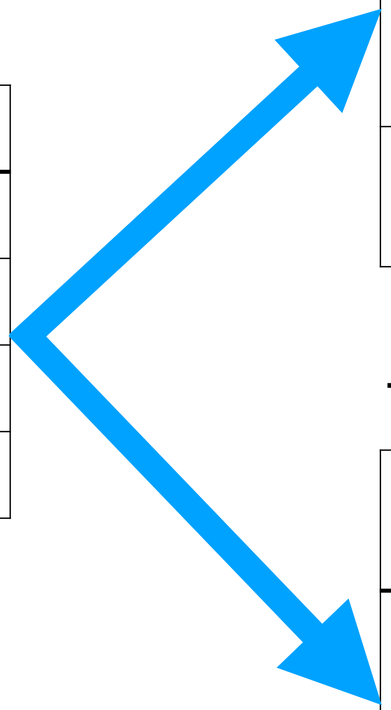
1. Generating and testing a null hypothesis.
2. Fitting and evaluating a model.

One approach: avoid double dipping entirely via sample splitting

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

One approach: avoid double dipping entirely via sample splitting

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



Train

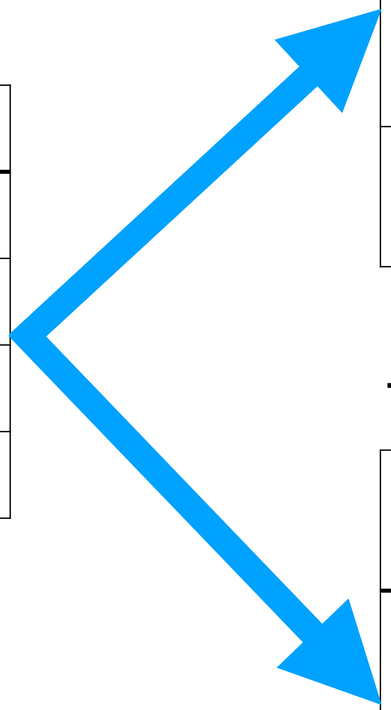
	Feature 1	Feature 2
Obs. 1	12	6
Obs. 2	31	8

Test

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

One approach: avoid double dipping entirely via 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

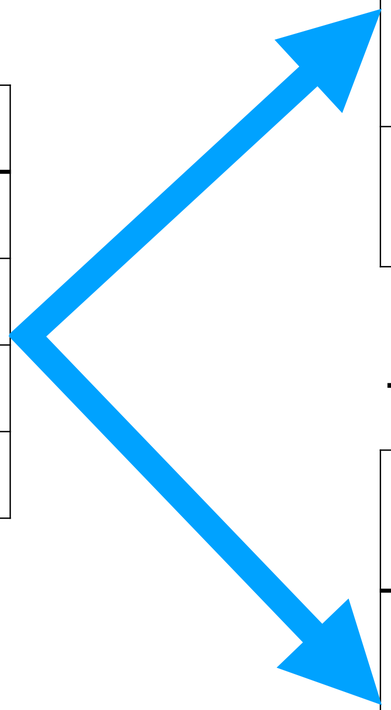
Test

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

Select hypothesis.

One approach: avoid double dipping entirely via 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.

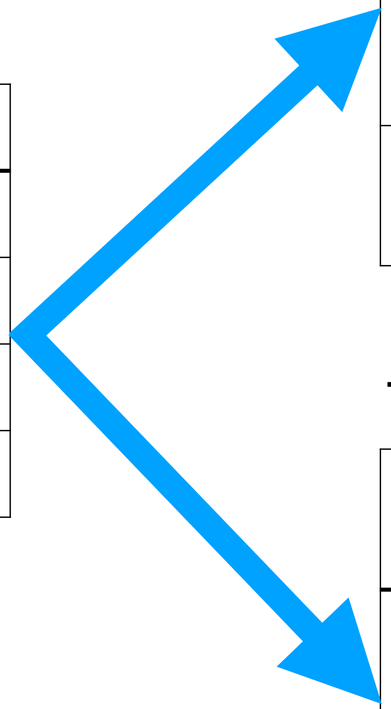
Test

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

Test hypothesis.

One approach: avoid double dipping entirely via 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.

Fit model.

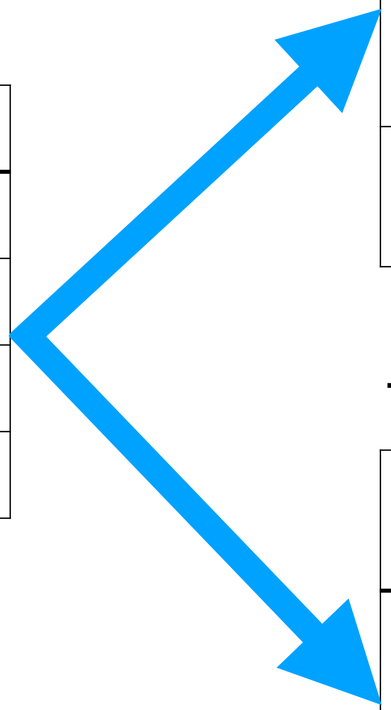
Test

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

Test hypothesis.

One approach: avoid double dipping entirely via 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.

Fit model.

Test

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

Test hypothesis.

Evaluate model.

Outline

1. **Motivation: settings where sample splitting doesn't work**
2. Poisson thinning
3. Data thinning
4. Application to single-cell RNA sequencing data
5. Ongoing work

Motivating example: identifying differentially expressed genes in scRNA-seq data

scRNA-seq dataset

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

Motivating example: identifying differentially expressed genes in scRNA-seq data

scRNA-seq dataset

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

Question: Which genes are differentially expressed across cell type?

Motivating example: identifying differentially expressed genes in scRNA-seq data

scRNA-seq dataset

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

Question: Which genes are differentially expressed across cell type?

Problem: Cell type is unobserved, and must be estimated via clustering.

Motivating example: identifying differentially expressed genes in scRNA-seq data

scRNA-seq dataset

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

Question: Which genes are differentially expressed across cell type?

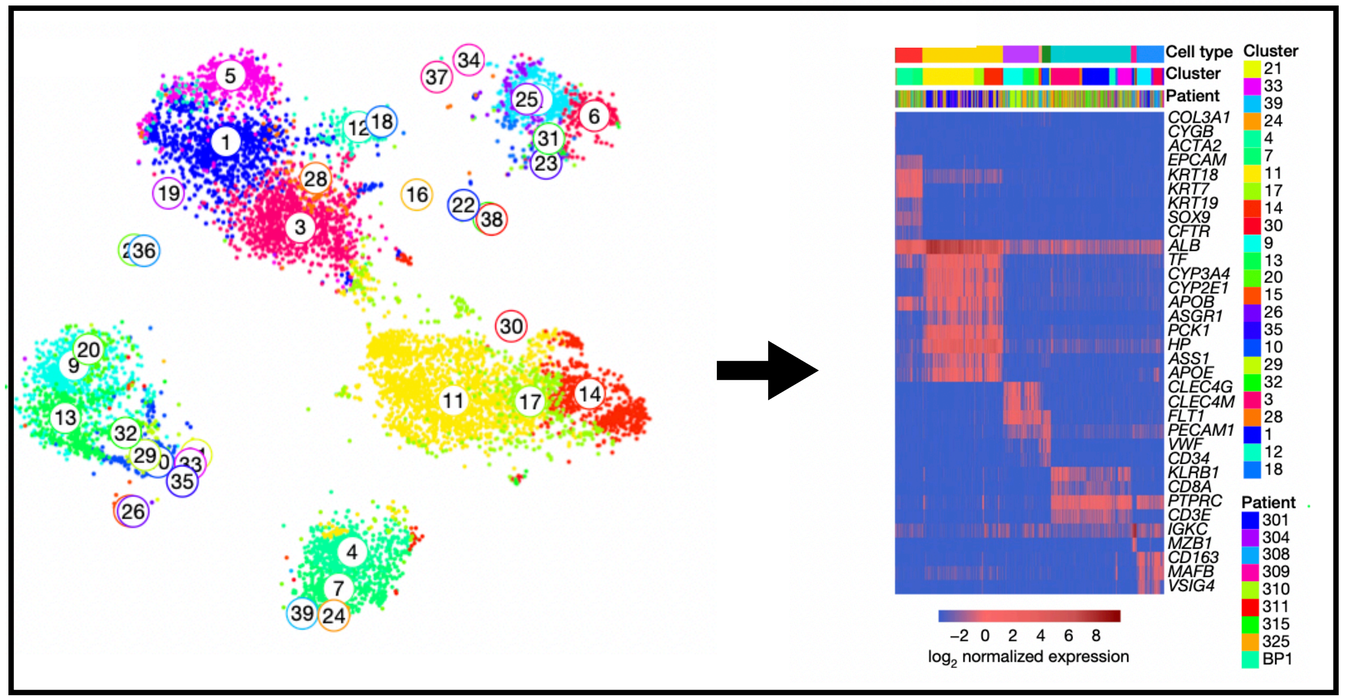
Problem: Cell type is unobserved, and must be estimated via clustering.

A human liver cell atlas reveals heterogeneity and epithelial progenitors

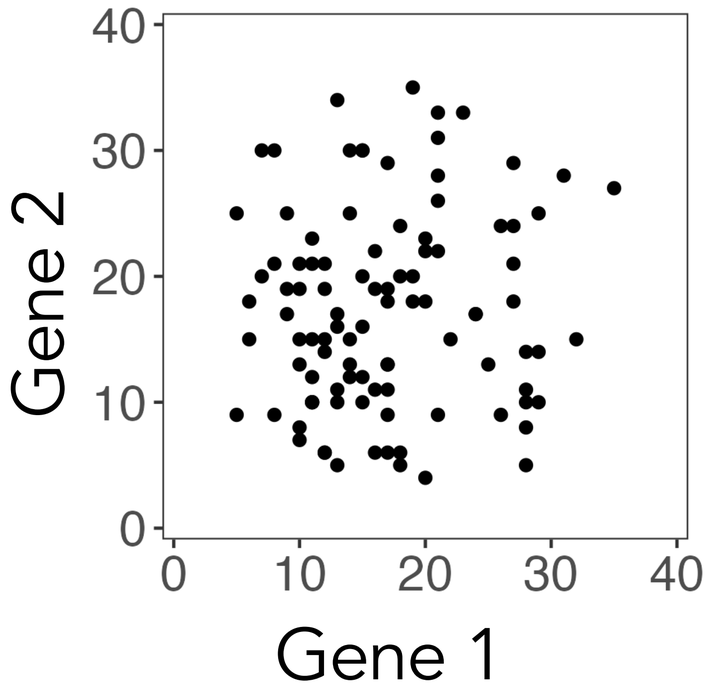
Nadim Aizarani, Antonio Saviano, Sagar, Laurent Mailly, Sarah Durand, Josip S. Herman, Patrick Pessaux, Thomas F. Baumert & Dominic Grün

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

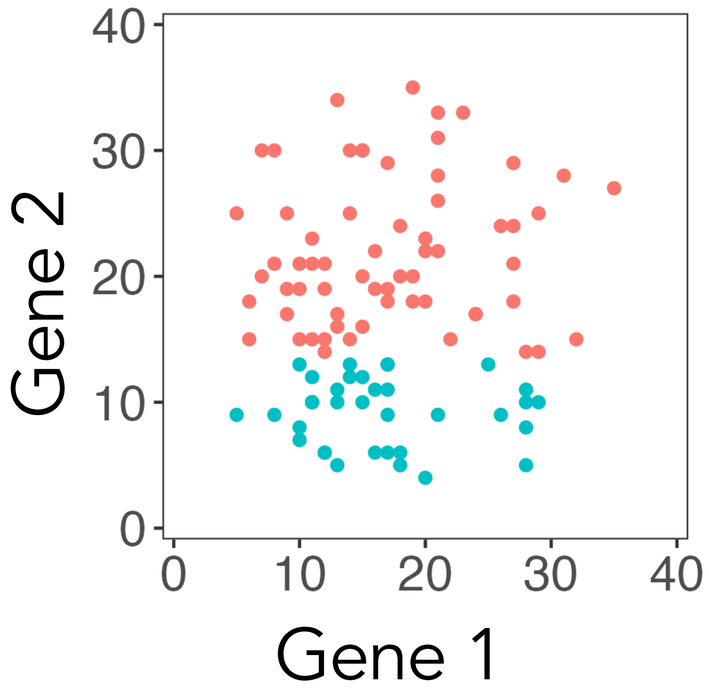
64k Accesses | **284** Citations | **321** Altmetric | Metrics



Naively testing for a difference in means across estimated clusters is an example of double dipping

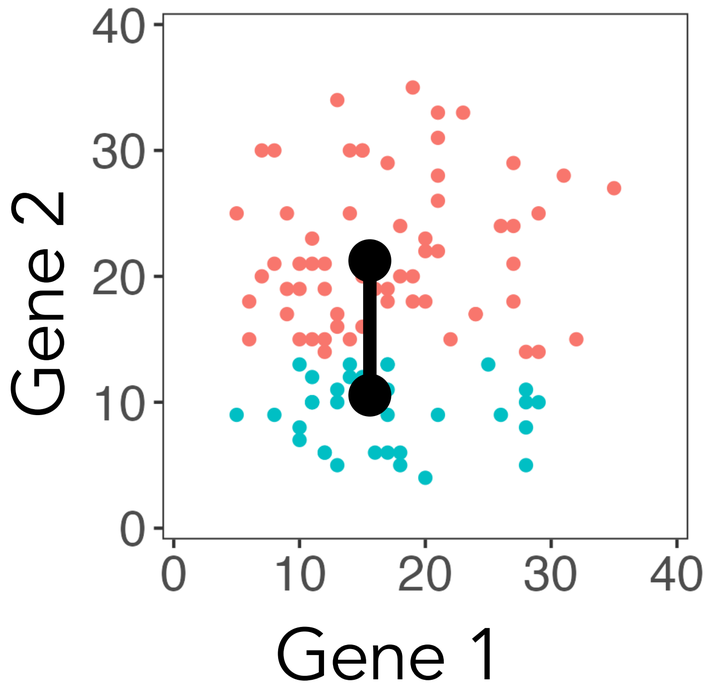


Naively testing for a difference in means across estimated clusters is an example of double dipping



Step 1: cluster the observations.

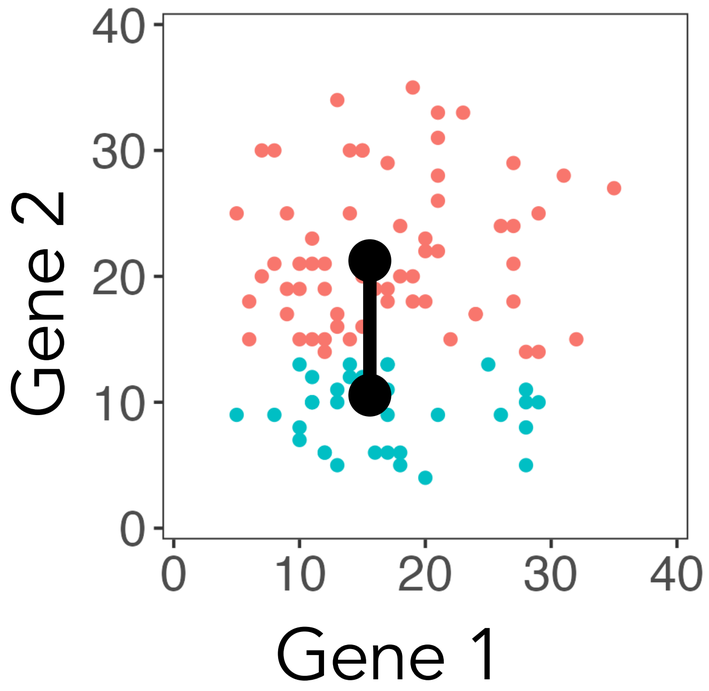
Naively testing for a difference in means across estimated clusters is an example of double dipping



Step 1: cluster the observations.

Generate H_0 :"the expected value of Gene 2 is the same between red cells and the blue cells."

Naively testing for a difference in means across estimated clusters is an example of double dipping

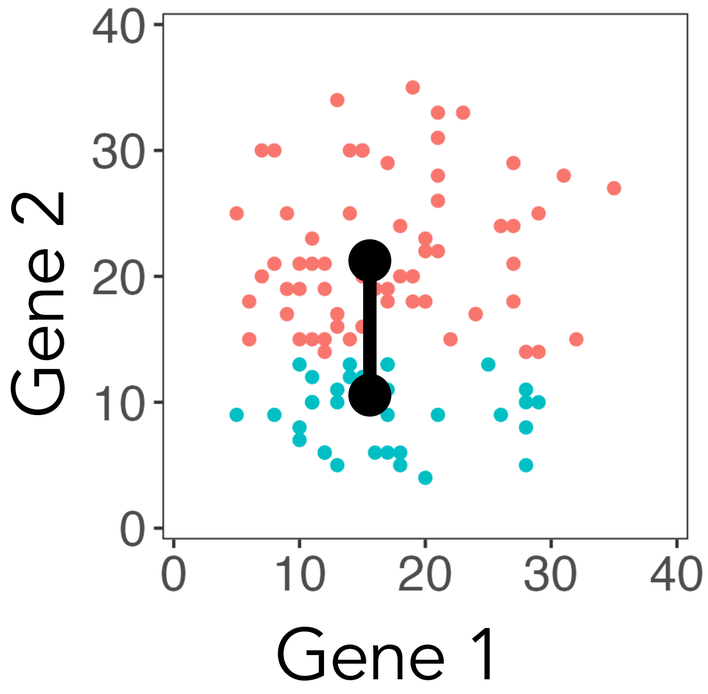


Step 1: cluster the observations.

Generate H_0 :"the expected value of Gene 2 is the same between red cells and the blue cells."

Step 2: test H_0 with a t-test.

Naively testing for a difference in means across estimated clusters is an example of double dipping



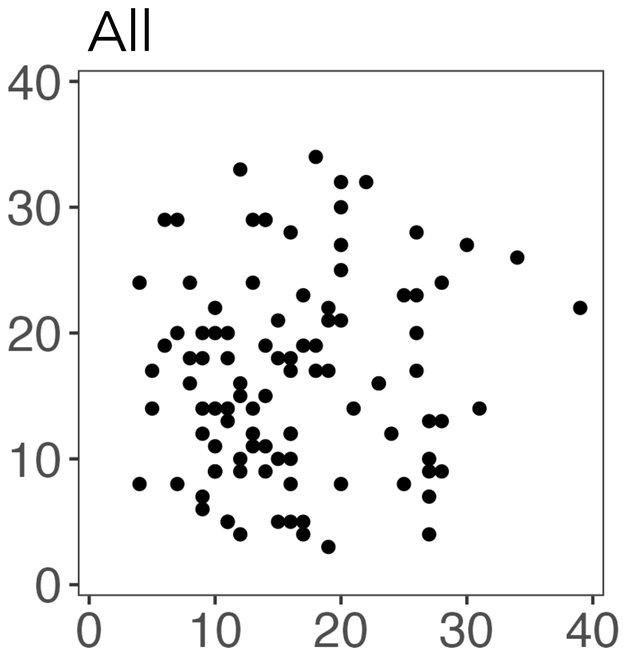
$p < 10^{-10}$ 😱

Step 1: cluster the observations.

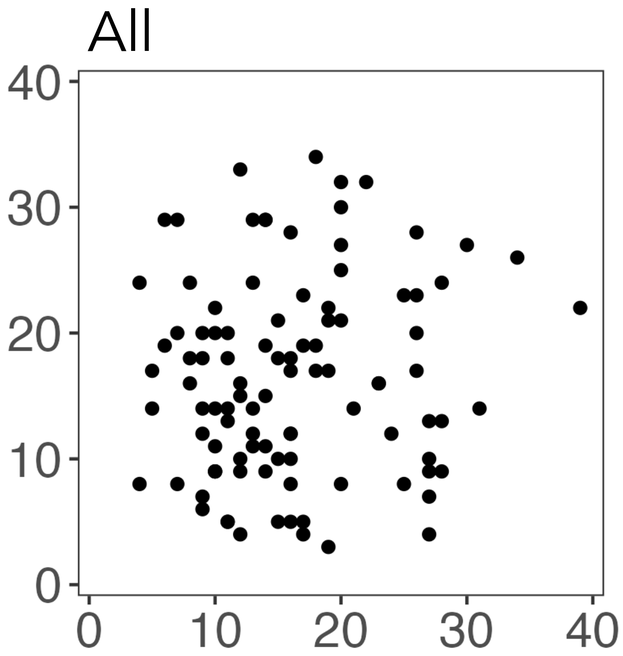
Generate H_0 :"the expected value of Gene 2 is the same between red cells and the blue cells."

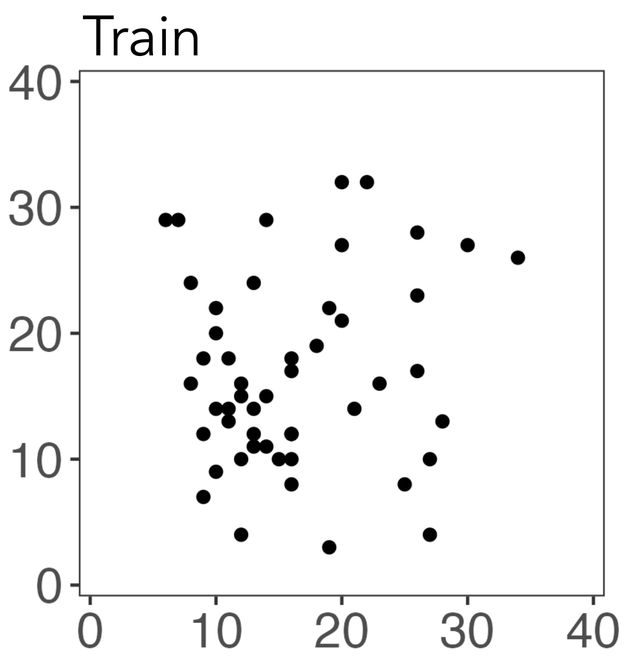
Step 2: test H_0 with a t-test.

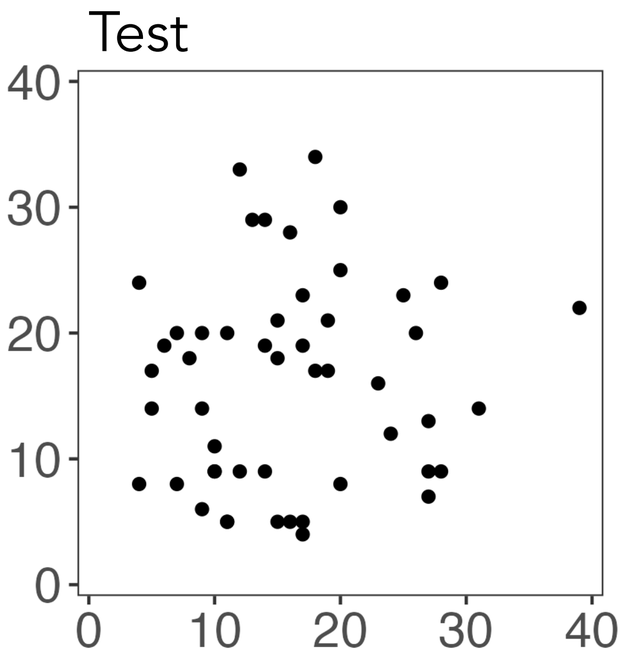
Sample splitting cannot be used in this context



Sample splitting cannot be used in this context

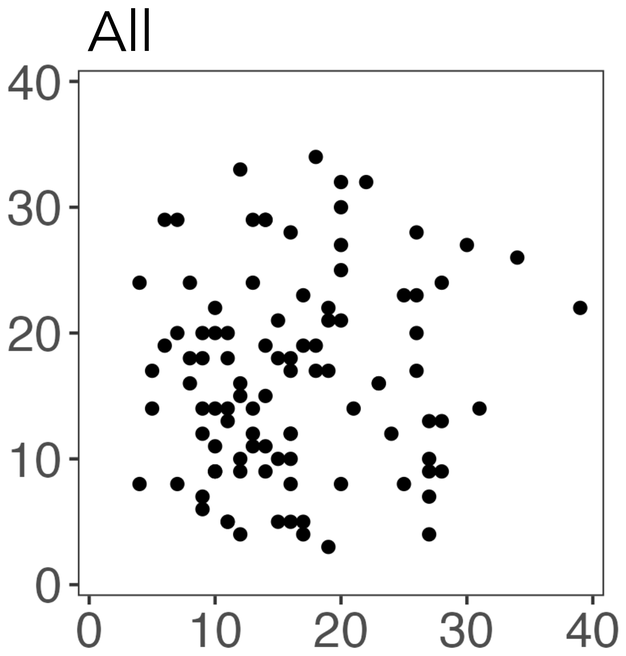


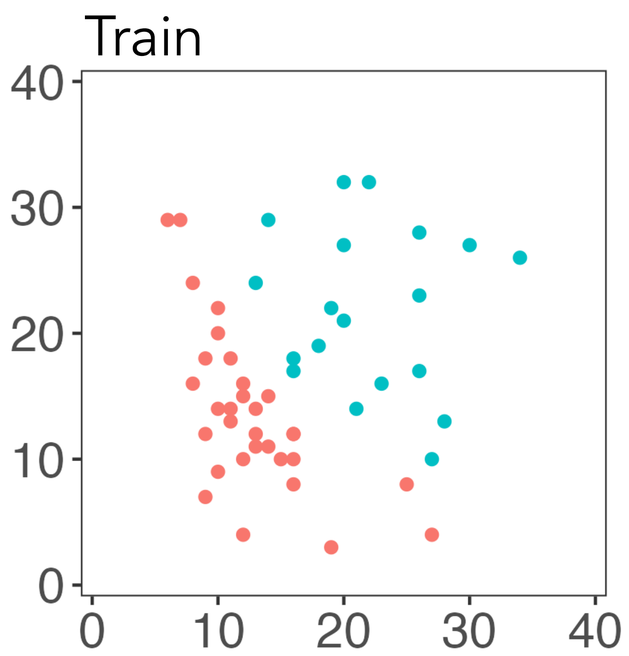


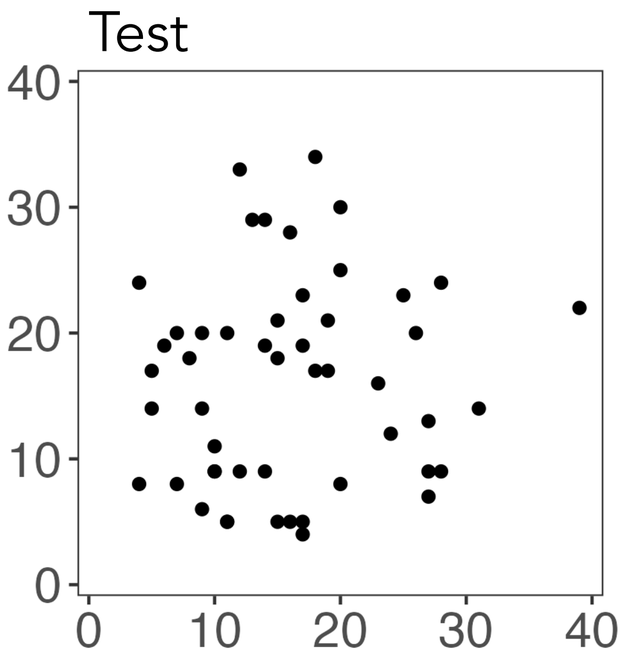


Step 1: split observations into train/test.

Sample splitting cannot be used in this context



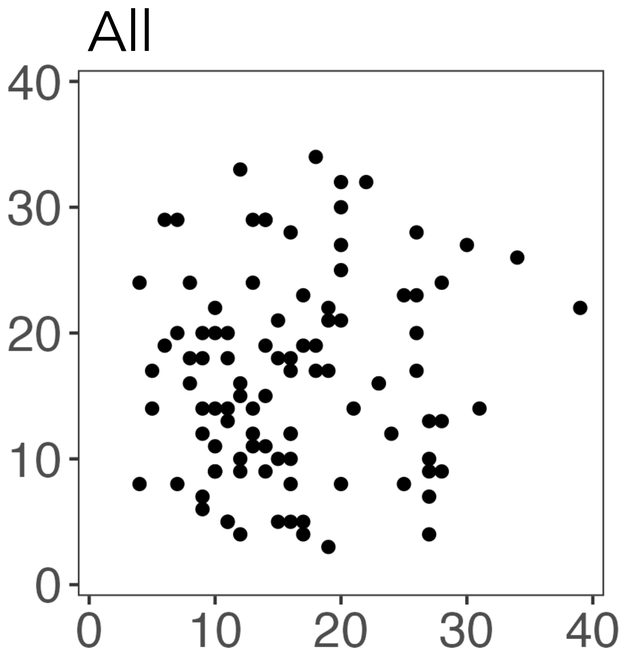


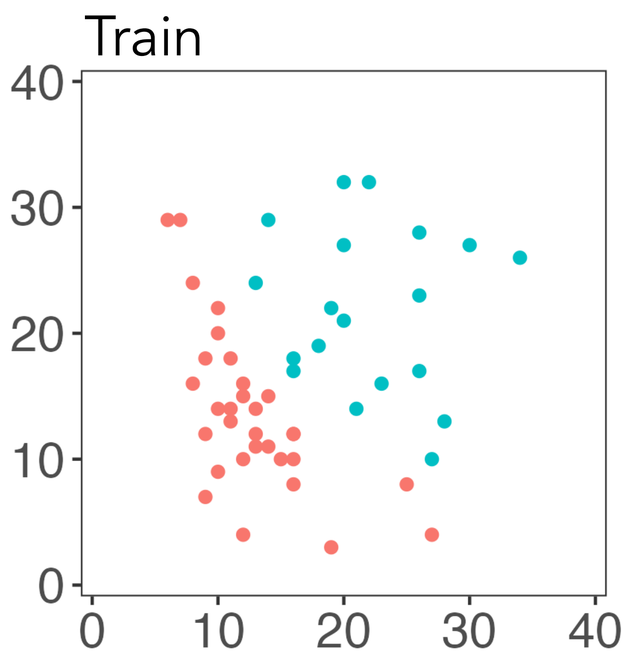


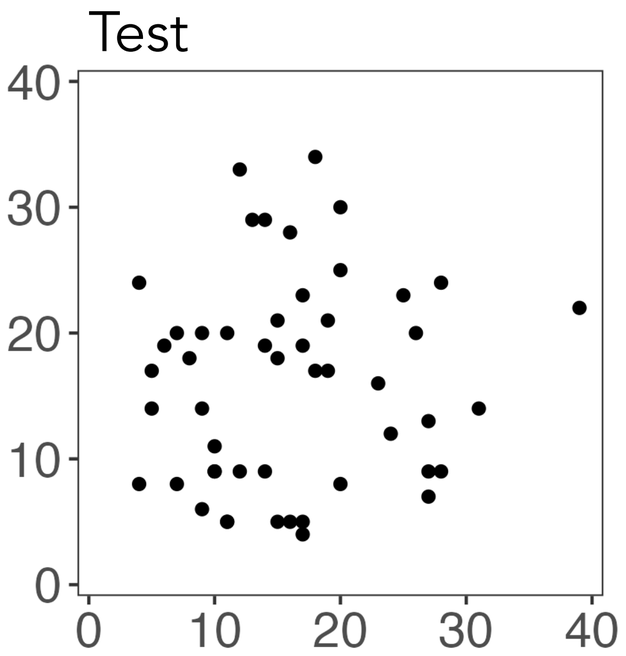
Step 1: split observations into train/test.

Step 2: cluster the training set.

Sample splitting cannot be used in this context





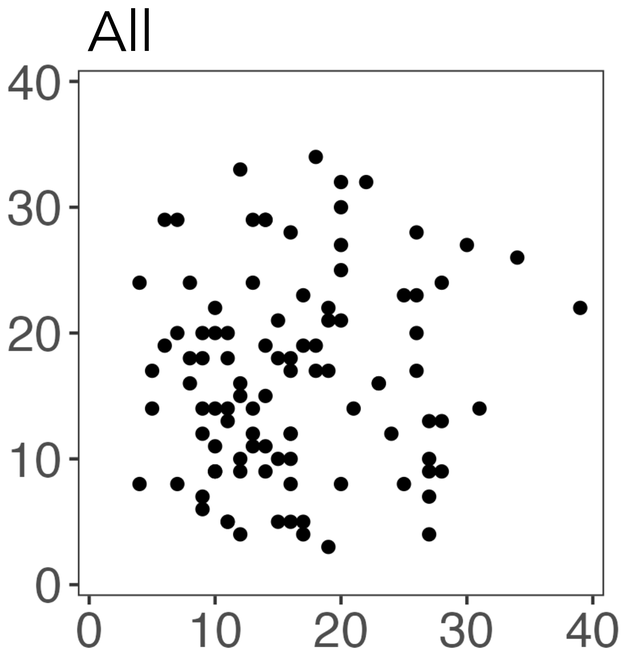


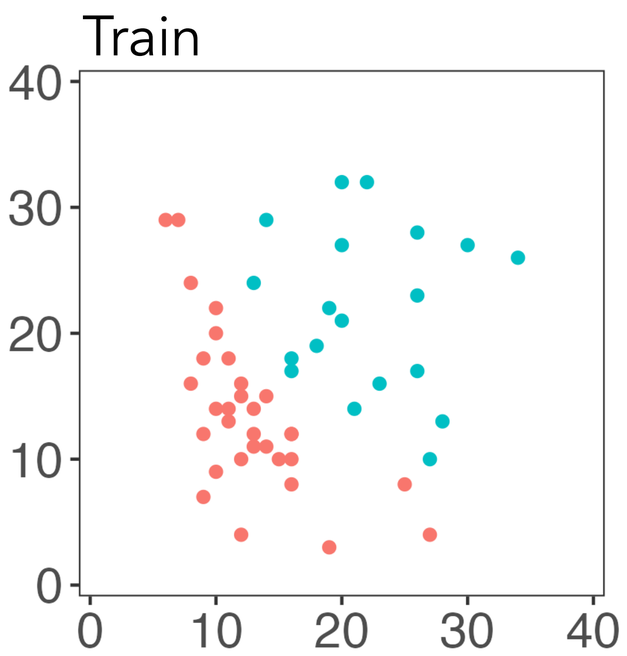
Step 1: split observations into train/test.

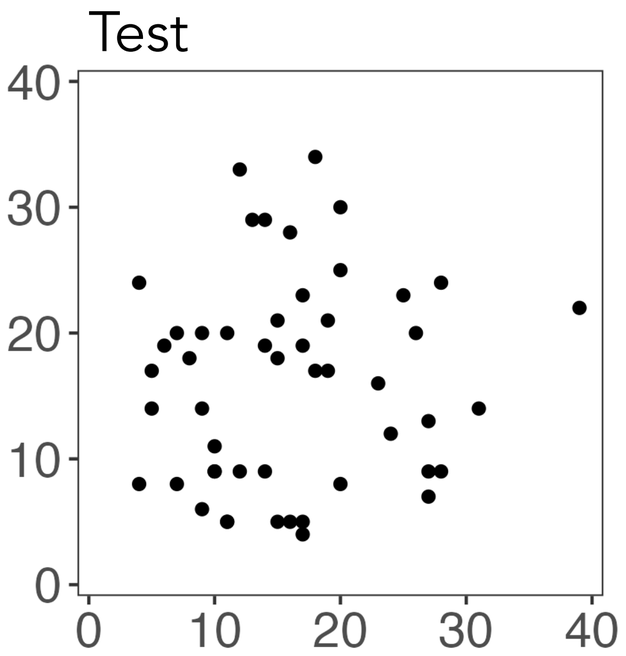
Step 2: cluster the training set.

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

Sample splitting cannot be used in this context







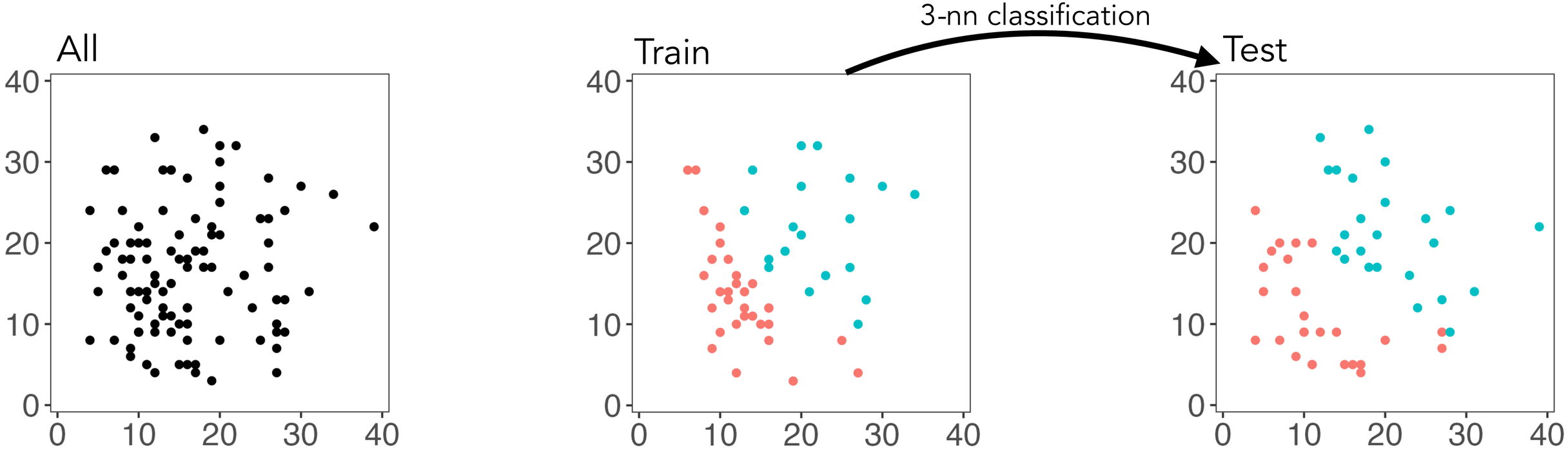
Step 1: split observations into train/test.

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 in this context



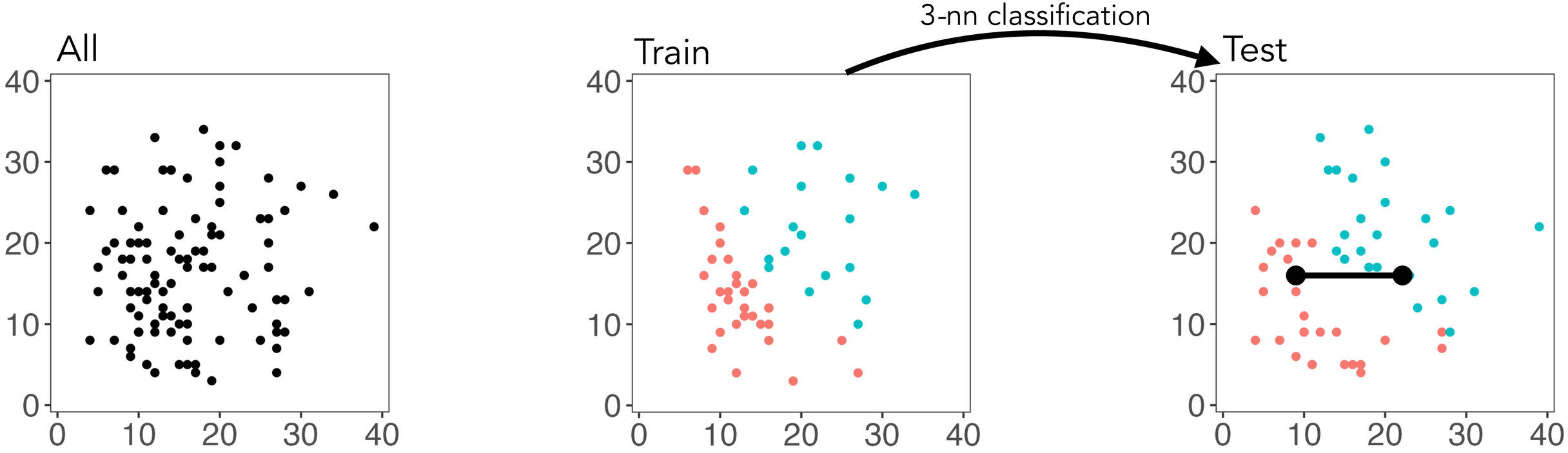
Step 1: split observations into train/test.

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 in this context



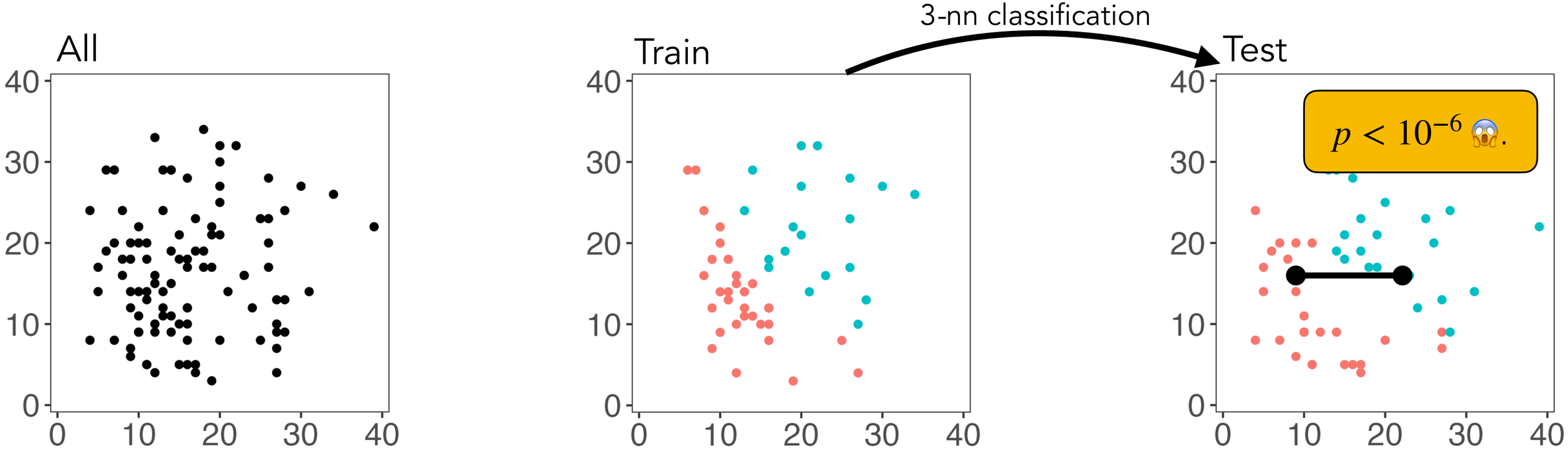
Step 1: split observations into train/test.

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 in this context



Step 1: split observations into train/test.

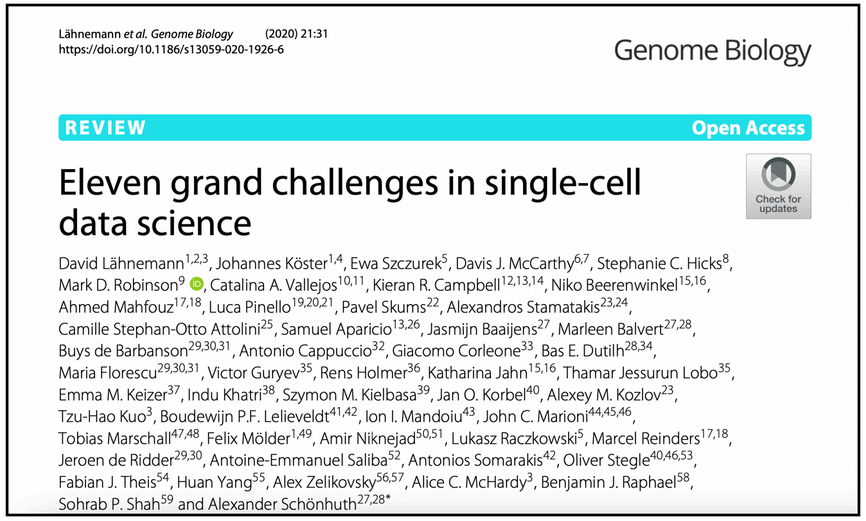
Step 2: cluster the training set.

Step 2.5: assign labels to observations in test set.

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

Gao, Bien, and Witten, 2022 (JASA).

Common scRNA-seq analysis pipelines ignore the double dipping problem



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

Genome Biology

REVIEW Open Access

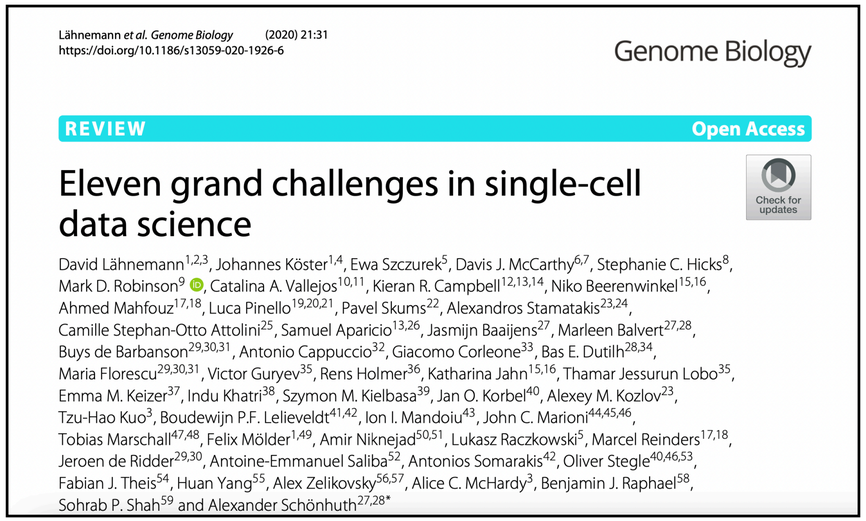
Eleven grand challenges in single-cell data science

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 Balvert[27,28], 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 Jessurun Lobo[35], Emma M. Keizer[37], Indu Khatri[38], Szymon M. Kielbasa[39], Jan O. Korbel[40], Alexey M. Kozlov[23], 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 Reinders[17,18], Jeroen de Ridder[29,30], Antoine-Emmanuel Saliba[52], Antonios Somarakis[42], Oliver Stegle[40,46,53], Fabian J. Theis[54], Huan Yang[55], Alex Zelikovsky[56,57], Alice C. McHardy[3], Benjamin J. Raphael[58], 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).

Common scRNA-seq analysis pipelines ignore the double dipping problem



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

Genome Biology

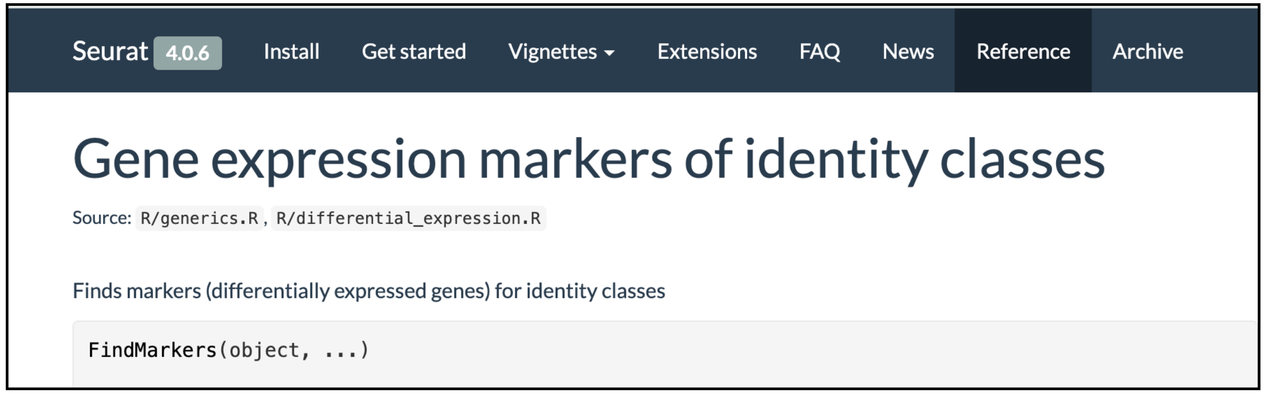
REVIEW Open Access

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 Balvert[27,28], 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 Jessurun Lobo[35], Emma M. Keizer[37], Indu Khatri[38], Szymon M. Kielbasa[39], Jan O. Korbel[40], Alexey M. Kozlov[23], 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 Reinders[17,18], Jeroen de Ridder[29,30], Antoine-Emmanuel Saliba[52], Antonios Somarakis[42], Oliver Stegle[40,46,53], Fabian J. Theis[54], Huan Yang[55], Alex Zelikovsky[56,57], Alice C. McHardy[3], Benjamin J. Raphael[58], 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).



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.

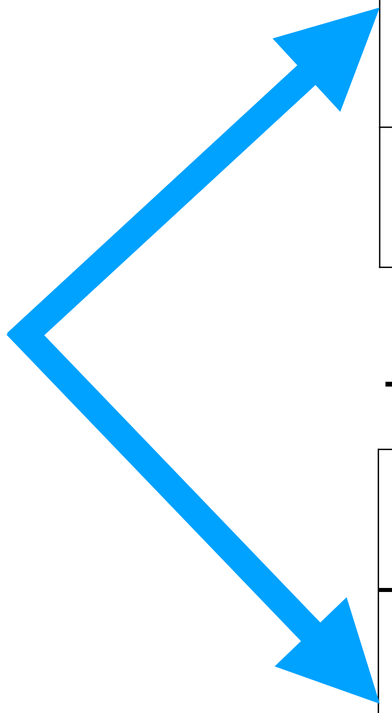
Outline

1. Motivation: settings where sample splitting doesn't work
2. **Poisson thinning**
3. Data thinning
4. Application to single-cell RNA sequencing data
5. Ongoing work

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

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

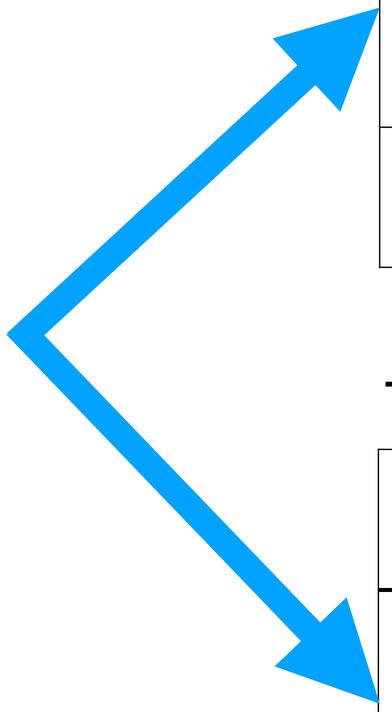
Test

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

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

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

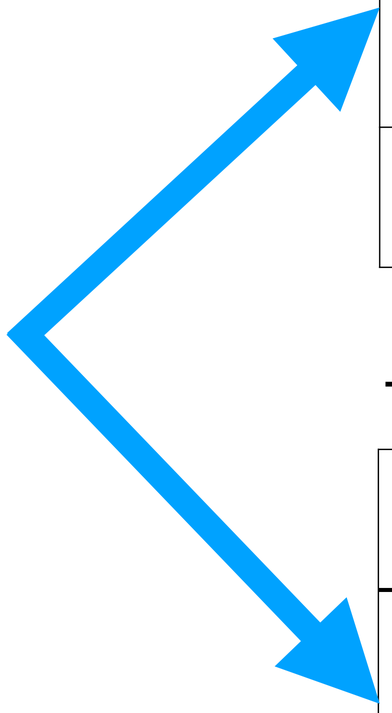
	Gene 1	Gene 2
Cell 3	11	5
Cell 4	22	21

Estimating clusters on training set

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

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	8
Cell 3	11	31
Cell 4	22	34

X_{ij}

$X^{(1)}$

	Gene 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

An alternative: Poisson thinning

X

	Gene 1	Gene 2
Cell 1	18	
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
	14	1
		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		5
ell 2	21	2
Cell 3	6	14
Cell 4	16	9

An alternative: Poisson thinning

X

	Gene 1	Gene 2
Cell 1	18	
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
	14	1
		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		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	
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
	14	1
		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
ell 1		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	
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
	14	1
		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
ell 1		5
ell 2	21	2
Cell 3	6	14
Cell 4	16	9

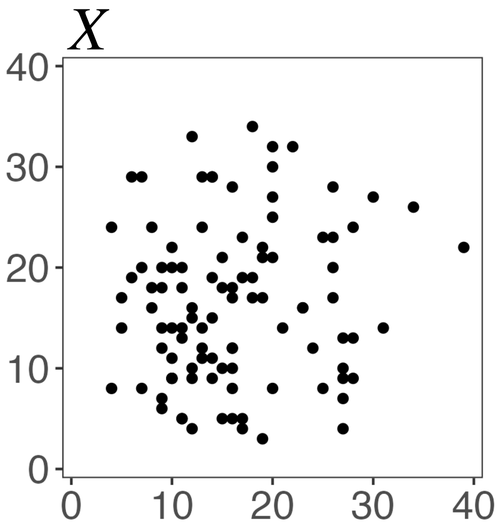
Test for difference in means.

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

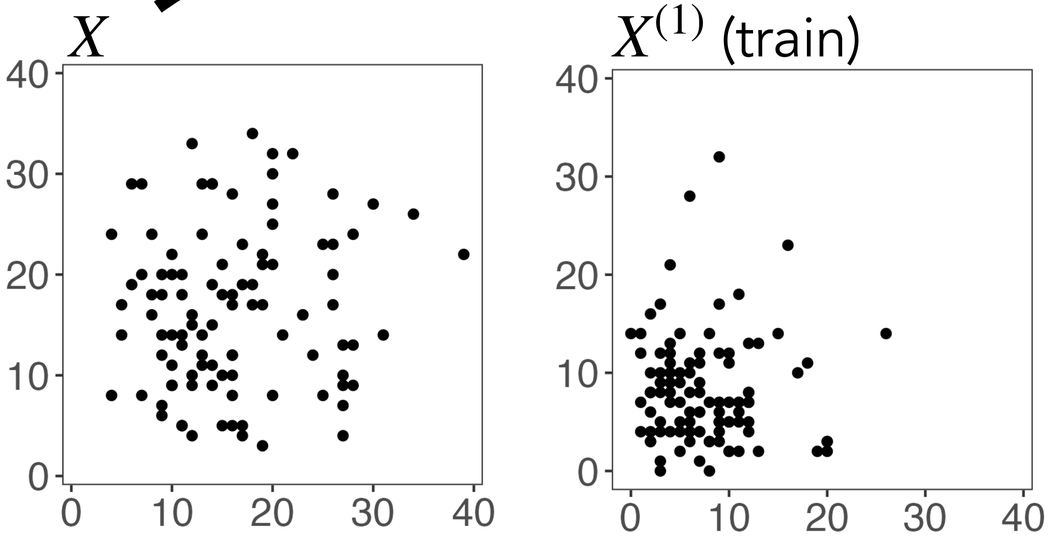
A very well-known result.

Thinning avoids the pitfall of sample splitting on our motivating examples



Thinning avoids the pitfall of sample splitting on our motivating examples

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

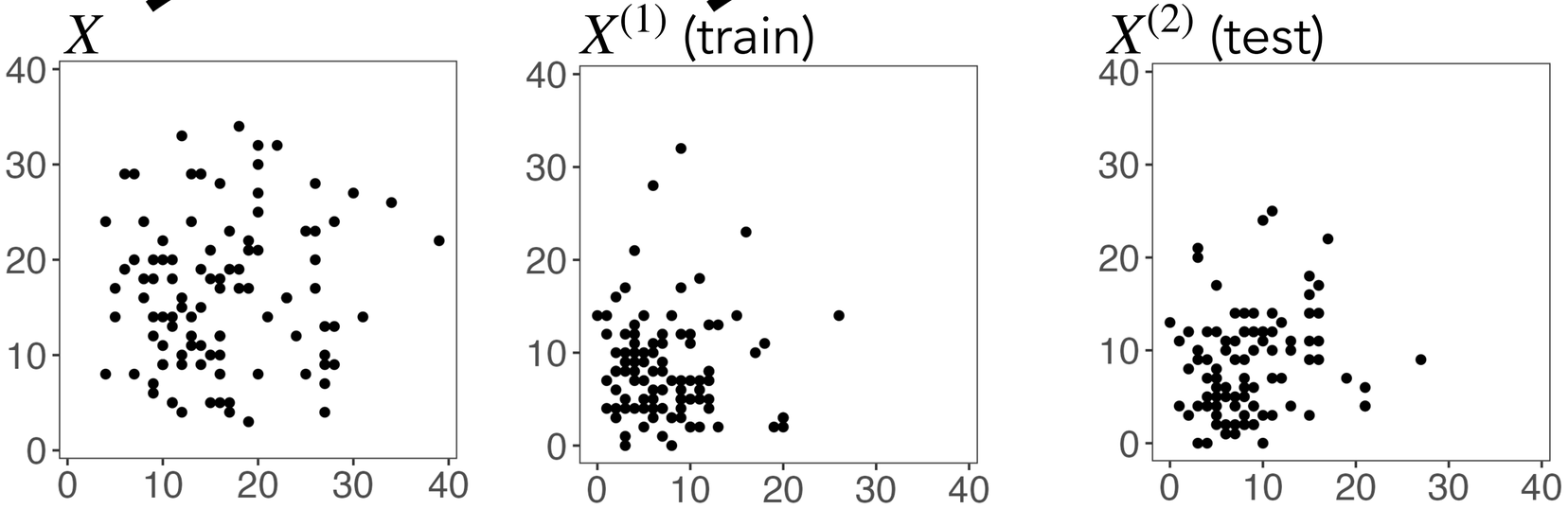


Step 1: thin observations into train/test.

Thinning avoids the pitfall of sample splitting on our motivating examples

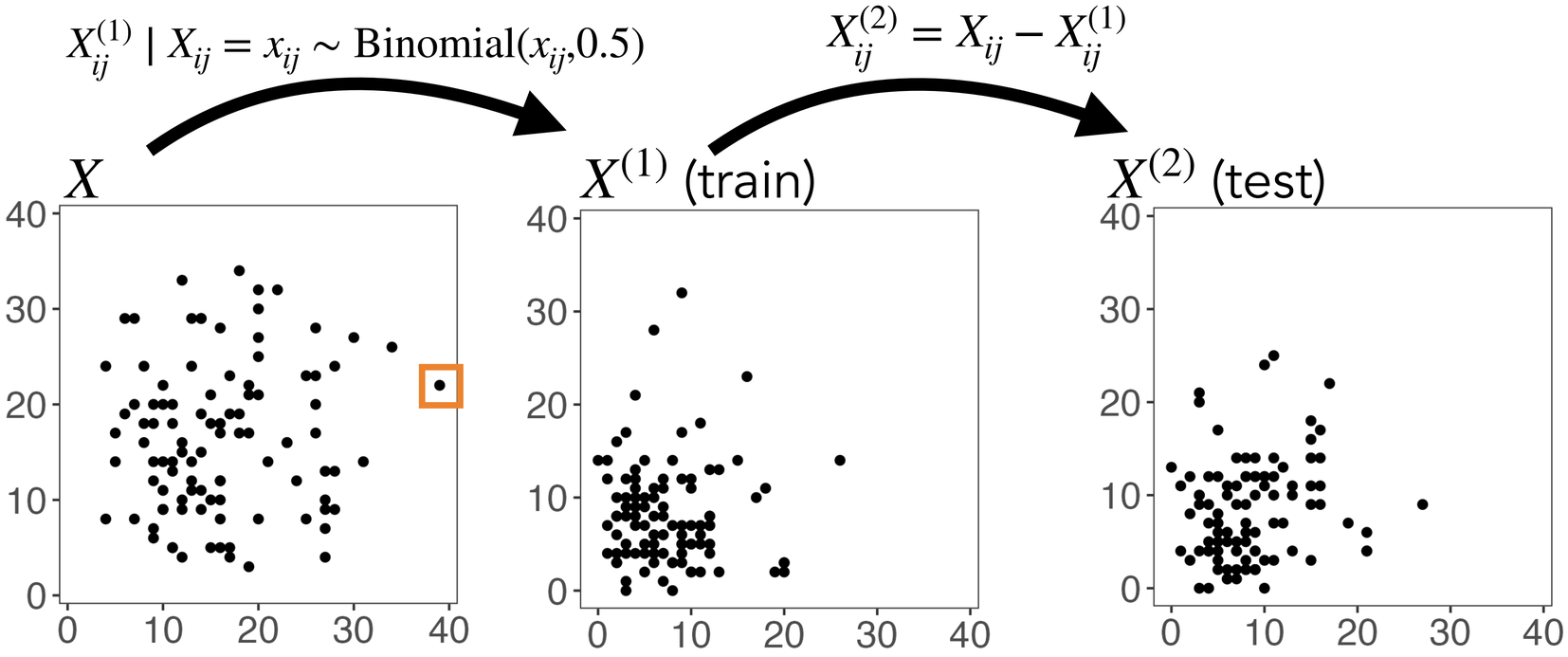
$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5)$

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



Step 1: thin observations into train/test.

Thinning avoids the pitfall of sample splitting on our motivating examples

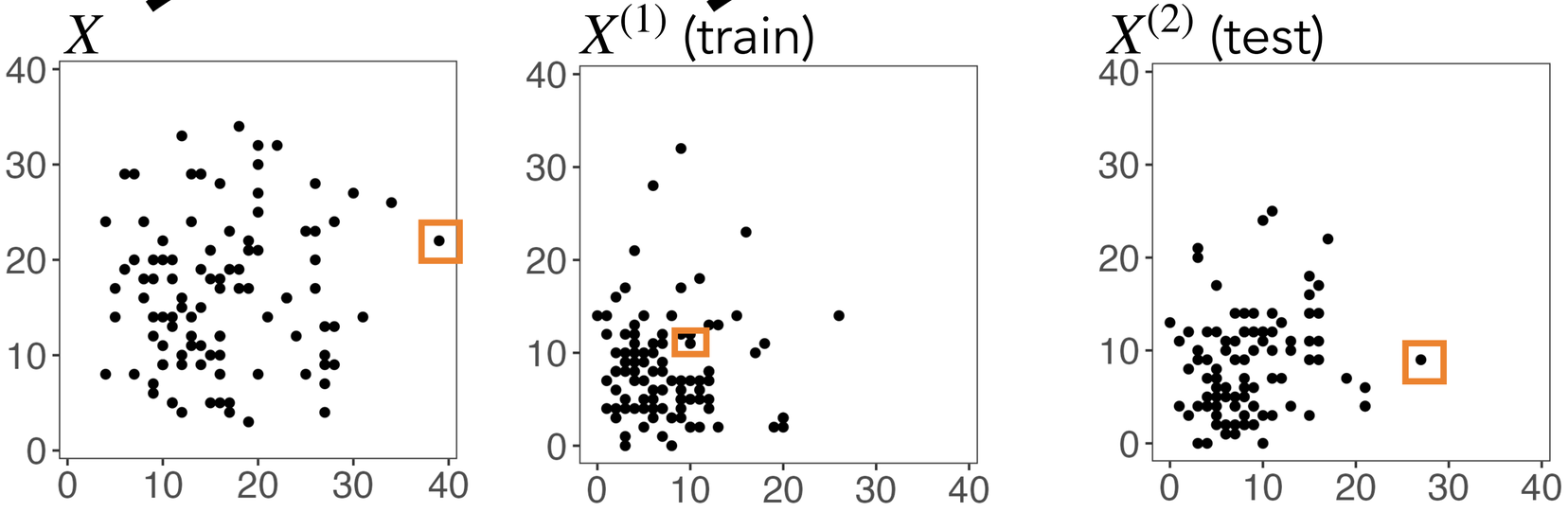


Step 1: thin observations into train/test.

Thinning avoids the pitfall of sample splitting on our motivating examples

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

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

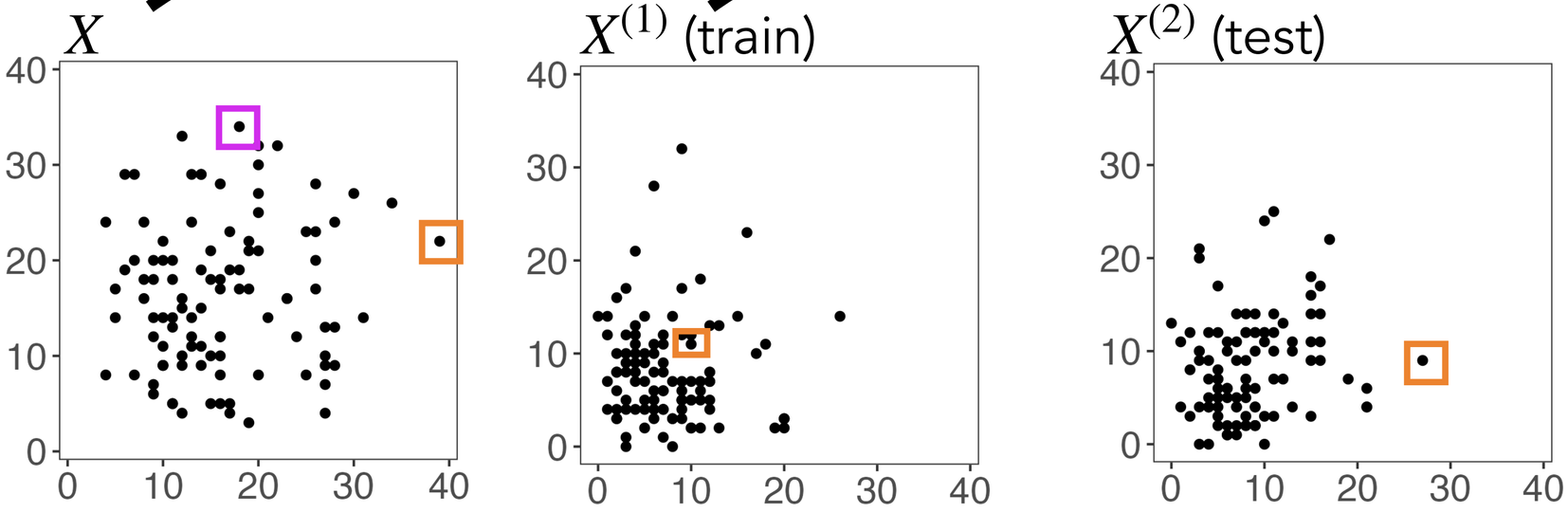


Step 1: thin observations into train/test.

Thinning avoids the pitfall of sample splitting on our motivating examples

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

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

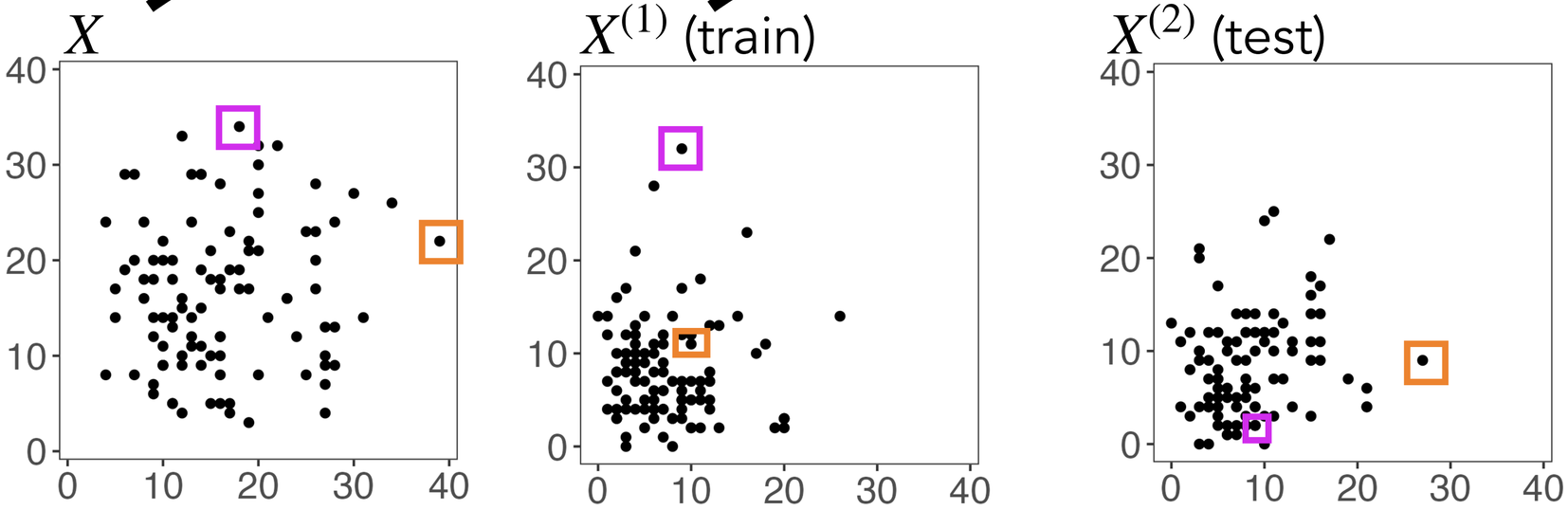


Step 1: thin observations into train/test.

Thinning avoids the pitfall of sample splitting on our motivating examples

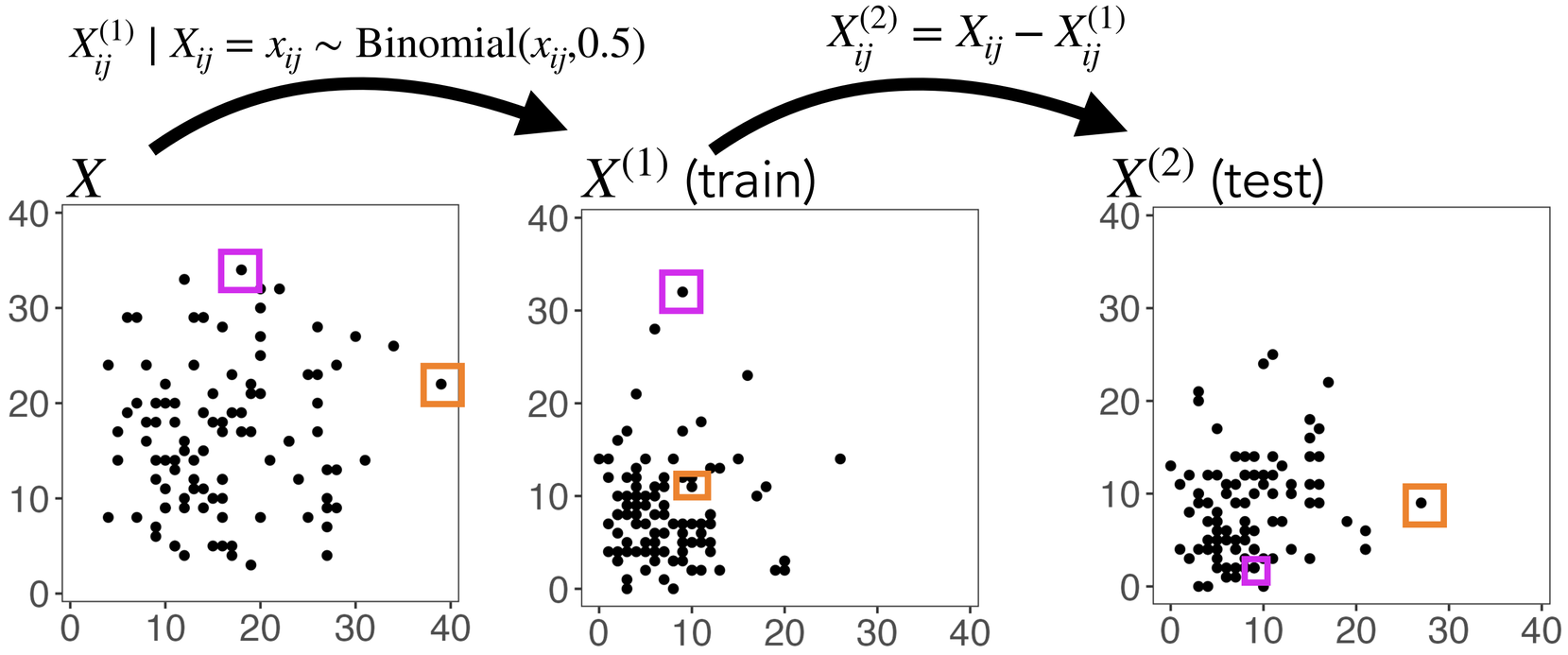
$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5)$

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



Step 1: thin observations into train/test.

Thinning avoids the pitfall of sample splitting on our motivating examples



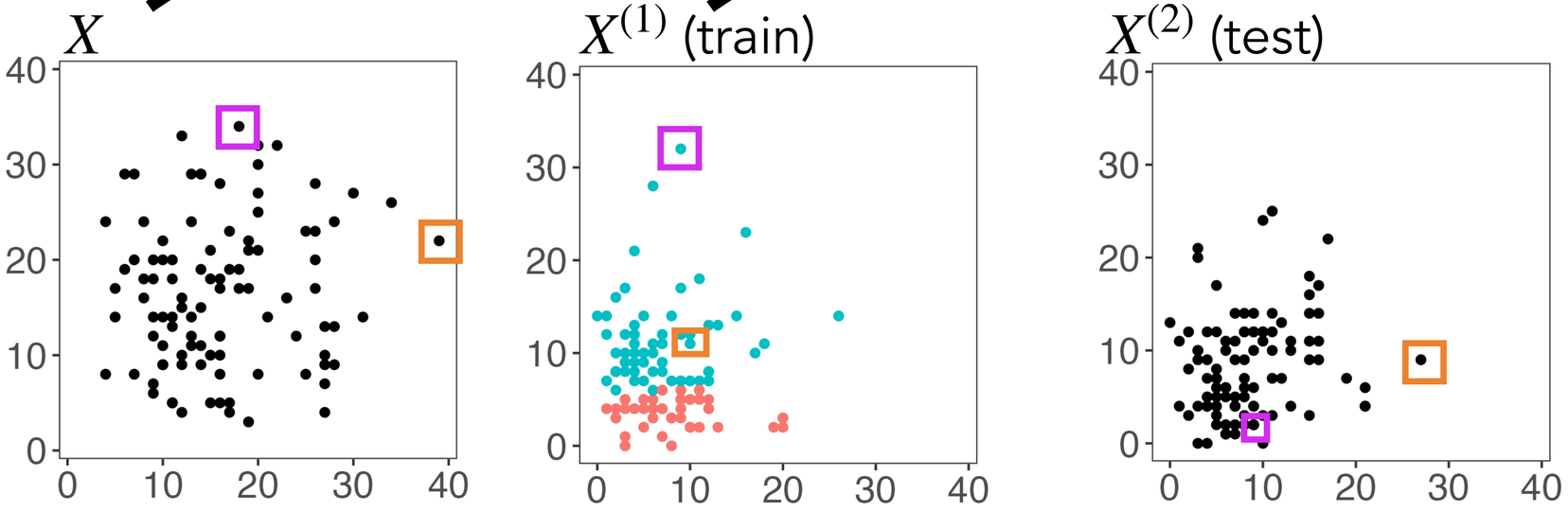
Step 1: thin observations into train/test.

Step 2: cluster the training set.

Thinning avoids the pitfall of sample splitting on our motivating examples

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

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



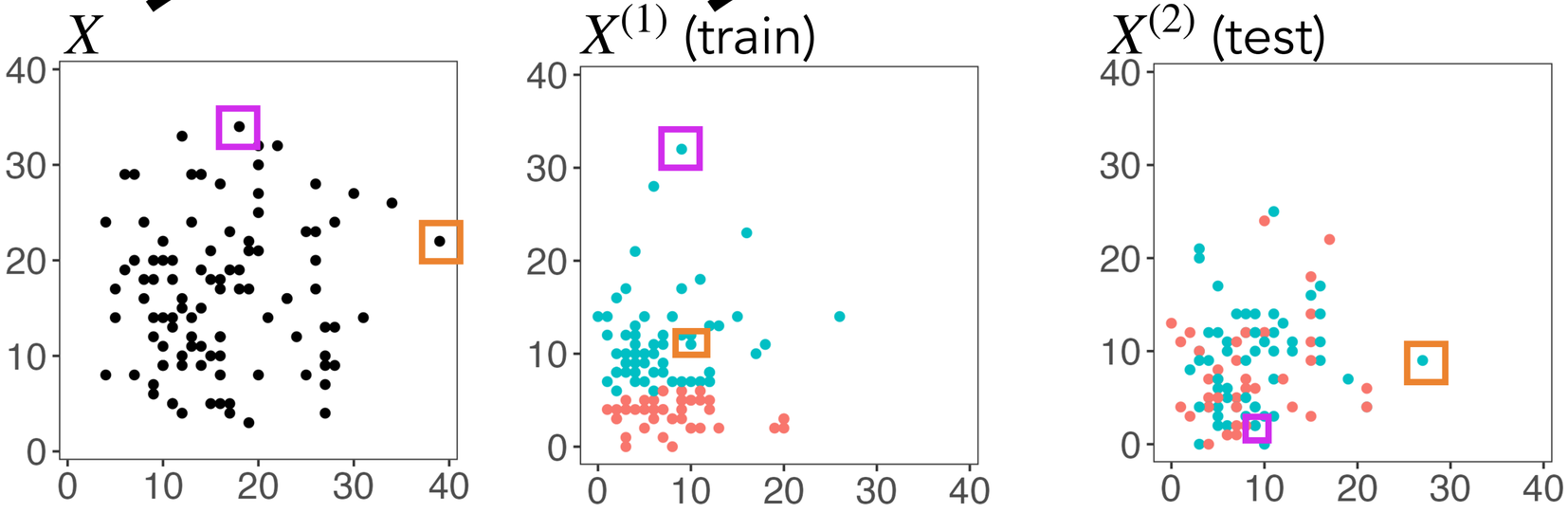
Step 1: thin observations into train/test.

Step 2: cluster the training set.

Thinning avoids the pitfall of sample splitting on our motivating examples

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

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



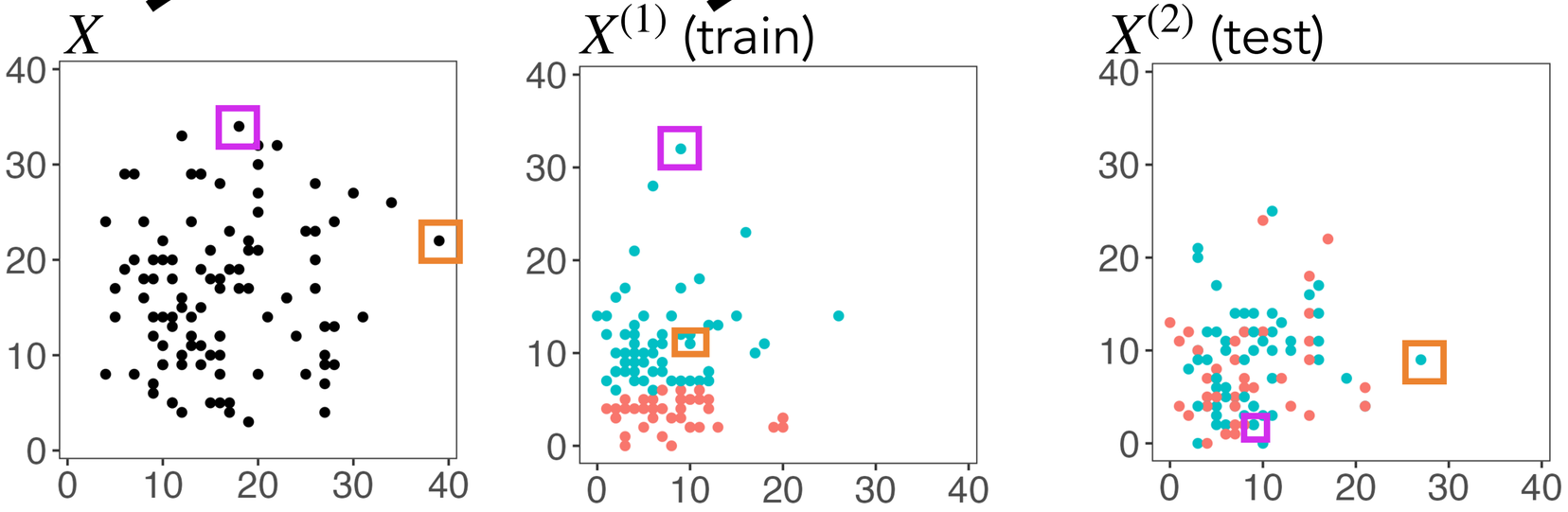
Step 1: thin observations into train/test.

Step 2: cluster the training set.

Thinning avoids the pitfall of sample splitting on our motivating examples

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

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



Step 1: thin observations into train/test.

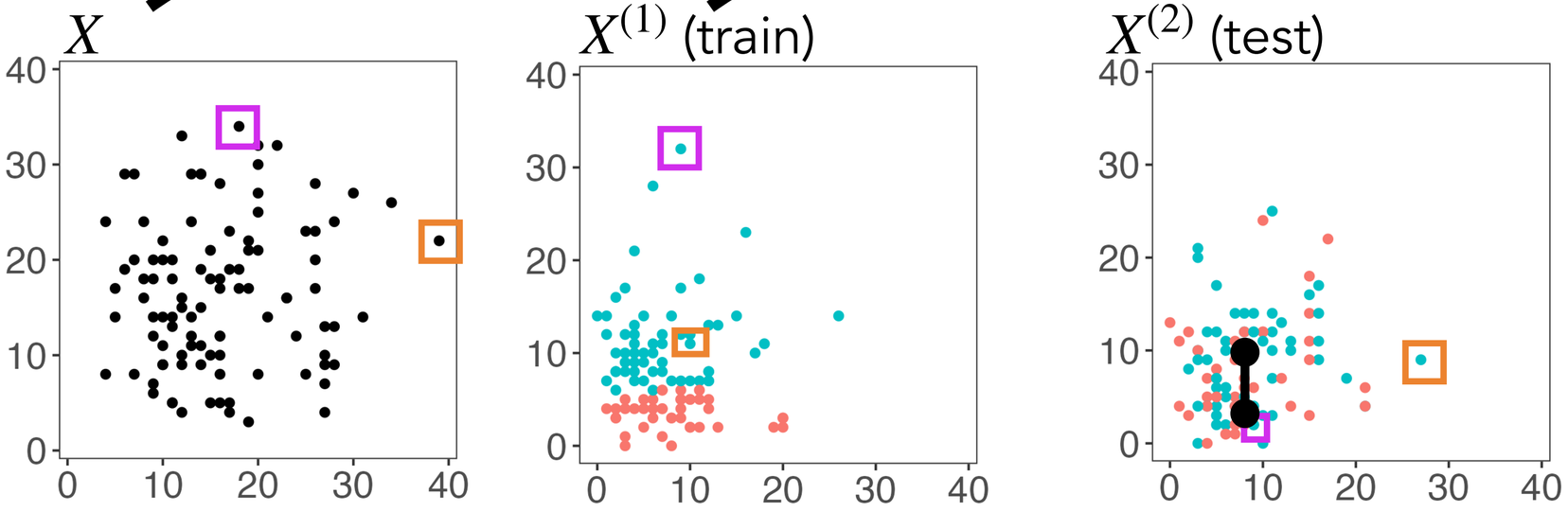
Step 2: cluster the training set.

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

Thinning avoids the pitfall of sample splitting on our motivating examples

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

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



Step 1: thin observations into train/test.

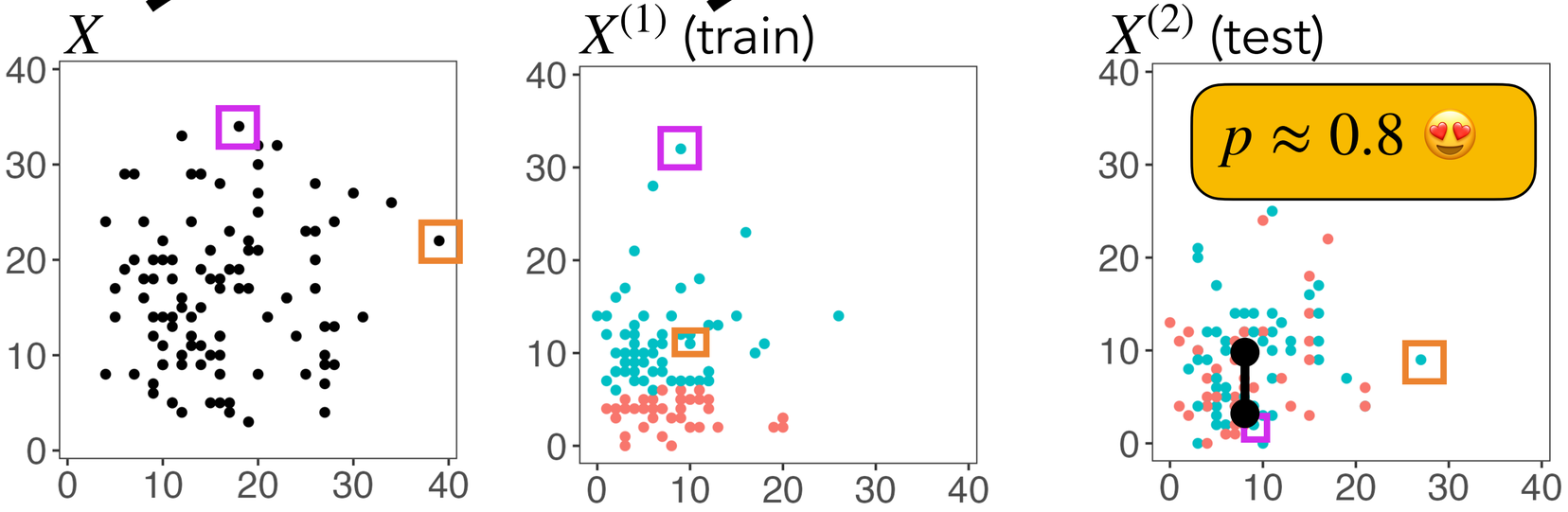
Step 2: cluster the training set.

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

Thinning avoids the pitfall of sample splitting on our motivating examples

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

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



Step 1: thin observations into train/test.

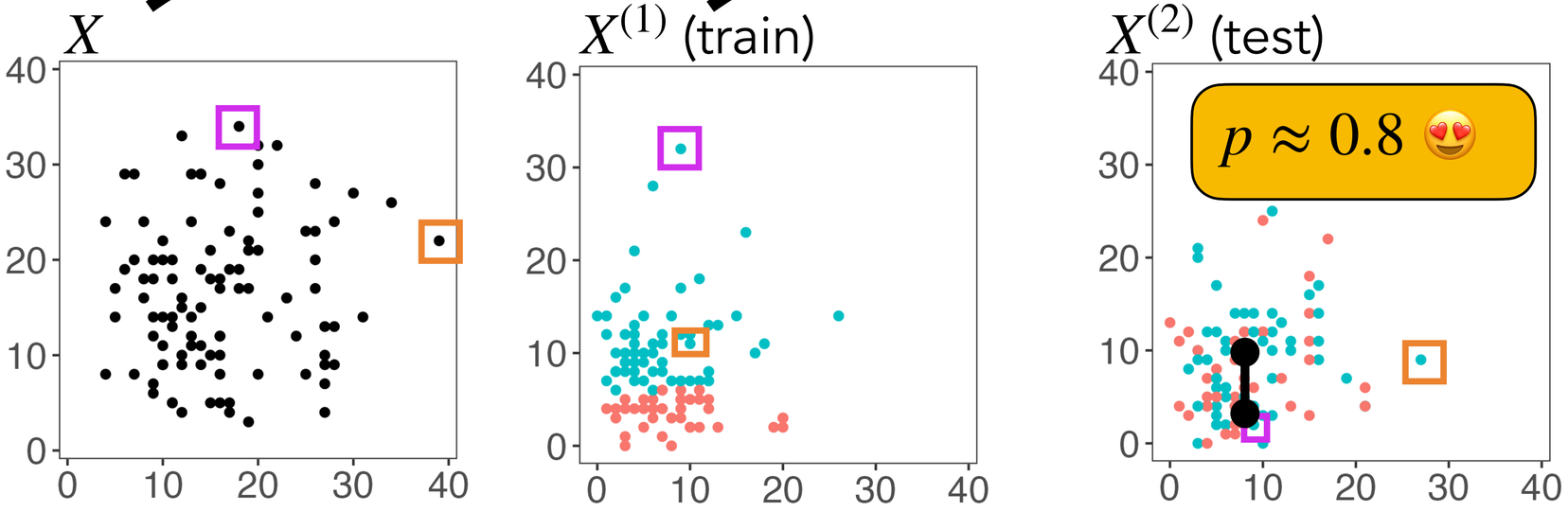
Step 2: cluster the training set.

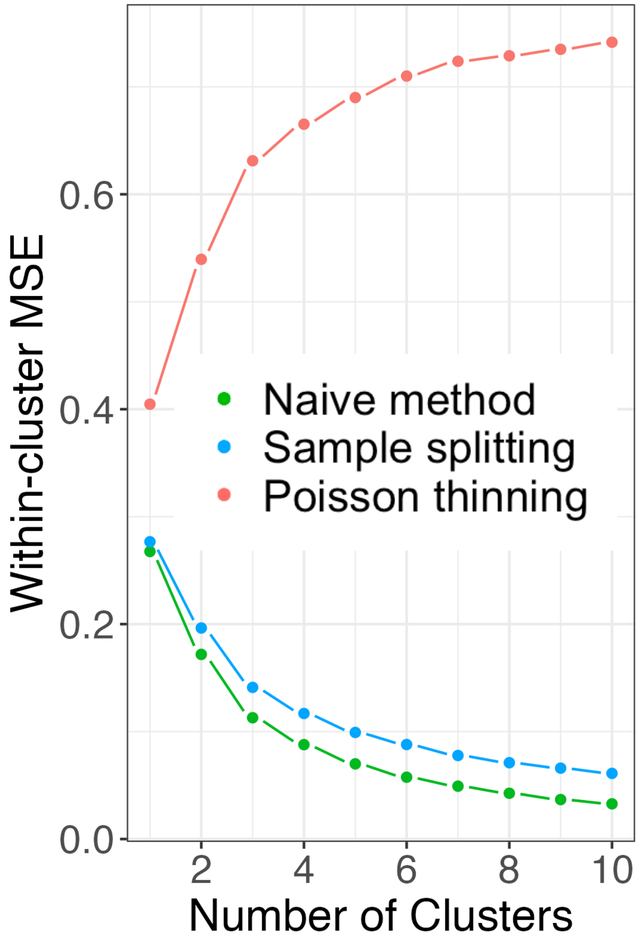
Step 3: test for difference in means on test set.

Thinning avoids the pitfall of sample splitting on our motivating examples

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

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



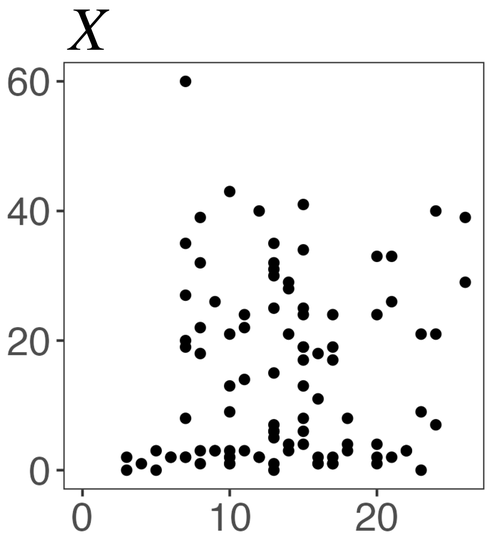


Step 1: thin observations into train/test.

Step 2: cluster the training set.

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

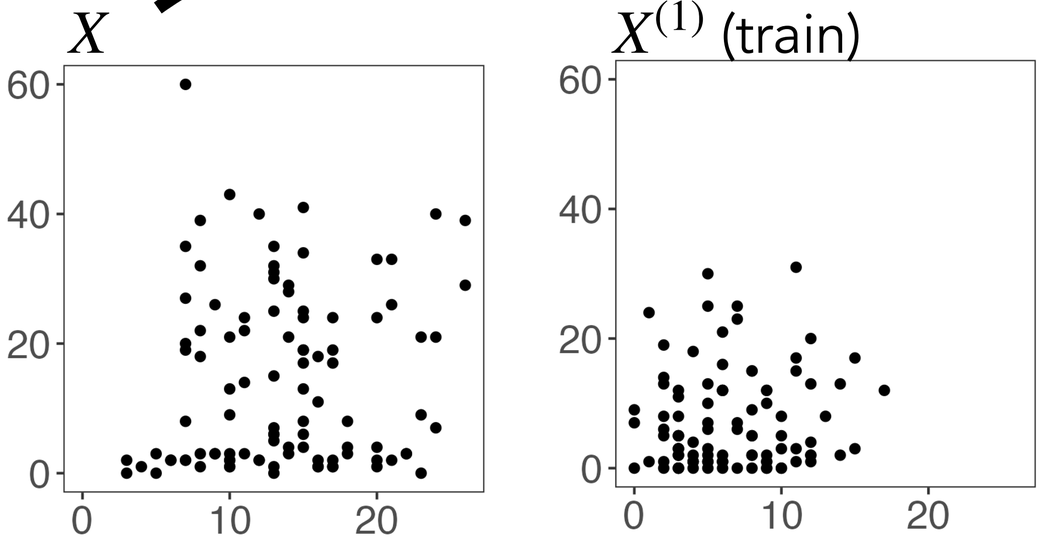
Thinning avoids the pitfall of sample splitting on our motivating examples



$$X_{i2} \sim \begin{cases} \text{Poisson}(3) & \text{if } i \leq 50 \\ \text{Poisson}(25) & \text{if } i > 50 \end{cases}$$

Thinning avoids the pitfall of sample splitting on our motivating examples

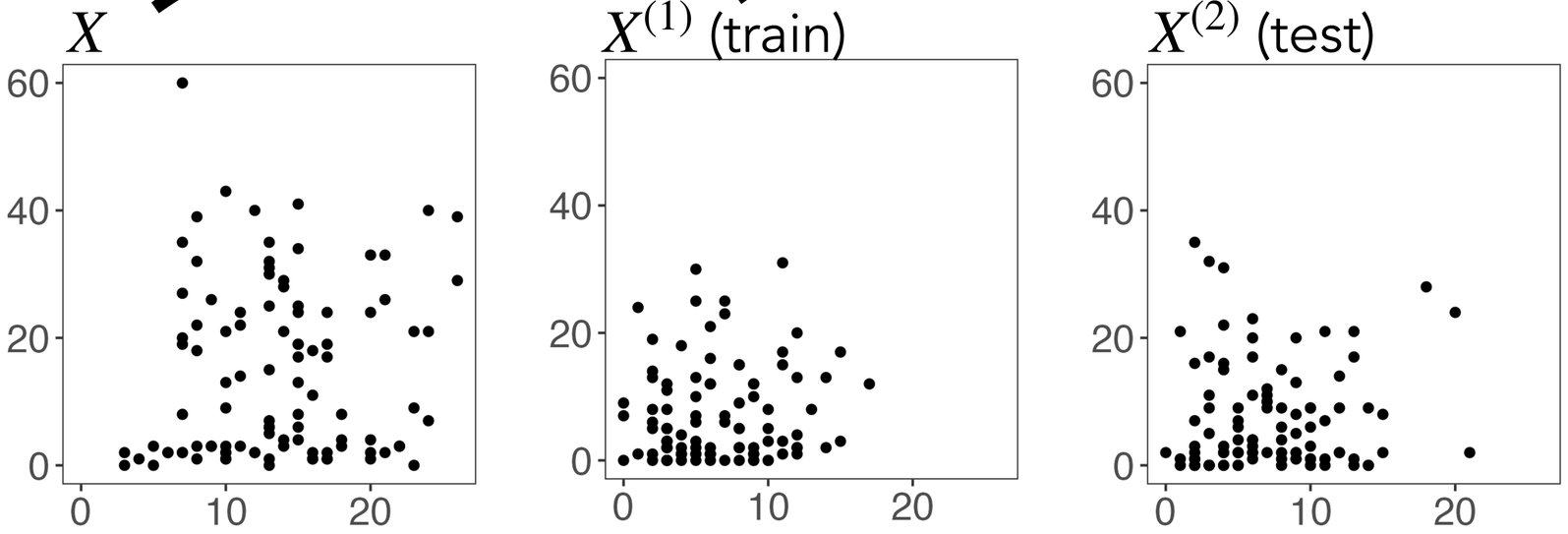
$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5)$$



$$X_{i2} \sim \begin{cases} \text{Poisson}(3) & \text{if } i \leq 50 \\ \text{Poisson}(25) & \text{if } i > 50 \end{cases}$$

Thinning avoids the pitfall of sample splitting on our motivating examples

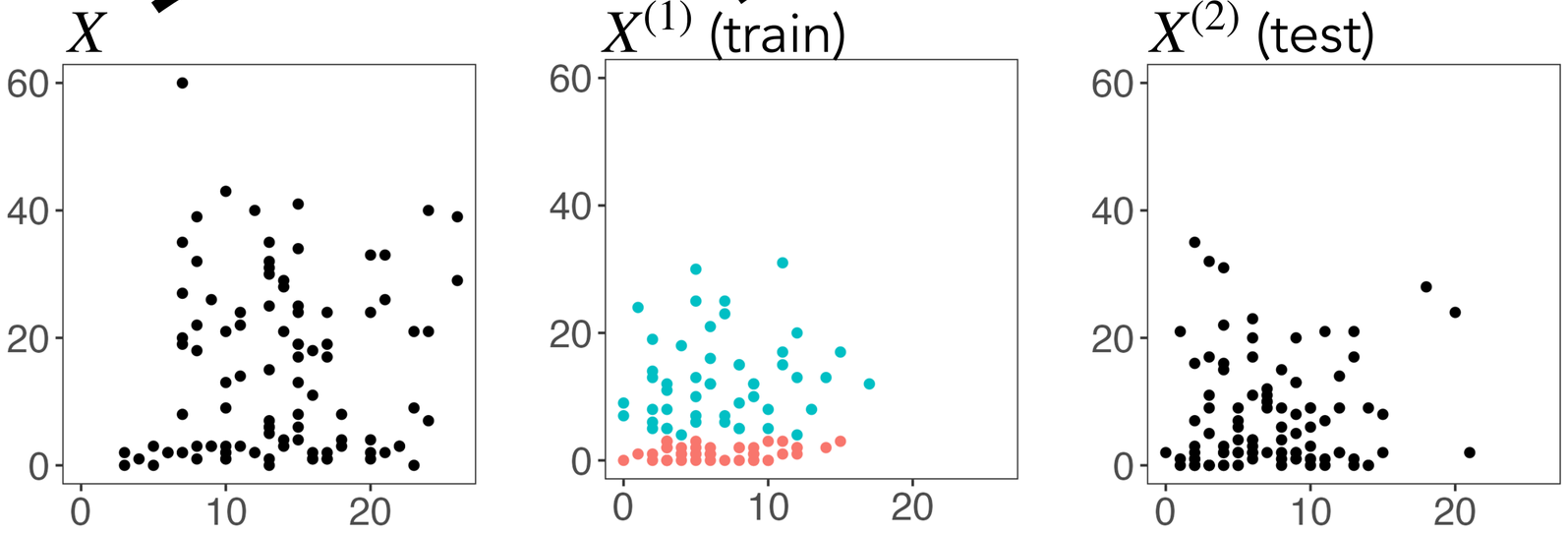
$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5) \qquad X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



$$X_{i2} \sim \begin{cases} \text{Poisson}(3) & \text{if } i \leq 50 \\ \text{Poisson}(25) & \text{if } i > 50 \end{cases}$$

Thinning avoids the pitfall of sample splitting on our motivating examples

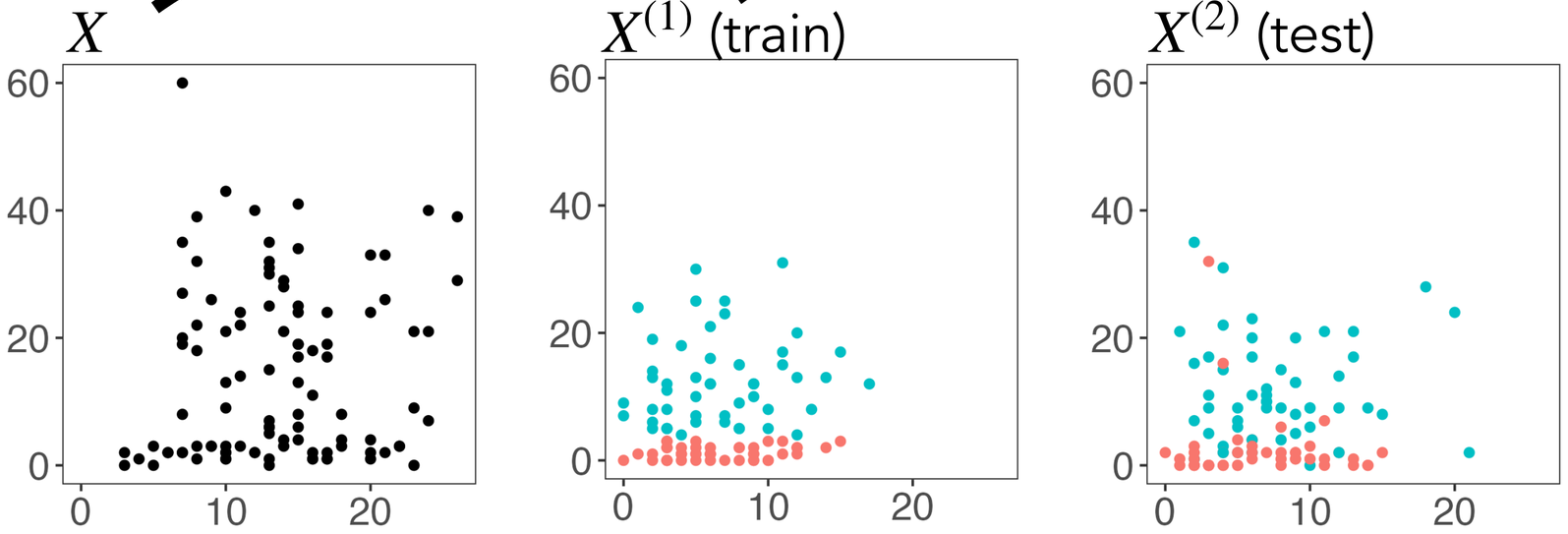
$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5) \qquad X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



$$X_{i2} \sim \begin{cases} \text{Poisson}(3) & \text{if } i \leq 50 \\ \text{Poisson}(25) & \text{if } i > 50 \end{cases}$$

Thinning avoids the pitfall of sample splitting on our motivating examples

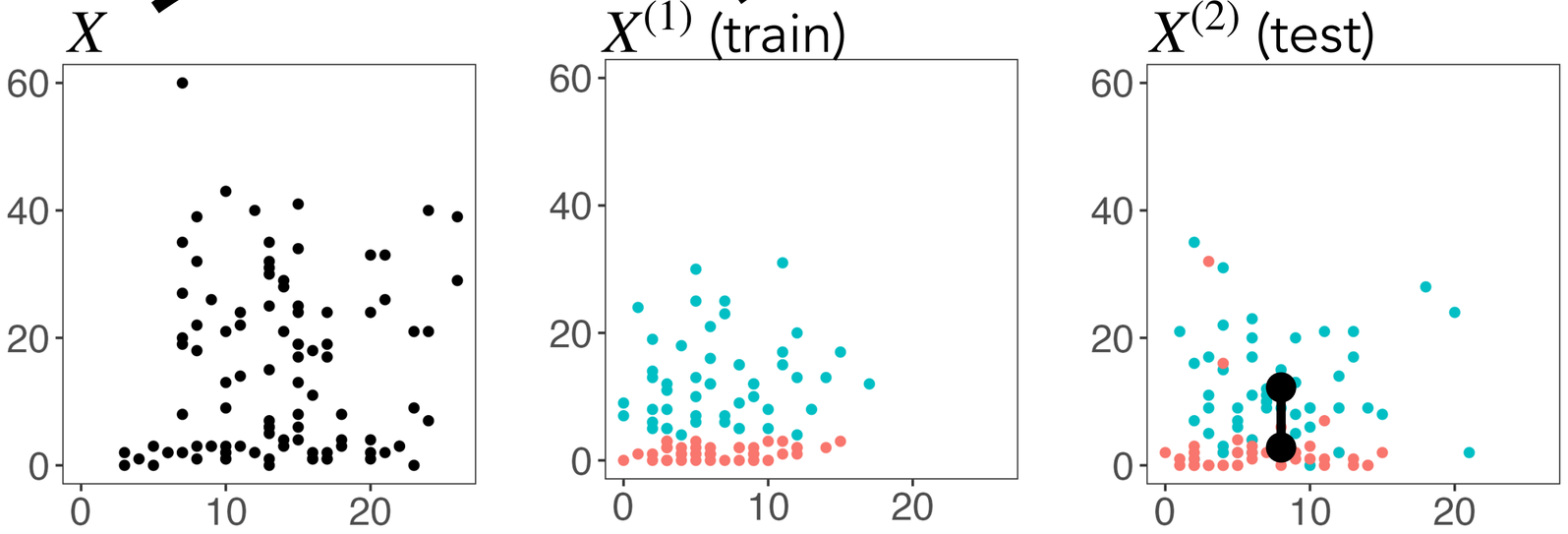
$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5) \qquad X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



$$X_{i2} \sim \begin{cases} \text{Poisson}(3) & \text{if } i \leq 50 \\ \text{Poisson}(25) & \text{if } i > 50 \end{cases}$$

Thinning avoids the pitfall of sample splitting on our motivating examples

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

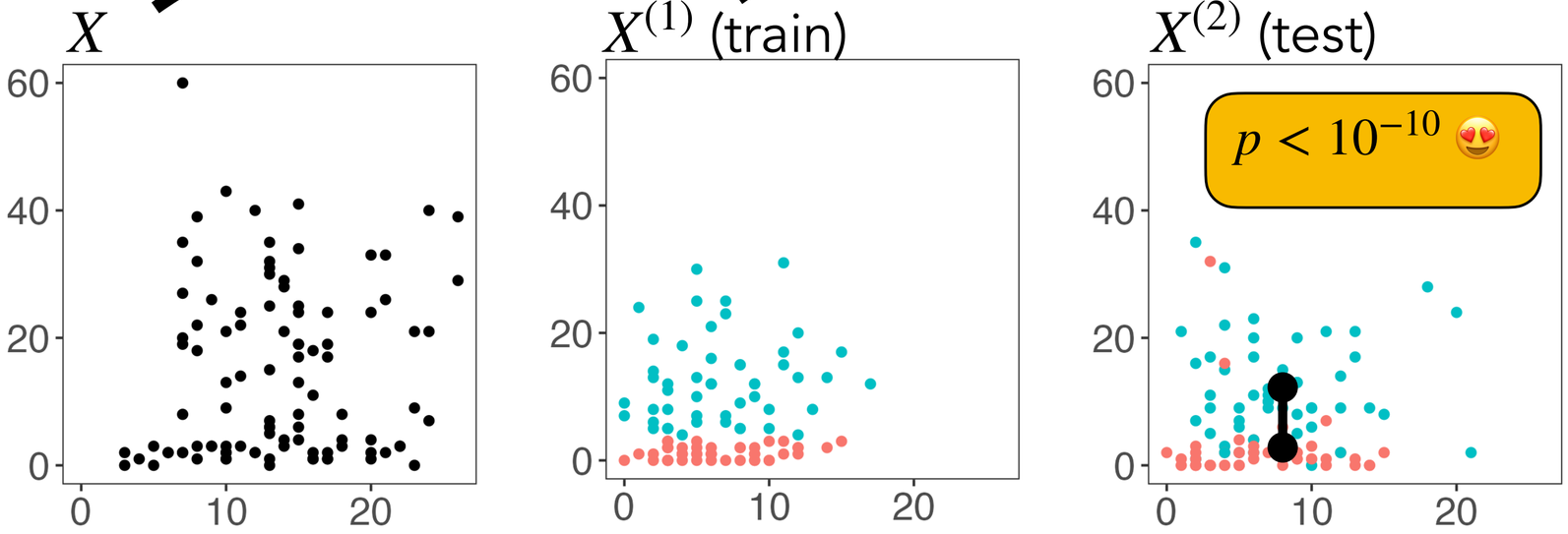


$$X_{i2} \sim \begin{cases} \text{Poisson}(3) & \text{if } i \leq 50 \\ \text{Poisson}(25) & \text{if } i > 50 \end{cases}$$

Thinning avoids the pitfall of sample splitting on our motivating examples

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

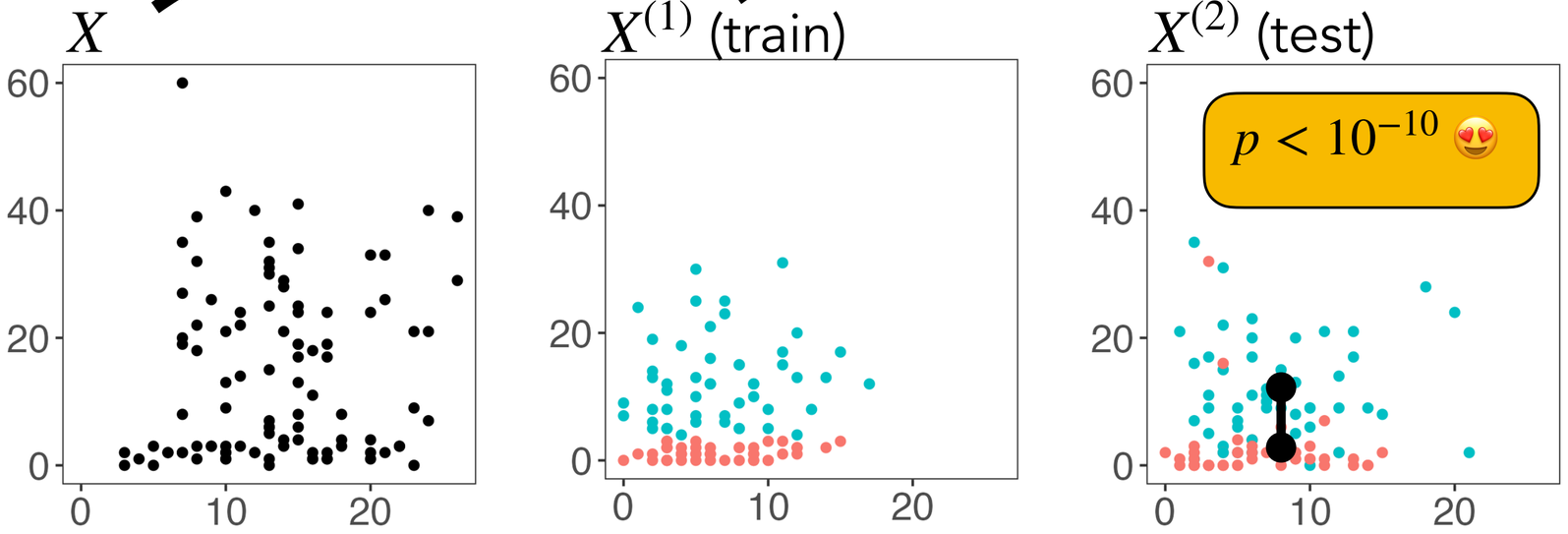
$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$

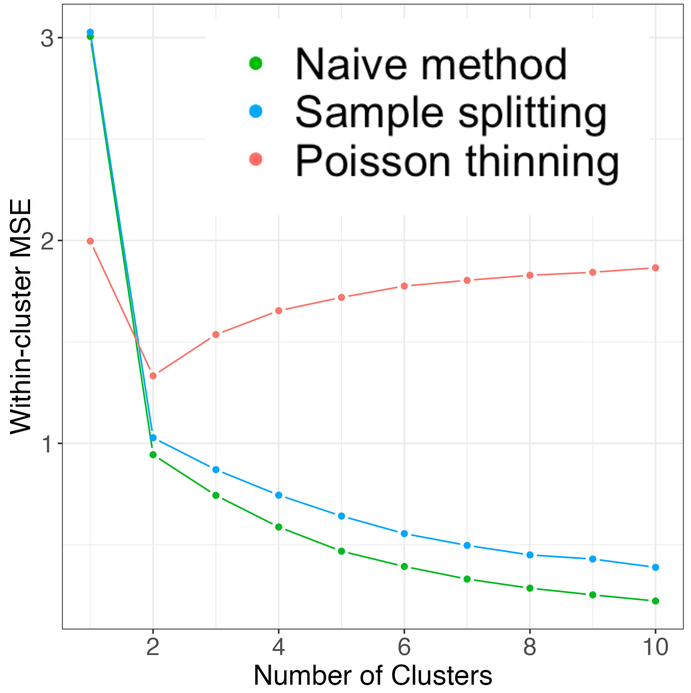


$$X_{i2} \sim \begin{cases} \text{Poisson}(3) & \text{if } i \leq 50 \\ \text{Poisson}(25) & \text{if } i > 50 \end{cases}$$

Thinning avoids the pitfall of sample splitting on our motivating examples

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





$$X_{i2} \sim \begin{cases} \text{Poisson}(3) & \text{if } i \leq 50 \\ \text{Poisson}(25) & \text{if } i > 50 \end{cases}$$

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/

But generalizations of Poisson thinning are needed, even if we only want to study scRNA-seq data

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*]

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.

Thinning approaches have the potential to be useful in a wide variety of settings, beyond our motivating example

Thinning approaches have the potential to be useful in a wide variety of settings, beyond our motivating example

1. Any unsupervised setting, where sample splitting is not an option.

Thinning approaches have the potential to be useful in a wide variety of settings, beyond our motivating example

1. Any unsupervised setting, where sample splitting is not an option.

2. Supervised settings where sample splitting is unsatisfying:

   - Fixed-X regression settings.
   - Non-IID data.
   - Data with outliers or influential points.

Thinning approaches have the potential to be useful in a wide variety of settings, beyond our motivating example

1. Any unsupervised setting, where sample splitting is not an option.

2. Supervised settings where sample splitting is unsatisfying:

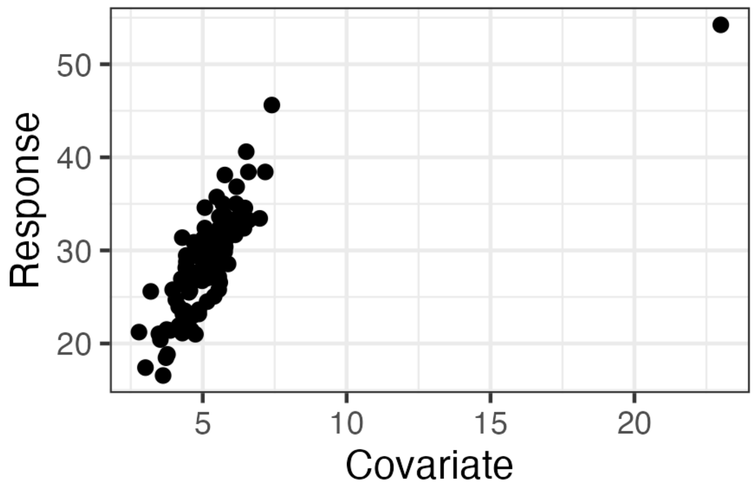
   - Fixed-X regression settings.
   - Non-IID data.
   - Data with outliers or influential points.

Thinning approaches have the potential to be useful in a wide variety of settings, beyond our motivating example

1. Any unsupervised setting, where sample splitting is not an option.

2. Supervised settings where sample splitting is unsatisfying:

   - Fixed-X regression settings.
   - Non-IID data.
   - Data with outliers or influential points.



Outline

1. Motivation: settings where sample splitting doesn't work
2. Poisson thinning
3. **Data thinning**
4. Application to single-cell RNA sequencing data
5. 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?

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

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

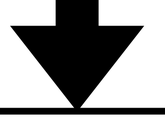
Data thinning for convolution-closed distributions

Data thinning for convolution-closed distributions

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

Data thinning for convolution-closed distributions

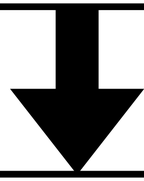
We know x could have arisen as $x' + x''$, where
$X' \sim F_{\epsilon\lambda}, \; X'' \sim F_{(1-\epsilon)\lambda}, \; X' \perp\!\!\!\perp X''.$



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 $X' \sim F_{\epsilon\lambda},\ X'' \sim F_{(1-\epsilon)\lambda},\ X' \perp\!\!\!\perp X''$.

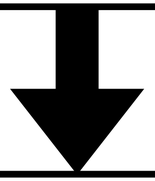


We observe realization x from $X \sim F_{\lambda}$.

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 $X' \sim F_{\epsilon\lambda},\ X'' \sim F_{(1-\epsilon)\lambda},\ X' \perp\!\!\!\perp X''$.



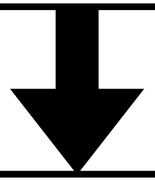
We observe realization x from $X \sim F_{\lambda}$.

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 $X' \sim F_{\epsilon\lambda},\ X'' \sim F_{(1-\epsilon)\lambda},\ X' \perp\!\!\!\perp X''$.



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

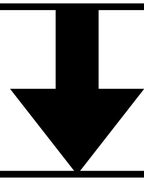
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

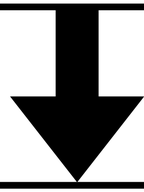
Let $G_{\epsilon,x}$ be the conditional distribution of $X' \mid X = x$.

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 $X' \sim F_{\epsilon\lambda},\ X'' \sim F_{(1-\epsilon)\lambda},\ X' \perp\!\!\!\perp X''$.

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

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

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

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

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

Theorem:

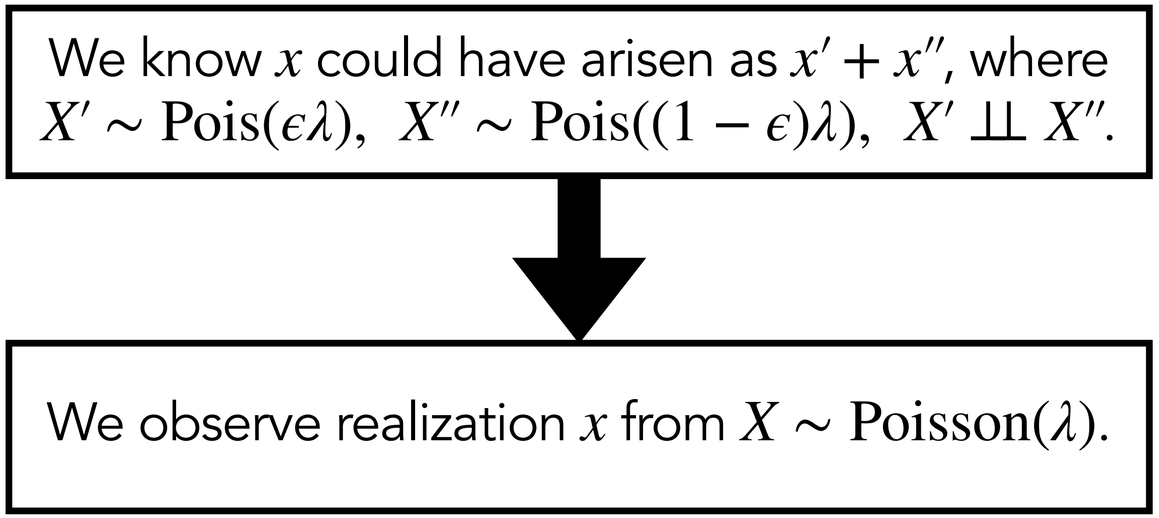
$X^{(1)} \sim F_{\epsilon\lambda},\ X^{(2)} \sim F_{(1-\epsilon)\lambda},\ X^{(1)} \perp\!\!\!\perp X^{(2)}$.

Data thinning for the Poisson distribution

Data thinning for the Poisson distribution

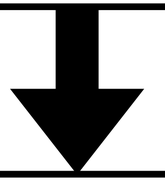
We observe realization x from $X \sim \text{Poisson}(\lambda)$.

Data thinning for the Poisson distribution



Data thinning for the Poisson distribution

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

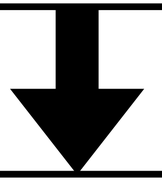


We observe realization x from $X \sim \text{Poisson}(\lambda)$.

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

Data thinning for the Poisson distribution

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



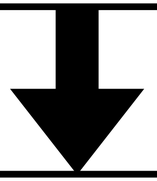
We observe realization x from $X \sim \text{Poisson}(\lambda)$.

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 the Poisson distribution

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



We observe realization x from $X \sim \text{Poisson}(\lambda)$.

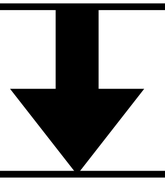
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

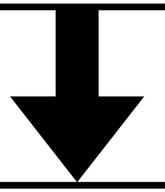
The conditional distribution of $X' \mid X = x$ is $\text{Binomial}(x, \epsilon)$.

Data thinning for the Poisson distribution

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



We observe realization x from $X \sim \text{Poisson}(\lambda)$.



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

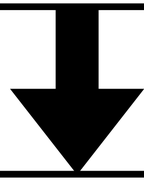
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

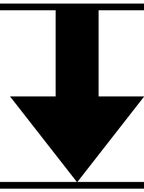
The conditional distribution of $X' \mid X = x$ is $\text{Binomial}(x, \epsilon)$.

Data thinning for the Poisson distribution

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



We observe realization x from $X \sim \text{Poisson}(\lambda)$.



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

Theorem:

$X^{(1)} \sim \text{Pois}(\epsilon\lambda),\ X^{(2)} \sim \text{Pois}((1-\epsilon)\lambda),\ \ X^{(1)} \perp\!\!\!\perp X^{(2)}$.

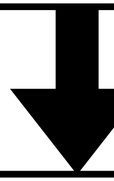
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

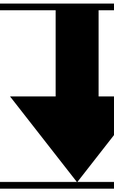
The conditional distribution of $X' \mid X = x$ is $\text{Binomial}(x, \epsilon)$.

Data thinning for the Poisson distribution

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



We observe realization x from $X \sim \text{Poisson}(\lambda)$.



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

Theorem:

$X^{(1)} \sim \text{Pois}(\epsilon\lambda),\ X^{(2)} \sim \text{Pois}((1-\epsilon)\lambda),\ \ X^{(1)} \perp\!\!\!\perp X^{(2)}$.

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''?

The conditional distribution of $X' \mid X = x$ is $\text{Binomial}(x, \epsilon)$.

We have recovered Poisson thinning!

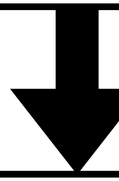
Data thinning for the negative binomial distribution

Data thinning for the negative binomial distribution

We observe realization x from $X \sim \text{NB}(\mu, b)$.

Data thinning for the negative binomial distribution

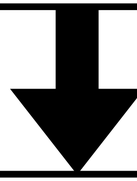
We know x could have arisen as $x' + x''$, where
$X' \sim \text{NB}(\epsilon\mu, \epsilon b), X'' \sim \text{NB}((1-\epsilon)\mu, (1-\epsilon)b), X' \perp\!\!\!\perp X''.$



We observe realization x from $X \sim \text{NB}(\mu, b)$.

Data thinning for the negative binomial distribution

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

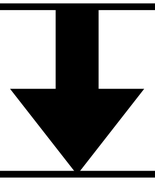


We observe realization x from $X \sim \text{NB}(\mu, b)$.

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

Data thinning for the negative binomial distribution

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



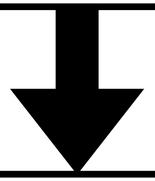
We observe realization x from $X \sim \text{NB}(\mu, b)$.

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 the negative binomial distribution

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



We observe realization x from $X \sim \text{NB}(\mu, b)$.

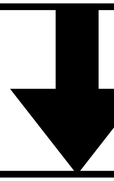
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

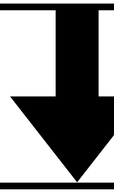
The conditional distribution of $X' \mid X = x$ is $\text{BetaBinomial}(x, \epsilon b, (1-\epsilon)b)$.

Data thinning for the negative binomial distribution

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



We observe realization x from $X \sim \text{NB}(\mu, b)$.



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

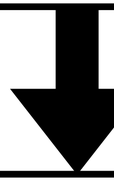
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

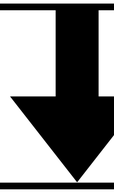
The conditional distribution of $X' \mid X = x$ is $\text{BetaBinomial}(x, \epsilon b, (1-\epsilon)b)$.

Data thinning for the negative binomial distribution

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



We observe realization x from $X \sim \text{NB}(\mu, b)$.



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

Theorem:

$X^{(1)} \sim \text{NB}(\epsilon\mu, \epsilon b)$, $X^{(2)} \sim \text{NB}((1-\epsilon)\mu, (1-\epsilon)b)$, $X^{(1)} \perp\!\!\!\perp X^{(2)}$.

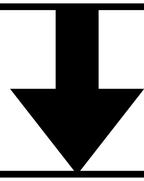
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

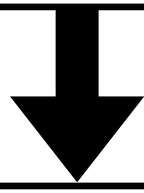
The conditional distribution of $X' \mid X = x$ is $\text{BetaBinomial}(x, \epsilon b, (1-\epsilon)b)$.

Data thinning for the negative binomial distribution

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



We observe realization x from $X \sim \text{NB}(\mu, b)$.



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

Theorem:

$X^{(1)} \sim \text{NB}(\epsilon\mu, \epsilon b)$, $X^{(2)} \sim \text{NB}((1-\epsilon)\mu, (1-\epsilon)b)$, $X^{(1)} \perp\!\!\!\perp X^{(2)}$.

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''?

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

This is a new result!

For many common distributions, the distribution $G_{\epsilon,x}$ has a simple form

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 $X^{(2)} = X - X^{(1)}$:
$\text{Poisson}(\lambda)$	$\text{Binomial}(x, \epsilon)$	$\text{Poisson}(\epsilon\lambda)$	$\text{Poisson}((1-\epsilon)\lambda)$

For many common distributions, the distribution $G_{\epsilon,x}$ has a simple form

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 $X^{(2)} = X - X^{(1)}$:
$\text{Poisson}(\lambda)$	$\text{Binomial}(x, \epsilon)$	$\text{Poisson}(\epsilon\lambda)$	$\text{Poisson}((1-\epsilon)\lambda)$

Related work on Poisson thinning:

- Sarkar and Stephens, 2021, Nature Genetics.
- Chen et al., 2021, arXiv:2108.03336
- Leiner et al., 2023, JASA
- Neufeld et al., 2022, Biostatistics.
- Oliveira, Lei, and Tibshirani, 2022, arXiv:2212.01943.

For many common distributions, the distribution $G_{\epsilon,x}$ has a simple form

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 $X^{(2)} = X - X^{(1)}$:
$\text{Poisson}(\lambda)$	$\text{Binomial}(x, \epsilon)$	$\text{Poisson}(\epsilon\lambda)$	$\text{Poisson}((1-\epsilon)\lambda)$
$\text{N}(\mu, \sigma^2)$	$\text{N}(\epsilon x, \epsilon(1-\epsilon)\sigma^2)$	$\text{N}(\epsilon\mu, \epsilon\sigma^2)$	$\text{N}((1-\epsilon)\mu, (1-\epsilon)\sigma^2)$

For many common distributions, the distribution $G_{\epsilon,x}$ has a simple form

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 $X^{(2)} = X - X^{(1)}$:
$\text{Poisson}(\lambda)$	$\text{Binomial}(x, \epsilon)$	$\text{Poisson}(\epsilon\lambda)$	$\text{Poisson}((1-\epsilon)\lambda)$
$\text{N}(\mu, \sigma^2)$	$\text{N}(\epsilon x, \epsilon(1-\epsilon)\sigma^2)$	$\text{N}(\epsilon\mu, \epsilon\sigma^2)$	$\text{N}((1-\epsilon)\mu, (1-\epsilon)\sigma^2)$

Related work on Gaussian thinning:

- Tian and Taylor, 2018, Annals of Statistics.
- Tian, 2020, Annals of Statistics.
- Rasines and Young, 2022, Biometrika.
- Leiner et al., 2023, JASA
- Oliveira, Lei, and Tibshirani, 2022, arXiv:2111.09447.

For many common distributions, the distribution $G_{\epsilon,x}$ has a simple form

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 $X^{(2)} = X - X^{(1)}$:
$\text{Poisson}(\lambda)$	$\text{Binomial}(x, \epsilon)$	$\text{Poisson}(\epsilon\lambda)$	$\text{Poisson}((1-\epsilon)\lambda)$
$\text{N}(\mu, \sigma^2)$	$\text{N}(\epsilon x, \epsilon(1-\epsilon)\sigma^2)$	$\text{N}(\epsilon\mu, \epsilon\sigma^2)$	$\text{N}((1-\epsilon)\mu, (1-\epsilon)\sigma^2)$
$\text{NegativeBinomial}(\mu, b)$	$\text{BetaBinomial}(x, \epsilon b, (1-\epsilon)b)$.	$\text{NegativeBinomial}(\epsilon\mu, \epsilon b)$	$\text{NegativeBinomial}((1-\epsilon)\mu, (1-\epsilon)b)$

For many common distributions, the distribution $G_{\epsilon,x}$ has a simple form

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

What if we get a nuisance parameter wrong?

Negative binomial thinning algorithm

Suppose $X \sim \text{NegBin}\left(\mu, b\right)$.

Draw

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

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

1) $X^{(1)} \sim \text{NegBin}\left(\epsilon\mu, \epsilon b\right)$.
2) $X^{(2)} \sim \text{NegBin}\left((1-\epsilon)\mu, (1-\epsilon)b\right)$
3) $X^{(1)} \perp\!\!\!\perp X^{(2)}$.

What if we get a nuisance parameter wrong?

Negative binomial thinning algorithm

Suppose $X \sim \text{NegBin}\left(\mu, b\right)$.

Draw

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

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

1) $X^{(1)} \sim \text{NegBin}\left(\epsilon\mu, \epsilon b\right)$.
2) $X^{(2)} \sim \text{NegBin}\left((1-\epsilon)\mu, (1-\epsilon)b\right)$
3) $X^{(1)} \perp\!\!\!\perp X^{(2)}$.

What if we get a nuisance parameter wrong?

Negative binomial thinning algorithm

Suppose $X \sim \text{NegBin}\left(\mu, b\right)$.

Draw

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

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

~~1) $X^{(1)} \sim \text{NegBin}\left(c\mu, cb\right)$.~~

~~2) $X^{(2)} \sim \text{NegBin}\left((1-c)\mu, (1-c)b\right)$~~

~~3) $X^{(1)} \perp\!\!\!\perp X^{(2)}$.~~

What if we get a nuisance parameter wrong?

Negative binomial thinning algorithm

Suppose $X \sim \text{NegBin}\left(\mu, b\right)$.

Draw

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

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

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

What if we get a nuisance parameter wrong?

Negative binomial thinning algorithm

Suppose $X \sim \text{NegBin}\left(\mu, b\right)$.

Draw

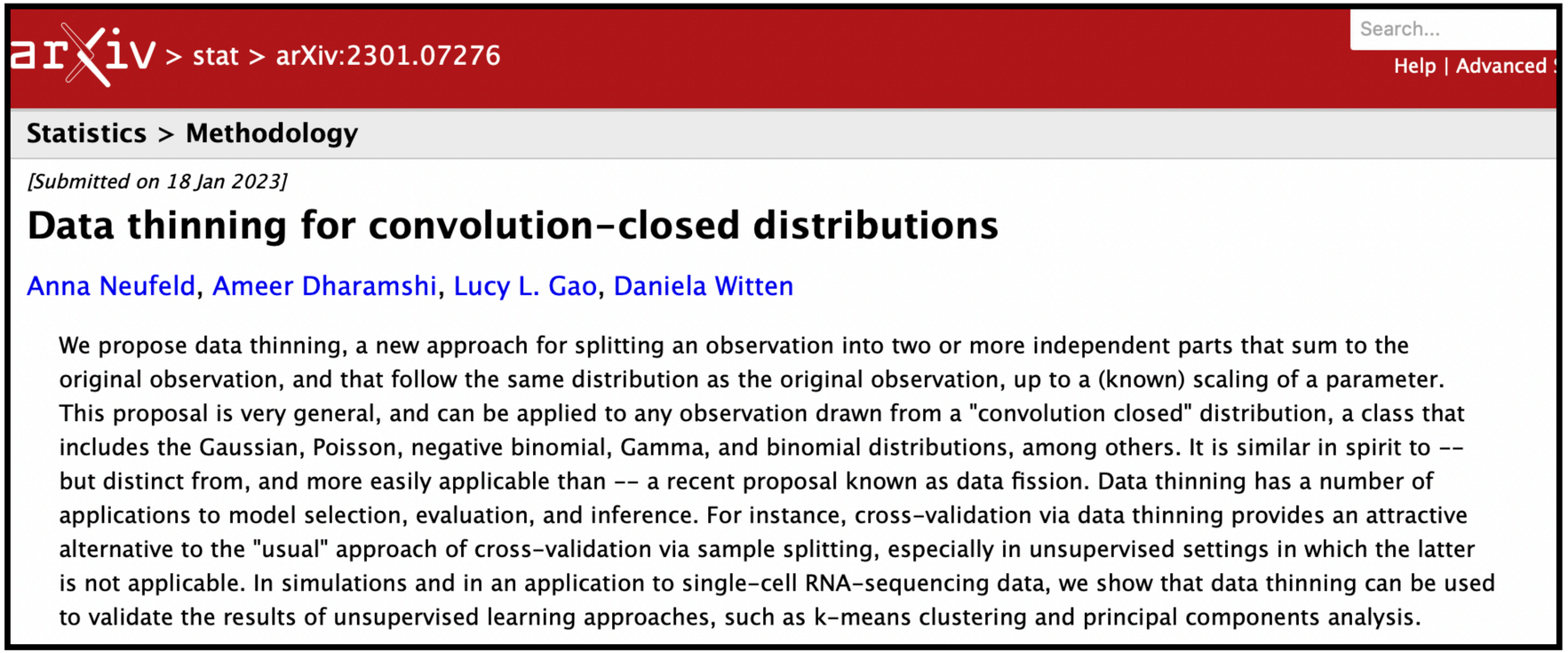
$X^{(1)} \sim \text{BetaBinomial}(x, \epsilon\tilde{b}, (1-\epsilon)\tilde{b}\,)$,

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

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

Similar results can be derived for other decompositions.

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
3. Data thinning
4. **Application to single-cell RNA sequencing data**
5. Ongoing work

How can we validate the results of clustering?

RESEARCH ARTICLE

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]†

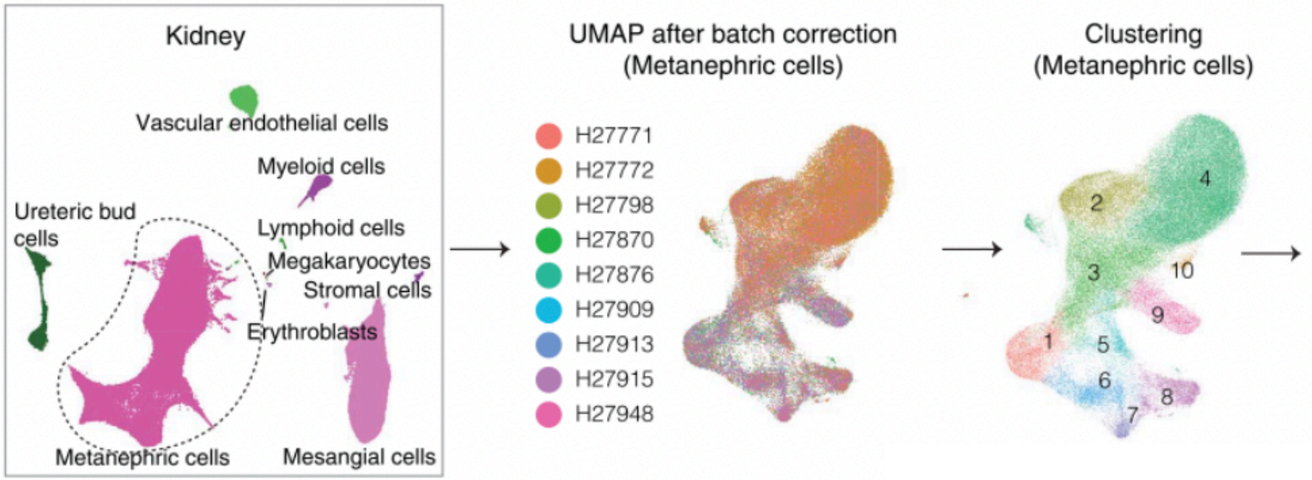
How can we validate the results of clustering?

RESEARCH ARTICLE

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]†



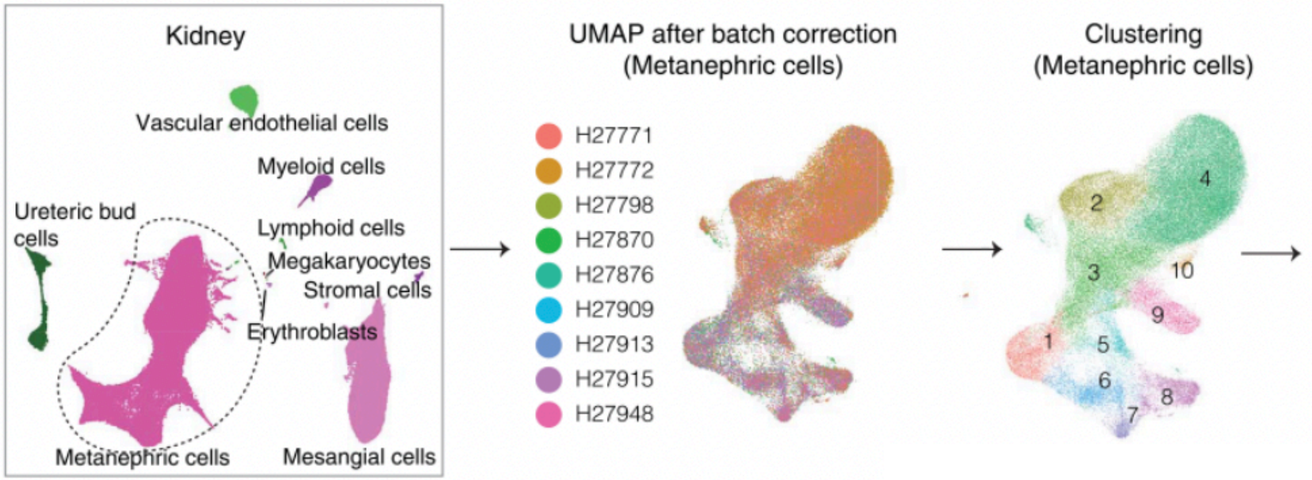
How can we validate the results of clustering?

RESEARCH ARTICLE

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]†



"Intradataset cross validation"

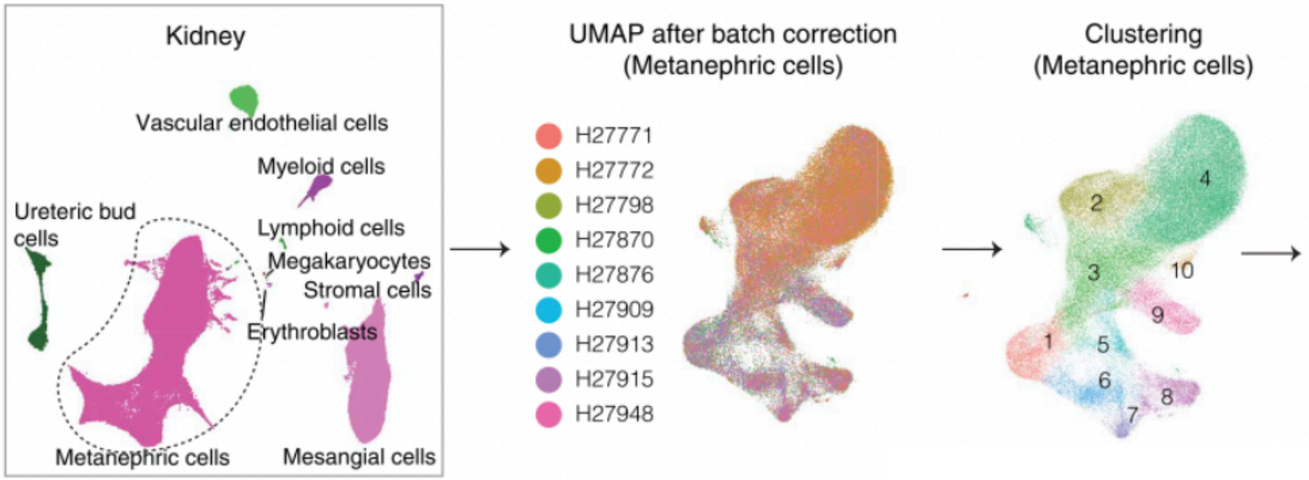
How can we validate the results of clustering?

RESEARCH ARTICLE

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]†



"Intradataset cross validation"

- **Step 1:** Cluster cells.

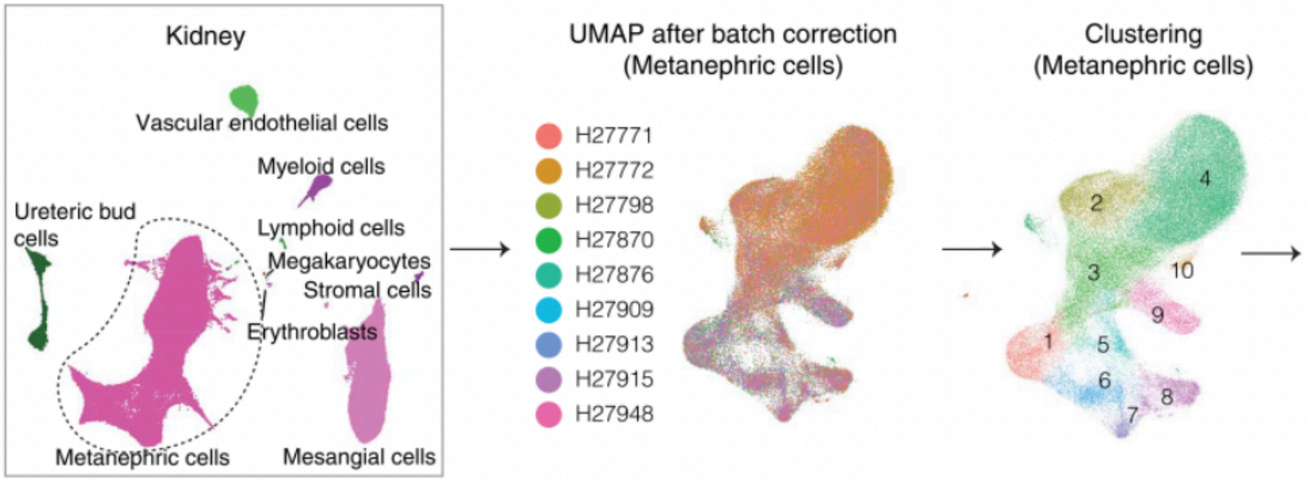
How can we validate the results of clustering?

RESEARCH ARTICLE

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]†



"Intradataset cross validation"

- **Step 1:** Cluster cells.
- **Step 2:** Treat clusters as truth. Do 5-fold cross validation with SVM.

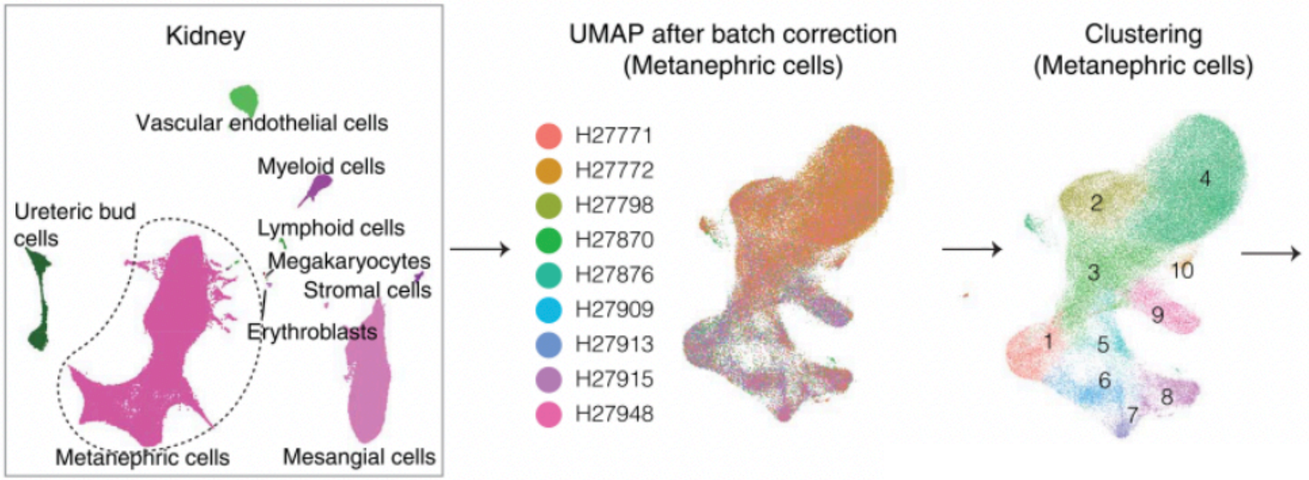
How can we validate the results of clustering?

RESEARCH ARTICLE

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]†



"Intradataset cross validation"

- **Step 1:** Cluster cells.
- **Step 2:** Treat clusters as truth. Do 5-fold cross validation with SVM.
- **Step 3:** Compare clusters to SVM predictions.

How can we validate the results of clustering?

RESEARCH ARTICLE

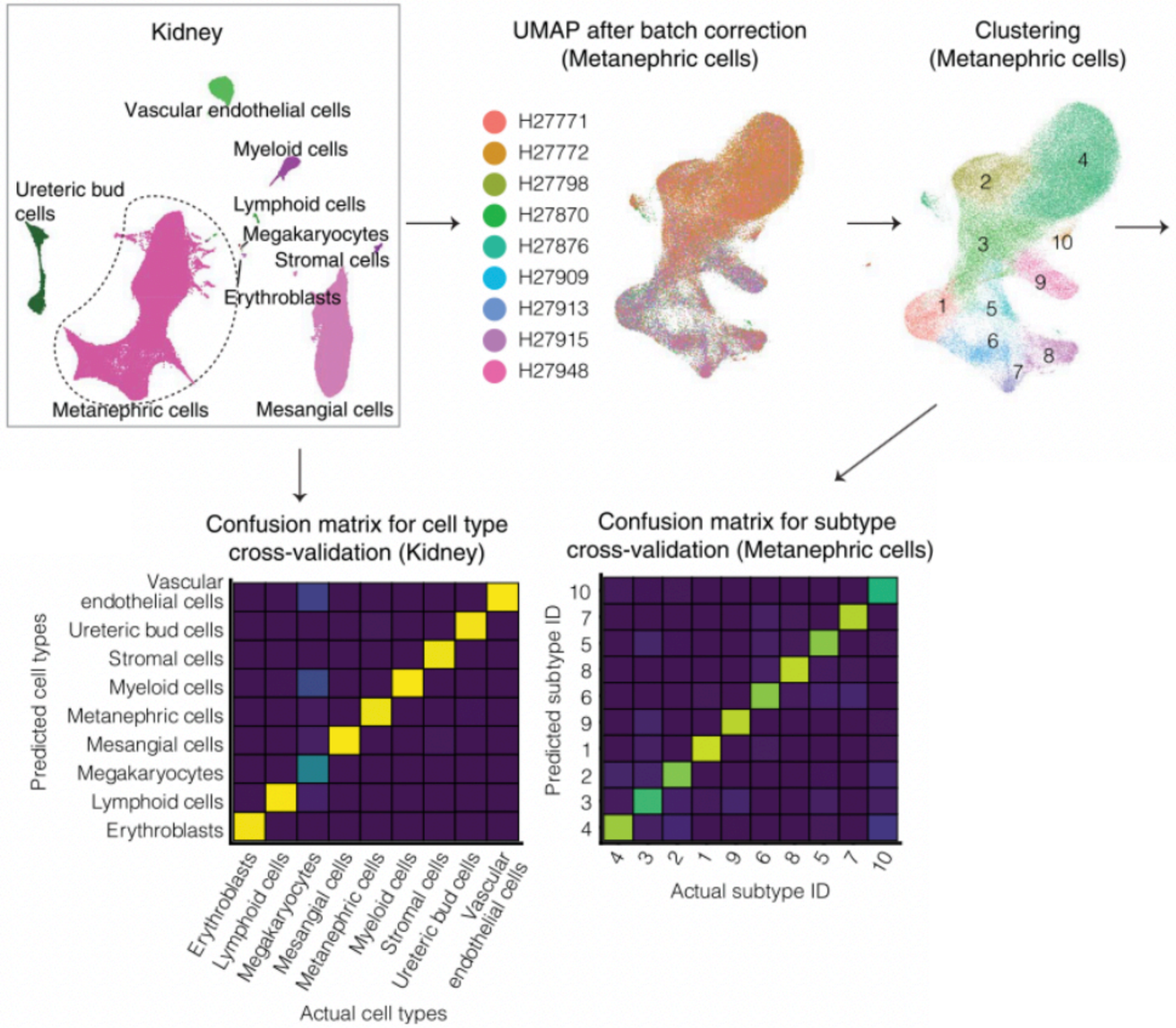
HUMAN GENOMICS

A human cell atlas of fetal gene expression

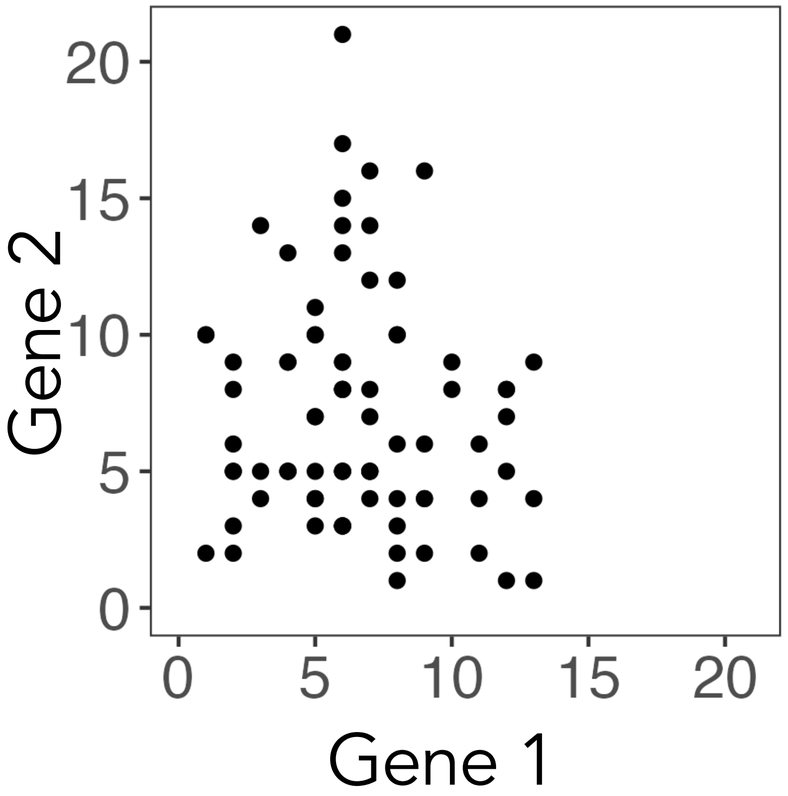
Junyue Cao[1]*, Diana R. O'Day[2], Hannah A. Pliner[3], Paul D. Kingsley[4], Mei Deng[2], Riza M. Daza[1], Michael A. Zager[3,5], Kimberly A. Aldinger[2,6], Ronnie Blecher-Gonen[1], Fan Zhang[7], Malte Spielmann[8,9], James Palis[4], Dan Doherty[2,3,6], Frank J. Steemers[7], Ian A. Glass[2,3,6], Cole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

"Intradataset cross validation"

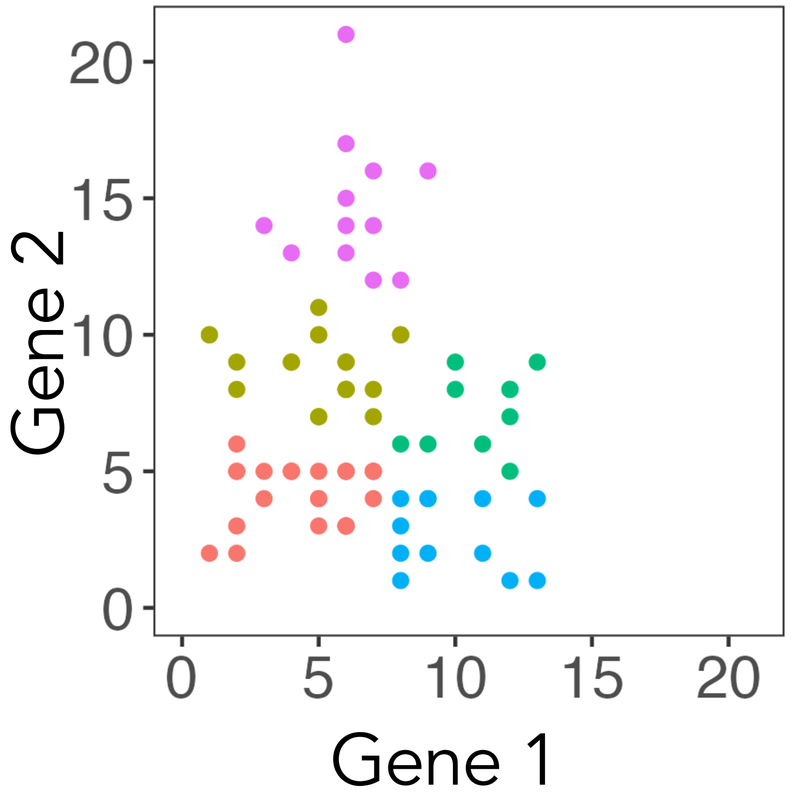
- **Step 1:** Cluster cells.
- **Step 2:** Treat clusters as truth. Do 5-fold cross validation with SVM.
- **Step 3:** Compare clusters to SVM predictions.



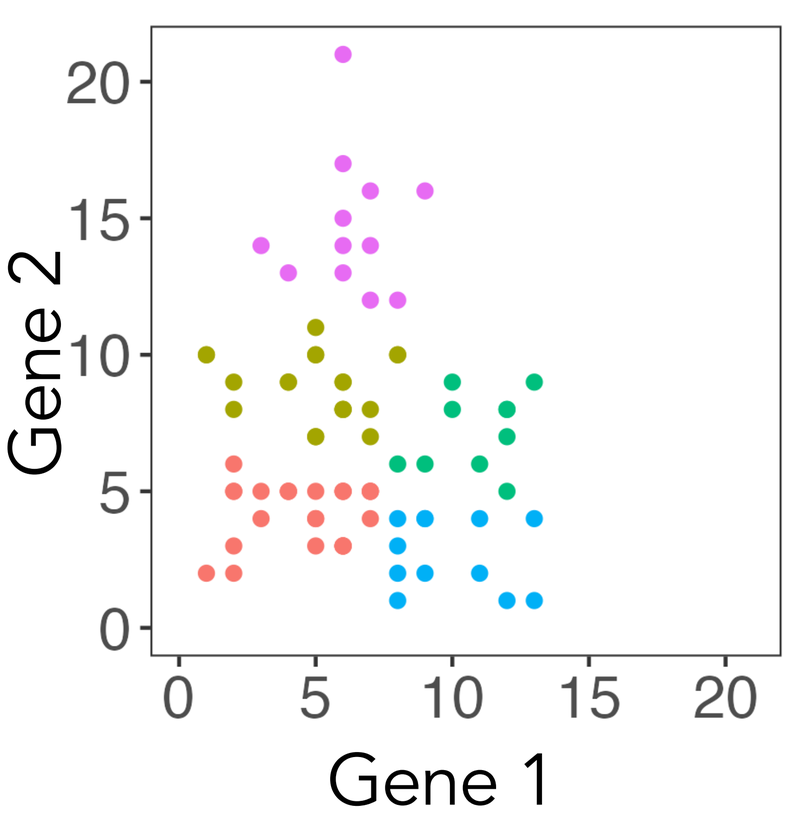
This cross validation procedure double dips

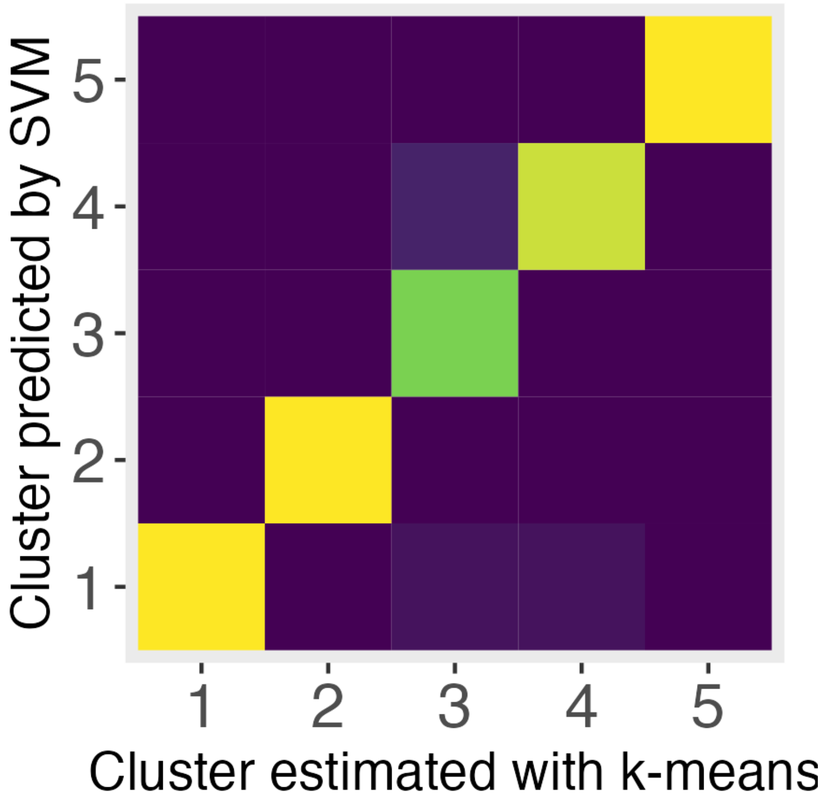


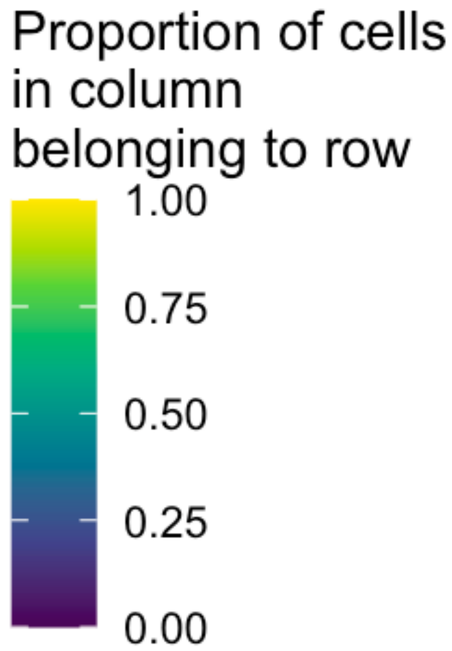
This cross validation procedure double dips



This cross validation procedure double dips

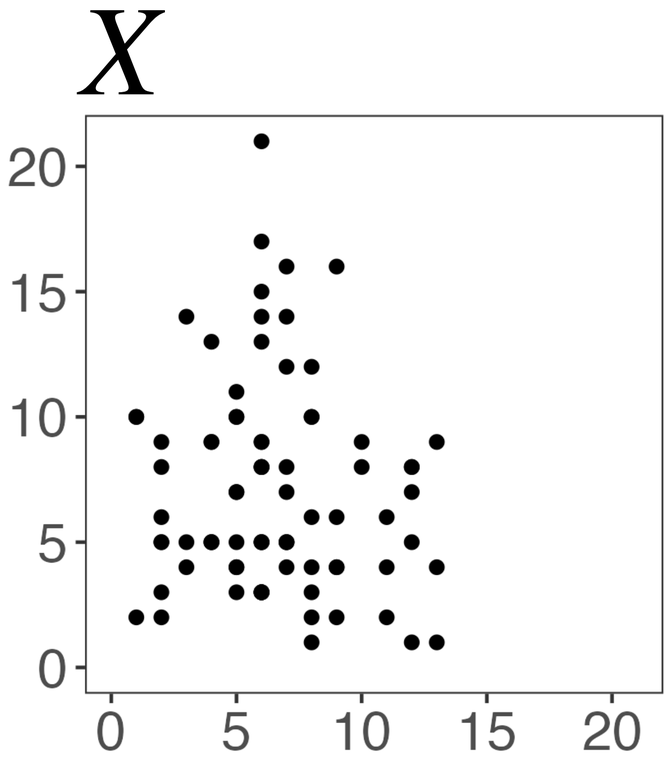




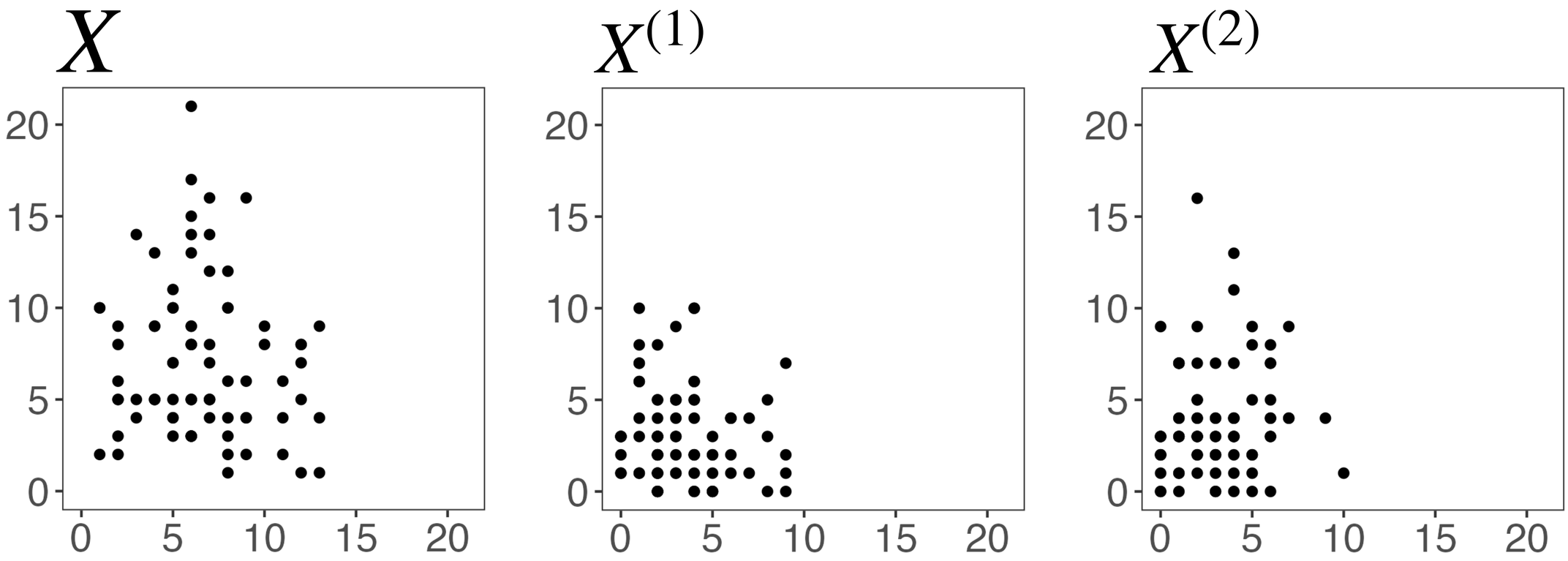


Classifier gets 96% accuracy to predict the five clusters, despite the fact that the five clusters are just random noise.

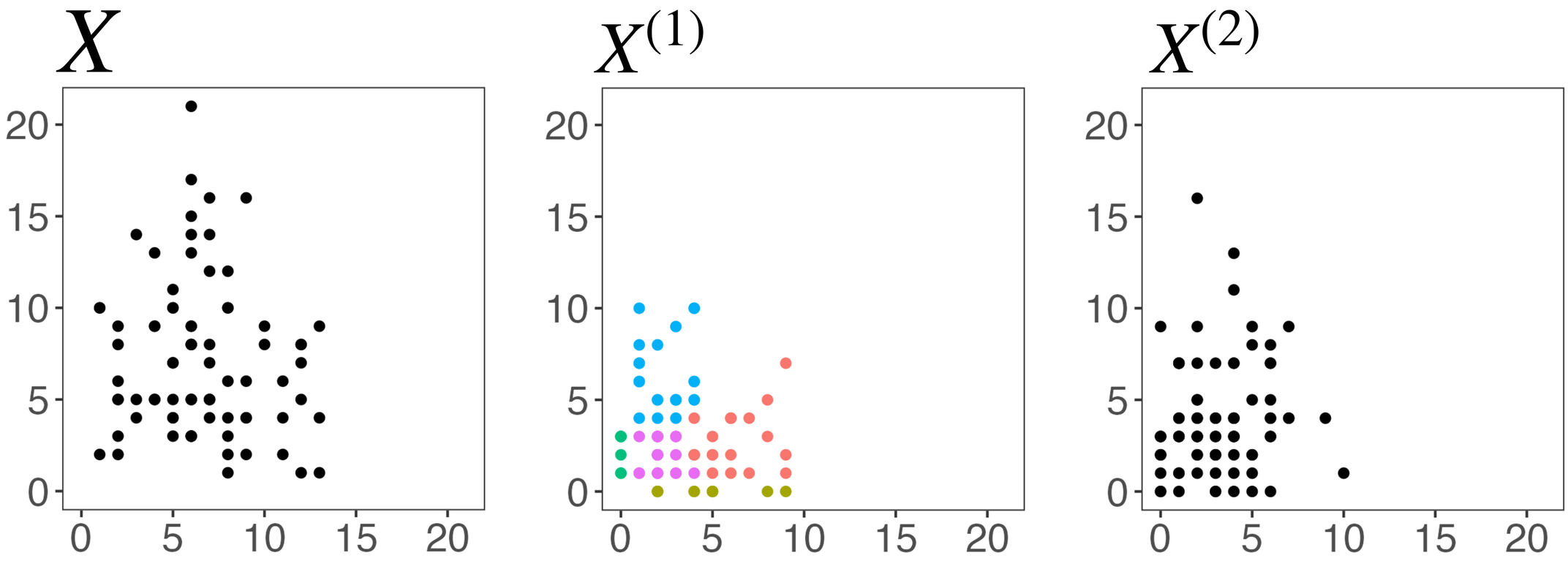
Data thinning provides a simple alternative



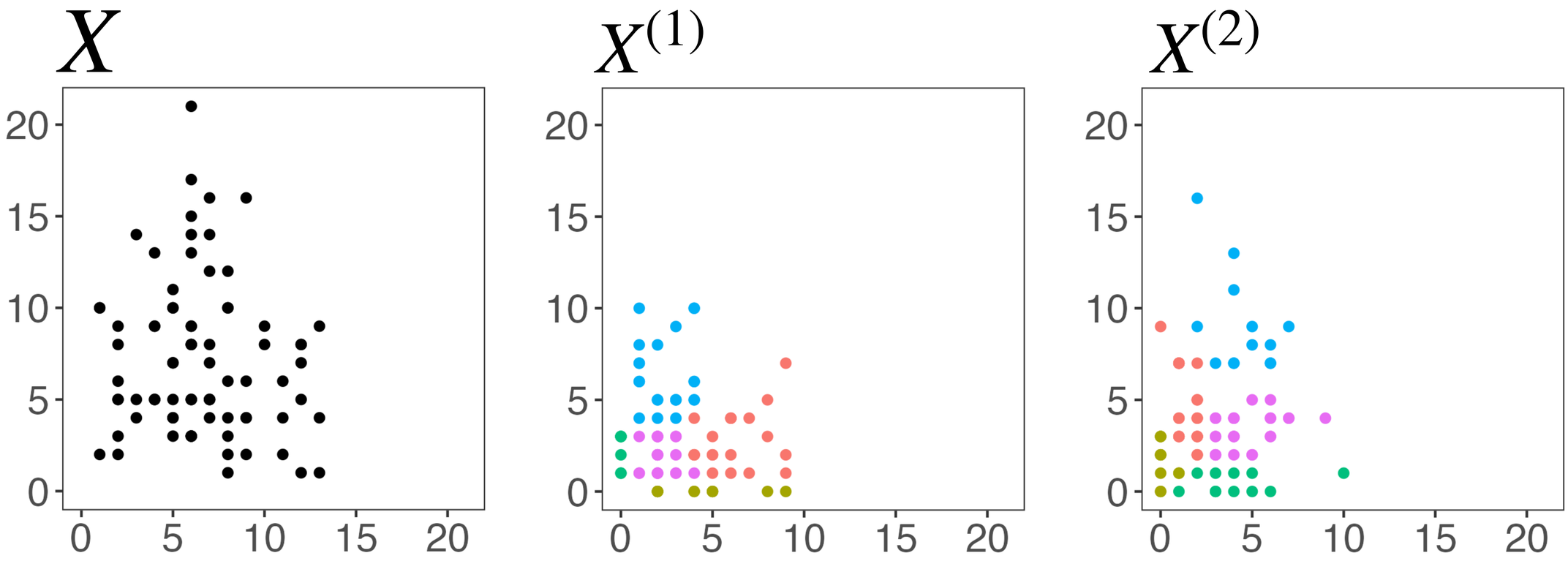
Data thinning provides a simple alternative



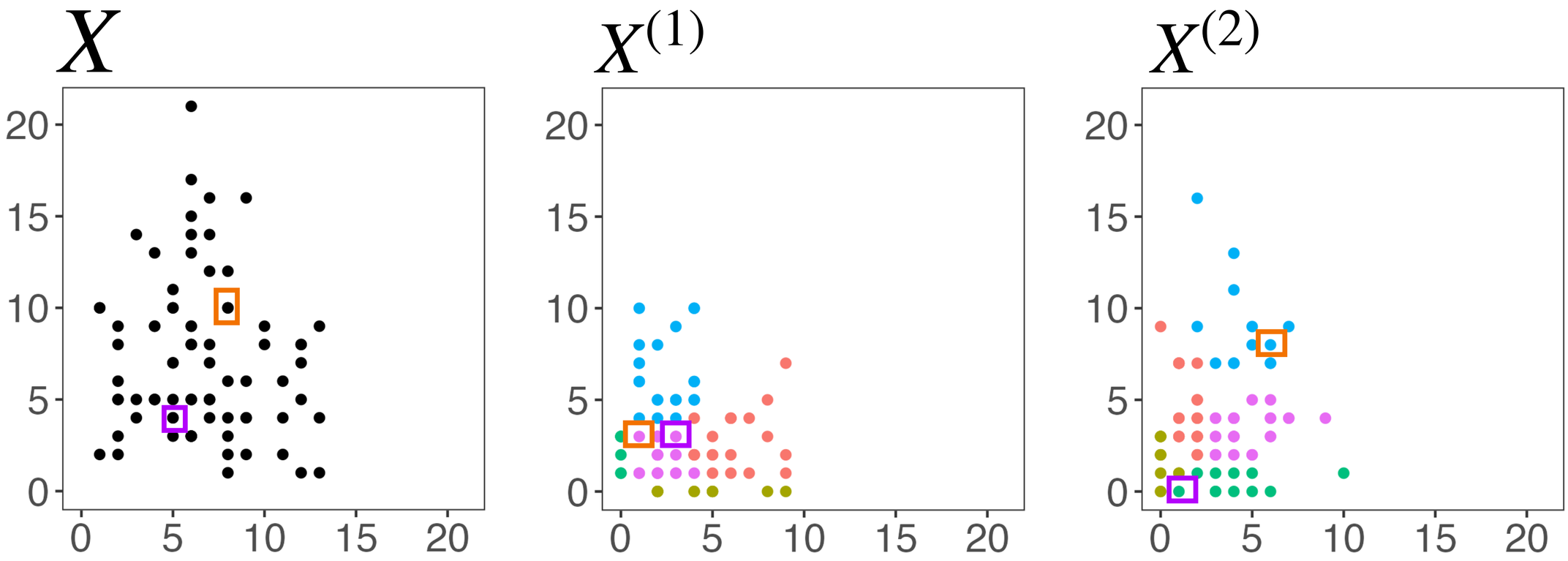
Data thinning provides a simple alternative



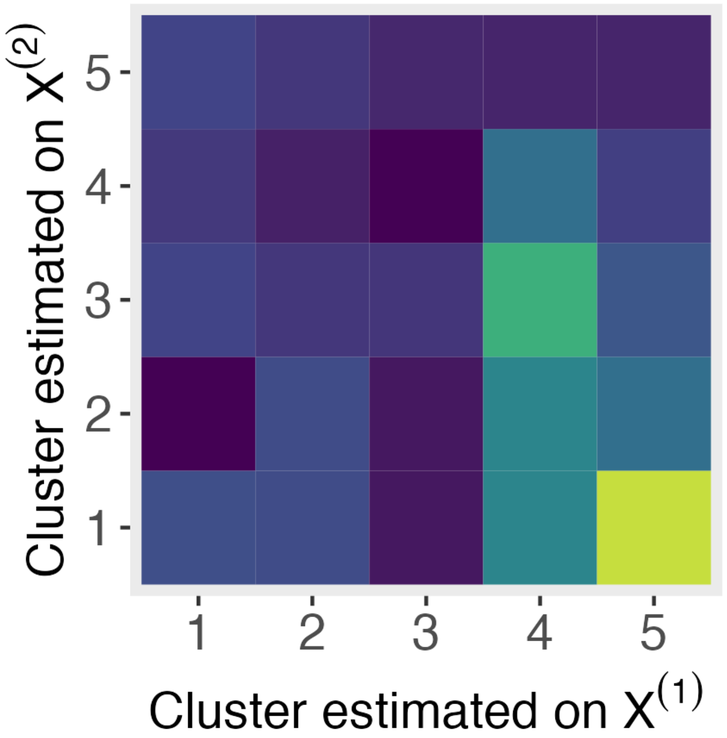
Data thinning provides a simple alternative

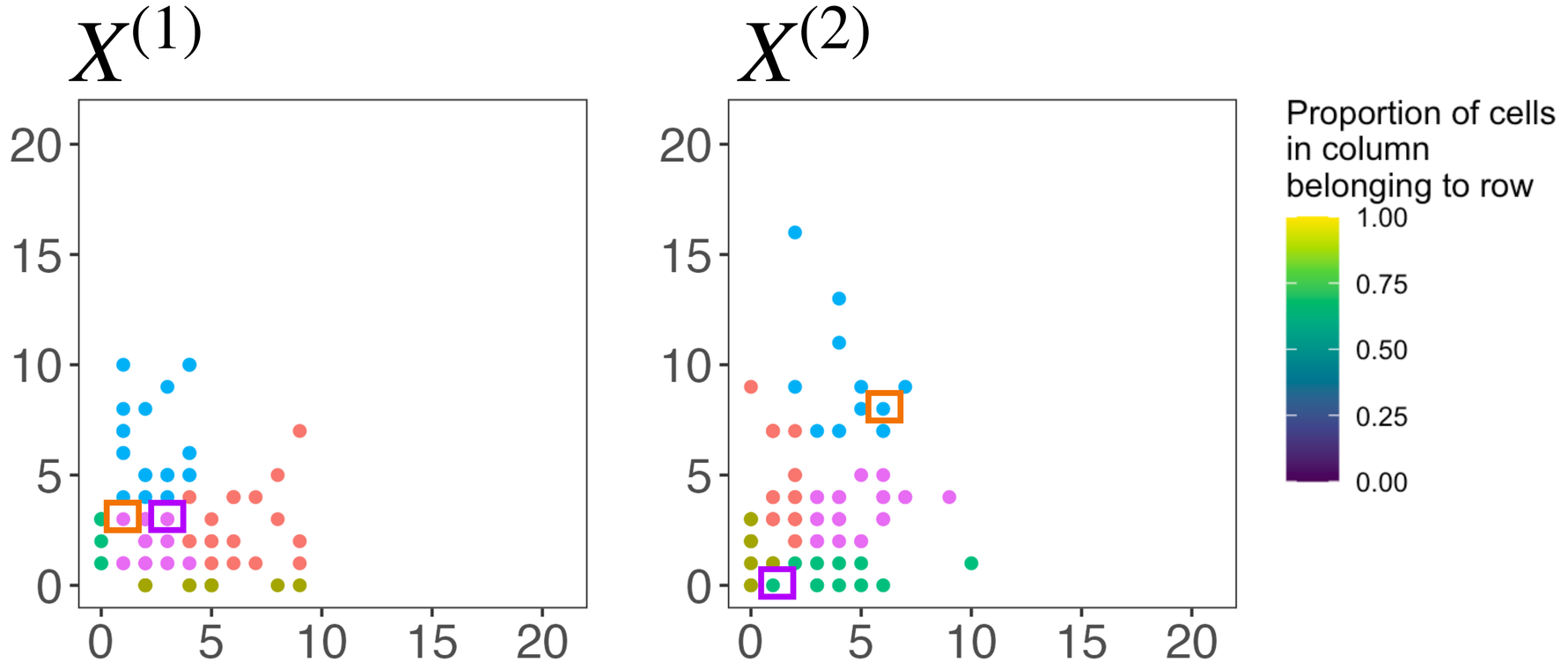


Data thinning provides a simple alternative



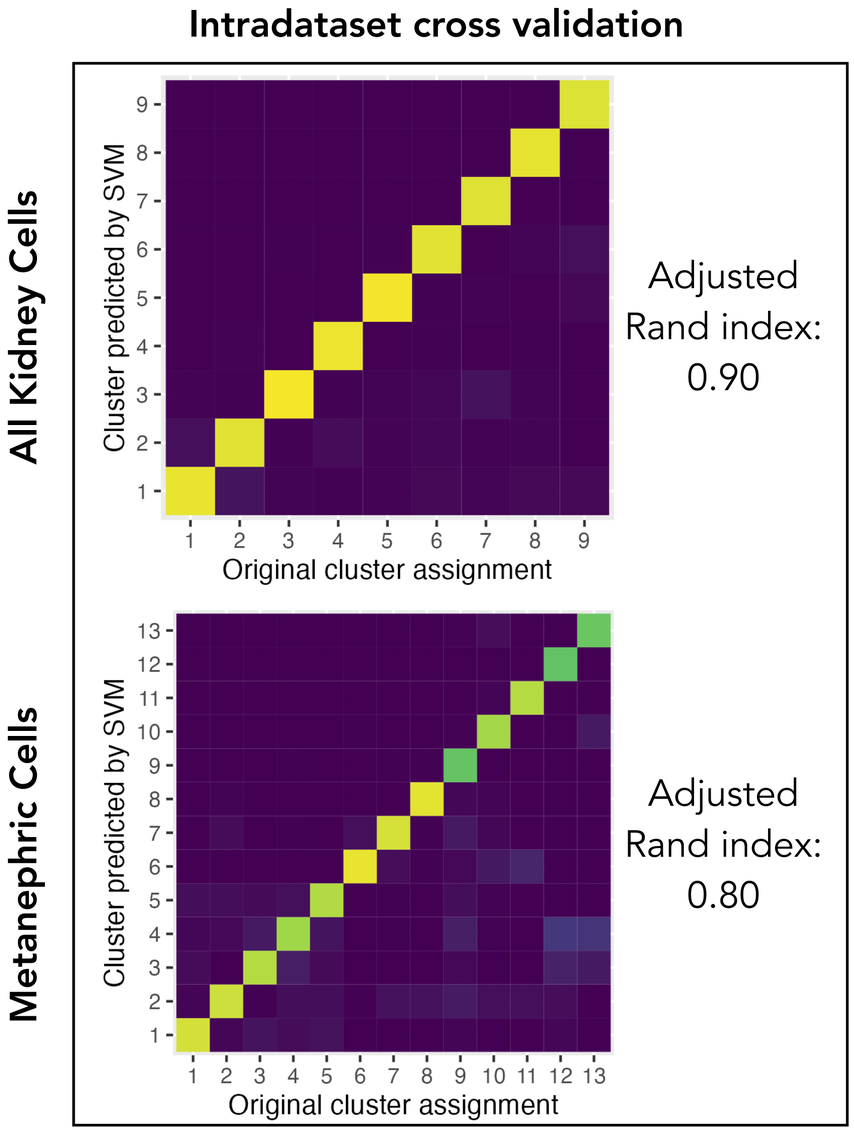
Data thinning provides a simple alternative

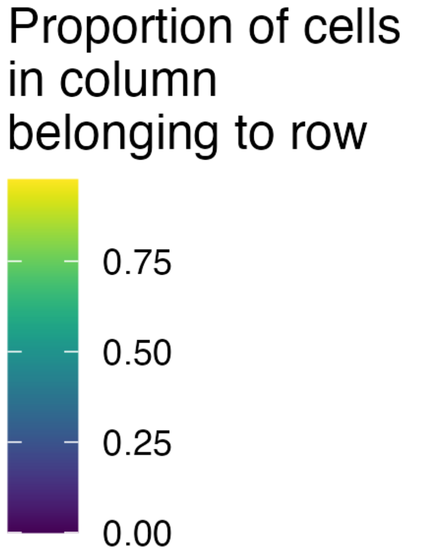




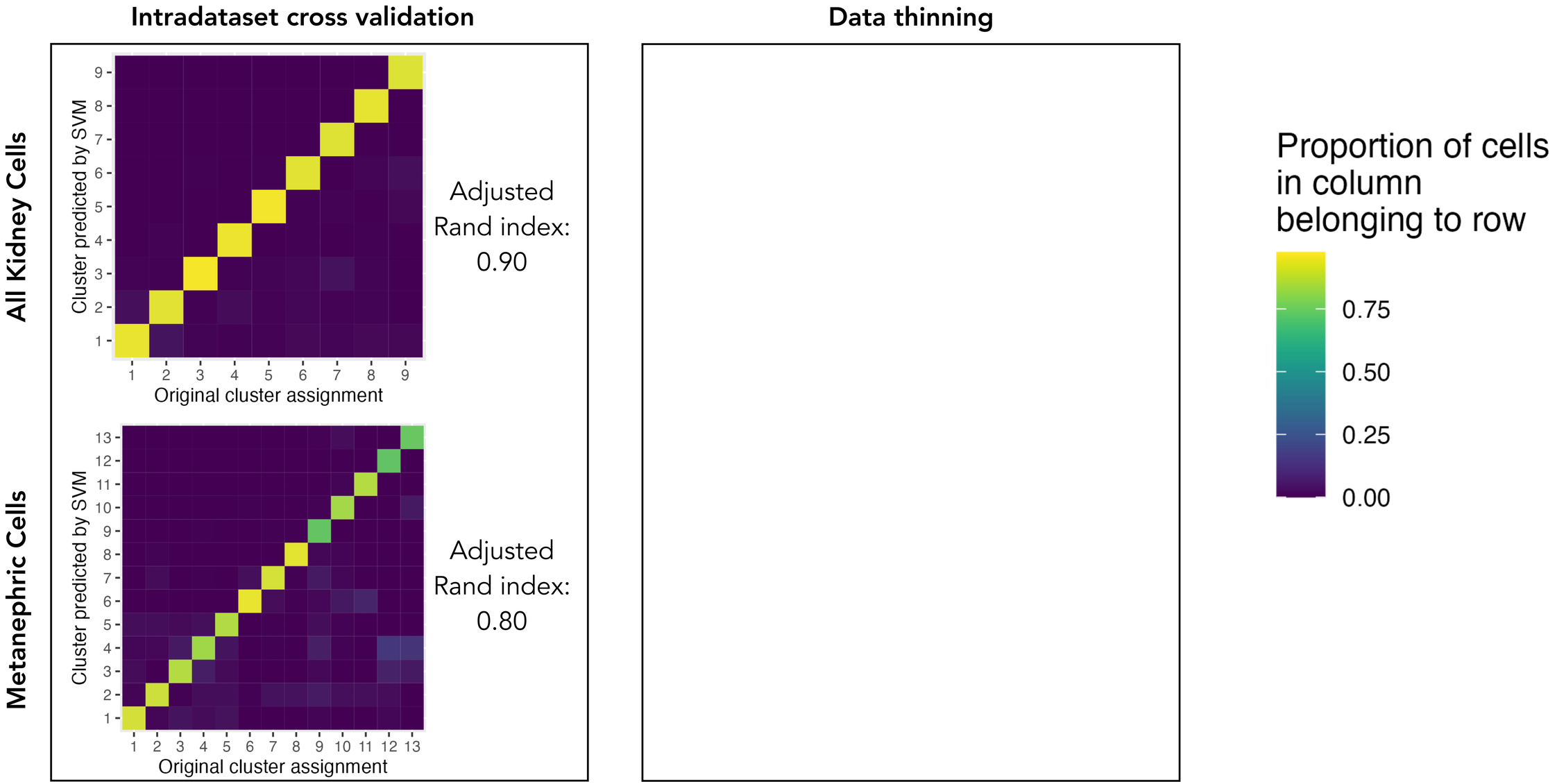
Adjusted Rand Index $= 0.01$

Re-analysis of Kidney cell data from fetal cell atlas

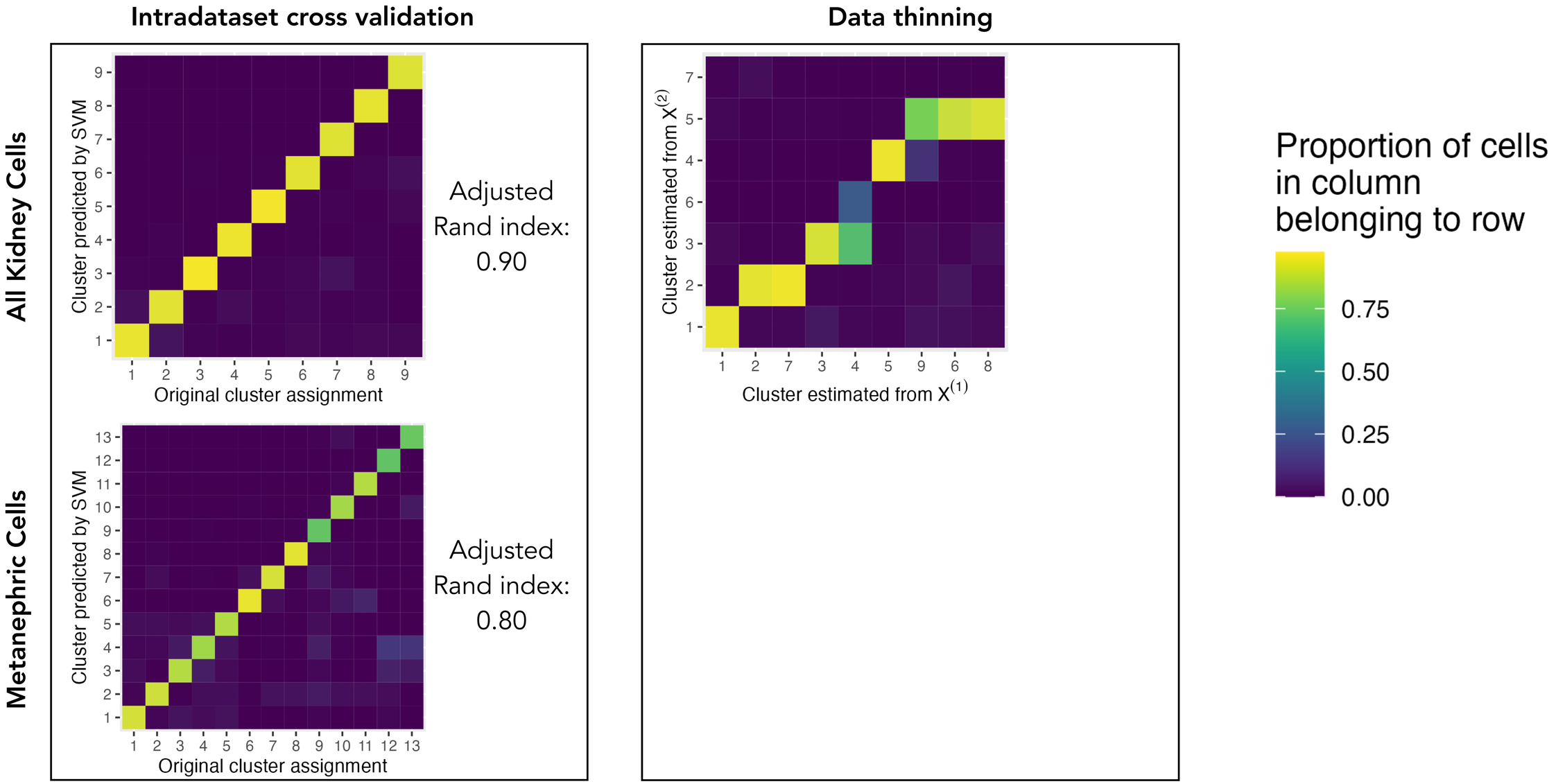




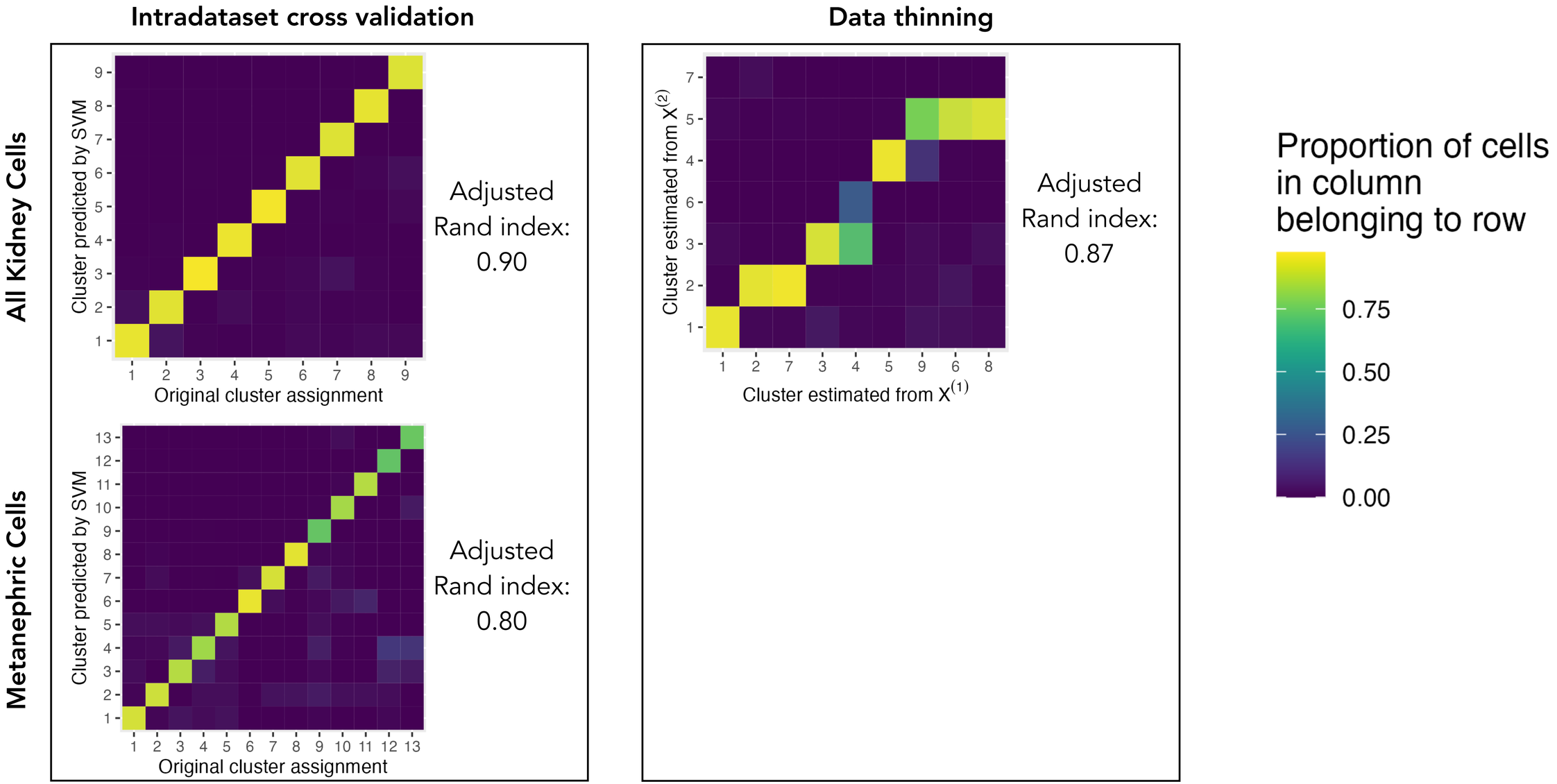
Re-analysis of Kidney cell data from fetal cell atlas



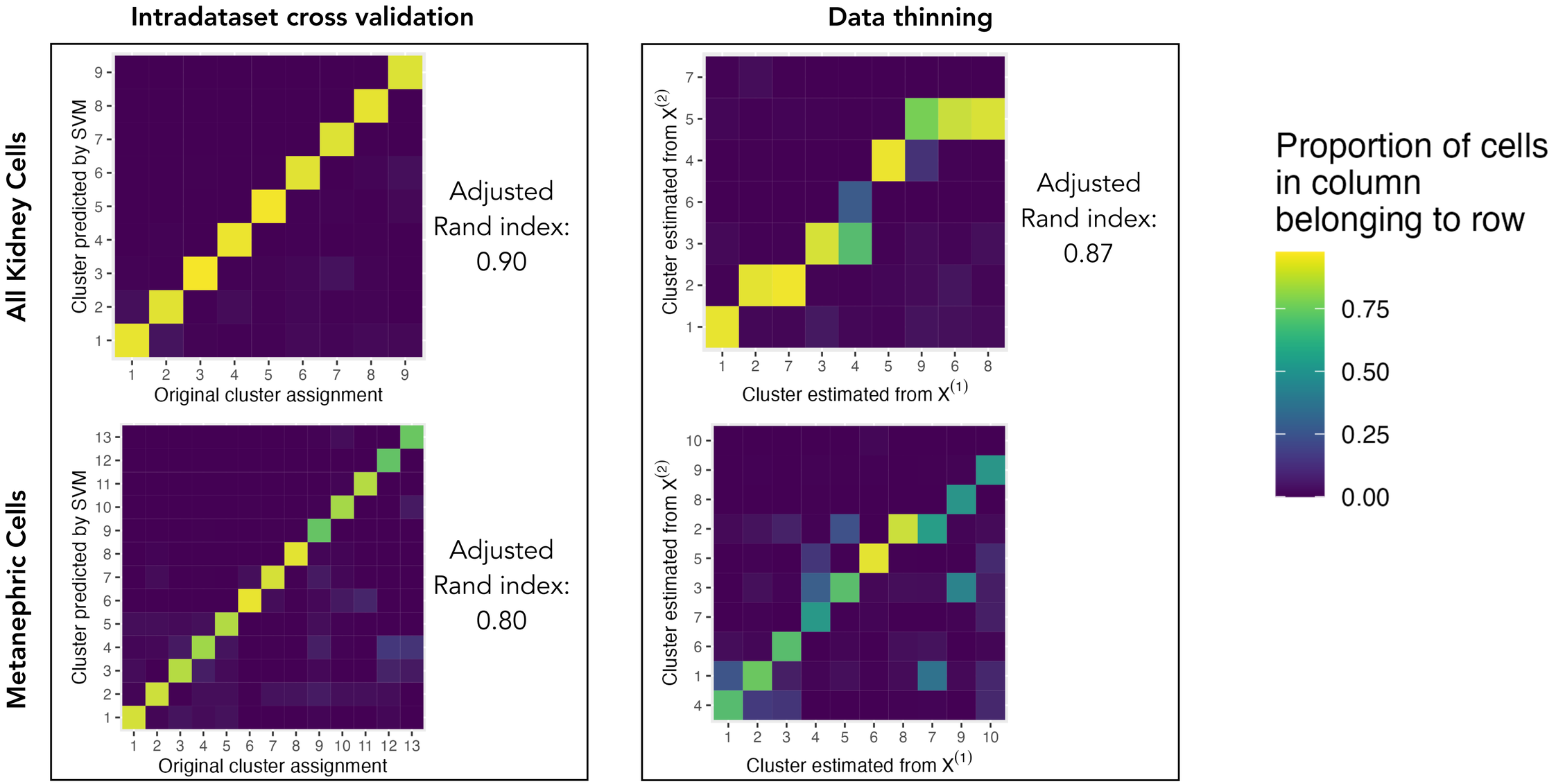
Re-analysis of Kidney cell data from fetal cell atlas



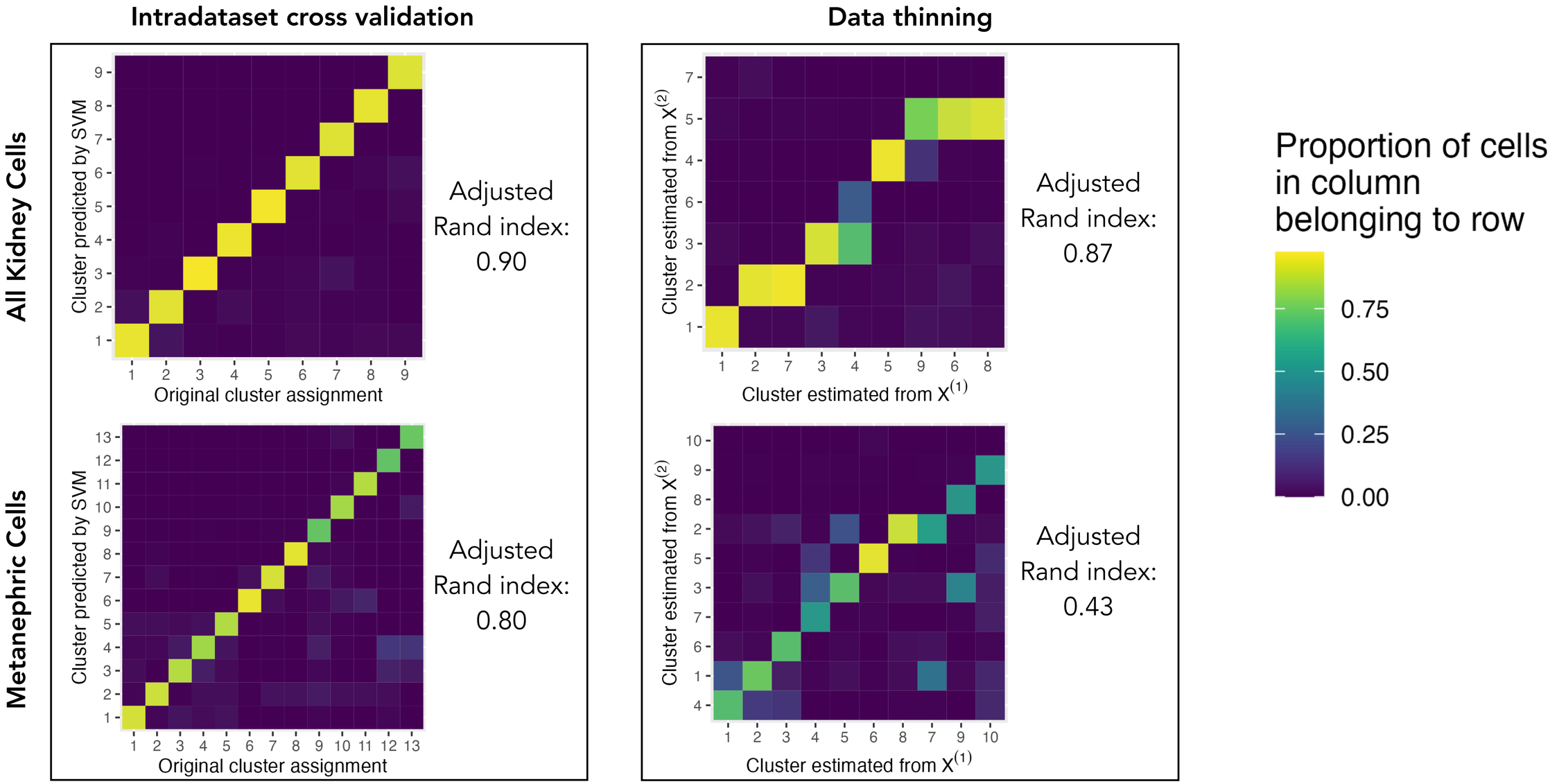
Re-analysis of Kidney cell data from fetal cell atlas



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 single-cell RNA sequencing data
5. **Ongoing work**

Three ways to avoid double dipping

1. Specialized methods, such as selective inference.

2. Sample splitting.

3. Data thinning.

Three ways to avoid double dipping

1. Specialized methods, such as selective inference.

   Requires a bespoke solution for every problem at hand.

2. Sample splitting.

3. Data thinning.

Three ways to avoid double dipping

1. Specialized methods, such as selective inference.

   Requires a bespoke solution for every problem at hand.

2. Sample splitting.

   Super flexible! Requires only an iid assumption on your data.

3. Data thinning.

Three ways to avoid double dipping

1. Specialized methods, such as selective inference.

   Requires a bespoke solution for every problem at hand.

2. Sample splitting.

   Super flexible! Requires only an iid assumption on your data.

   Not an option in unsupervised settings; unsatisfying in other settings.

3. Data thinning.

Three ways to avoid double dipping

1. Specialized methods, such as selective inference.

   Requires a bespoke solution for every problem at hand.

2. Sample splitting.

   Super flexible! Requires only an iid assumption on your data.

   Not an option in unsupervised settings; unsatisfying in other settings.

3. Data thinning.

   No bespoke solutions needed; works in supervised and unsupervised settings.

Three ways to avoid double dipping

1. Specialized methods, such as selective inference.

   Requires a bespoke solution for every problem at hand.

2. Sample splitting.

   Super flexible! Requires only an iid assumption on your data.

   Not an option in unsupervised settings; unsatisfying in other settings.

3. Data thinning.

   No bespoke solutions needed; works in supervised and unsupervised settings.

   Requires distributional assumptions and knowledge of nuisance parameters.

Three ways to avoid double dipping

1. Specialized methods, such as selective inference.

   Requires a bespoke solution for every problem at hand.

2. Sample splitting.

   Super flexible! Requires only an iid assumption on your data.

   Not an option in unsupervised settings; unsatisfying in other settings.

3. Data thinning.

   No bespoke solutions needed; works in supervised and unsupervised settings.

   Requires distributional assumptions and knowledge of nuisance parameters.

   Limited to convolution-closed distributions?

Revisiting the goals of data thinning

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

Revisiting the goals of data thinning

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

In our previous recipe:

(3) $X = X^{(1)} + X^{(2)}$.

Revisiting the goals of data thinning

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

In our previous recipe:

~~**(3)** $X = X^{(1)} + X^{(2)}$.~~

Revisiting the goals of data thinning

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

In our previous recipe:

~~**(3)** $X = X^{(1)} + X^{(2)}$.~~ **(3)** $X = T(X^{(1)}, X^{(2)})$.

Revisiting the goals of data thinning

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

In our previous recipe:

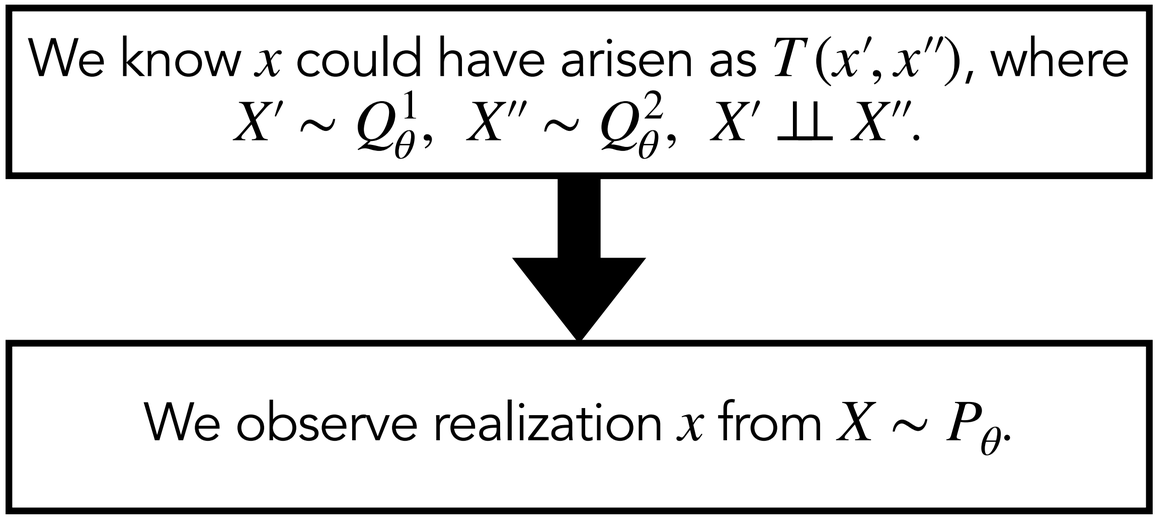
~~**(3)** $X = X^{(1)} + X^{(2)}$.~~ **(3)** $X = T(X^{(1)}, X^{(2)})$.

Generalized thinning with non-additive decompositions

Generalized thinning with non-additive decompositions

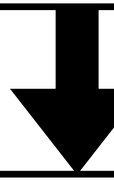
We observe realization x from $X \sim P_\theta$.

Generalized thinning with non-additive decompositions



Generalized thinning with non-additive decompositions

We know x could have arisen as $T(x', x'')$, where $X' \sim Q_\theta^1,\ X'' \sim Q_\theta^2,\ X' \perp\!\!\!\perp X''$.

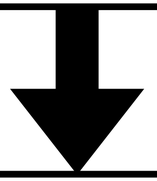


We observe realization x from $X \sim P_\theta$.

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

Generalized thinning with non-additive decompositions

We know x could have arisen as $T(x', x'')$, where $X' \sim Q_\theta^1,\ X'' \sim Q_\theta^2,\ X' \perp\!\!\!\perp X''$.

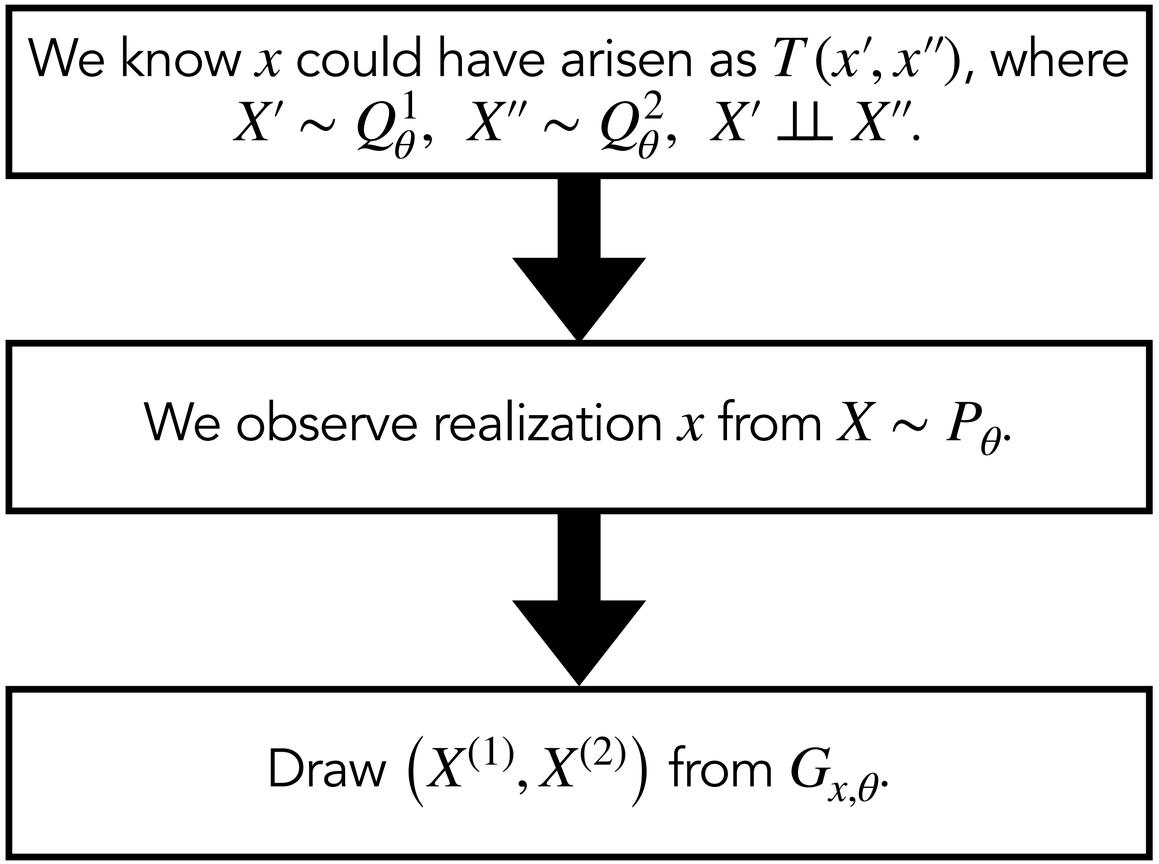


We observe realization x from $X \sim P_\theta$.

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

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

Generalized thinning with non-additive decompositions

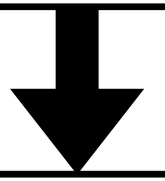


Can we work backwards to recover x' and x''?

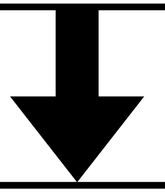
Let $G_{x,\theta}$ be the conditional distribution of $(X', X'') \mid X = x$.

Generalized thinning with non-additive decompositions

We know x could have arisen as $T(x', x'')$, where $X' \sim Q_\theta^1,\ X'' \sim Q_\theta^2,\ X' \perp\!\!\!\perp X''$.



We observe realization x from $X \sim P_\theta$.



Draw $\left(X^{(1)}, X^{(2)}\right)$ from $G_{x,\theta}$.

Theorem:

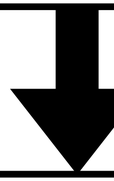
$X^{(1)} \sim Q_\theta^1, \quad X^{(2)} \sim Q_\theta^2, \quad X^{(1)} \perp\!\!\!\perp X^{(2)}$.

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

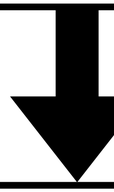
Let $G_{x,\theta}$ be the conditional distribution of $(X', X'') \mid X = x$.

Generalized thinning with non-additive decompositions

We know x could have arisen as $T(x', x'')$, where $X' \sim Q_\theta^1,\ X'' \sim Q_\theta^2,\ X' \perp\!\!\!\perp X''$.



We observe realization x from $X \sim P_\theta$.



Draw $\left(X^{(1)}, X^{(2)}\right)$ from $G_{x,\theta}$.

Theorem:
$X^{(1)} \sim Q_\theta^1, \quad X^{(2)} \sim Q_\theta^2, \quad X^{(1)} \perp\!\!\!\perp X^{(2)}$.

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

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

Key idea: If $X = T(X', X'')$ is sufficient for θ in the joint of (X', X''), then $G_{x,\theta}$ does not depend on θ.

The list of distributions we can thin is extensive

Family	Distribution P_θ, where $X \sim P_\theta$.	Distribution $Q_\theta^{(k)}$ where $X^{(k)} \overset{ind.}{\sim} Q_\theta^{(k)}$.	Sufficient statistic T (sufficient for θ)
Natural exponential family (in parameter θ)	$N(\theta, \sigma^2)$	$\mathrm{N}(\epsilon_k\theta, \epsilon_k\sigma^2)$	$\sum_{k=1}^K X^{(k)}$
	$\mathrm{Poisson}(\theta)$	$\mathrm{Poisson}(\epsilon_k\theta)$	
	$\mathrm{NegBin}(r, \theta)$	$\mathrm{NegBin}(\epsilon_k r, \theta)$	
	$\mathrm{Binomial}(r, \theta)$	$\mathrm{Binomial}(\epsilon_k r, \theta)$	
	$\mathrm{Gamma}(\alpha, \theta)$	$\mathrm{Gamma}(\epsilon_k\alpha, \theta)$	
	$\mathrm{N}_p(\boldsymbol{\theta}, \Sigma)$	$N_p(\epsilon_k\boldsymbol{\theta}, \epsilon_k\Sigma)$	
	$\mathrm{Multinomial}_p(r, \boldsymbol{\theta})$	$\mathrm{Multinomial}_p(\epsilon_k r, \boldsymbol{\theta})$	
	$\mathrm{Gamma}(K/2, \theta)$	$N(0, \frac{1}{2\theta})$	$\sum_{k=1}^K \left(X^{(k)}\right)^2$
	$\mathrm{Gamma}(K, \theta)$	$\mathrm{Weibull}(\theta^{-\frac{1}{\nu}}, \nu)$	$\sum_{k=1}^K \left(X^{(k)}\right)^\nu$
General exponential family (in parameter θ)	$\mathrm{Beta}(\theta, \beta)$	$\mathrm{Beta}\left(\frac{1}{K}\theta + \frac{k-1}{K}, \frac{1}{K}\beta\right)$	$\left(\Pi_{k=1}^K X^{(k)}\right)^{1/K}$
	$\mathrm{Beta}(\alpha, \theta)$	$\mathrm{Beta}\left(\frac{1}{K}\alpha, \frac{1}{K}\theta + \frac{k-1}{K}\right)$	$\left(\Pi_{k=1}^K \left(1 - X^{(k)}\right)\right)^{1/K}$
	$\mathrm{Gamma}(\theta, \beta)$	$\mathrm{Gamma}(\frac{1}{K}\theta + \frac{k-1}{K}, \frac{1}{K}\beta)$	$\left(\Pi_{k=1}^K X^{(k)}\right)^{1/K}$
	$\mathrm{Weibull}(\theta, \nu)$	$\mathrm{Gamma}(\frac{1}{K}, \theta^{-\nu})$	$\left(\sum_{k=1}^K X^{(k)}\right)^{1/\nu}$
	$\mathrm{Pareto}(\nu, \theta)$	$\mathrm{Gamma}(\frac{1}{K}, \theta)$	$\nu \times \mathrm{Exp}\left(\sum_{k=1}^K X^{(k)}\right)$
	$\mathrm{N}(0,\theta)$	$\mathrm{Gamma}(\frac{1}{2K}, \frac{1}{2\theta})$	$X^2 = \sum_{k=1}^K X^{(k)}$
	$\mathrm{N}_K(\theta_1 1_K, \theta_2 I_K)$	$N(\theta_1, \theta_2)$	sample mean and variance
Truncated support family	$\mathrm{Unif}(0, \theta)$	$\theta \cdot \mathrm{Beta}(\frac{1}{K}, 1)$	$\max\left(X^{(1)}, \ldots, X^{(K)}\right)$
	$\theta \cdot \mathrm{Beta}(\alpha, 1)$	$\theta \cdot \mathrm{Beta}(\frac{\alpha}{K}, 1)$	
	$\theta + \mathrm{Exp}(\lambda)$	$\theta + \mathrm{Exp}(\lambda/\mathrm{K})$	$\min\left(X^{(1)}, \ldots, X^{(K)}\right)$
Non-parametric	F^n	F^{n_k}	$\mathrm{sort}(X^{(1)}, \ldots, X^{(K)})$

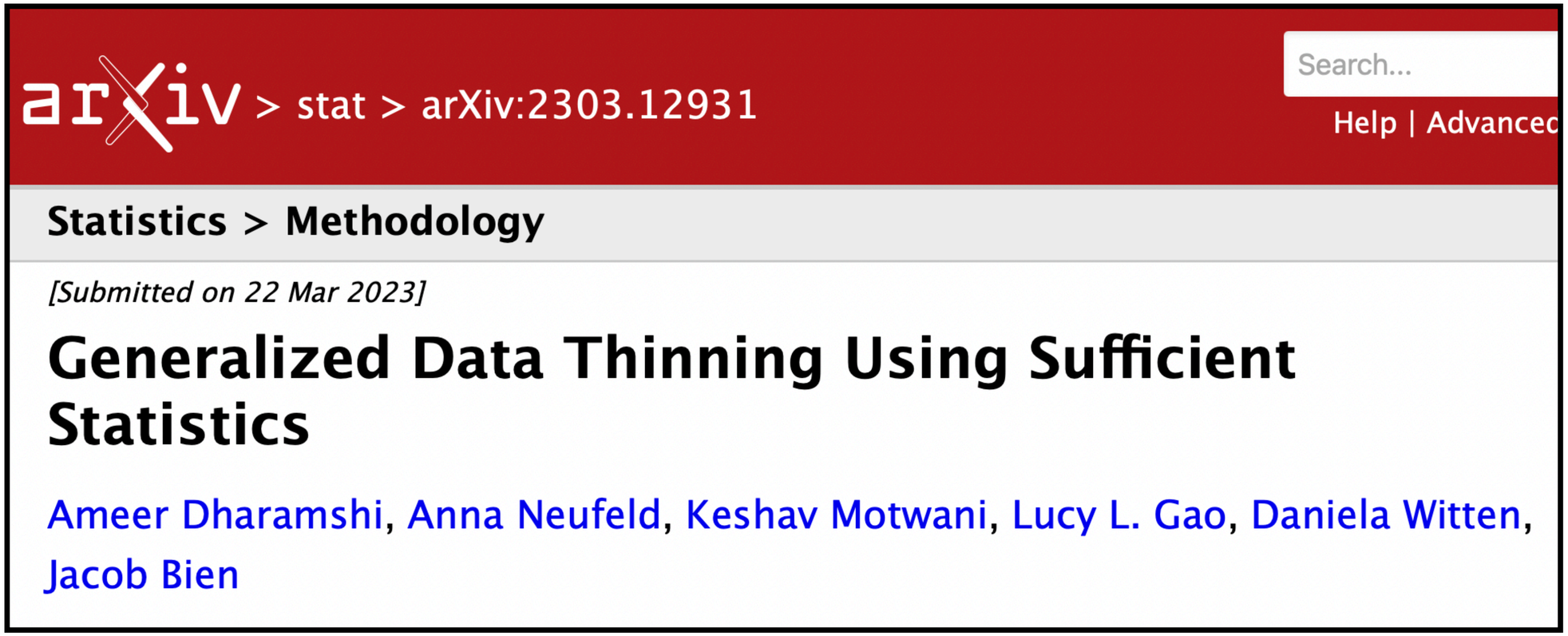
The list of distributions we can thin is extensive

Family	Distribution P_θ, where $X \sim P_\theta$.	Distribution $Q_\theta^{(k)}$ where $X^{(k)} \overset{ind.}{\sim} Q_\theta^{(k)}$.	Sufficient statistic T (sufficient for θ)
Natural exponential family (in parameter θ)	$N(\theta, \sigma^2)$	$\mathrm{N}(\epsilon_k\theta, \epsilon_k\sigma^2)$	$\sum_{k=1}^K X^{(k)}$
	$\mathrm{Poisson}(\theta)$	$\mathrm{Poisson}(\epsilon_k\theta)$	
	$\mathrm{NegBin}(r, \theta)$	$\mathrm{NegBin}(\epsilon_k r, \theta)$	
	$\mathrm{Binomial}(r, \theta)$	$\mathrm{Binomial}(\epsilon_k r, \theta)$	
	$\mathrm{Gamma}(\alpha, \theta)$	$\mathrm{Gamma}(\epsilon_k\alpha, \theta)$	
	$\mathrm{N}_p(\boldsymbol{\theta}, \Sigma)$	$N_p(\epsilon_k\boldsymbol{\theta}, \epsilon_k\Sigma)$	
	$\mathrm{Multinomial}_p(r, \boldsymbol{\theta})$	$\mathrm{Multinomial}_p(\epsilon_k r, \boldsymbol{\theta})$	
	$\mathrm{Gamma}(K/2, \theta)$	$N(0, \frac{1}{2\theta})$	$\sum_{k=1}^K \left(X^{(k)}\right)^2$
	$\mathrm{Gamma}(K, \theta)$	$\mathrm{Weibull}(\theta^{-\frac{1}{\nu}}, \nu)$	$\sum_{k=1}^K \left(X^{(k)}\right)^\nu$
General exponential family (in parameter θ)	$\mathrm{Beta}(\theta, \beta)$	$\mathrm{Beta}\left(\frac{1}{K}\theta + \frac{k-1}{K}, \frac{1}{K}\beta\right)$	$\left(\Pi_{k=1}^K X^{(k)}\right)^{1/K}$
	$\mathrm{Beta}(\alpha, \theta)$	$\mathrm{Beta}\left(\frac{1}{K}\alpha, \frac{1}{K}\theta + \frac{k-1}{K}\right)$	$\left(\Pi_{k=1}^K \left(1 - X^{(k)}\right)\right)^{1/K}$
	$\mathrm{Gamma}(\theta, \beta)$	$\mathrm{Gamma}(\frac{1}{K}\theta + \frac{k-1}{K}, \frac{1}{K}\beta)$	$\left(\Pi_{k=1}^K X^{(k)}\right)^{1/K}$
	$\mathrm{Weibull}(\theta, \nu)$	$\mathrm{Gamma}(\frac{1}{K}, \theta^{-\nu})$	$\left(\sum_{k=1}^K X^{(k)}\right)^{1/\nu}$
	$\mathrm{Pareto}(\nu, \theta)$	$\mathrm{Gamma}(\frac{1}{K}, \theta)$	$\nu \times \mathrm{Exp}\left(\sum_{k=1}^K X^{(k)}\right)$
	$\mathrm{N}(0,\theta)$	$\mathrm{Gamma}(\frac{1}{2K}, \frac{1}{2\theta})$	$X^2 = \sum_{k=1}^K X^{(k)}$
	$\mathrm{N}_K(\theta_1 1_K, \theta_2 I_K)$	$N(\theta_1, \theta_2)$	sample mean and variance
Truncated support family	$\mathrm{Unif}(0, \theta)$	$\theta \cdot \mathrm{Beta}(\frac{1}{K}, 1)$	$\max\left(X^{(1)}, \ldots, X^{(K)}\right)$
	$\theta \cdot \mathrm{Beta}(\alpha, 1)$	$\theta \cdot \mathrm{Beta}(\frac{\alpha}{K}, 1)$	
	$\theta + \mathrm{Exp}(\lambda)$	$\theta + \mathrm{Exp}(\lambda/\mathrm{K})$	$\min\left(X^{(1)}, \ldots, X^{(K)}\right)$
Non-parametric	F^n	F^{n_k}	$\mathrm{sort}(X^{(1)}, \ldots, X^{(K)})$

The list of distributions we can thin is extensive

Family	Distribution P_θ, where $X \sim P_\theta$.	Distribution $Q_\theta^{(k)}$ where $X^{(k)} \overset{ind.}{\sim} Q_\theta^{(k)}$.	Sufficient statistic T (sufficient for θ)
Natural exponential family (in parameter θ)	$N(\theta, \sigma^2)$	$\mathrm{N}(\epsilon_k\theta, \epsilon_k\sigma^2)$	$\sum_{k=1}^K X^{(k)}$
	$\mathrm{Poisson}(\theta)$	$\mathrm{Poisson}(\epsilon_k\theta)$	
	$\mathrm{NegBin}(r, \theta)$	$\mathrm{NegBin}(\epsilon_k r, \theta)$	
	$\mathrm{Binomial}(r, \theta)$	$\mathrm{Binomial}(\epsilon_k r, \theta)$	
	$\mathrm{Gamma}(\alpha, \theta)$	$\mathrm{Gamma}(\epsilon_k\alpha, \theta)$	
	$\mathrm{N}_p(\boldsymbol{\theta}, \Sigma)$	$N_p(\epsilon_k\boldsymbol{\theta}, \epsilon_k\Sigma)$	
	$\mathrm{Multinomial}_p(r, \boldsymbol{\theta})$	$\mathrm{Multinomial}_p(\epsilon_k r, \boldsymbol{\theta})$	
	$\mathrm{Gamma}(K/2, \theta)$	$N(0, \frac{1}{2\theta})$	$\sum_{k=1}^K \left(X^{(k)}\right)^2$
	$\mathrm{Gamma}(K, \theta)$	$\mathrm{Weibull}(\theta^{-\frac{1}{\nu}}, \nu)$	$\sum_{k=1}^K \left(X^{(k)}\right)^\nu$
General exponential family (in parameter θ)	$\mathrm{Beta}(\theta, \beta)$	$\mathrm{Beta}\left(\frac{1}{K}\theta + \frac{k-1}{K}, \frac{1}{K}\beta\right)$	$\left(\Pi_{k=1}^K X^{(k)}\right)^{1/K}$
	$\mathrm{Beta}(\alpha, \theta)$	$\mathrm{Beta}\left(\frac{1}{K}\alpha, \frac{1}{K}\theta + \frac{k-1}{K}\right)$	$\left(\Pi_{k=1}^K \left(1 - X^{(k)}\right)\right)^{1/K}$
	$\mathrm{Gamma}(\theta, \beta)$	$\mathrm{Gamma}(\frac{1}{K}\theta + \frac{k-1}{K}, \frac{1}{K}\beta)$	$\left(\Pi_{k=1}^K X^{(k)}\right)^{1/K}$
	$\mathrm{Weibull}(\theta, \nu)$	$\mathrm{Gamma}(\frac{1}{K}, \theta^{-\nu})$	$\left(\sum_{k=1}^K X^{(k)}\right)^{1/\nu}$
	$\mathrm{Pareto}(\nu, \theta)$	$\mathrm{Gamma}(\frac{1}{K}, \theta)$	$\nu \times \mathrm{Exp}\left(\sum_{k=1}^K X^{(k)}\right)$
	$\mathrm{N}(0,\theta)$	$\mathrm{Gamma}(\frac{1}{2K}, \frac{1}{2\theta})$	$X^2 = \sum_{k=1}^K X^{(k)}$
	$\mathrm{N}_K(\theta_1 1_K, \theta_2 I_K)$	$N(\theta_1, \theta_2)$	sample mean and variance
Truncated support family	$\mathrm{Unif}(0, \theta)$	$\theta \cdot \mathrm{Beta}(\frac{1}{K}, 1)$	$\max\left(X^{(1)}, \ldots, X^{(K)}\right)$
	$\theta \cdot \mathrm{Beta}(\alpha, 1)$	$\theta \cdot \mathrm{Beta}(\frac{\alpha}{K}, 1)$	
	$\theta + \mathrm{Exp}(\lambda)$	$\theta + \mathrm{Exp}(\lambda/\mathrm{K})$	$\min\left(X^{(1)}, \ldots, X^{(K)}\right)$
Non-parametric	F^n	F^{n_k}	$\mathrm{sort}(X^{(1)}, \ldots, X^{(K)})$

We are working on additional extensions and applications



Acknowledgements

Daniela Witten
University of Washington

Lucy Gao
University of British Columbia

Ameer Dharamshi
University of Washington

Keshav Motwani
University of Washington

Alexis Battle
Johns Hopkins

Joshua Popp
Johns Hopkins

Jacob Bien
USC

Questions?