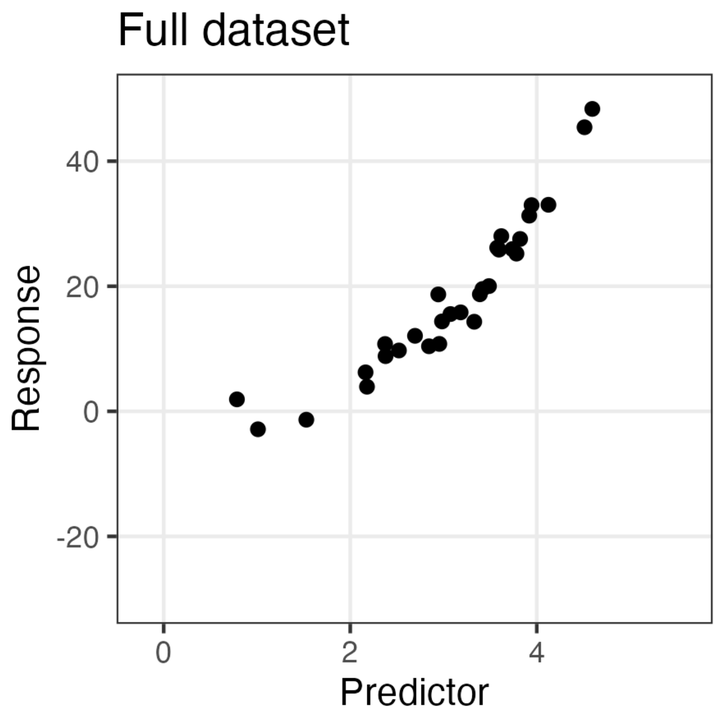
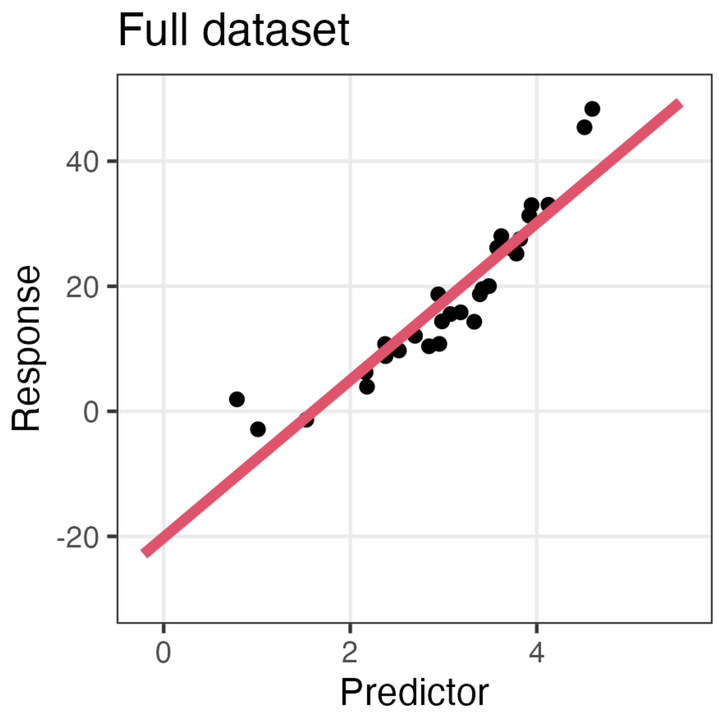
Data thinning to avoid double dipping

Anna Neufeld
Postdoctoral Research Fellow
Fred Hutchinson Cancer Center
November 2, 2023

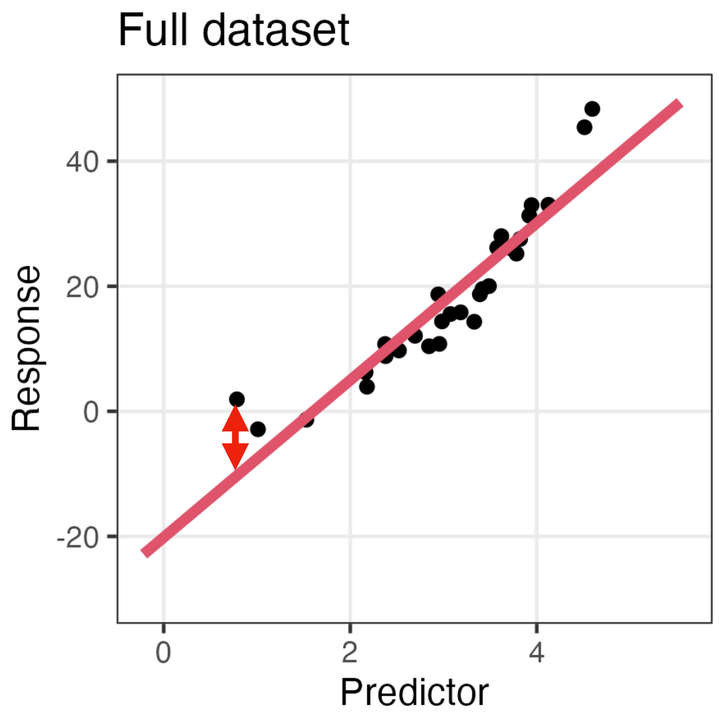
A familiar example: selecting the degree for a polynomial regression model



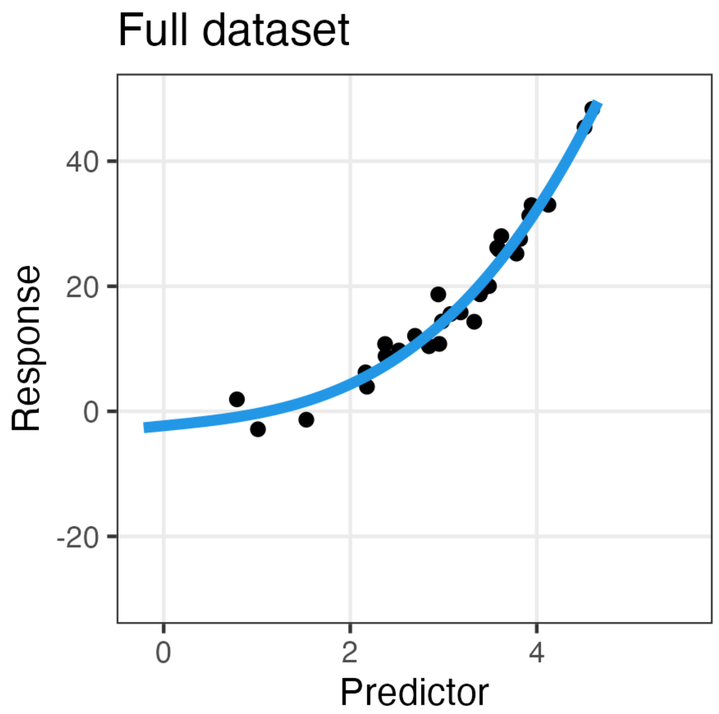
A familiar example: selecting the degree for a polynomial regression model



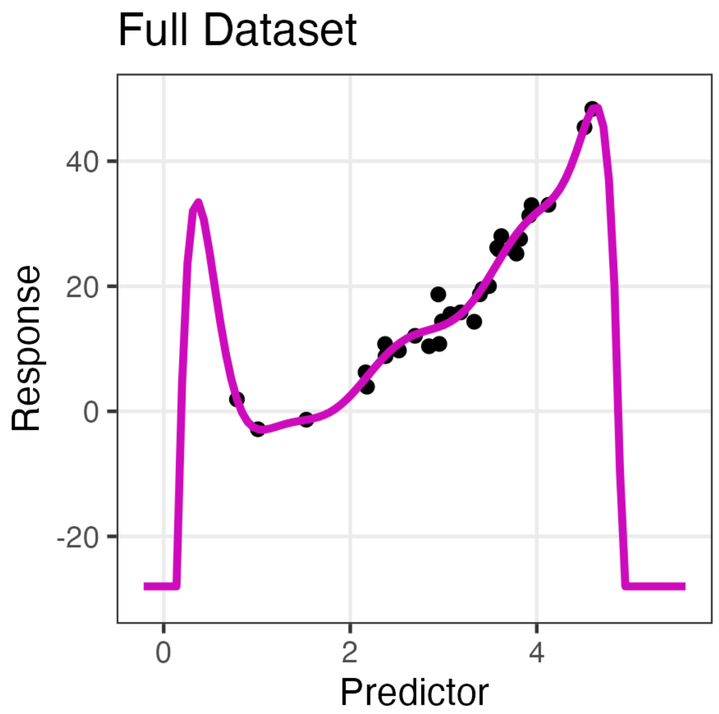
A familiar example: selecting the degree for a polynomial regression model



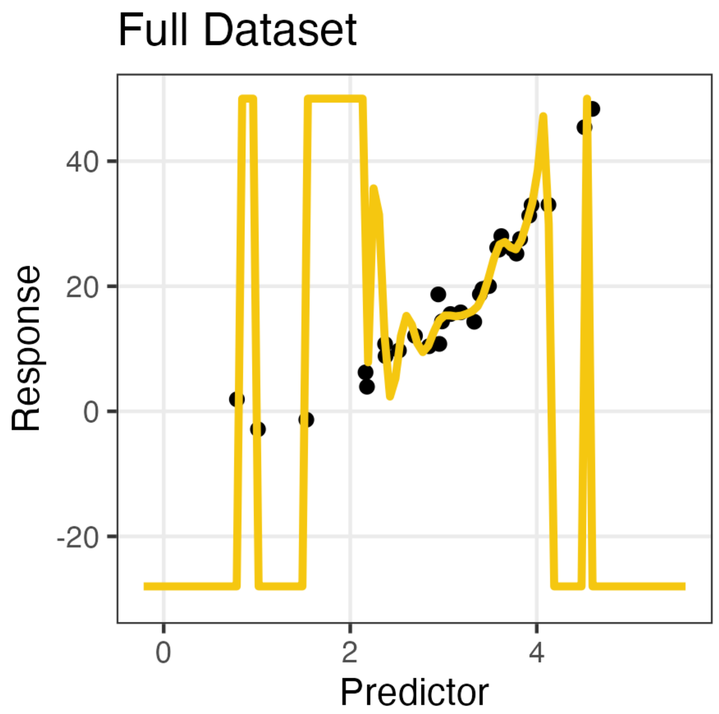
A familiar example: selecting the degree for a polynomial regression model



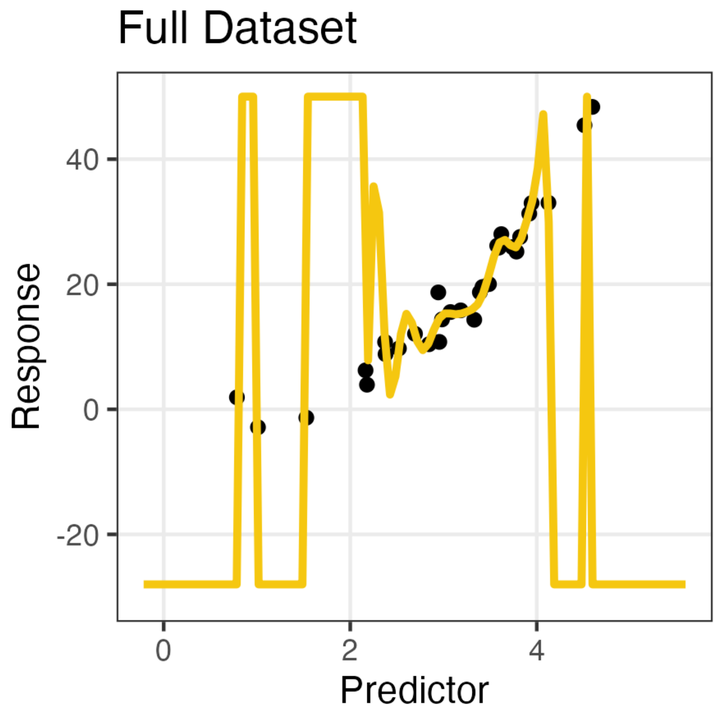
A familiar example: selecting the degree for a polynomial regression model

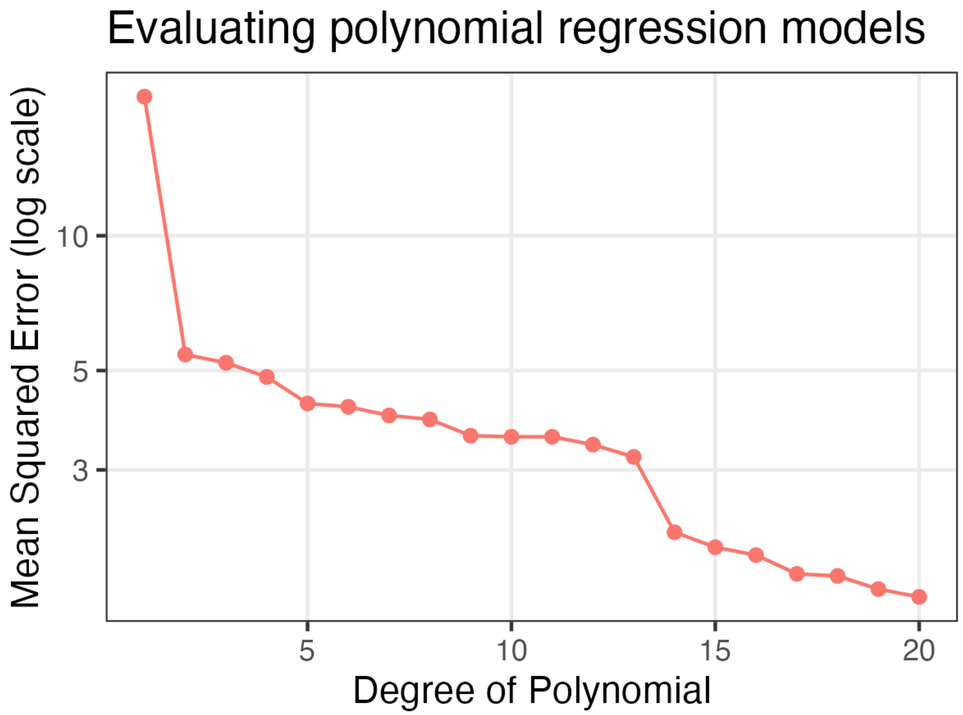


A familiar example: selecting the degree for a polynomial regression model

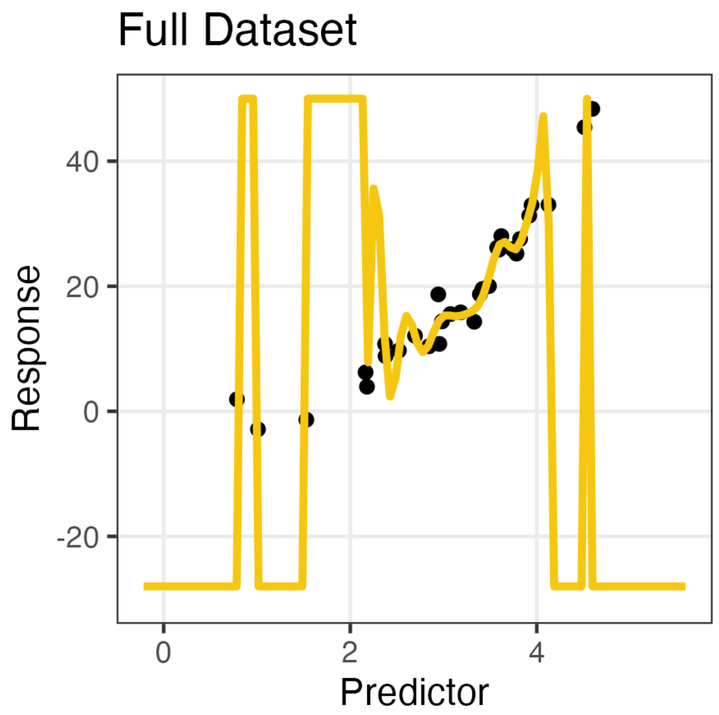


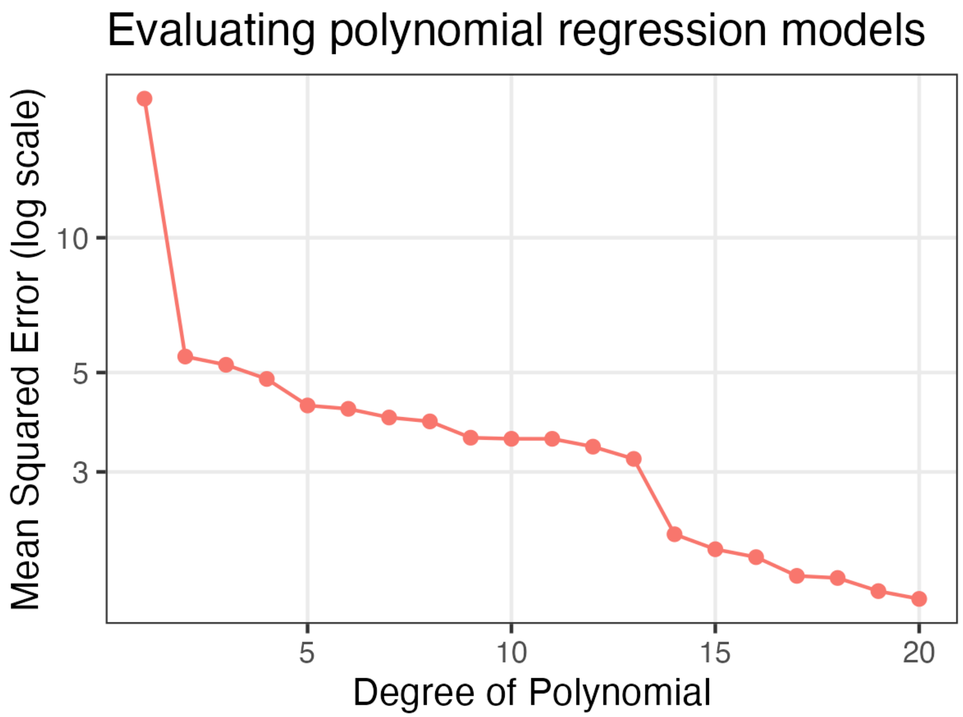
A familiar example: selecting the degree for a polynomial regression model





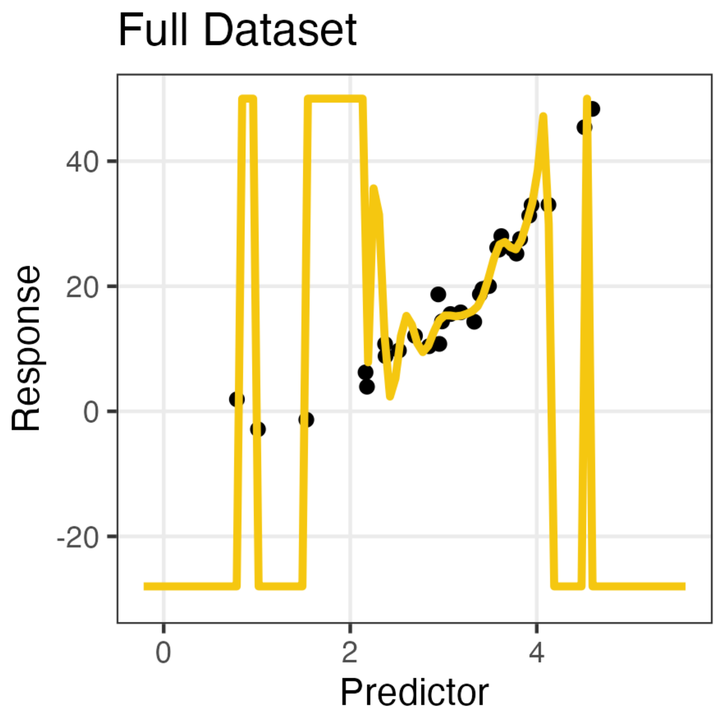
A familiar example: selecting the degree for a polynomial regression model

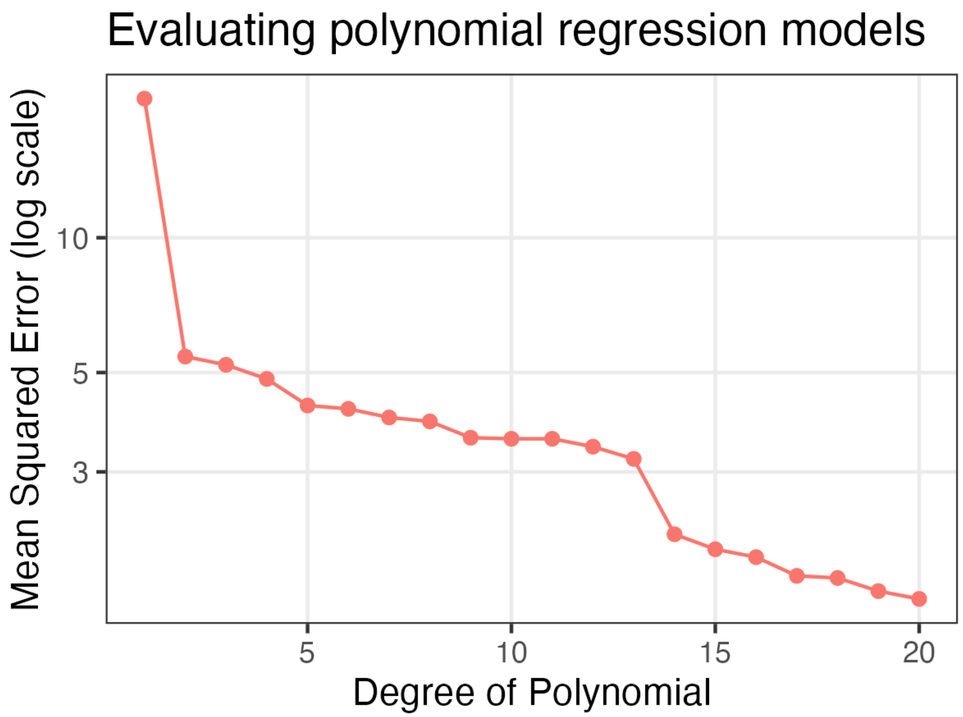




When we use the same data to fit and evaluate a model, more complex models appear better.

A familiar example: selecting the degree for a polynomial regression model

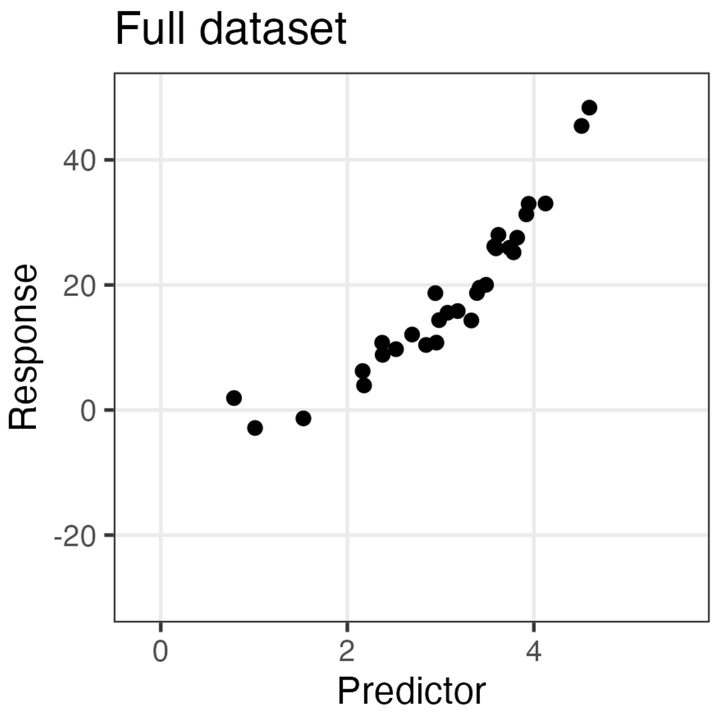




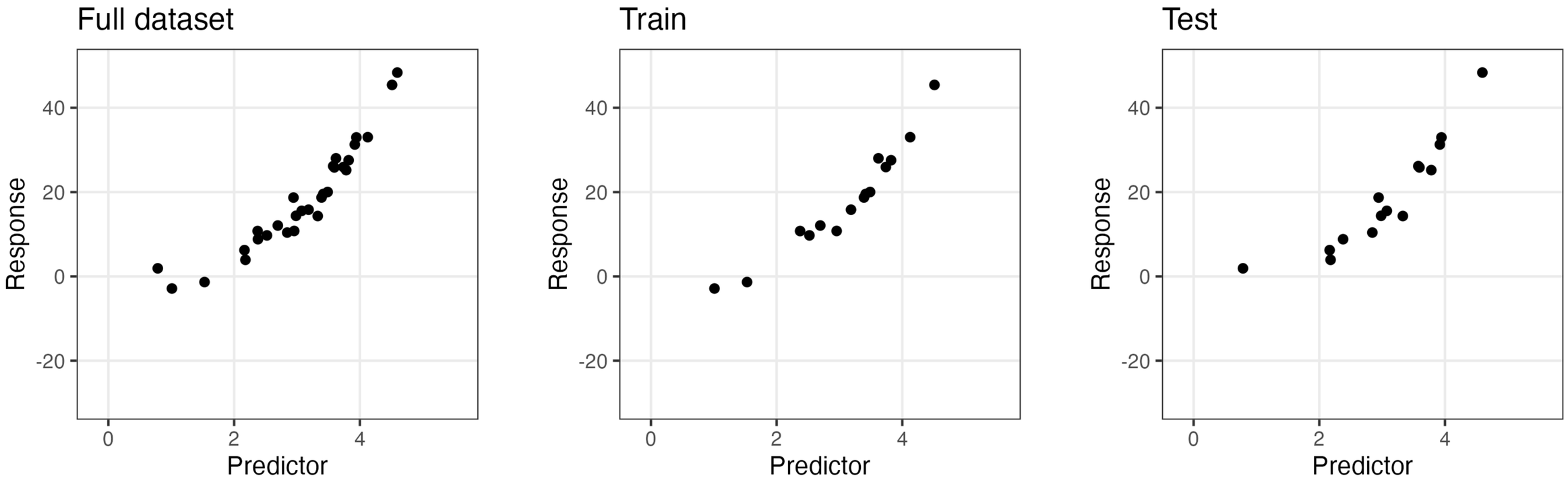
Double Dipping: Using the same data for two tasks, such as:

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

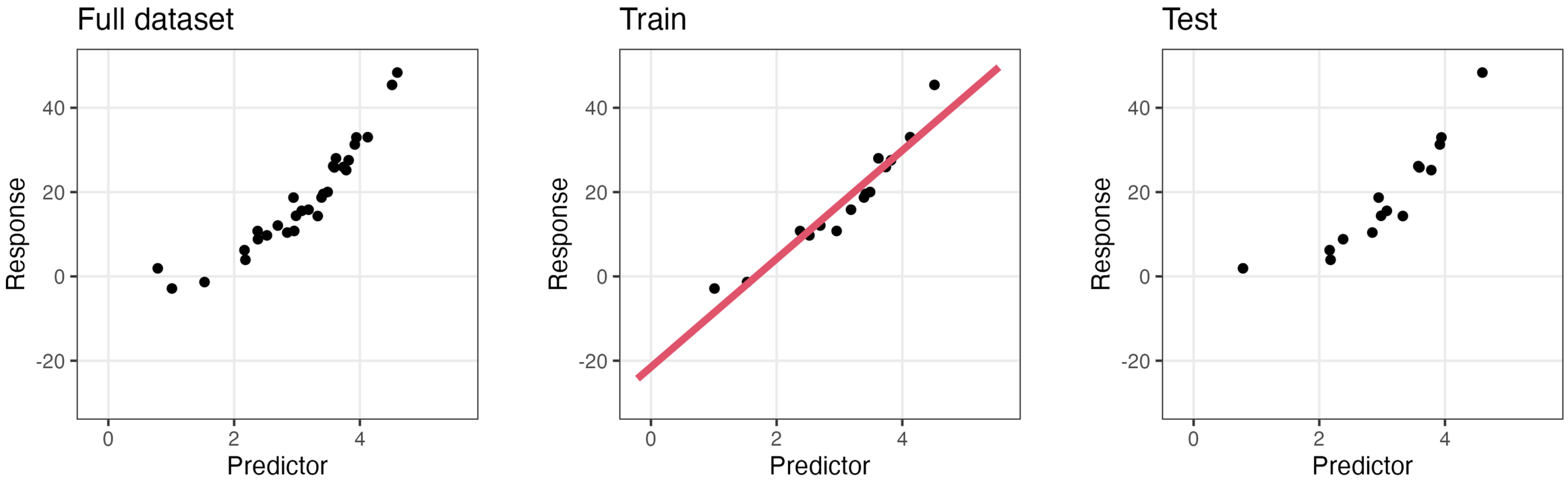
We can often avoid double dipping through sample splitting



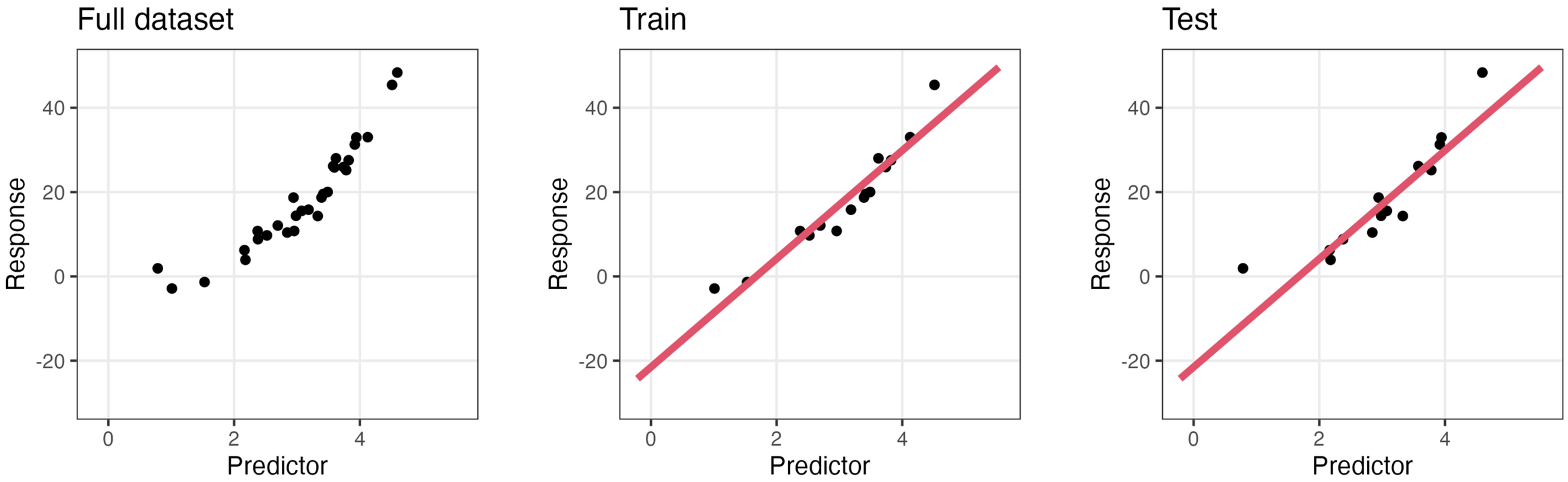
We can often avoid double dipping through sample splitting



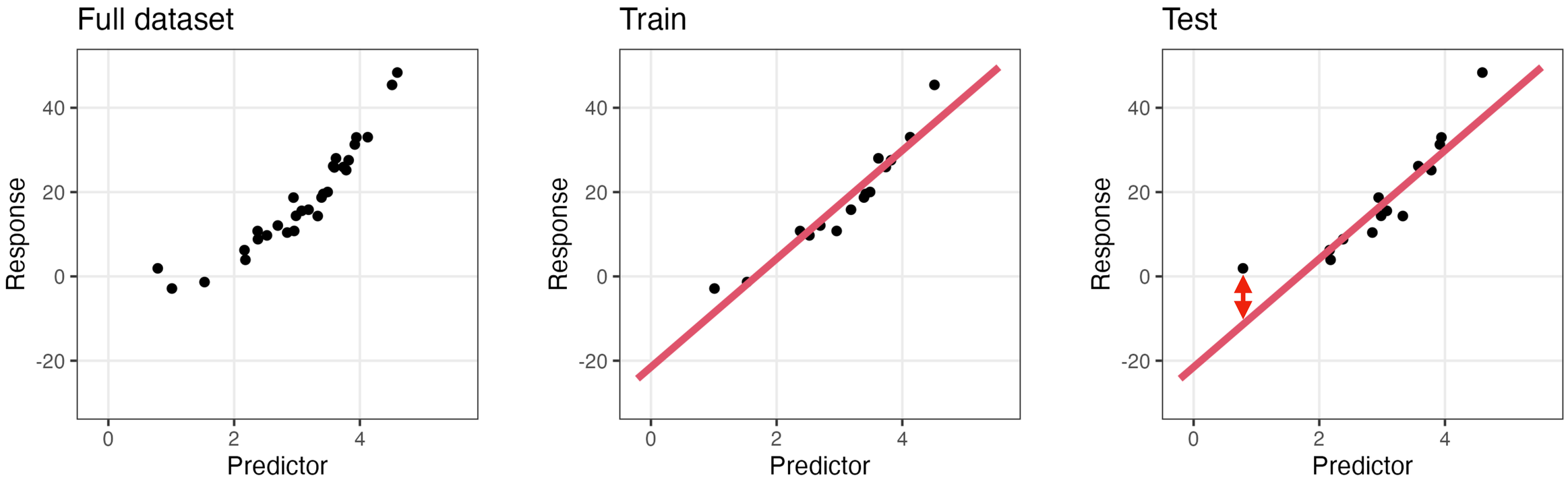
We can often avoid double dipping through sample splitting



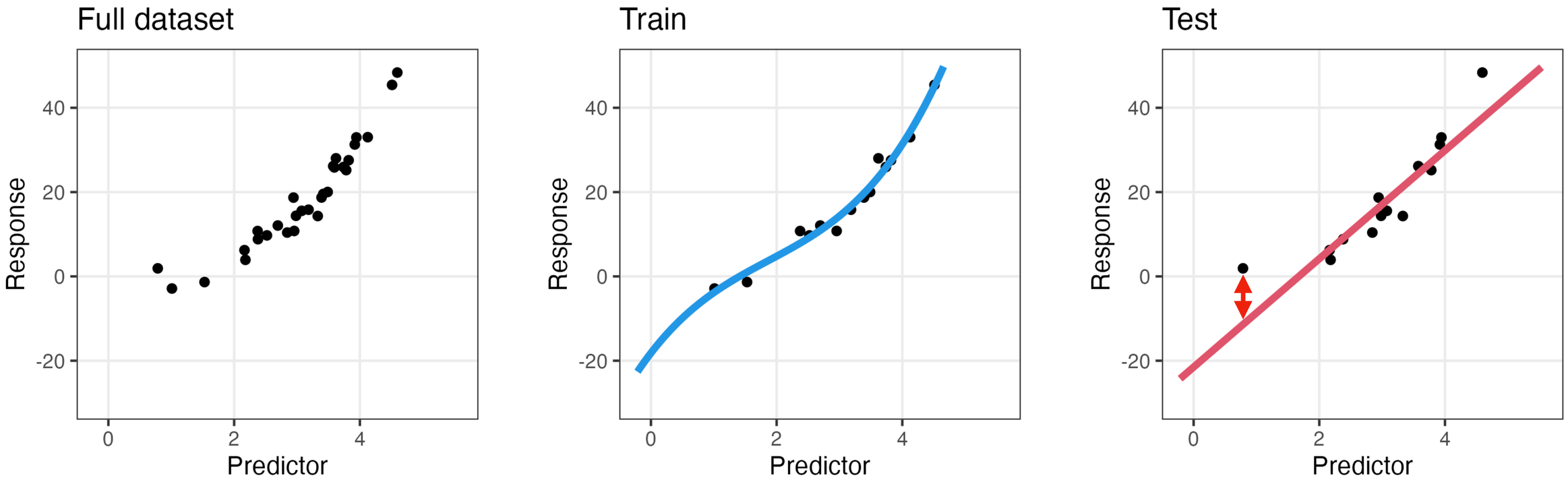
We can often avoid double dipping through sample splitting



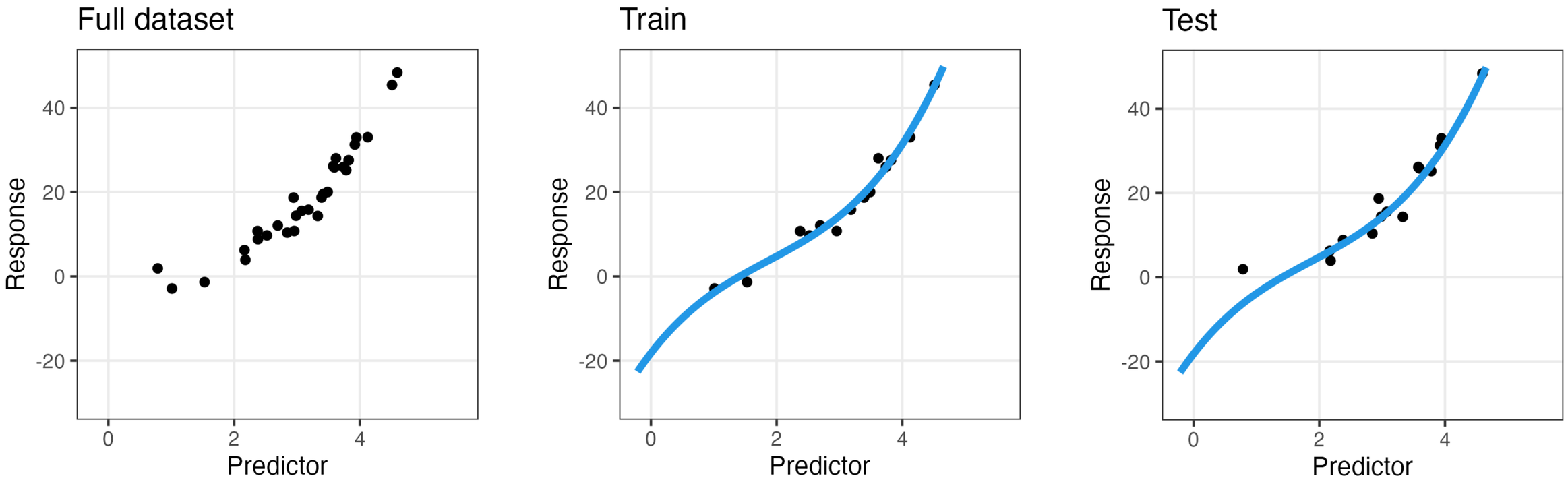
We can often avoid double dipping through sample splitting



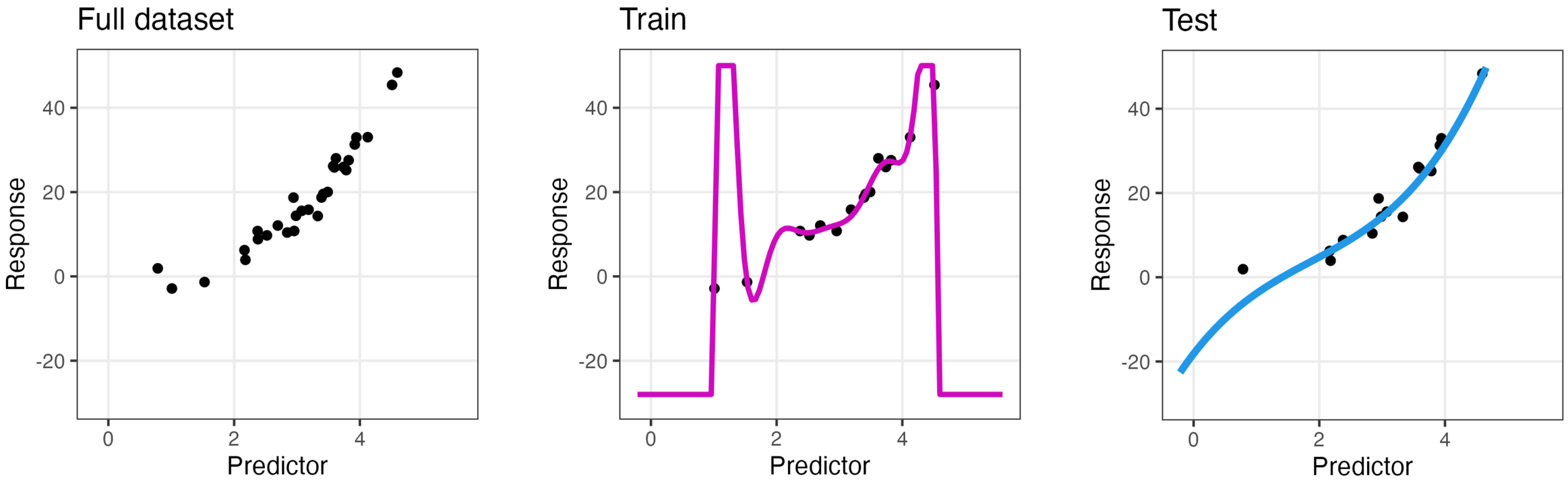
We can often avoid double dipping through sample splitting



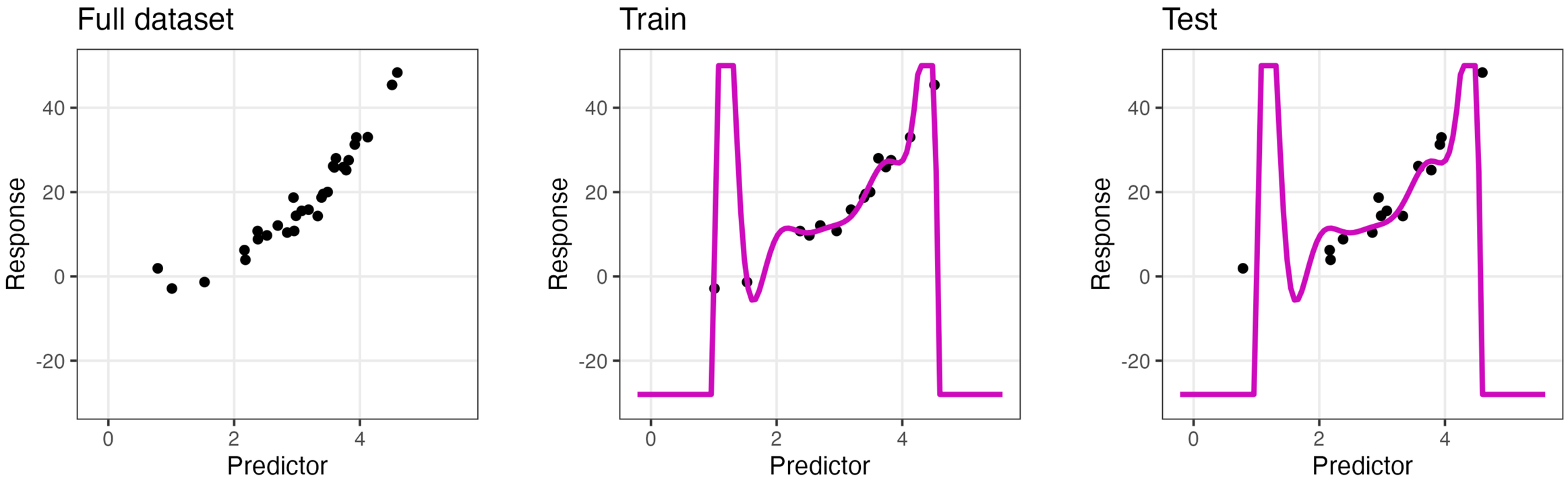
We can often avoid double dipping through sample splitting



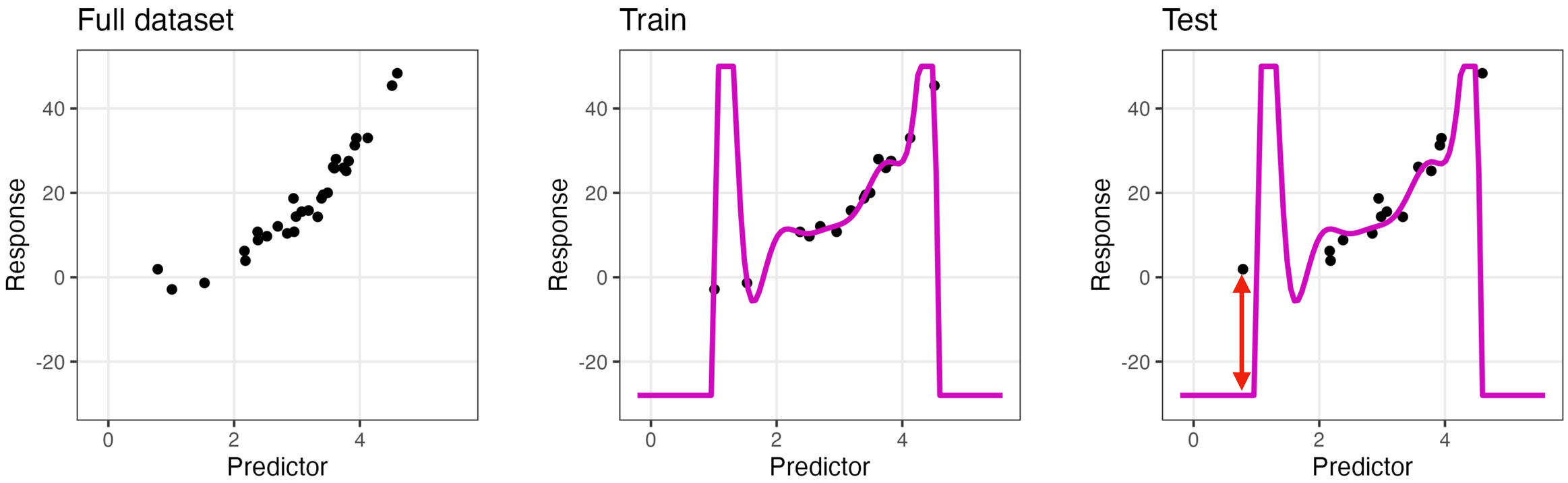
We can often avoid double dipping through sample splitting



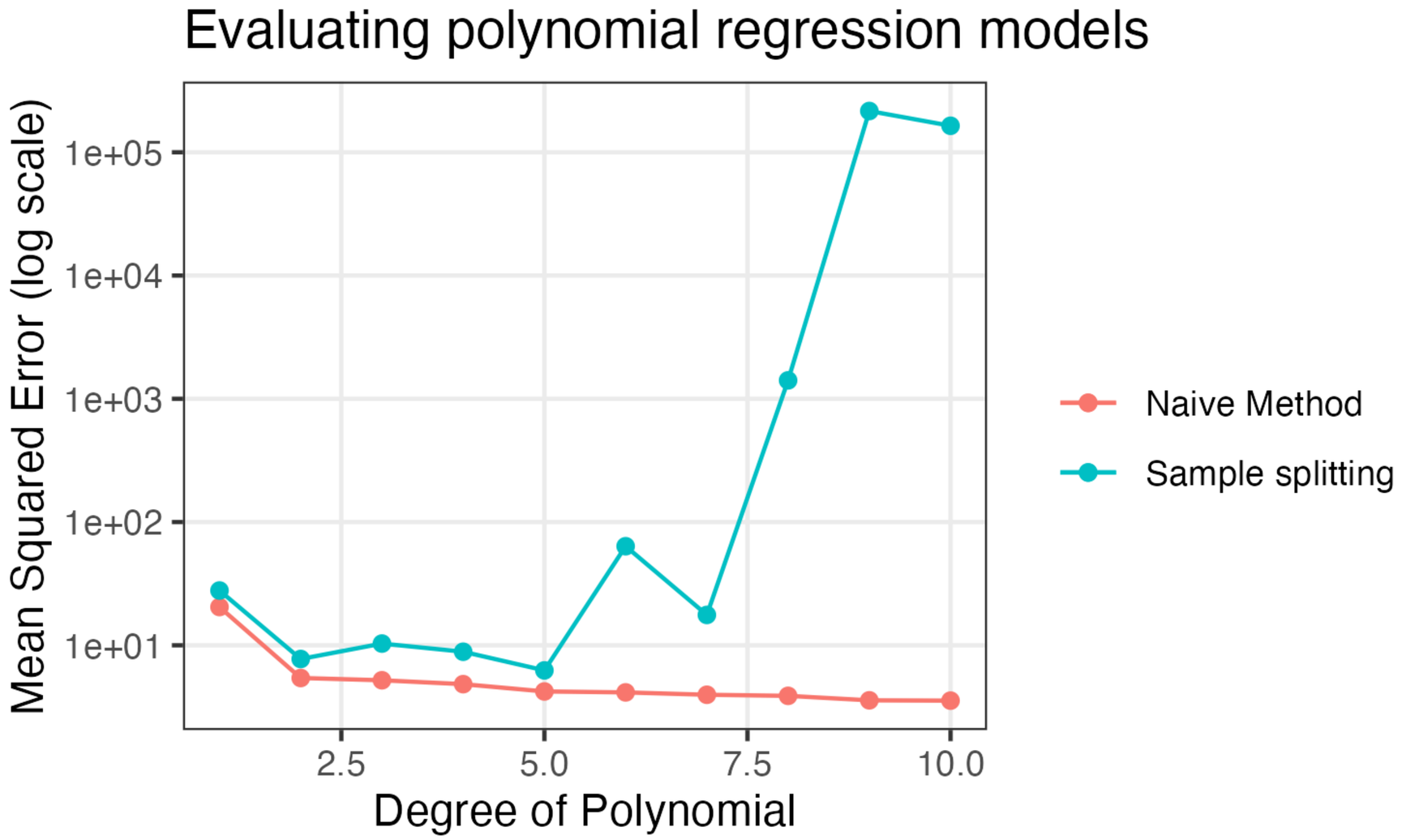
We can often avoid double dipping through sample splitting



We can often avoid double dipping through sample splitting



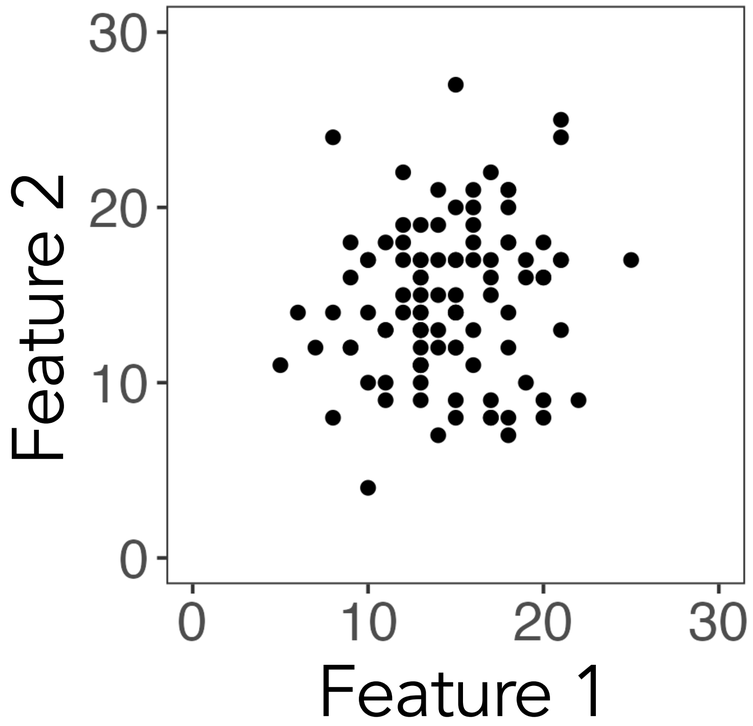
Sample splitting allows us to avoid double dipping in Example #1



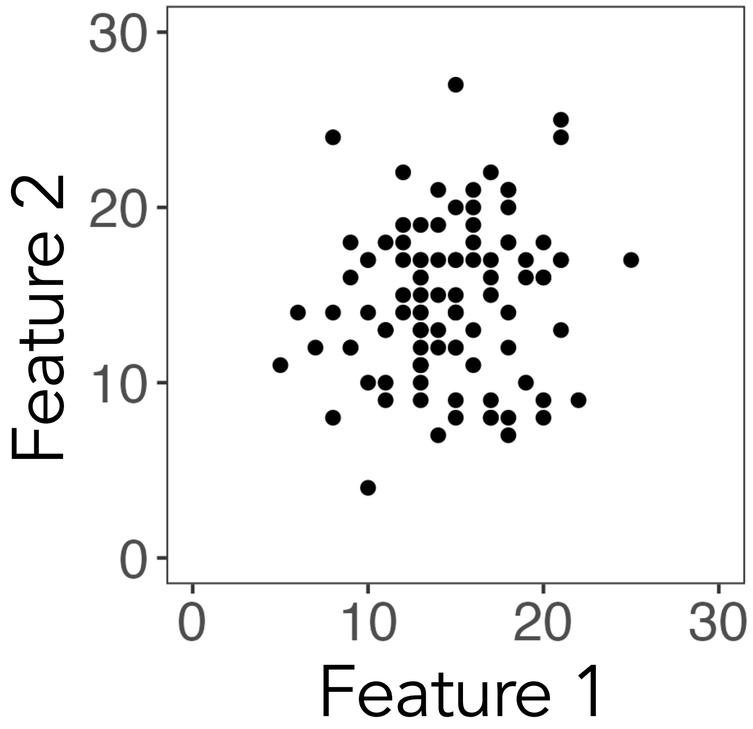
Outline

1. **Motivation: settings where sample splitting doesn’t work**
2. Poisson thinning
3. Data thinning
4. Real data application
5. Ongoing work

Example 2: how many clusters are in our data?

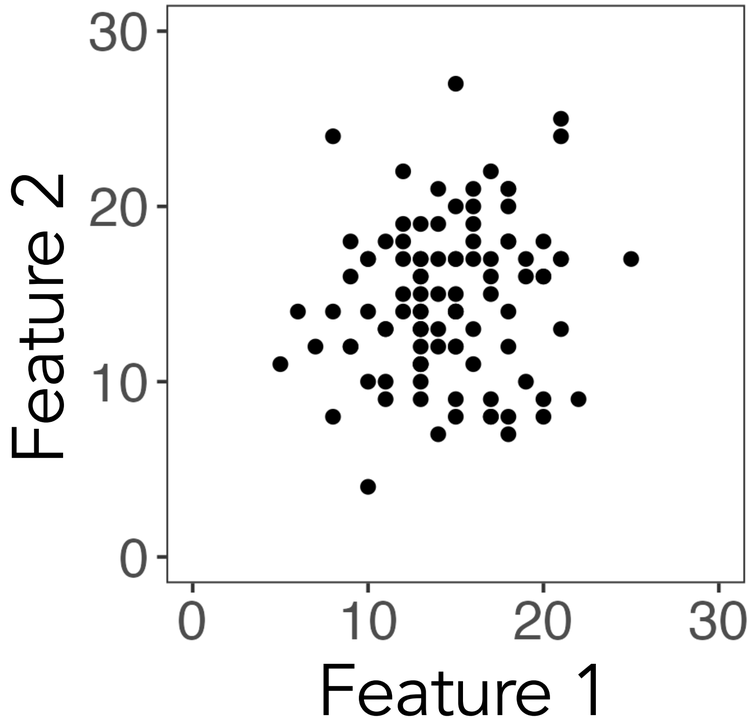


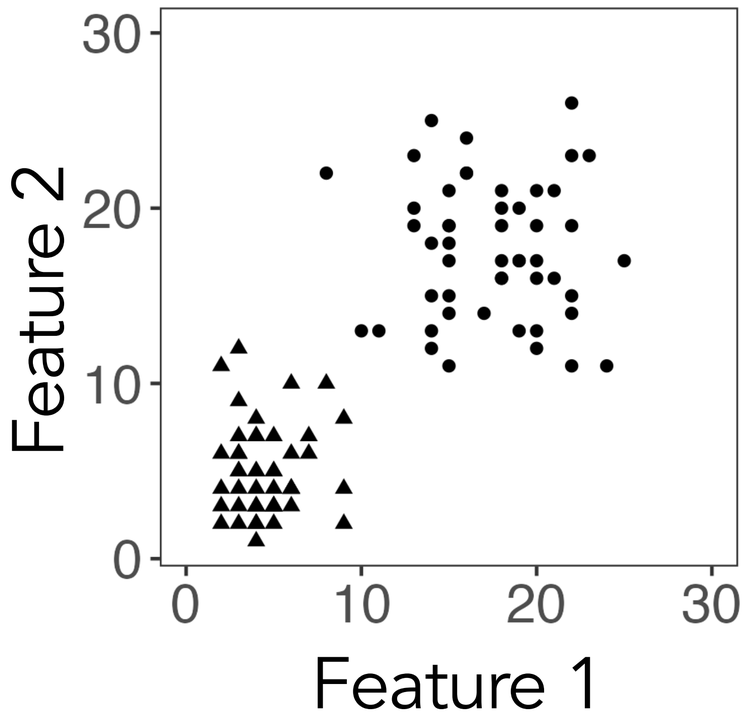
Example 2: how many clusters are in our data?



One true cluster: all observations drawn from same distribution.

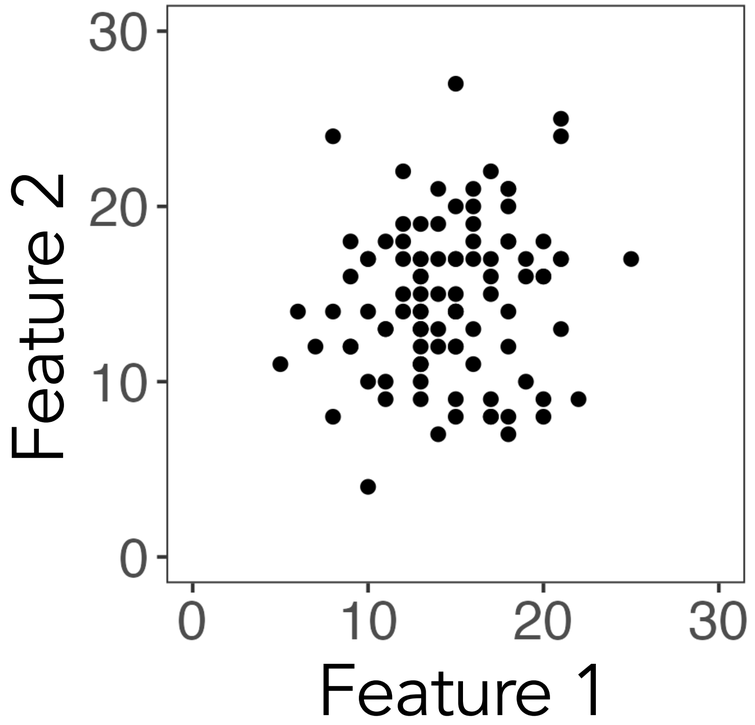
Example 2: how many clusters are in our data?

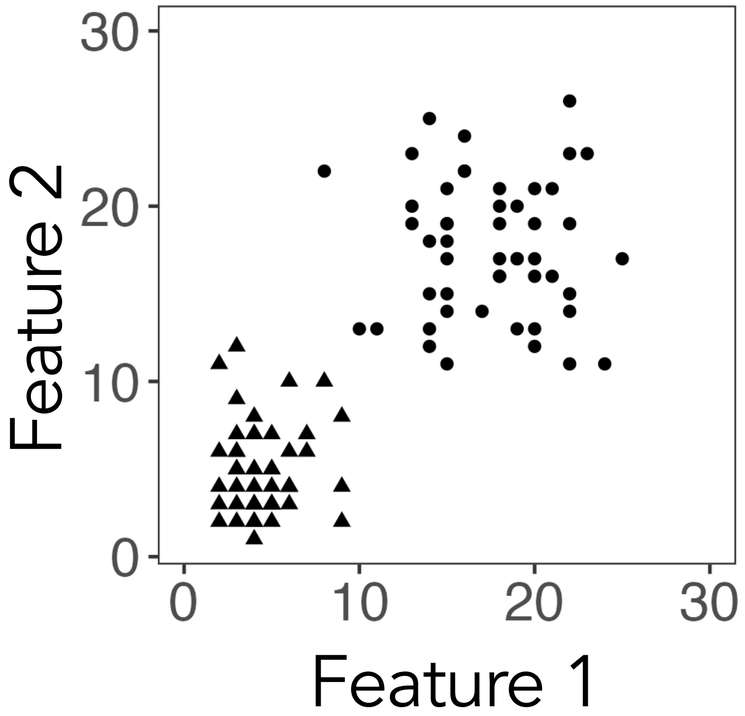


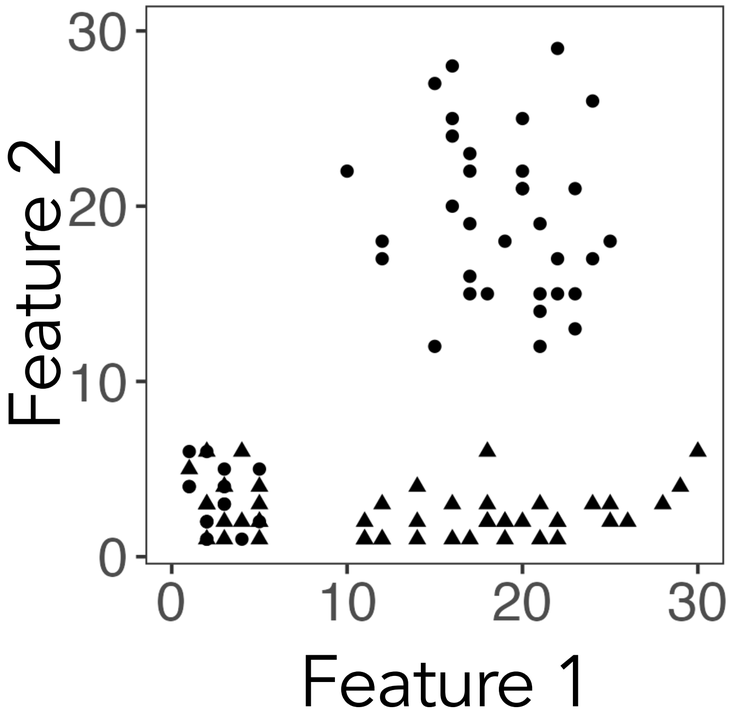


Two true clusters.

Example 2: how many clusters are in our data?

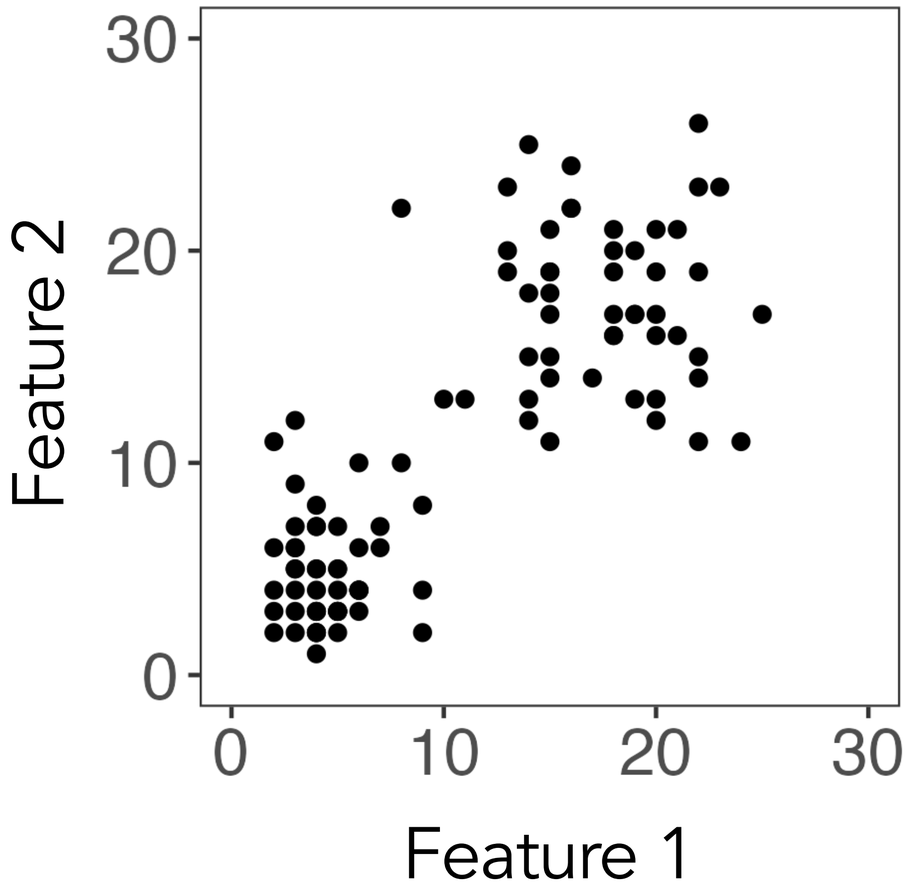




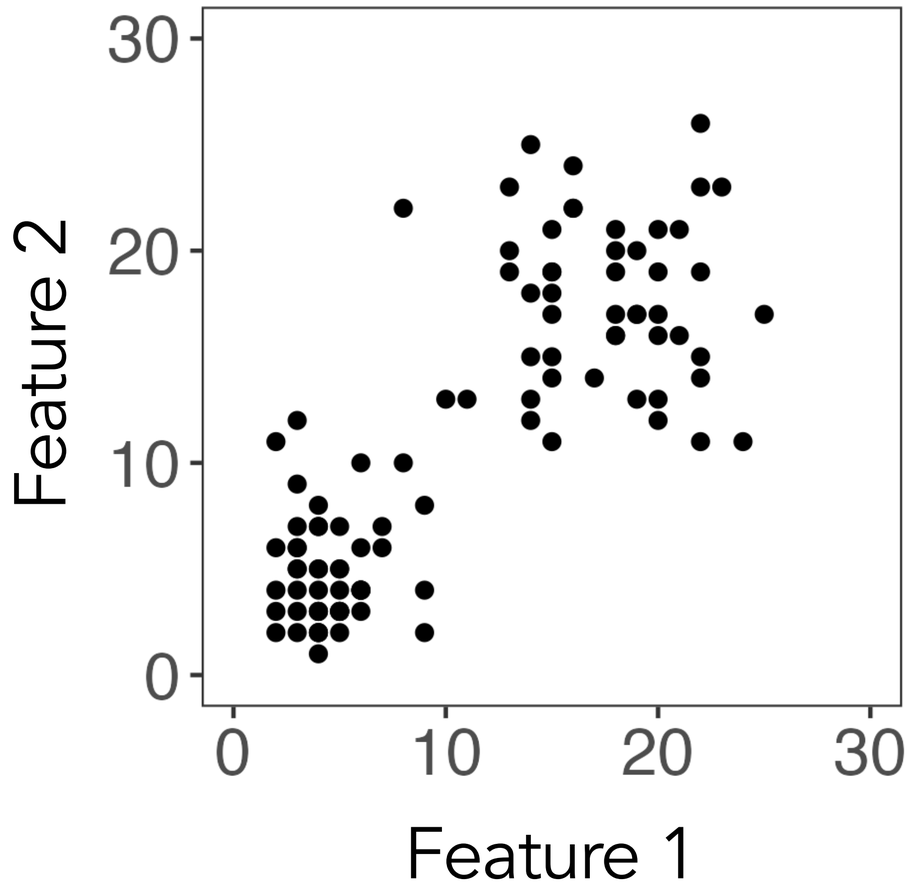


Three true clusters.

A brief introduction to k-means clustering

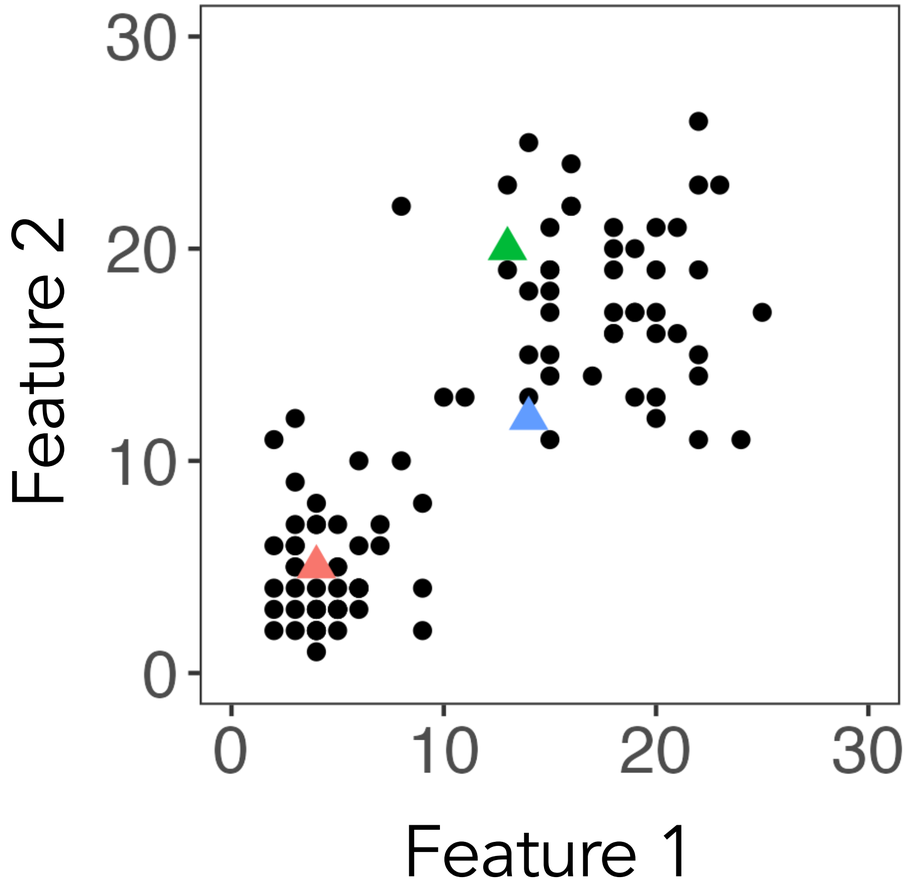


A brief introduction to k-means clustering



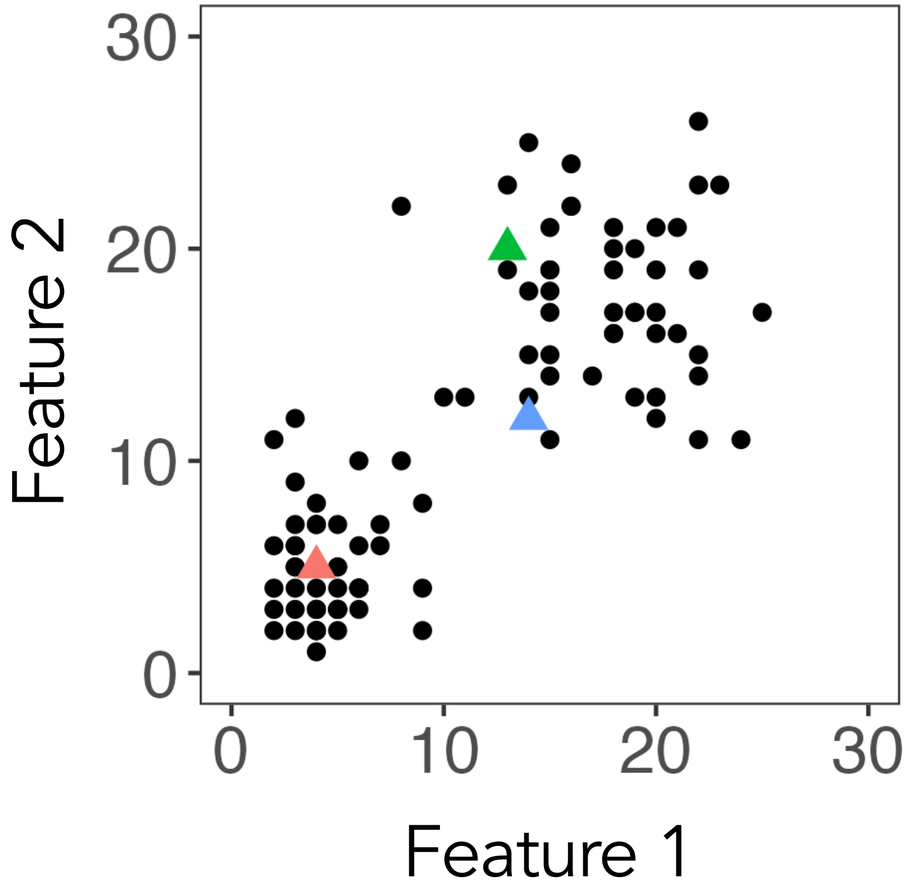
Step 1: Randomly initialize k cluster centers.

A brief introduction to k-means clustering



Step 1: Randomly initialize k cluster centers.

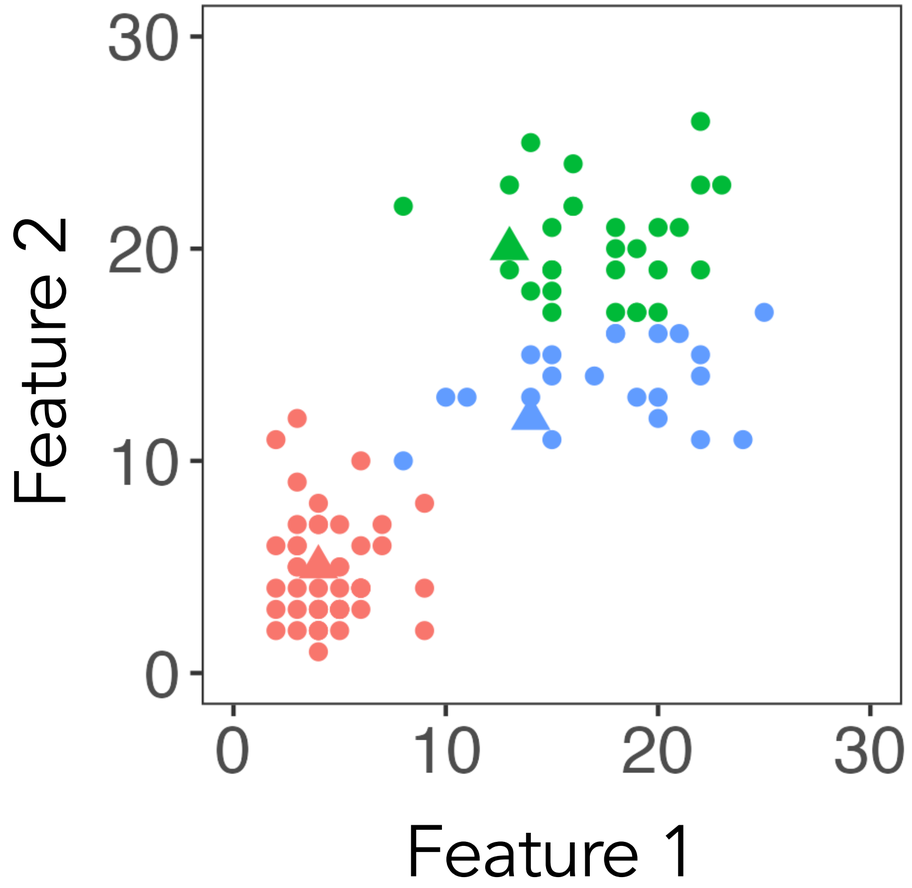
A brief introduction to k-means clustering



Step 1: Randomly initialize k cluster centers.

Step 2: Assign each point to the cluster whose center is nearest.

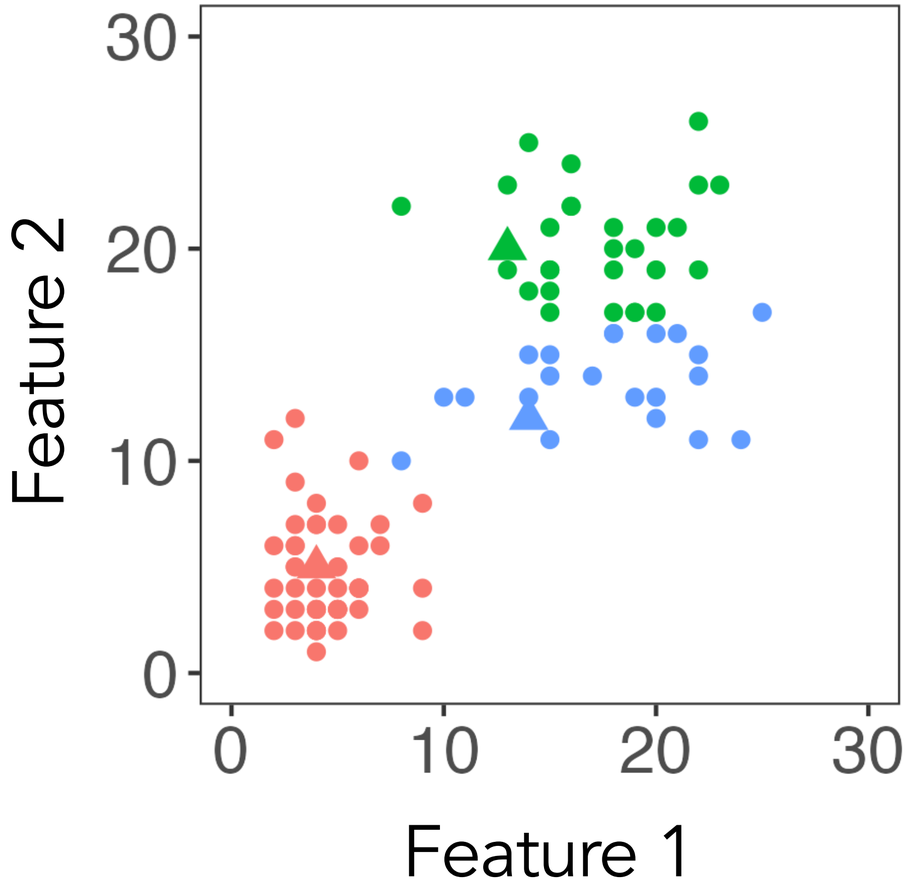
A brief introduction to k-means clustering



Step 1: Randomly initialize k cluster centers.

Step 2: Assign each point to the cluster whose center is nearest.

A brief introduction to k-means clustering

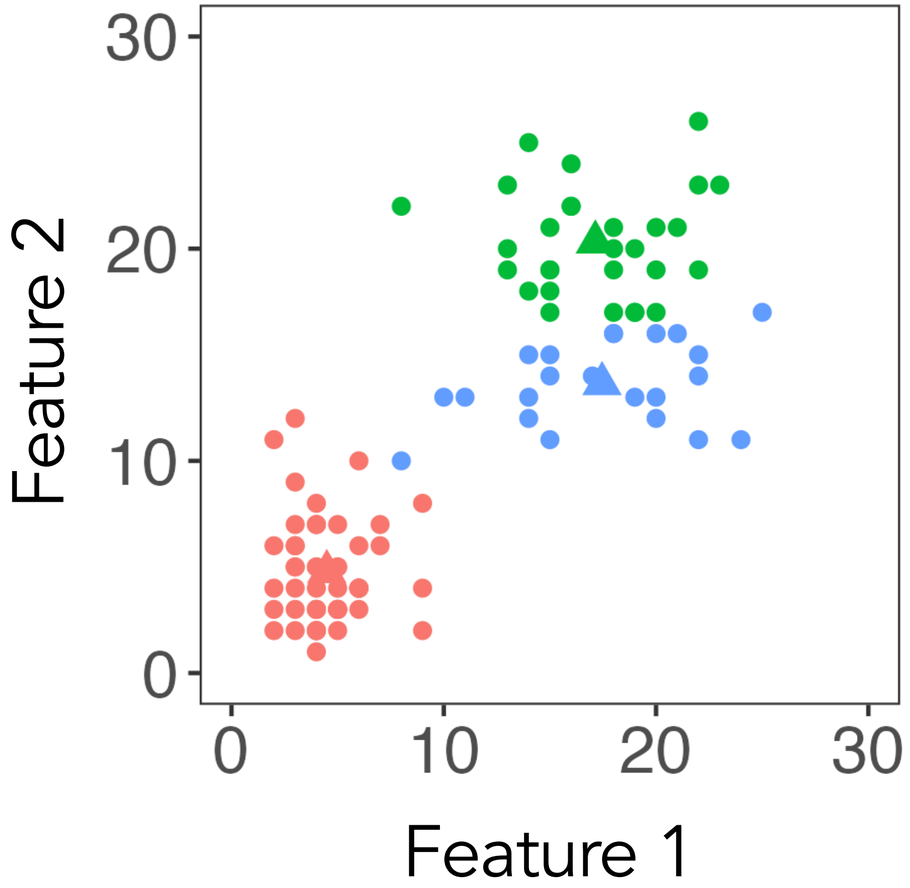


Step 1: Randomly initialize k cluster centers.

Step 2: Assign each point to the cluster whose center is nearest.

Step 3: Update centers to be the mean of all points in that cluster.

A brief introduction to k-means clustering

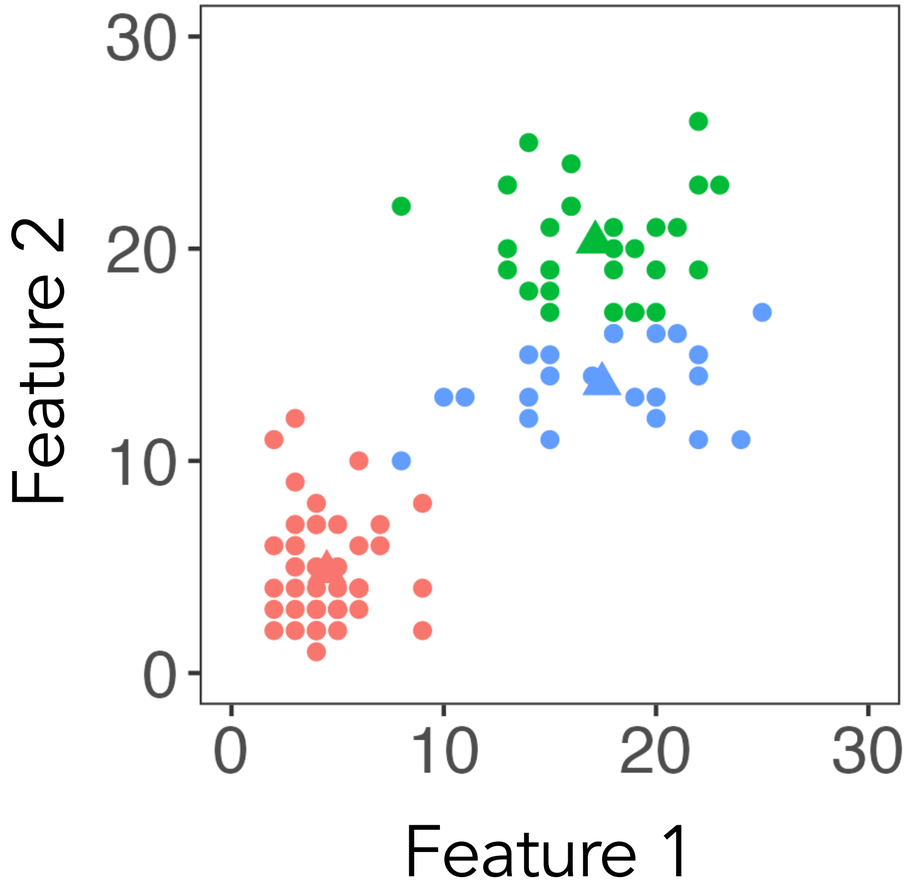


Step 1: Randomly initialize k cluster centers.

Step 2: Assign each point to the cluster whose center is nearest.

Step 3: Update centers to be the mean of all points in that cluster.

A brief introduction to k-means clustering



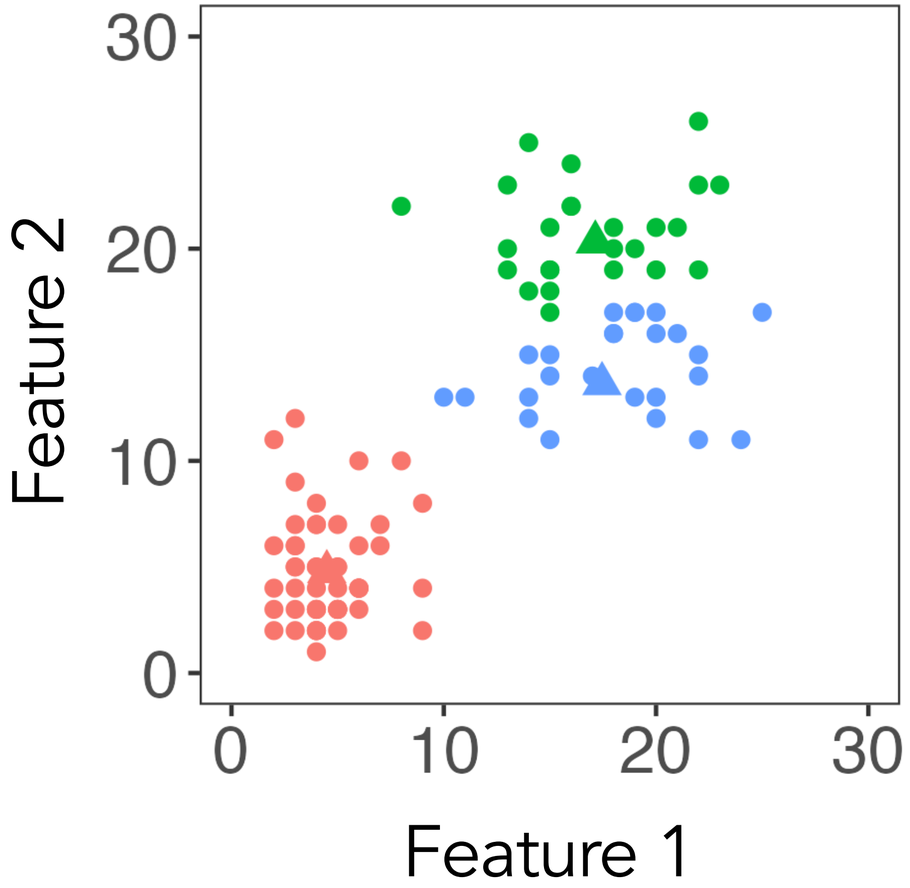
Step 1: Randomly initialize k cluster centers.

Step 2: Assign each point to the cluster whose center is nearest.

Step 3: Update centers to be the mean of all points in that cluster.

Repeat steps 2 and 3 until centers stop moving.

A brief introduction to k-means clustering



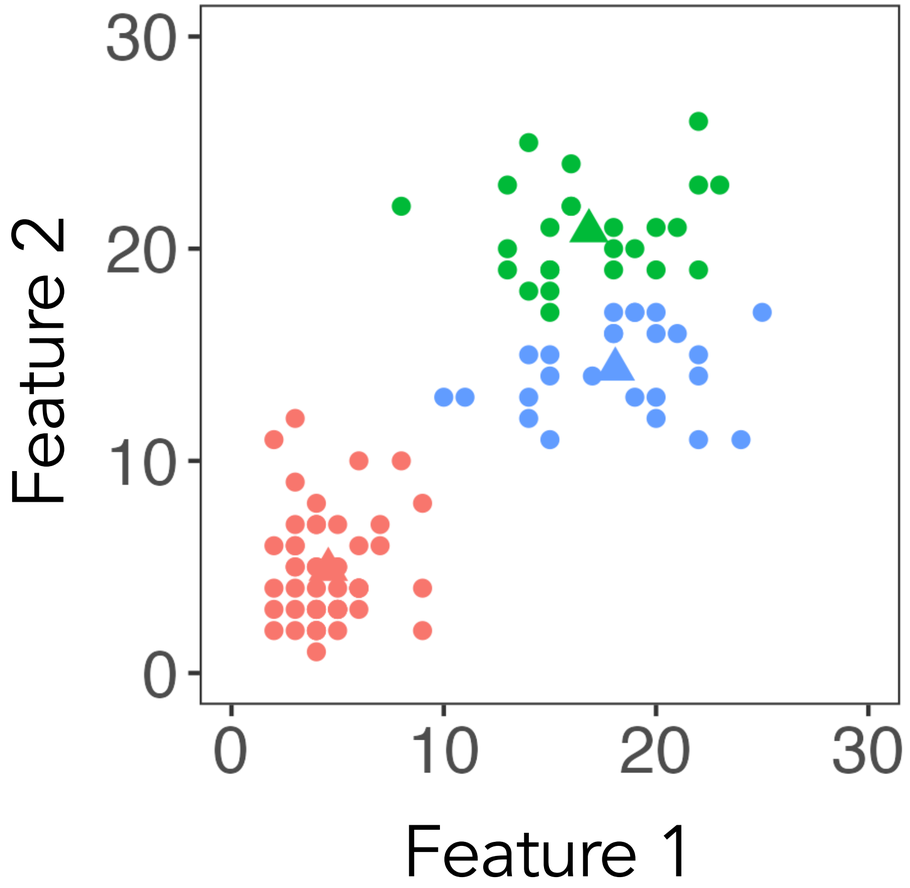
Step 1: Randomly initialize k cluster centers.

Step 2: Assign each point to the cluster whose center is nearest.

Step 3: Update centers to be the mean of all points in that cluster.

Repeat steps 2 and 3 until centers stop moving.

A brief introduction to k-means clustering



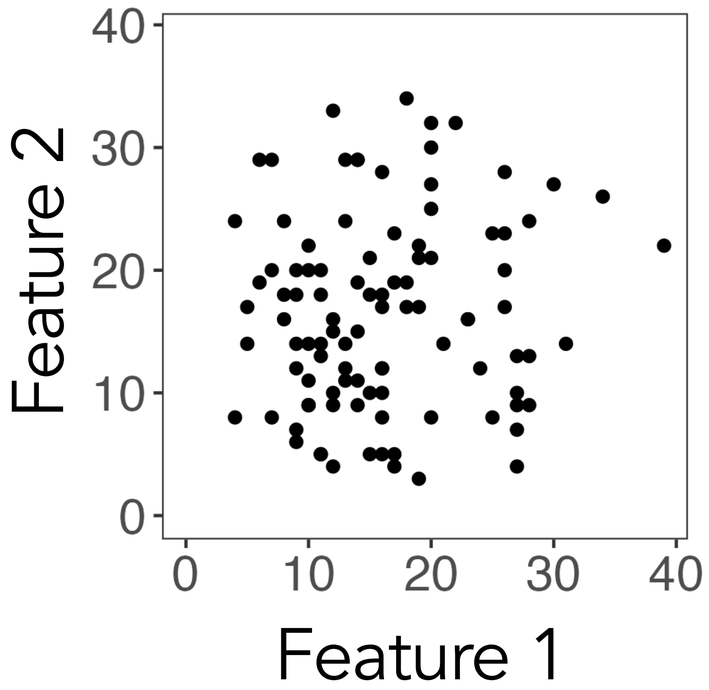
Step 1: Randomly initialize k cluster centers.

Step 2: Assign each point to the cluster whose center is nearest.

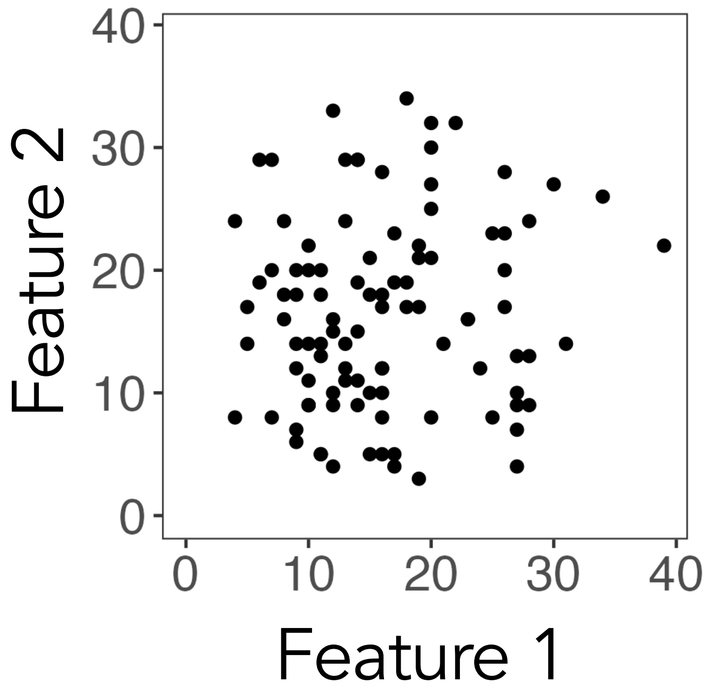
Step 3: Update centers to be the mean of all points in that cluster.

Repeat steps 2 and 3 until centers stop moving.

Example 2: how many clusters are in our data?

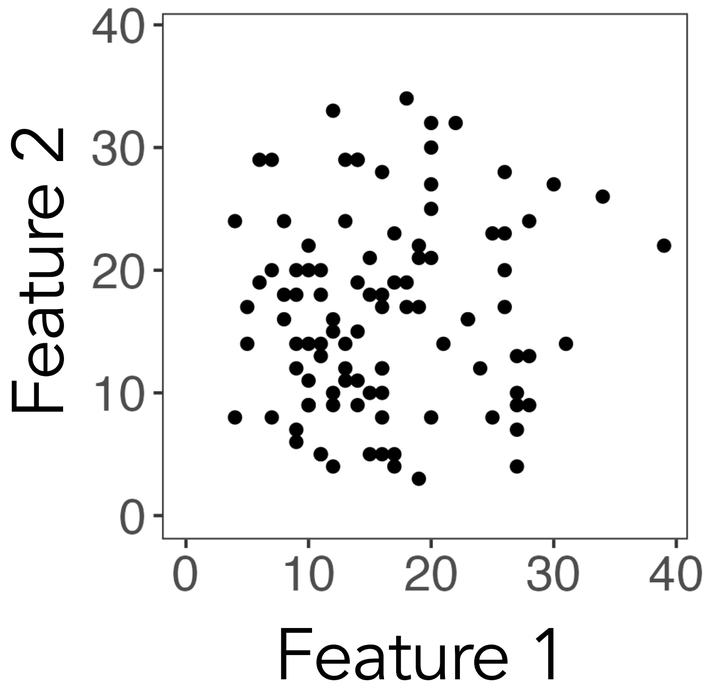


Example 2: how many clusters are in our data?



For several values of k:

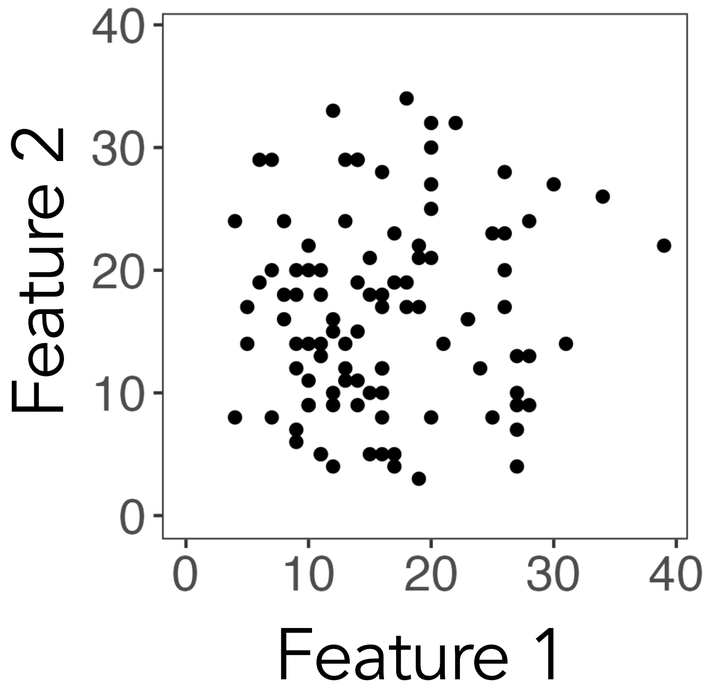
Example 2: how many clusters are in our data?



For several values of k:

Step 1: use k-means to fit a model with k clusters.

Example 2: how many clusters are in our data?

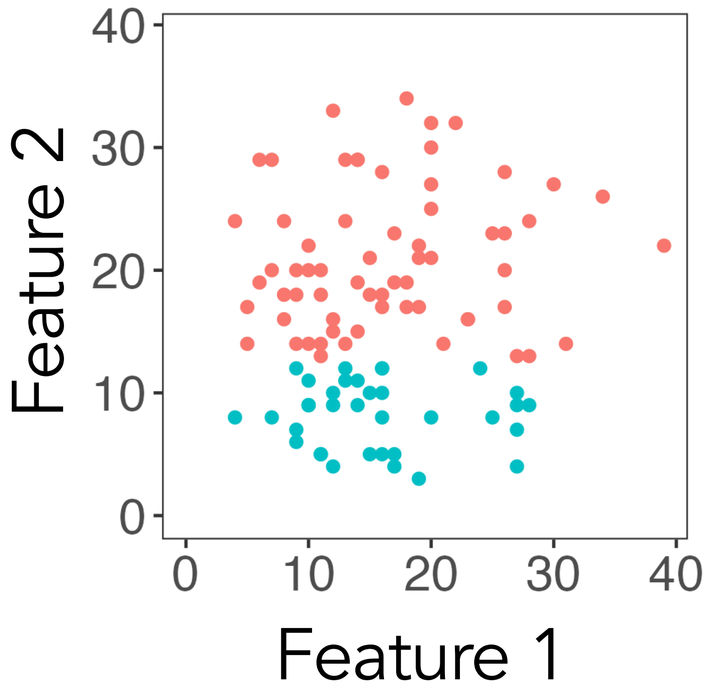


For several values of k:

Step 1: use k-means to fit a model with k clusters.

Step 2: evaluate model using within-cluster mean squared error.

Example 2: how many clusters are in our data?

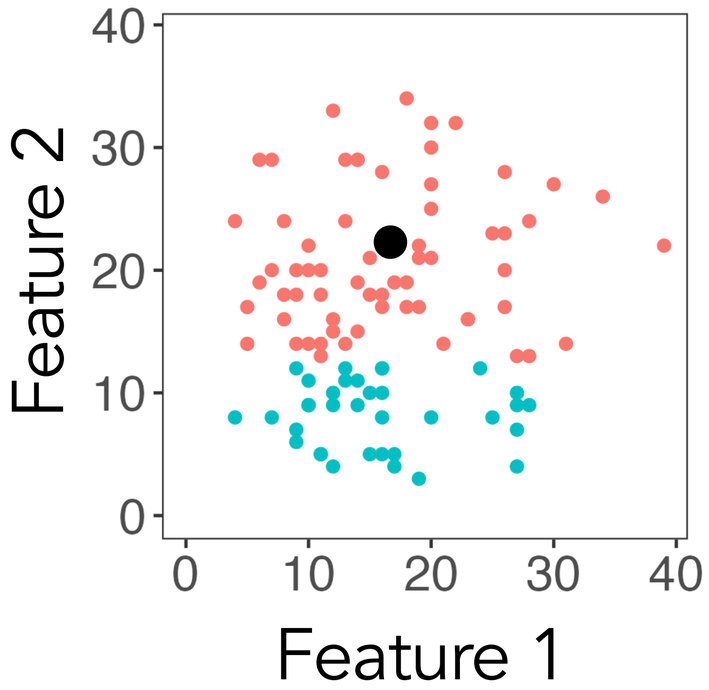


For several values of k:

Step 1: use k-means to fit a model with k clusters.

Step 2: evaluate model using within-cluster mean squared error.

Example 2: how many clusters are in our data?

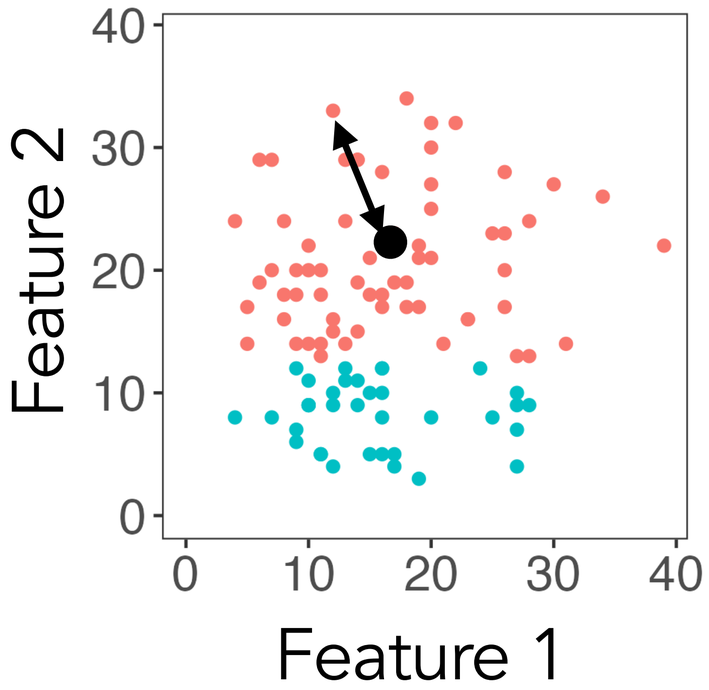


For several values of k:

Step 1: use k-means to fit a model with k clusters.

Step 2: evaluate model using within-cluster mean squared error.

Example 2: how many clusters are in our data?

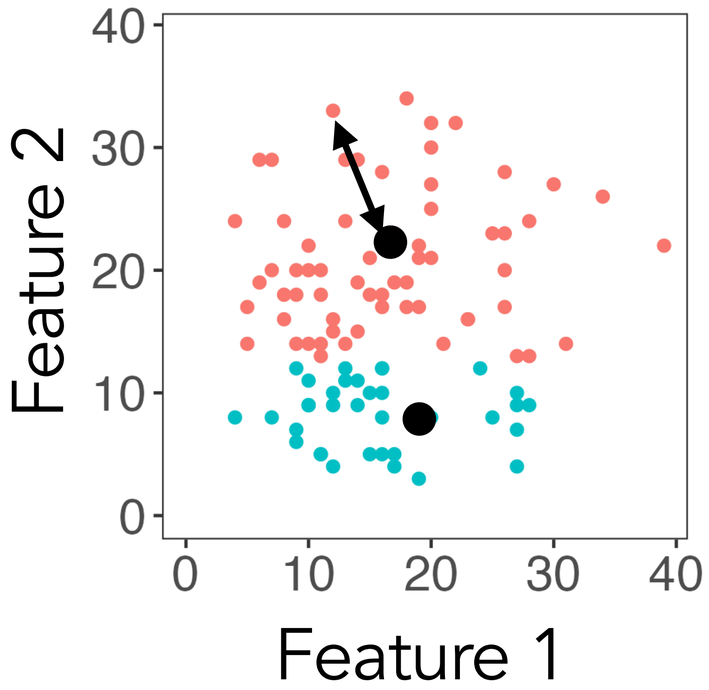


For several values of k:

Step 1: use k-means to fit a model with k clusters.

Step 2: evaluate model using within-cluster mean squared error.

Example 2: how many clusters are in our data?

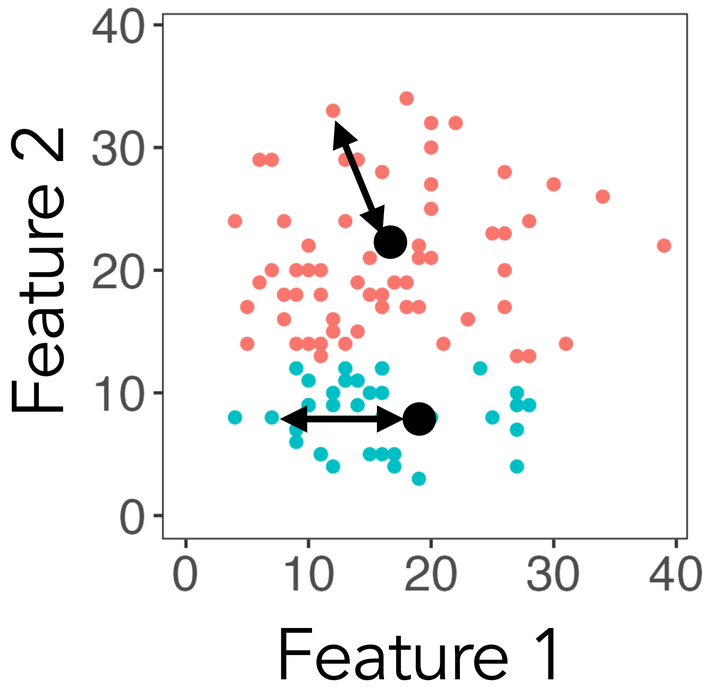


For several values of k:

Step 1: use k-means to fit a model with k clusters.

Step 2: evaluate model using within-cluster mean squared error.

Example 2: how many clusters are in our data?

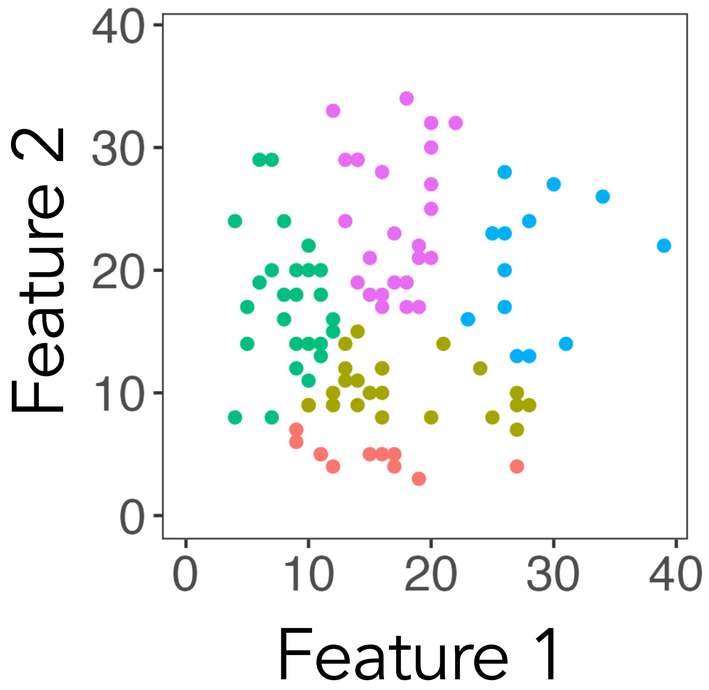


For several values of k:

Step 1: use k-means to fit a model with k clusters.

Step 2: evaluate model using within-cluster mean squared error.

Example 2: how many clusters are in our data?

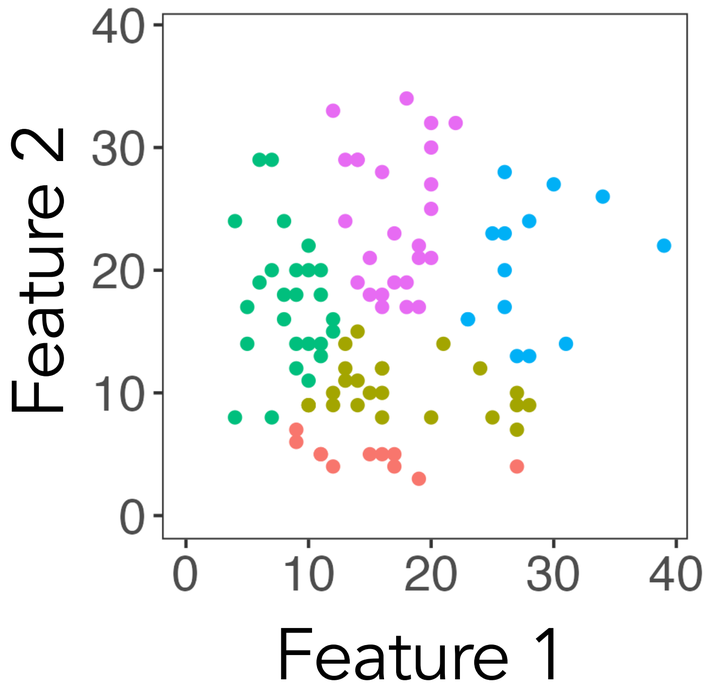


For several values of k:

Step 1: use k-means to fit a model with k clusters.

Step 2: evaluate model using within-cluster mean squared error.

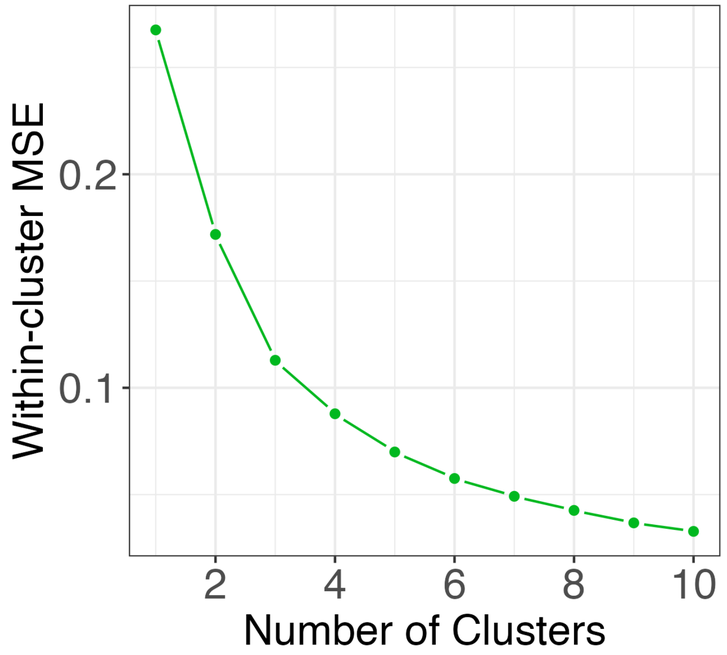
Example 2: how many clusters are in our data?



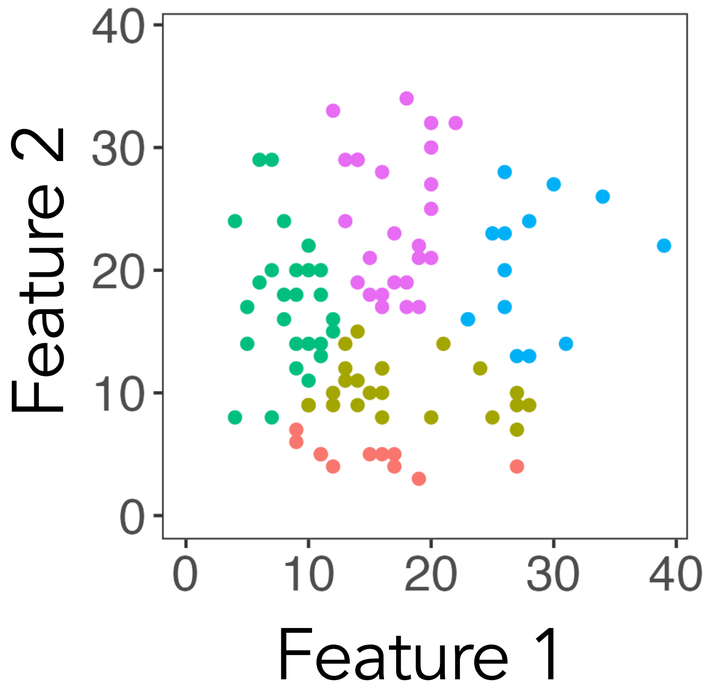
For several values of k:

Step 1: use k-means to fit a model with k clusters.

Step 2: evaluate model using within-cluster mean squared error.



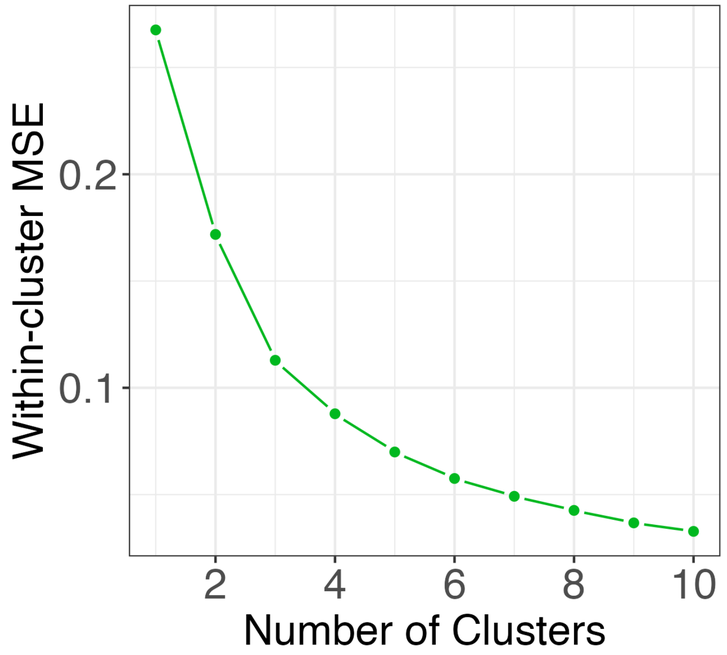
Example 2: how many clusters are in our data?



For several values of k:

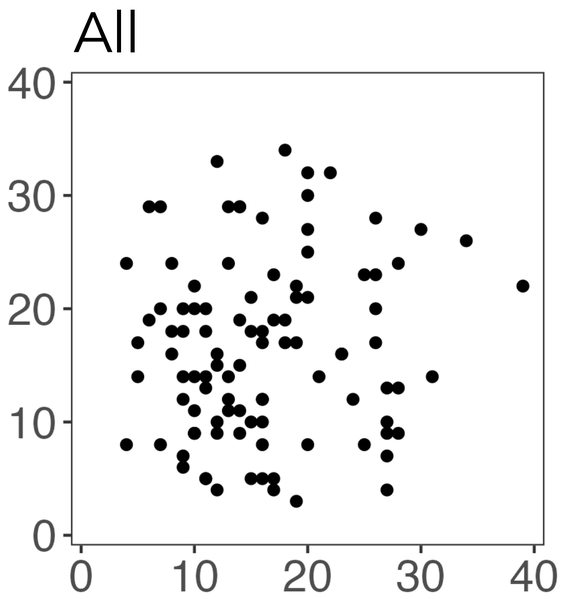
Step 1: use k-means to fit a model with k clusters.

Step 2: evaluate model using within-cluster mean squared error.



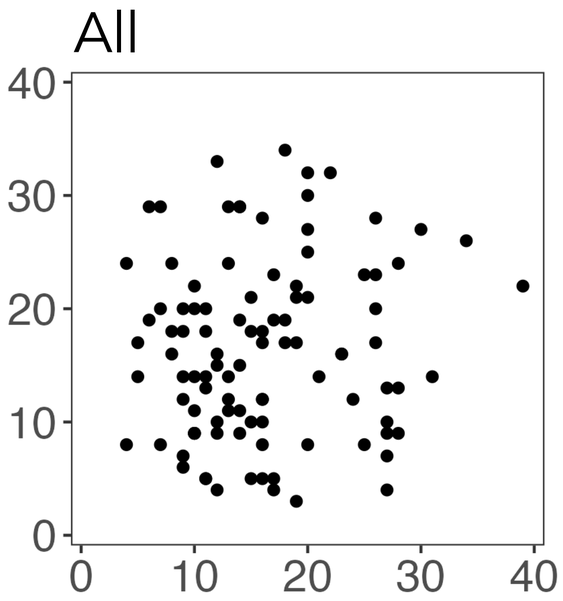
When we use the same data to fit and evaluate a model, more complex models appear better.

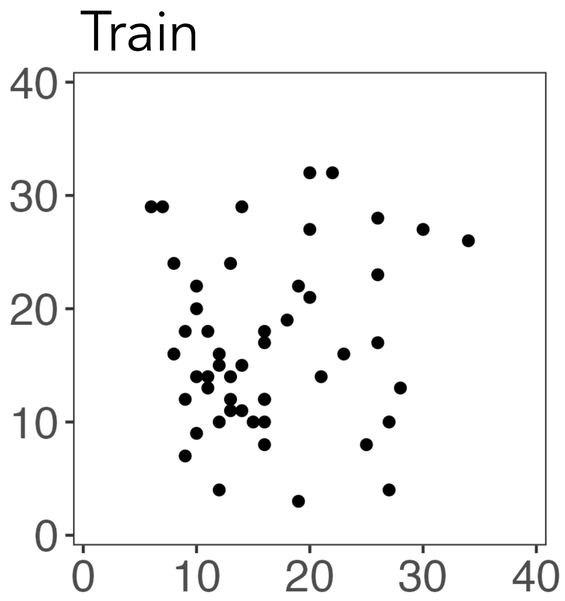
Sample splitting cannot be used for Example 2

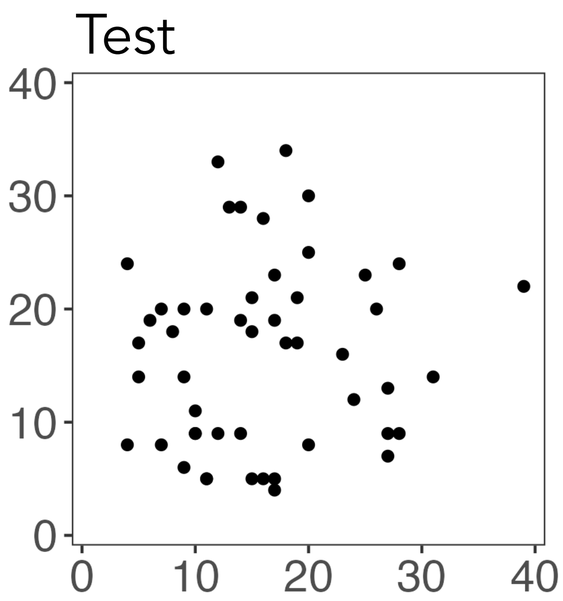


Step 1: split observations into train/test.

Sample splitting cannot be used for Example 2

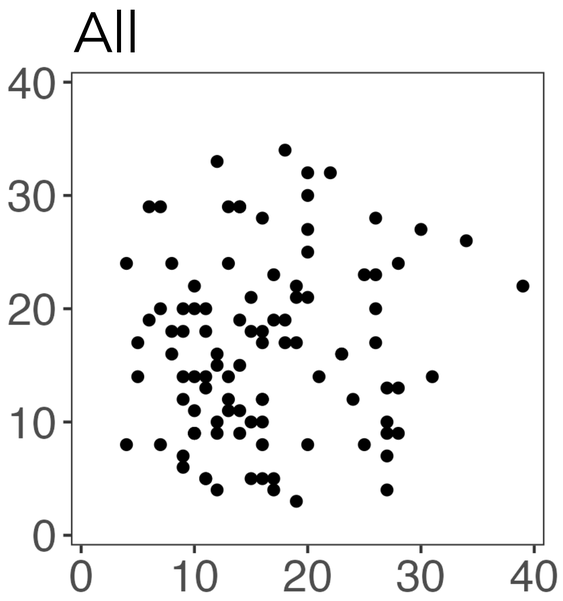


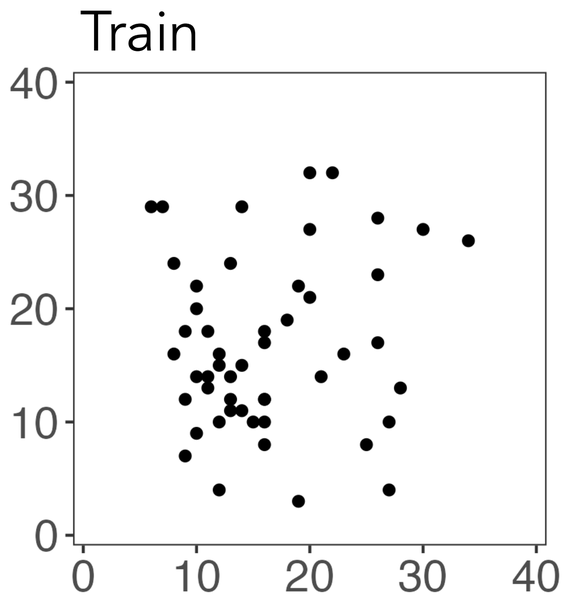


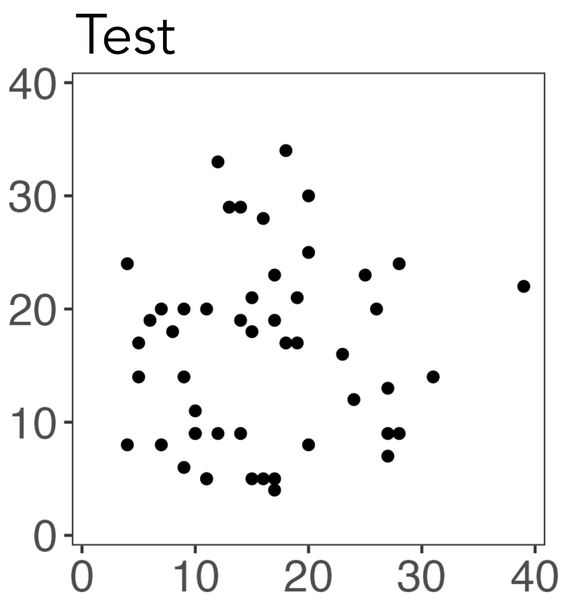


Step 1: split observations into train/test.

Sample splitting cannot be used for Example 2



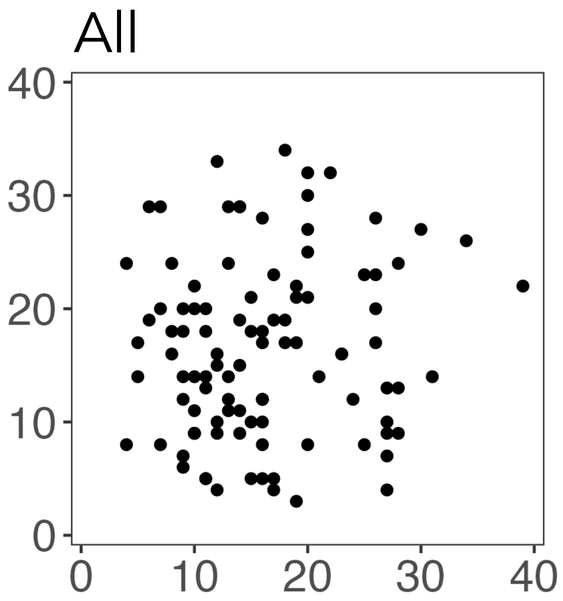


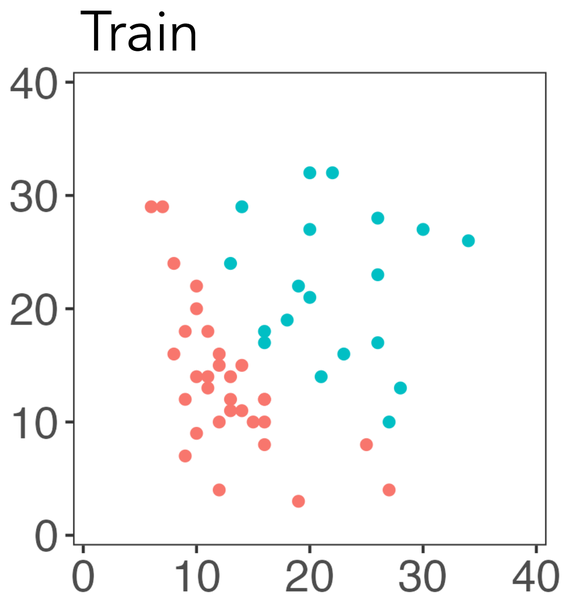


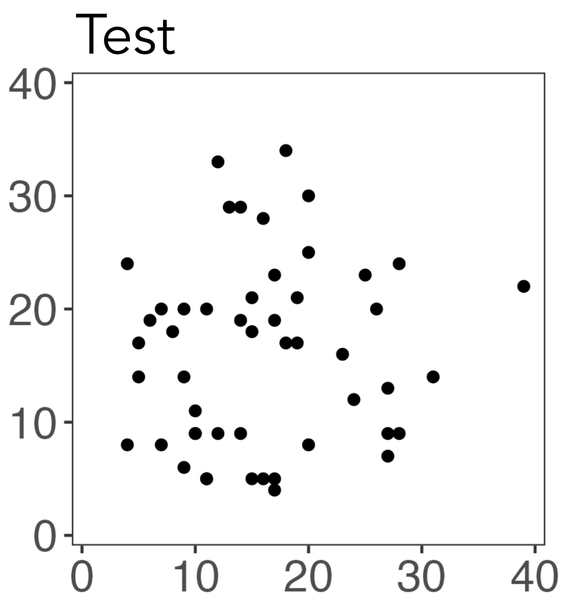
Step 1: split observations into train/test.

Step 2: cluster the training set.

Sample splitting cannot be used for Example 2



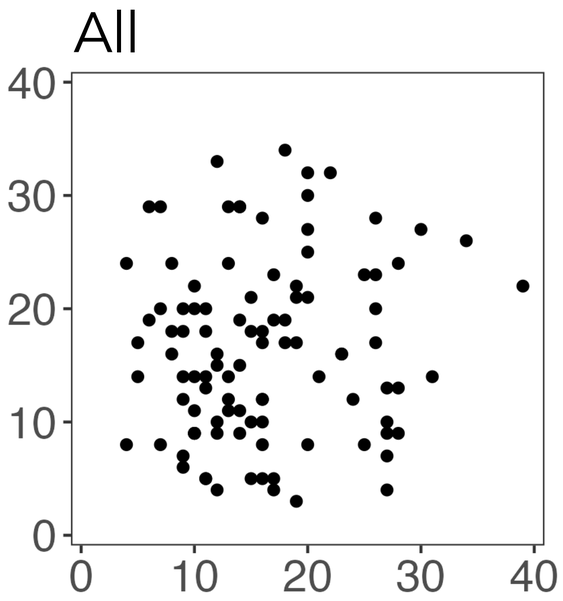


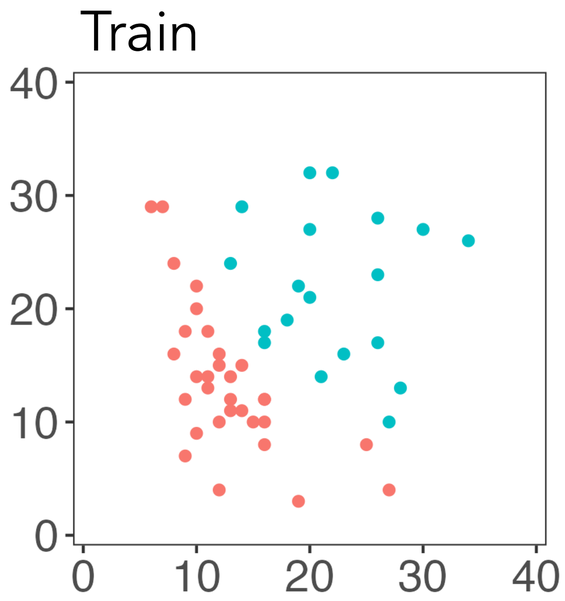


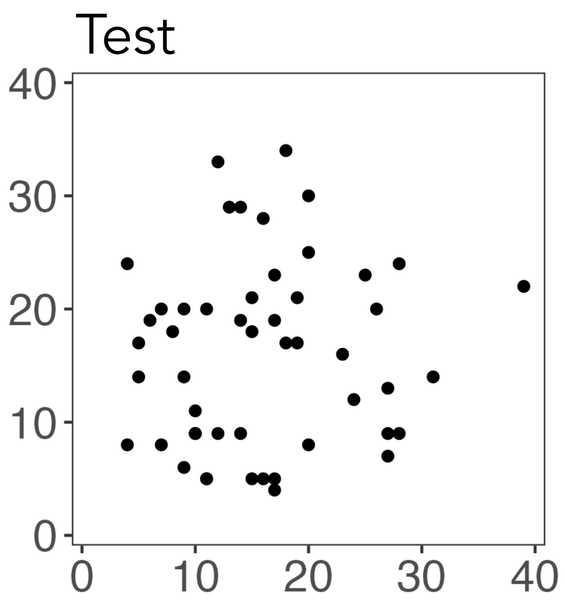
Step 1: split observations into train/test.

Step 2: cluster the training set.

Sample splitting cannot be used for Example 2





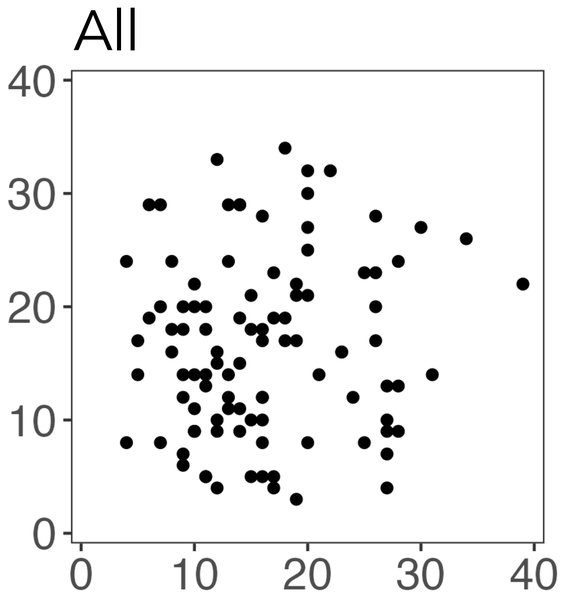


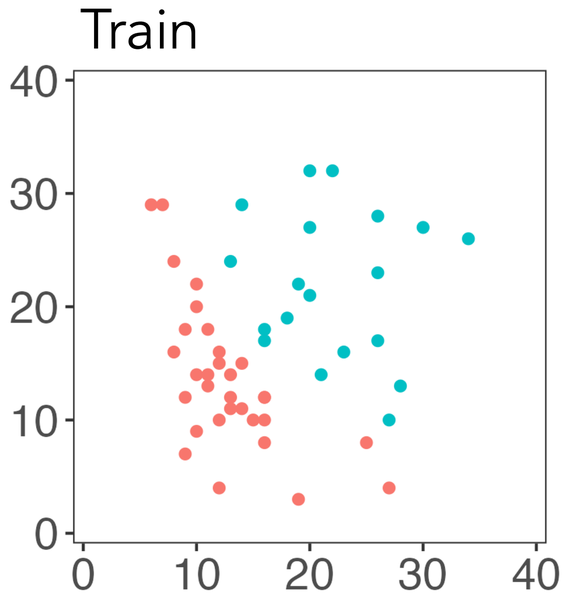
Step 1: split observations into train/test.

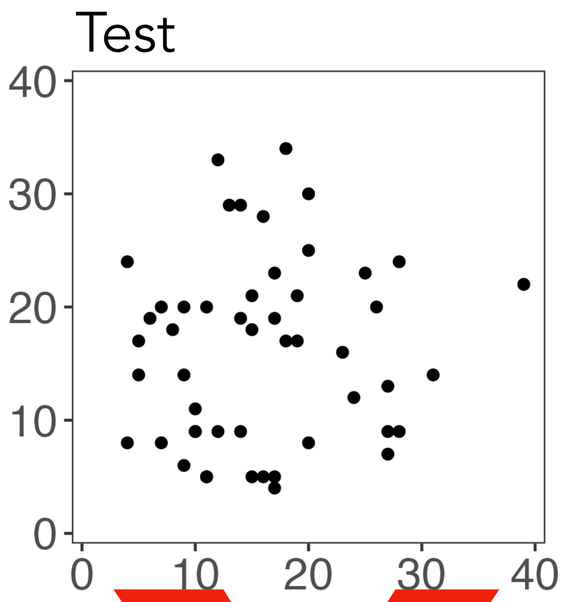
Step 2: cluster the training set.

Step 3: evaluate clusters using test set.

Sample splitting cannot be used for Example 2





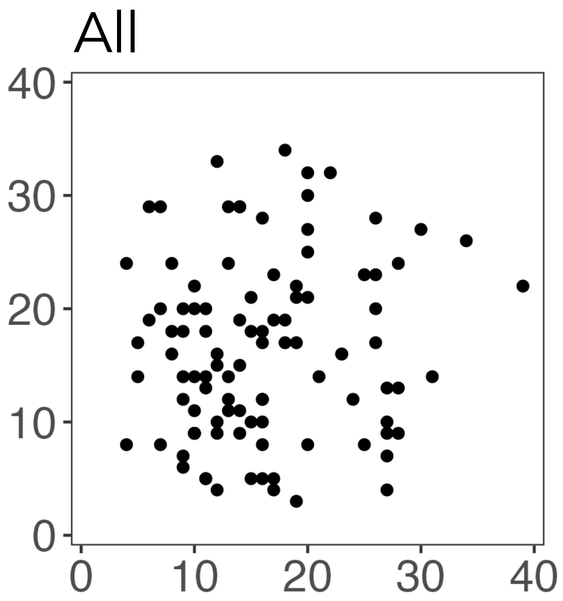


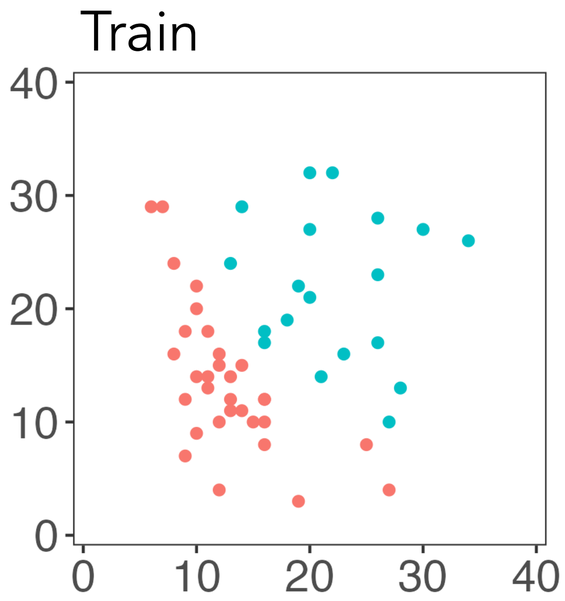
Step 1: split observations into train/test.

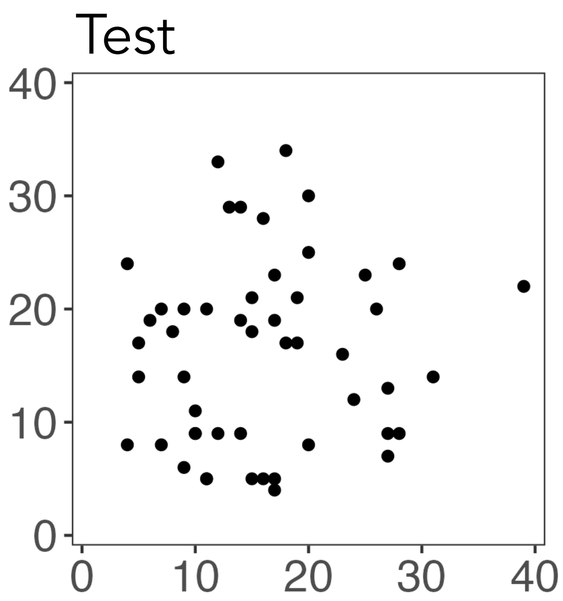
Step 2: cluster the training set.

St 3: evalu clusters usin set.

Attempts to salvage this approach end up double dipping







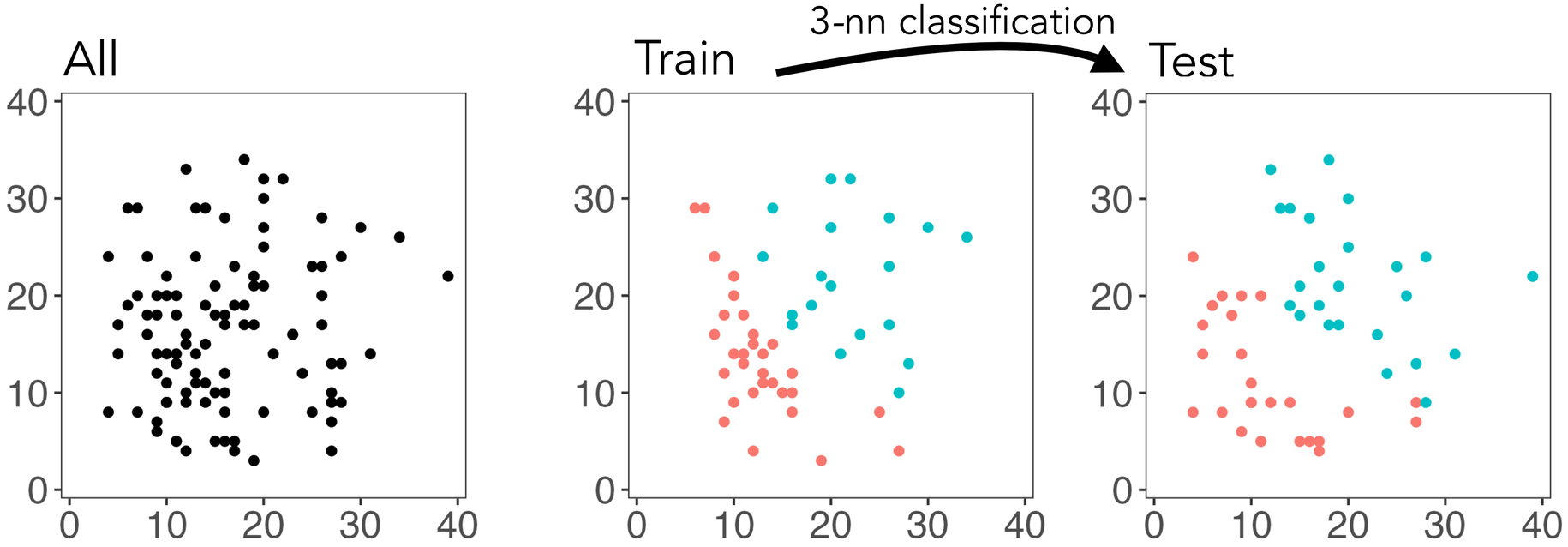
Step 1: split observations into train/test.

Step 2: cluster the training set.

Step 2.5: assign labels to observations in test set.

Step 3: evaluate clusters using test set.

Attempts to salvage this approach end up double dipping



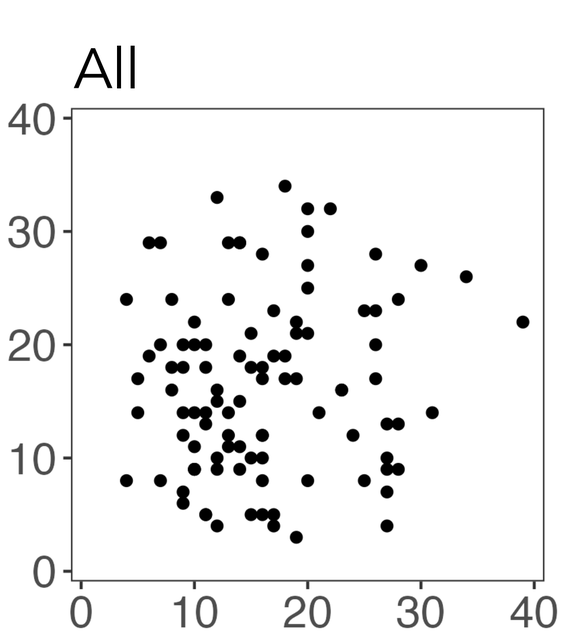
Step 1: split observations into train/test.

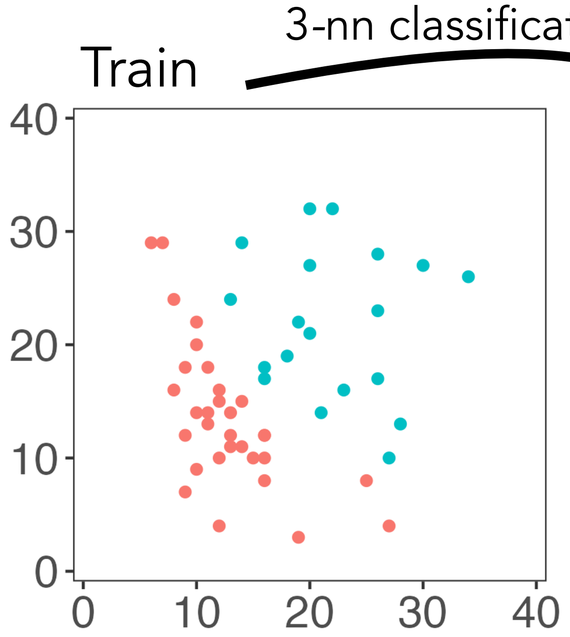
Step 2: cluster the training set.

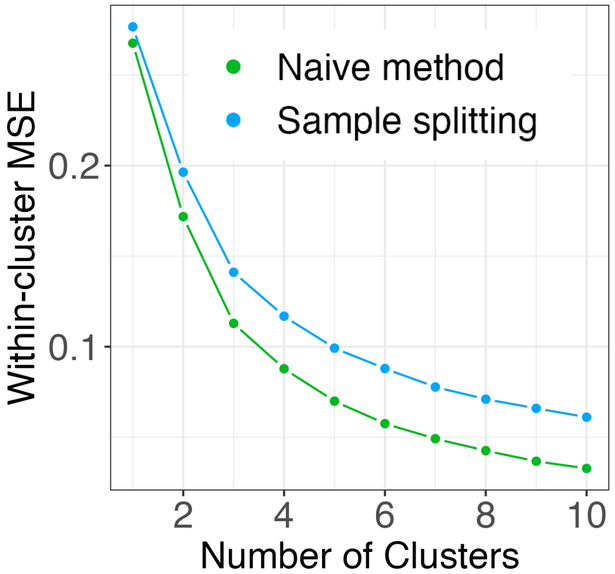
Step 2.5: assign labels to observations in test set.

Step 3: evaluate clusters using test set.

Attempts to salvage this approach end up double dipping







Step 1: split observations into train/test.

Step 2: cluster the training set.

Step 2.5: assign labels to observations in test set.

Step 3: evaluate clusters using test set.

Example 2 is an important problem in many applications

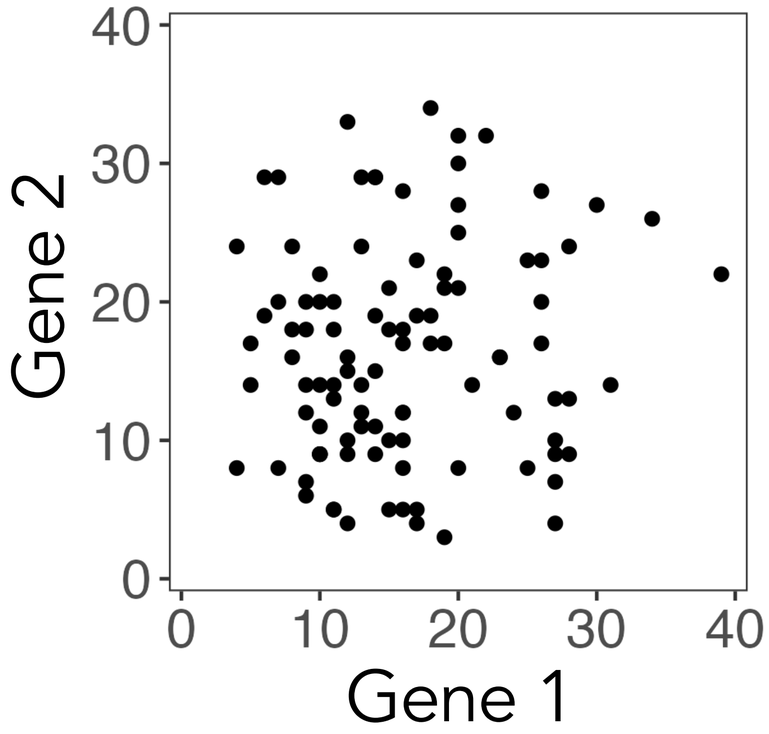
scRNA-seq dataset X

	Gene 1	**Gene 2**
Cell 1	18	6
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34
…	…	…
Cell 100	40	21

Example 2 is an important problem in many applications

scRNA-seq dataset X

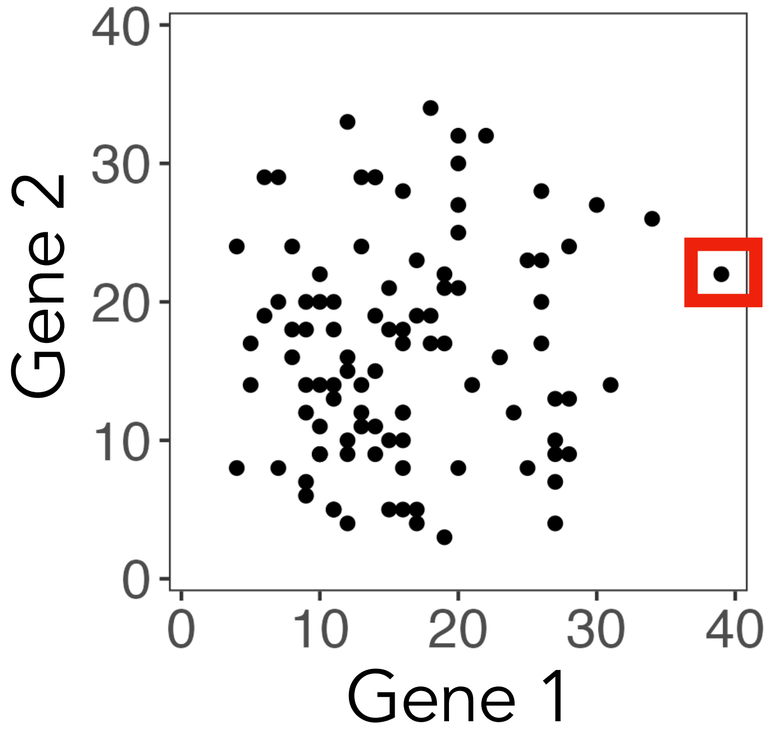
	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34
…	…	…
Cell 100	40	21



Example 2 is an important problem in many applications

scRNA-seq dataset X

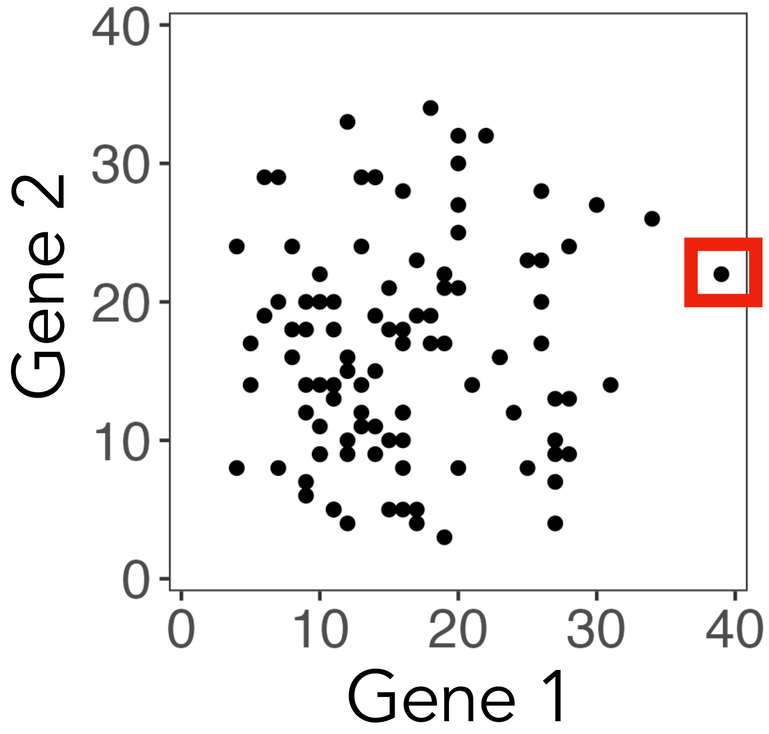
	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34
…	…	…
Cell 100	40	21

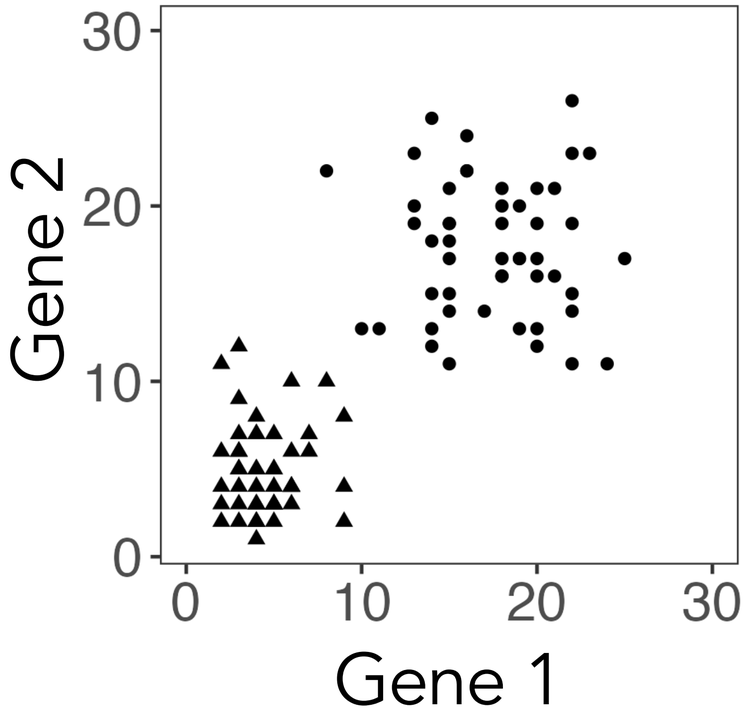


Example 2 is an important problem in many applications

scRNA-seq dataset X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34
…	…	…
Cell 100	40	21

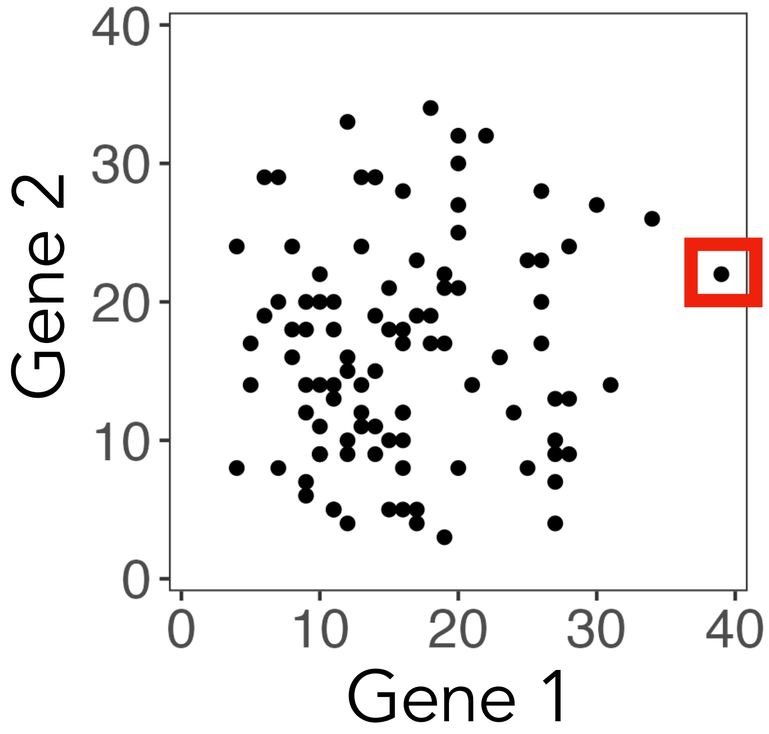


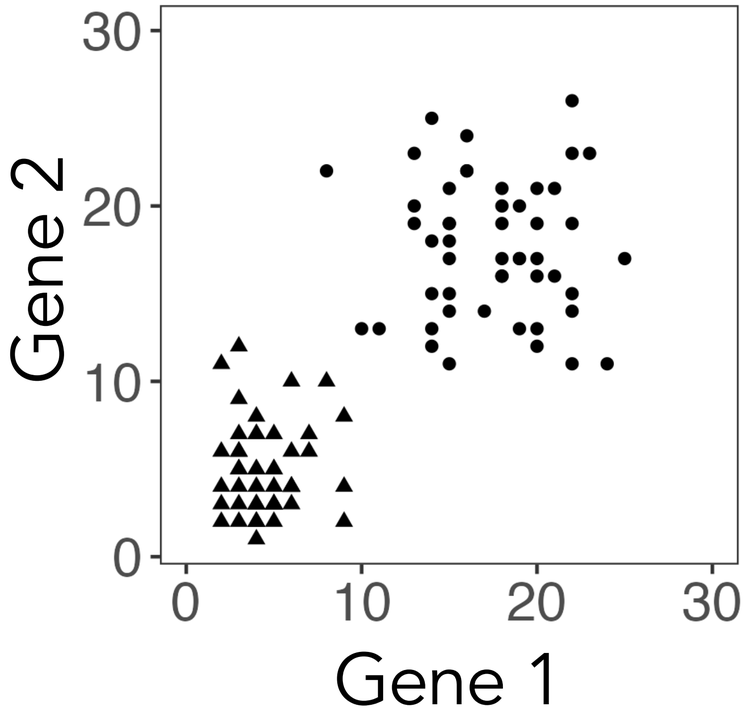


Example 2 is an important problem in many applications

scRNA-seq dataset X

	Gene 1	Gene 2
Cell 1	18	6
Cell 2	31	8
Cell 3	11	31
Cell 4	22	34
…	…	…
Cell 100	40	21





Goal: how many distinct cell types (clusters) exist in this data?

Example 2 is an important problem in many applications

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

Genome Biology

RESEARCH Open Access

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

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

*Correspondence:
pengyi.yang@sydney.edu.au
[3] Charles Perkins Centre, University of Sydney, Sydney, NSW 2006, Australia
Full list of author information is available at the end of the article

Abstract

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

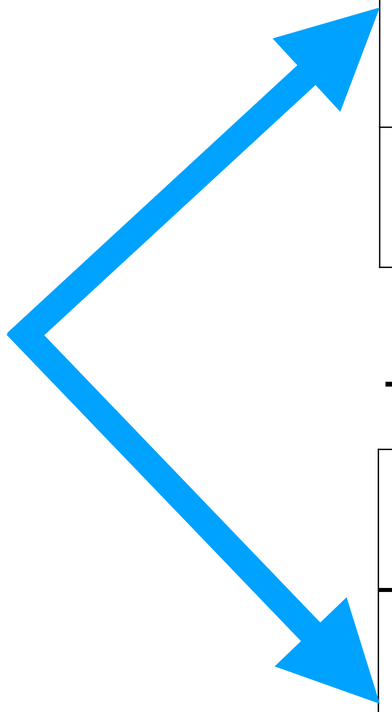
Outline

1. Motivation: settings where sample splitting doesn’t work
2. **Poisson thinning**
3. Data thinning
4. Real data application
5. Ongoing work

Reminder: sample splitting does not help us determine the number of clusters in a dataset

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

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

$X^{(1)}$ (train)

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

$X^{(2)}$ (test)

	Gene 1	Gene 2
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}

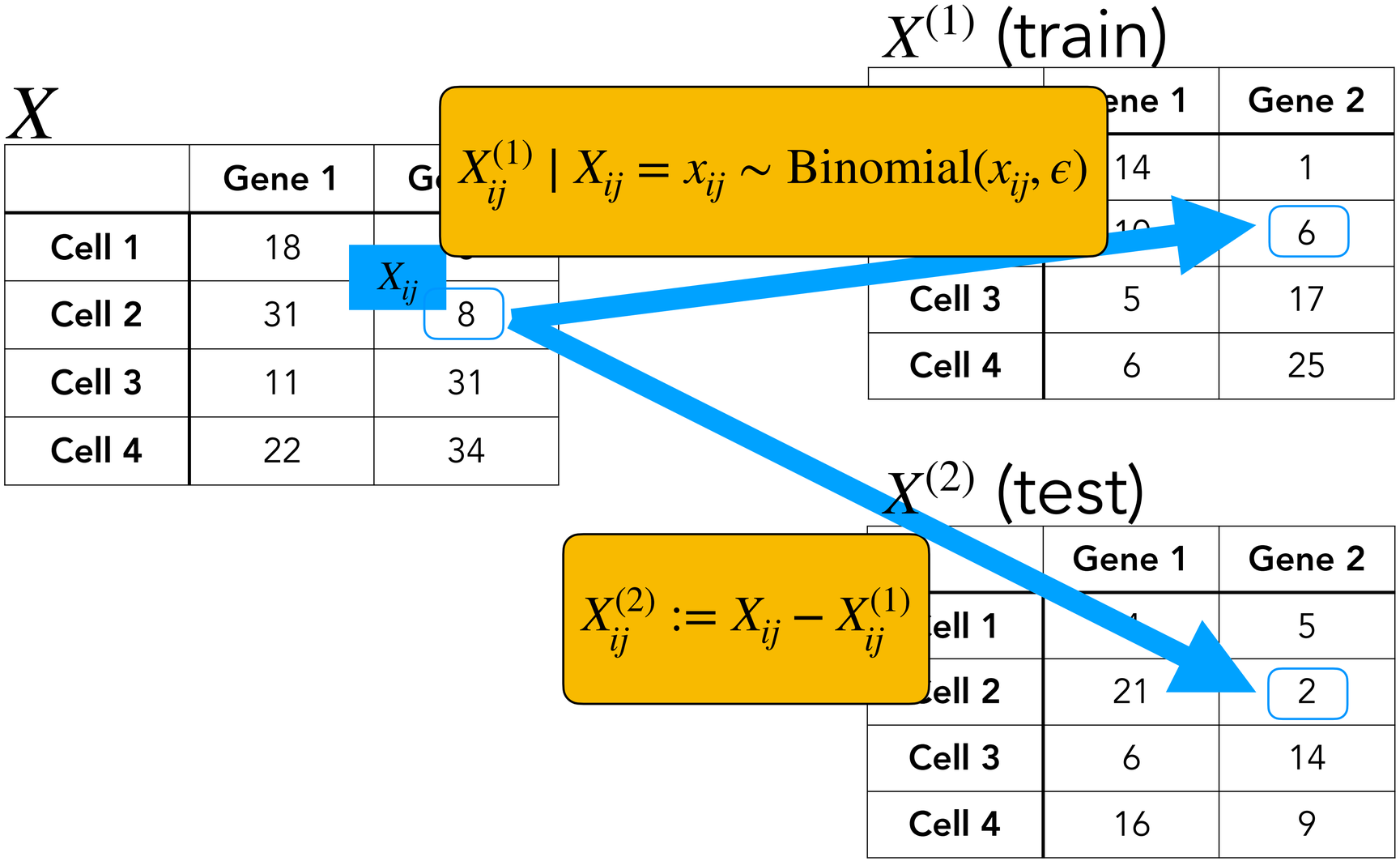
$X^{(1)}$ (train)

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

$X^{(2)}$ (test)

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

An alternative: Poisson thinning



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}

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

$X^{(1)}$ (train)

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

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

$X^{(2)}$ (test)

	Gene 1	Gene 2
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}

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

$X^{(1)}$ (train)

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

Estimate clusters.

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

$X^{(2)}$ (test)

	Gene 1	Gene 2
ell 1	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}

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

$X^{(1)}$ (train)

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

Estimate clusters.

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

$X^{(2)}$ (test)

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

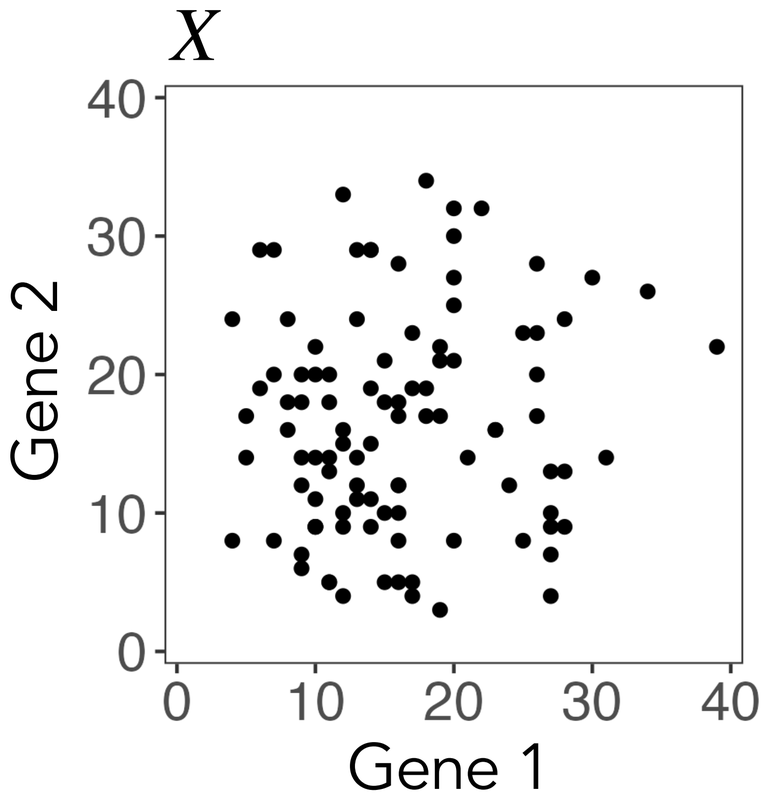
Evaluate clusters.

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

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

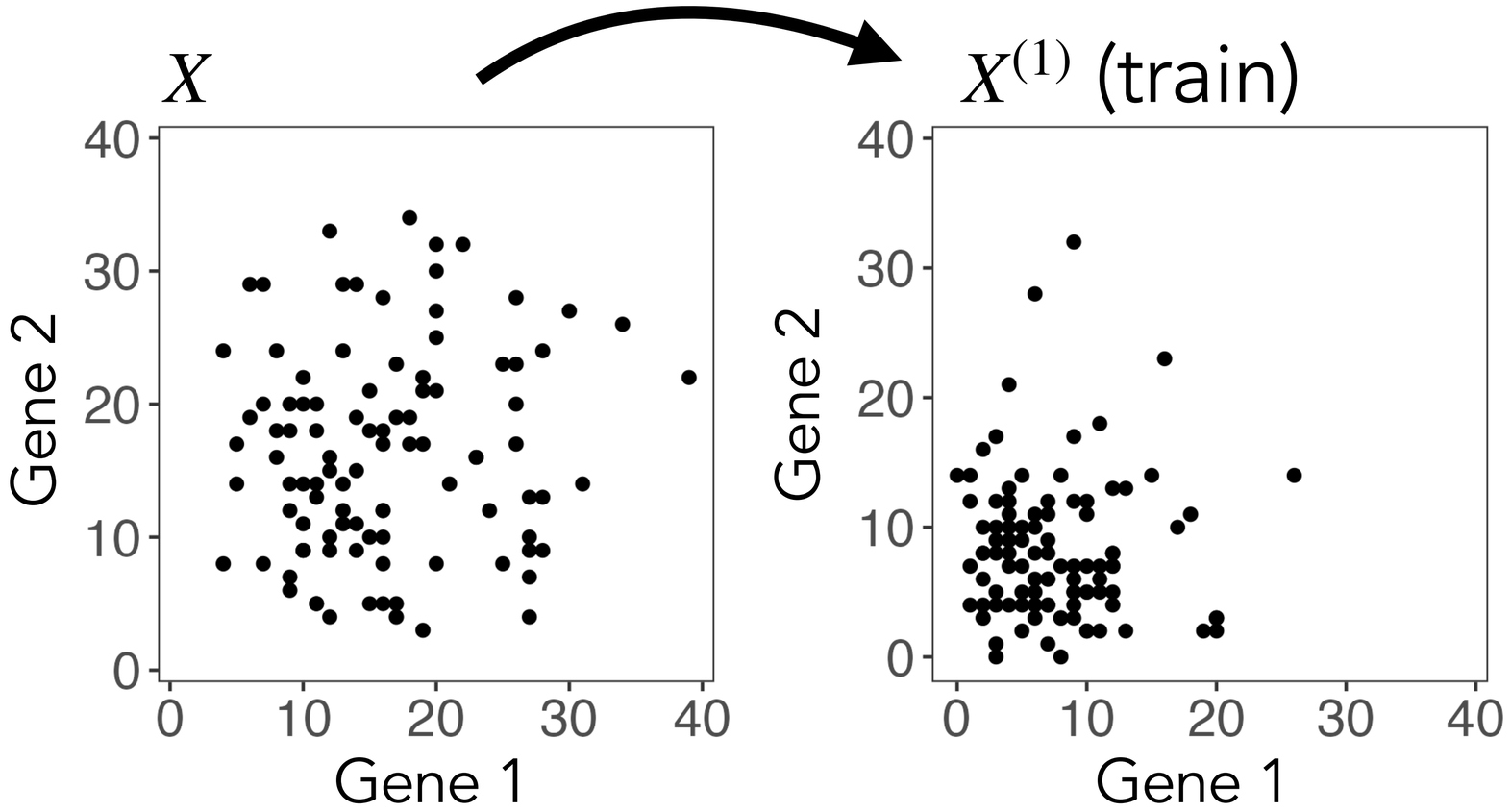
A very well-known result.

Visualizing thinning on a dataset with one true cluster



Visualizing thinning on a dataset with one true cluster

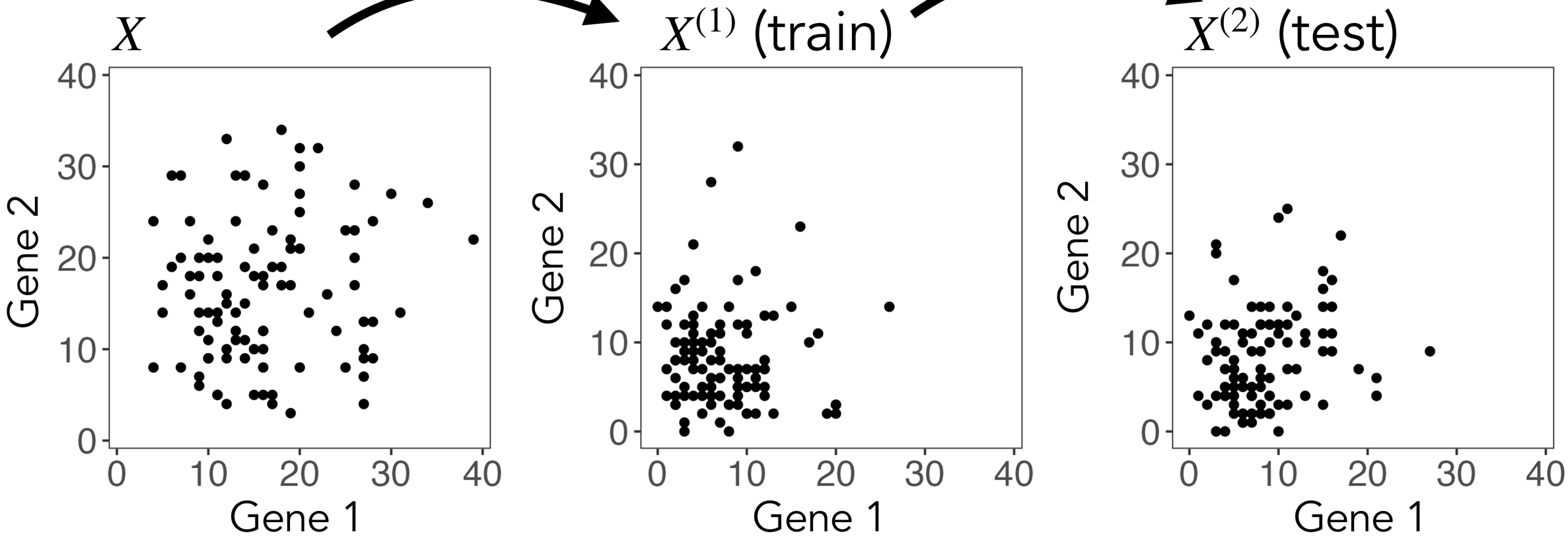
$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5)$$



Visualizing thinning on a dataset with one true cluster

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

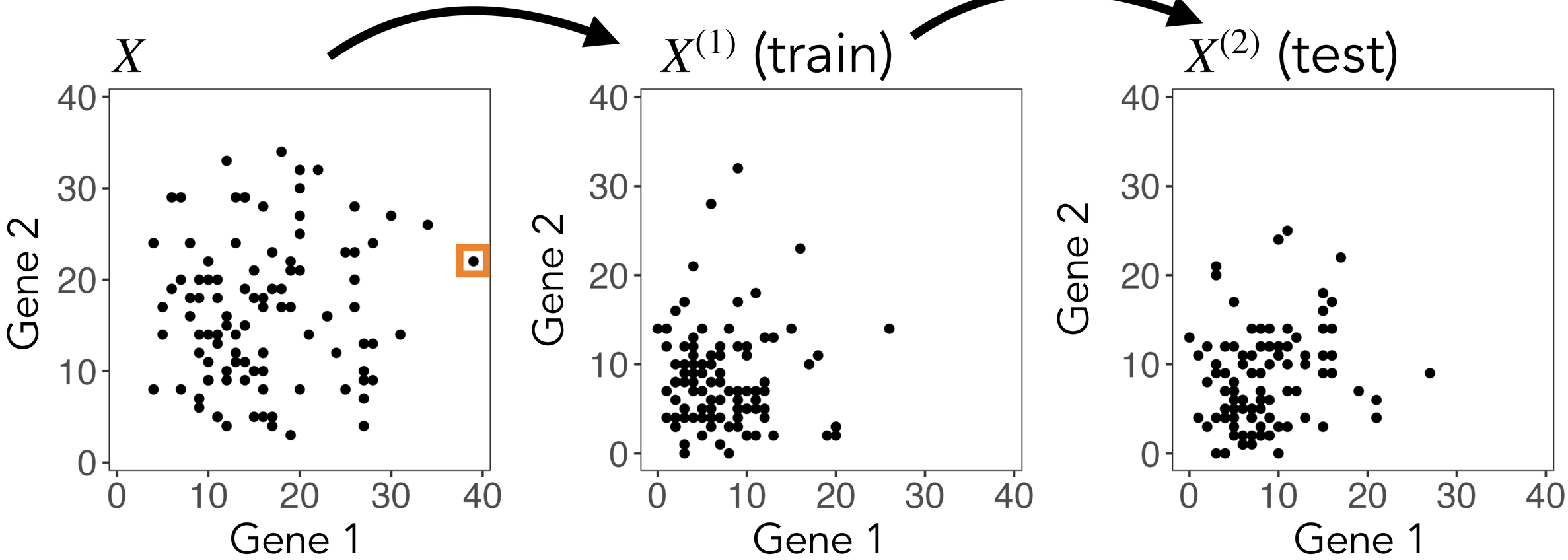
$$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



Visualizing thinning on a dataset with one true cluster

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

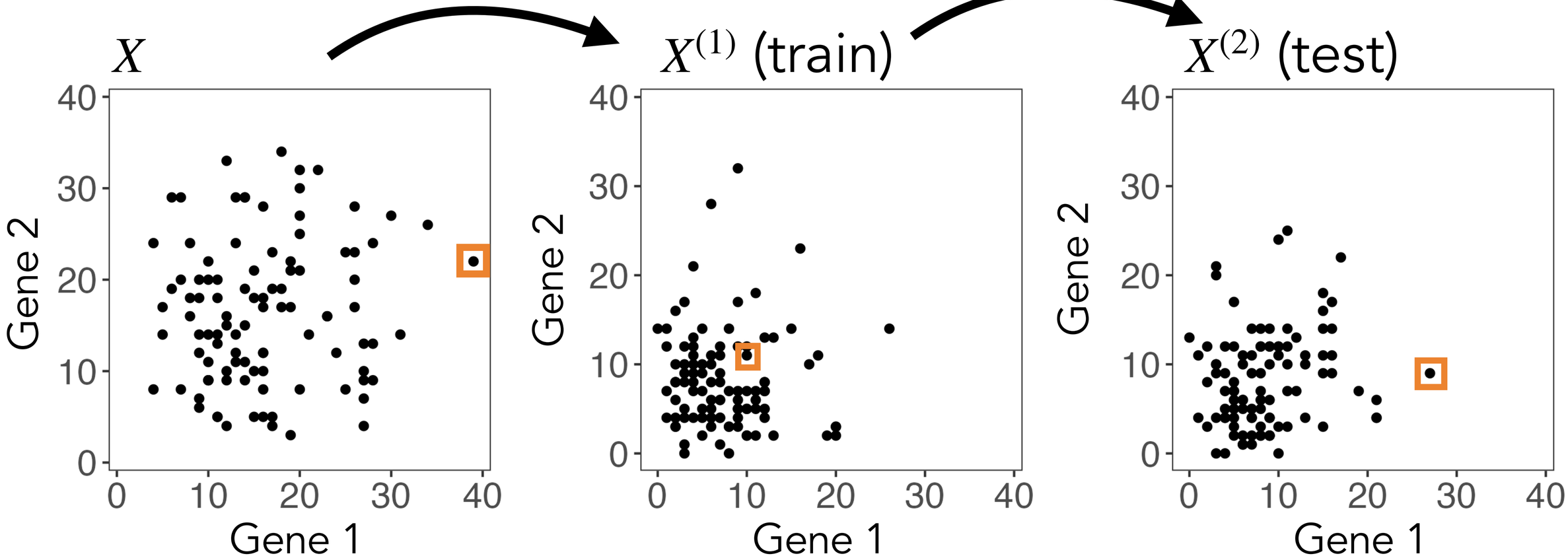
$$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



Visualizing thinning on a dataset with one true cluster

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

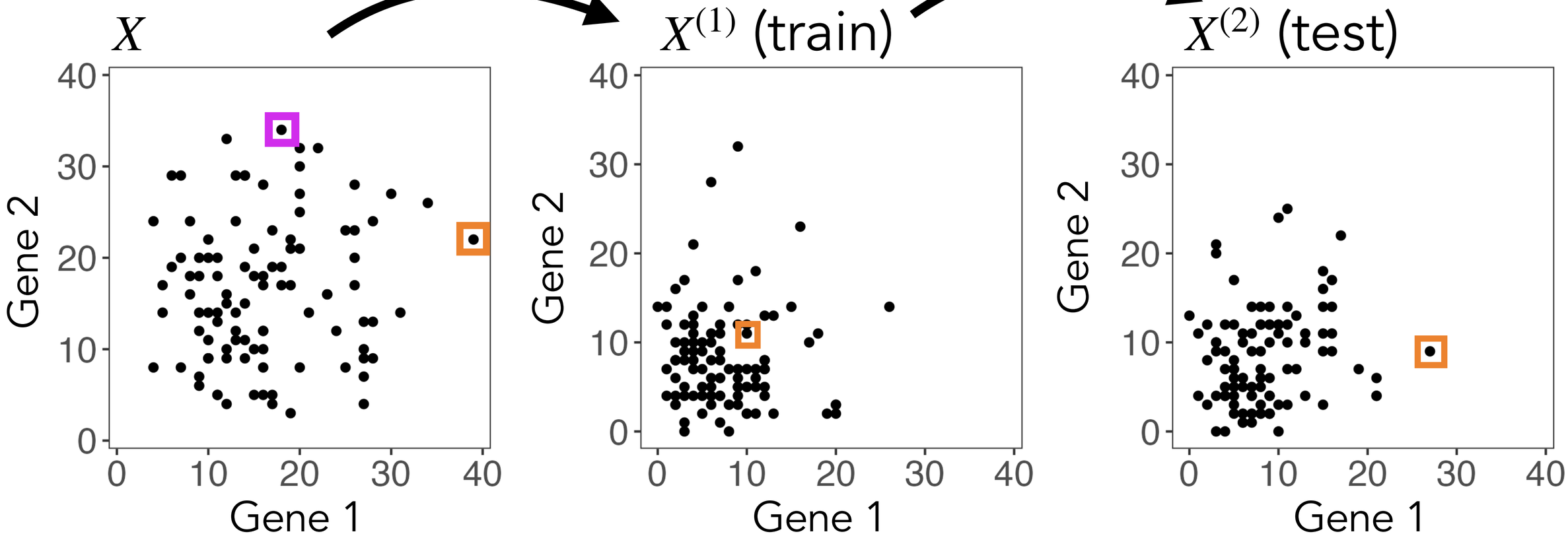
$$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



Visualizing thinning on a dataset with one true cluster

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

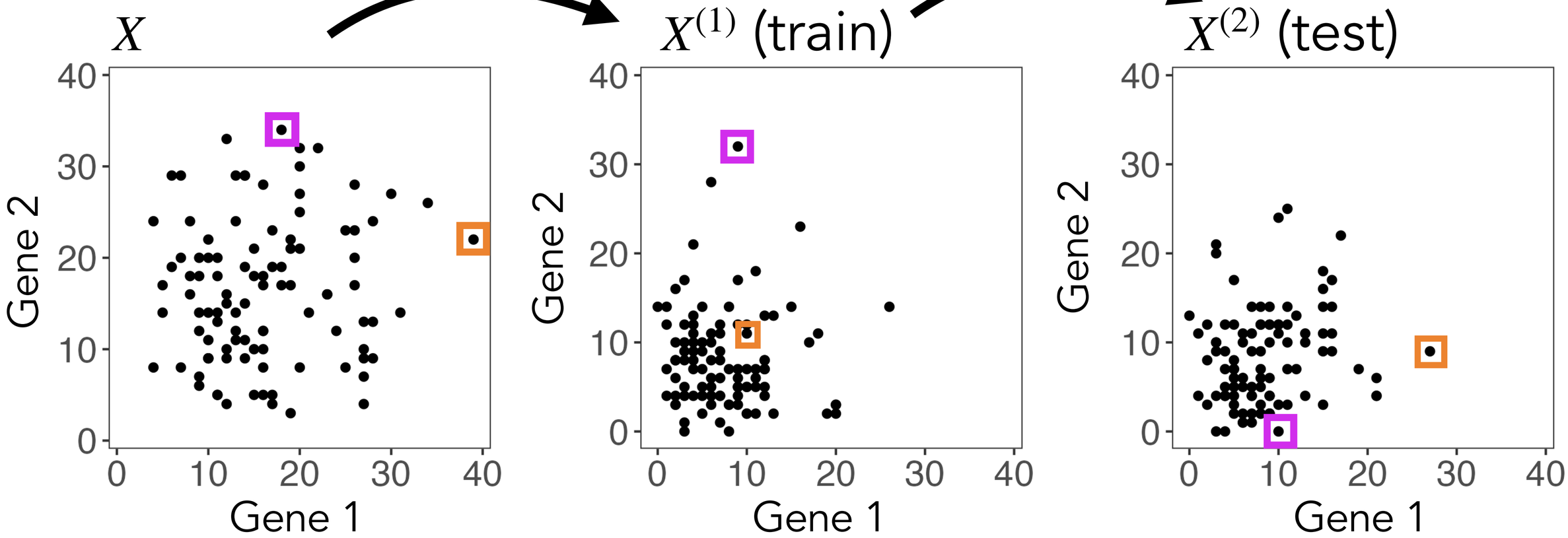
$$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



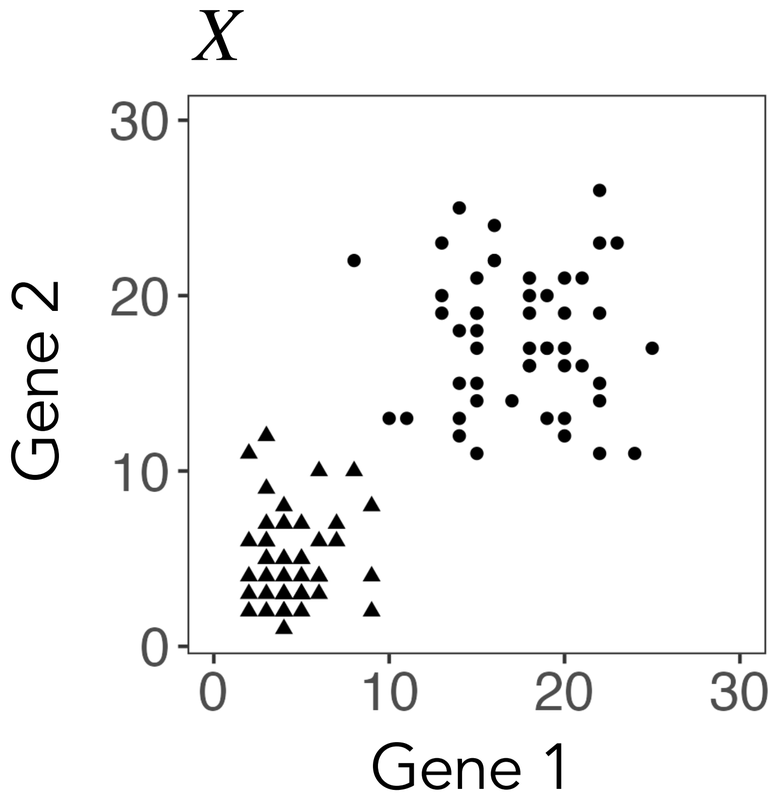
Visualizing thinning on a dataset with one true cluster

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

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

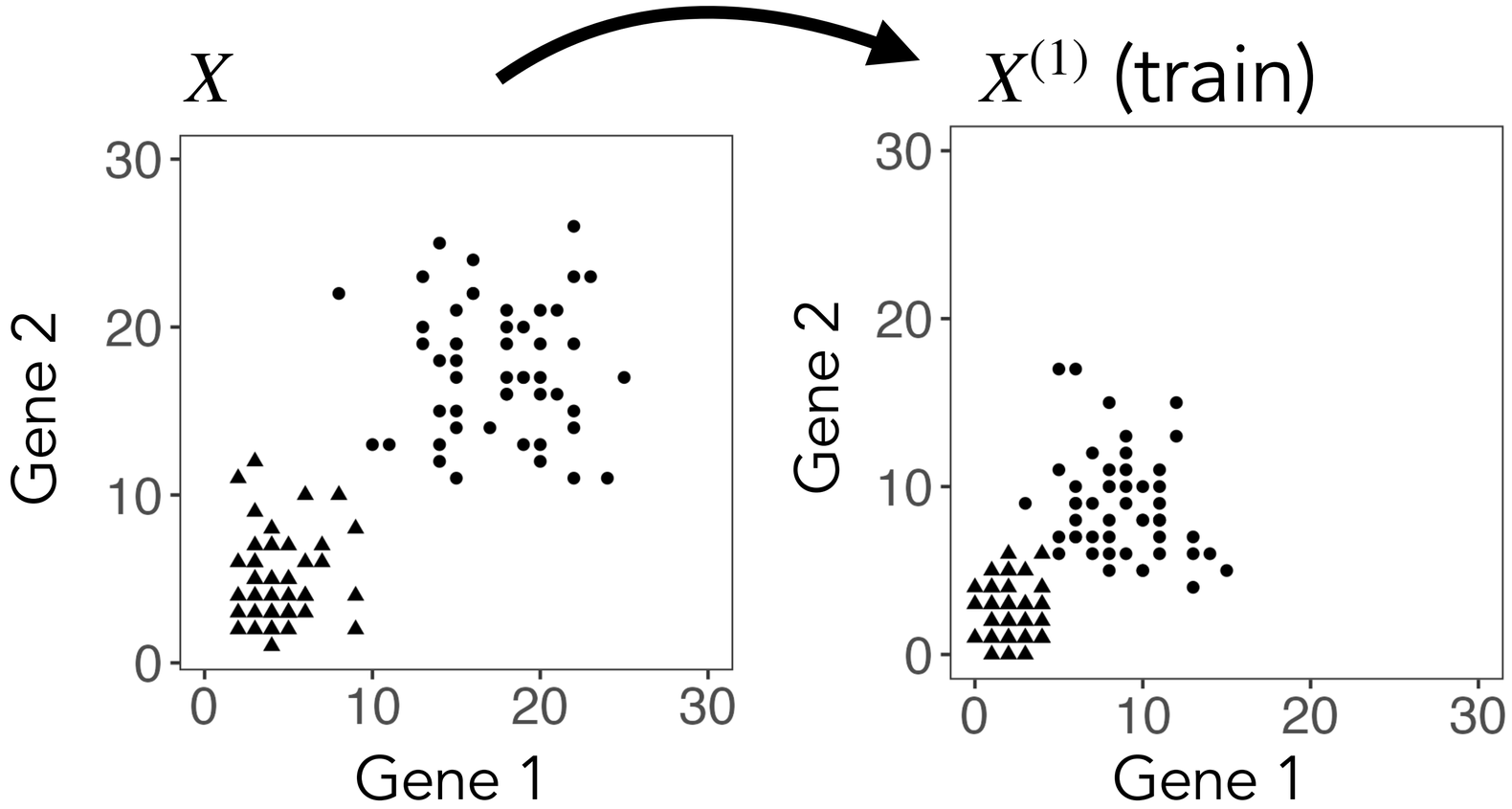


Visualizing thinning on a dataset with two true clusters



Visualizing thinning on a dataset with two true clusters

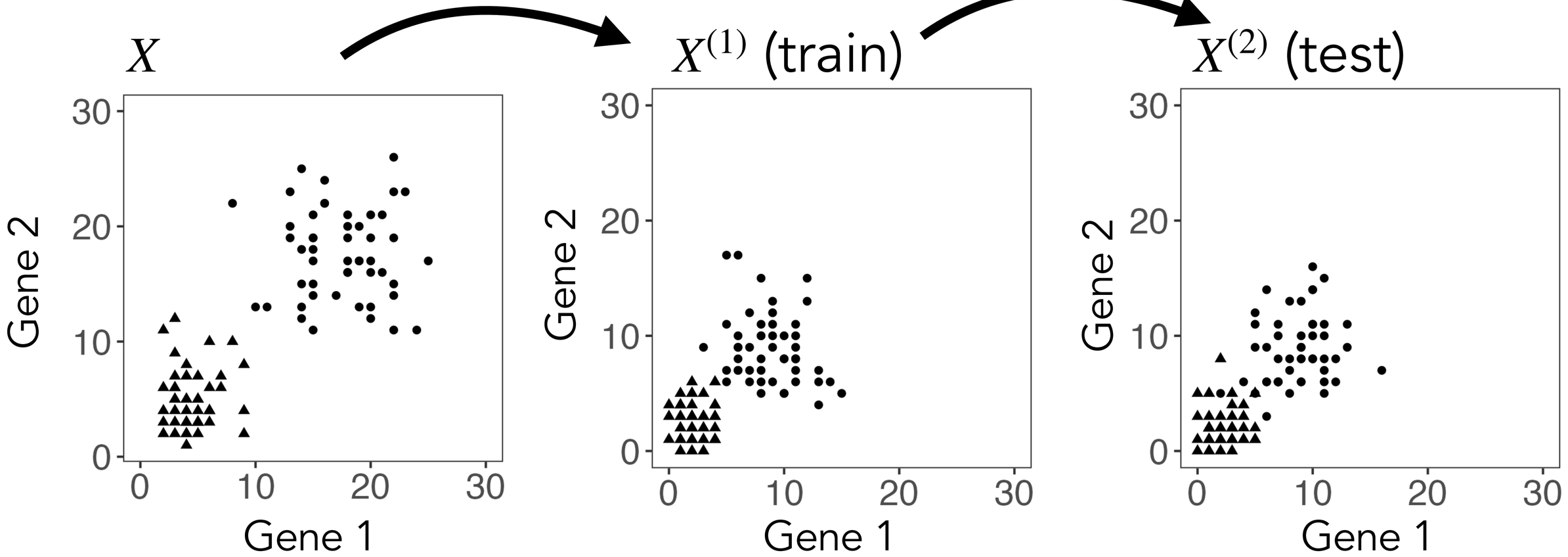
$$X_{ij}^{(1)} \mid X_{ij} = x_{ij} \sim \text{Binomial}(x_{ij}, 0.5)$$



Visualizing thinning on a dataset with two true clusters

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

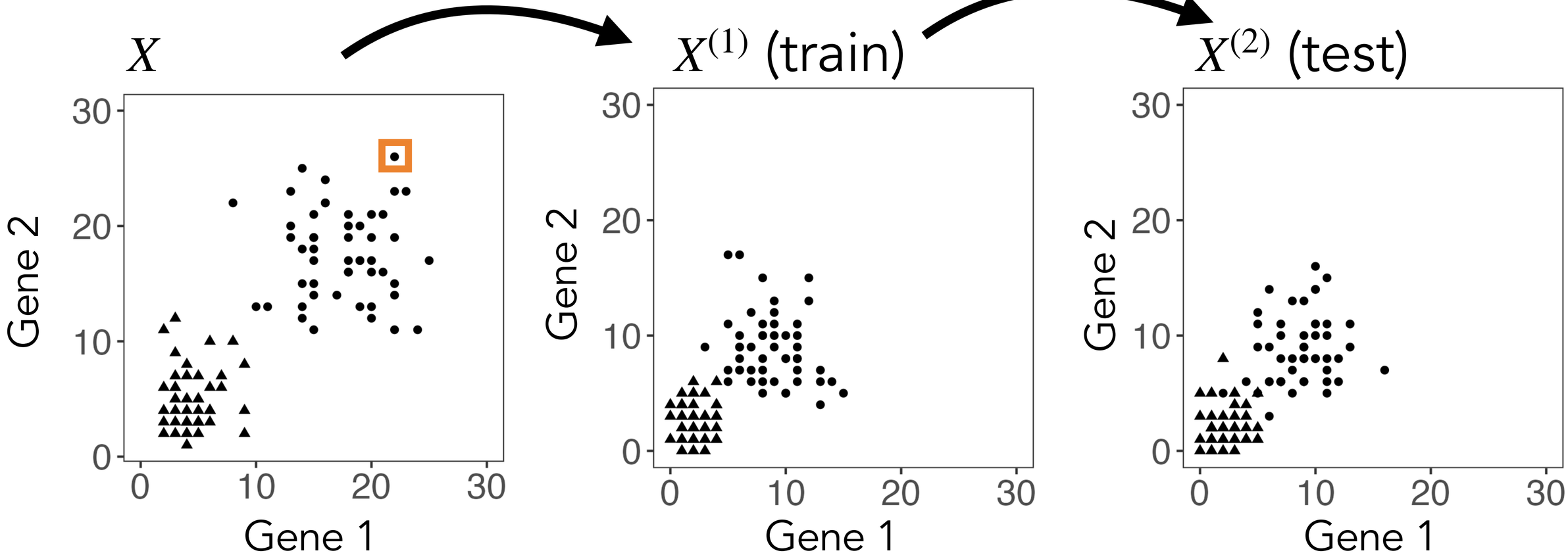
$$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



Visualizing thinning on a dataset with two true clusters

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

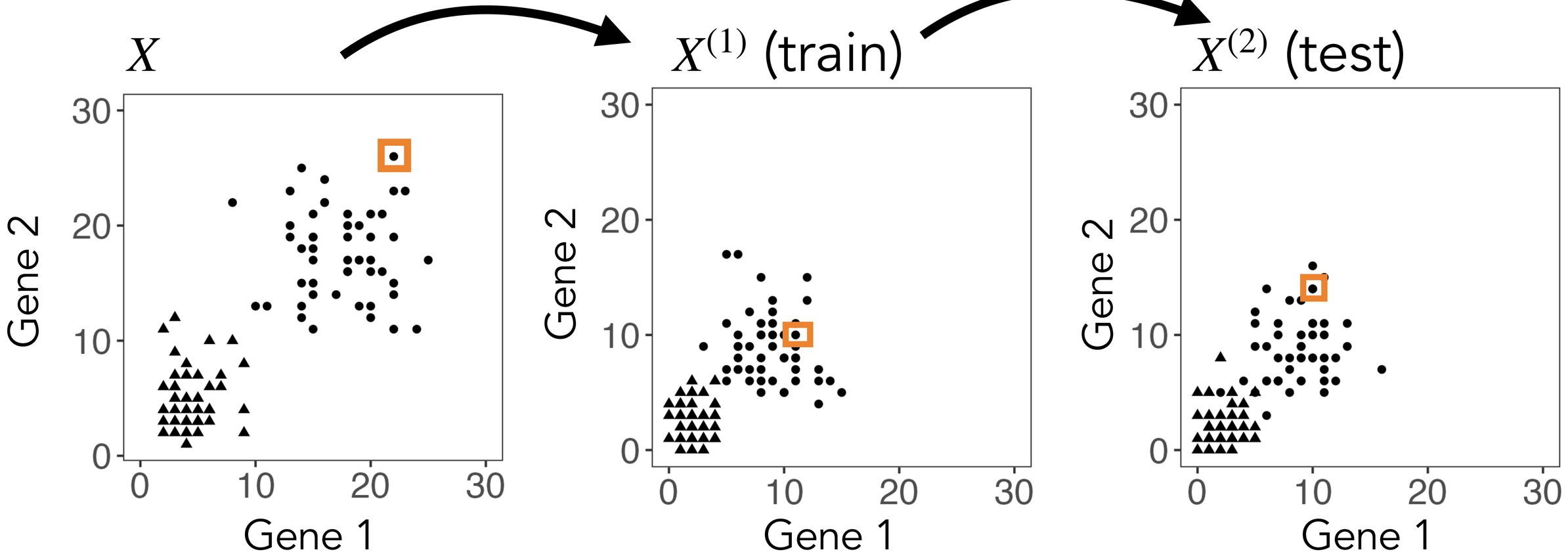
$$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



Visualizing thinning on a dataset with two true clusters

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

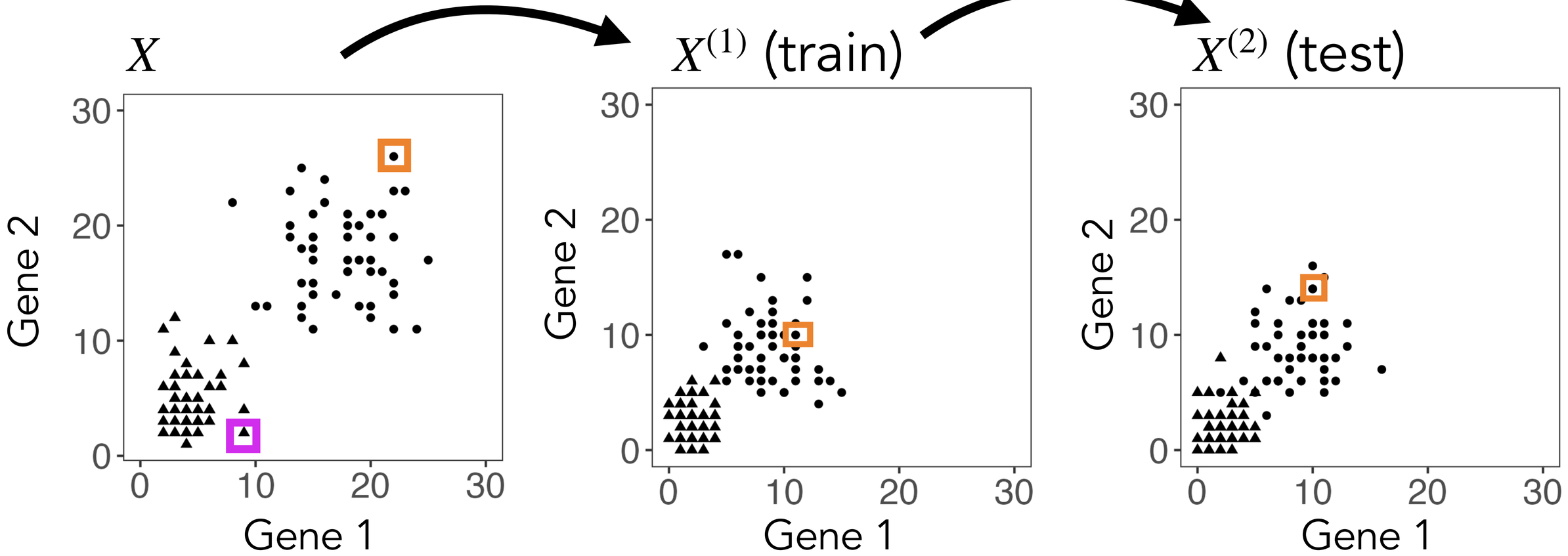
$$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



Visualizing thinning on a dataset with two true clusters

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

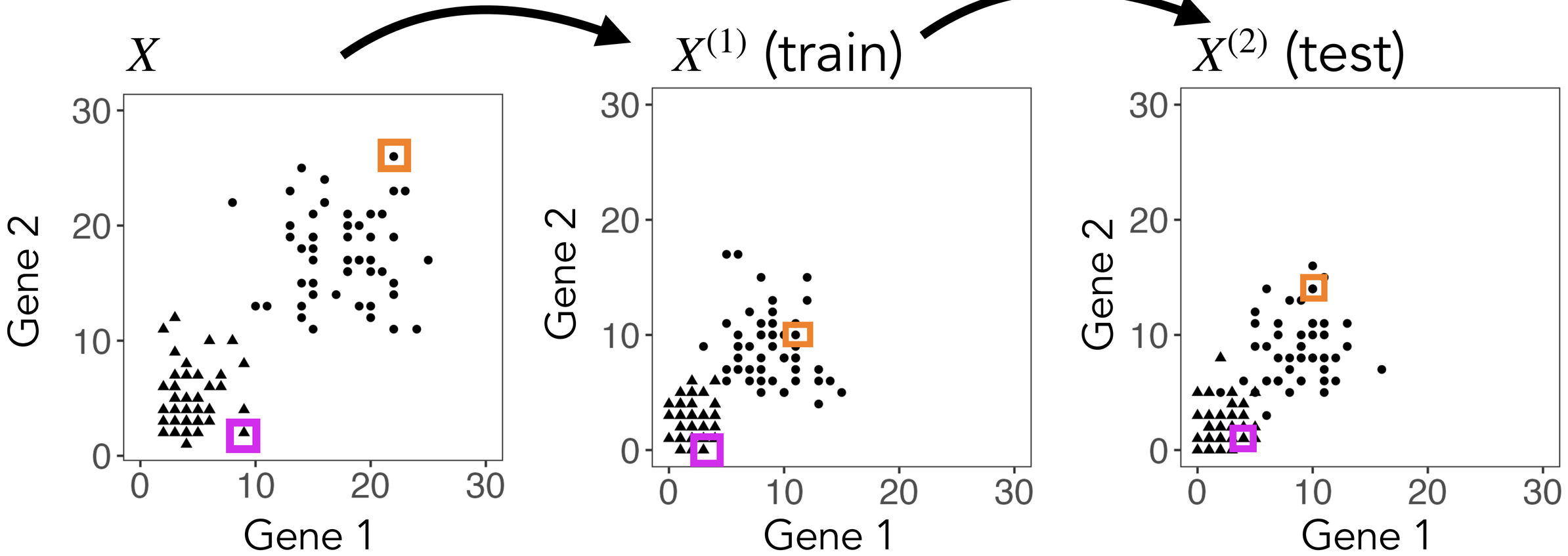
$$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$



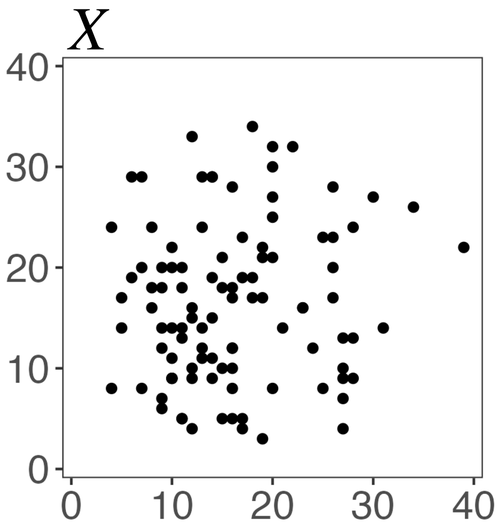
Visualizing thinning on a dataset with two true clusters

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

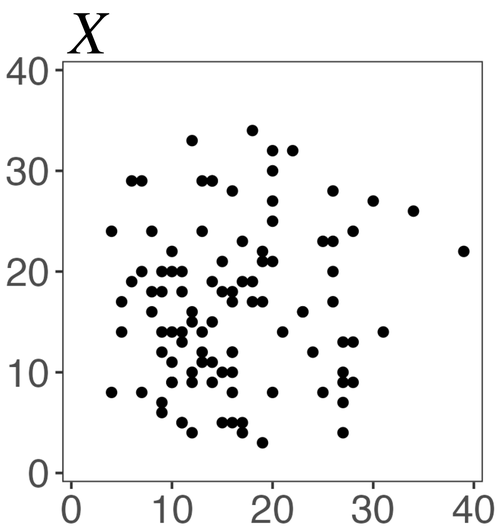
$$X_{ij}^{(2)} = X_{ij} - X_{ij}^{(1)}$$

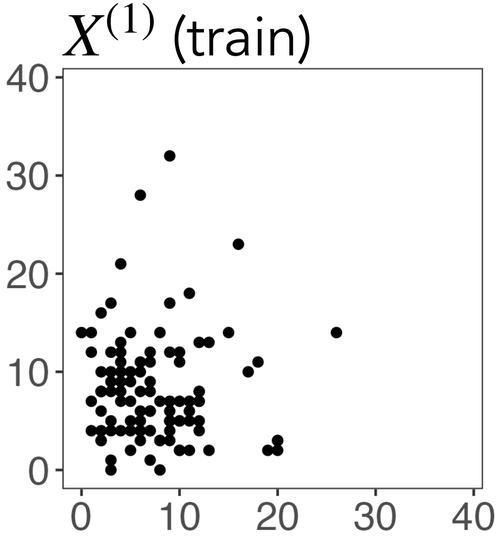


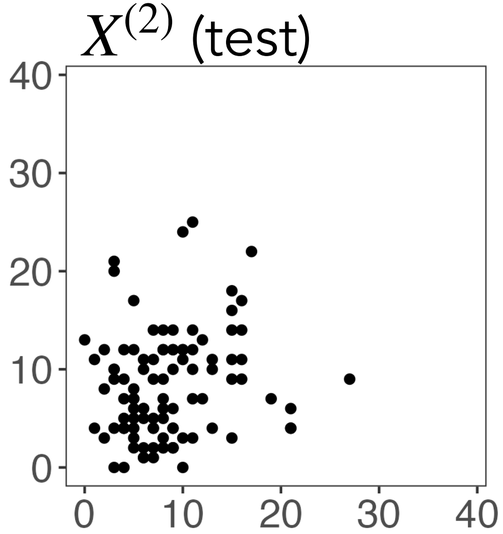
Thinning avoids the pitfall of sample splitting in Example 2



Thinning avoids the pitfall of sample splitting in Example 2

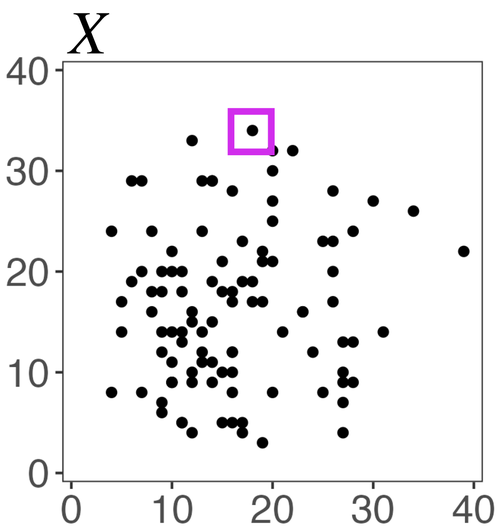


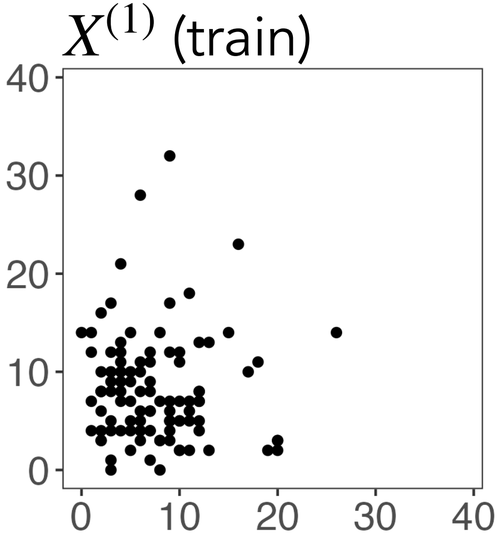


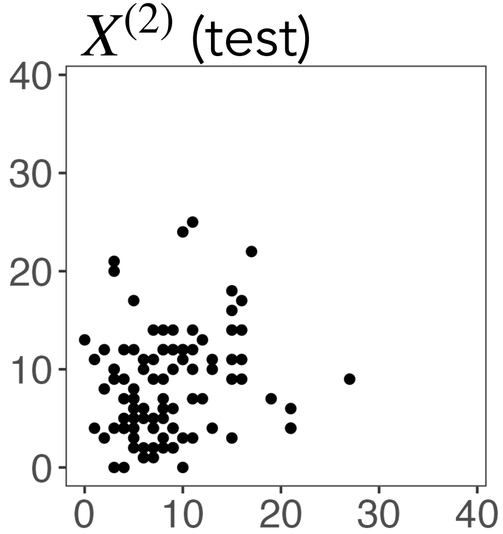


Step 1: thin observations into train/test.

Thinning avoids the pitfall of sample splitting in Example 2

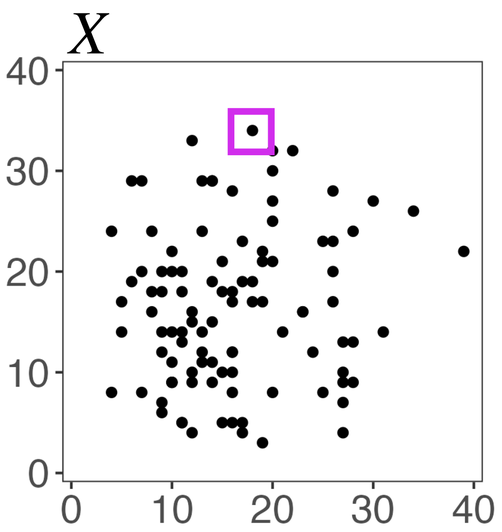


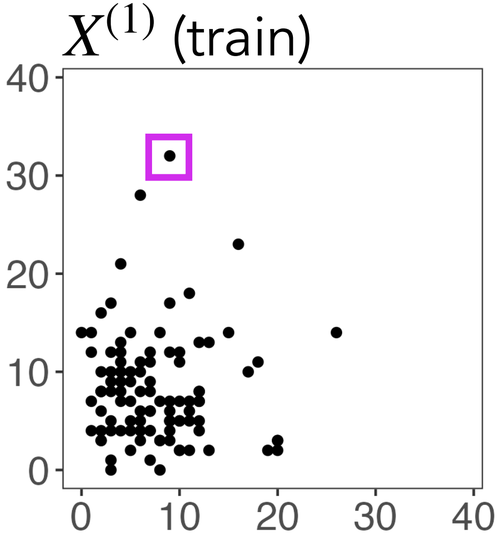


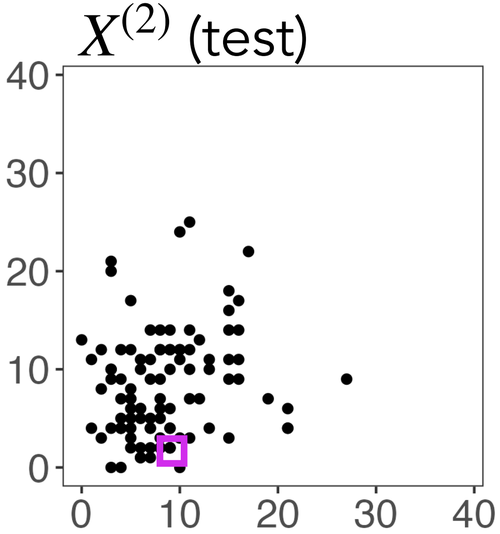


Step 1: thin observations into train/test.

Thinning avoids the pitfall of sample splitting in Example 2

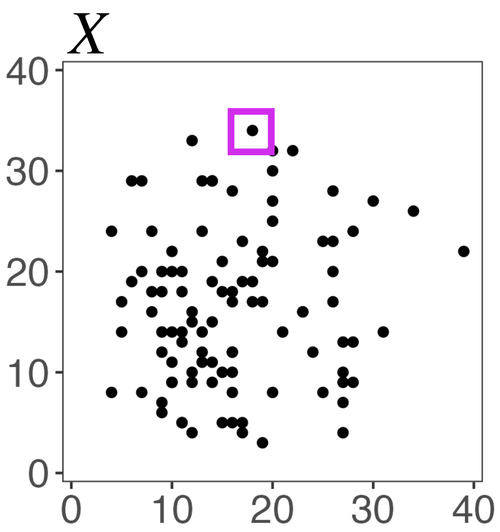


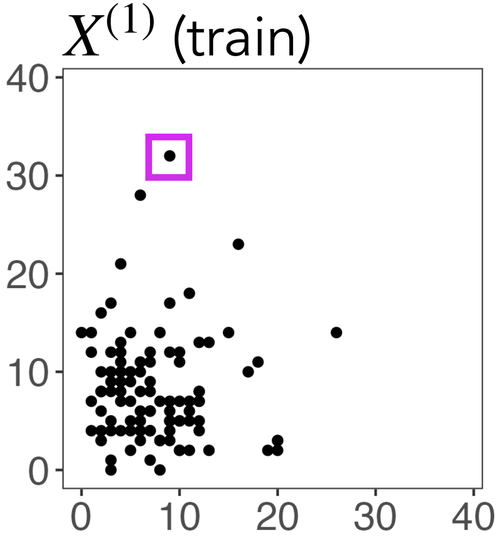


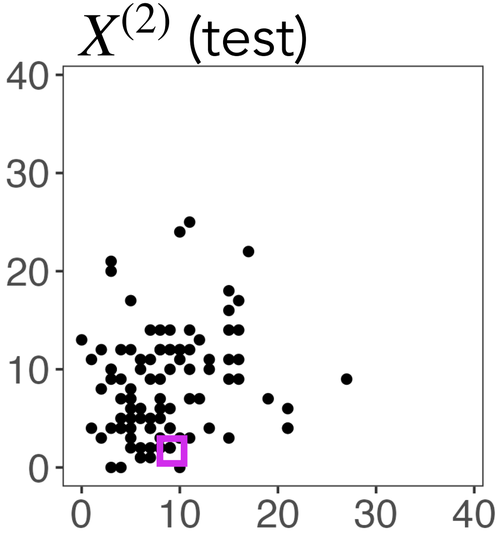


Step 1: thin observations into train/test.

Thinning avoids the pitfall of sample splitting in Example 2



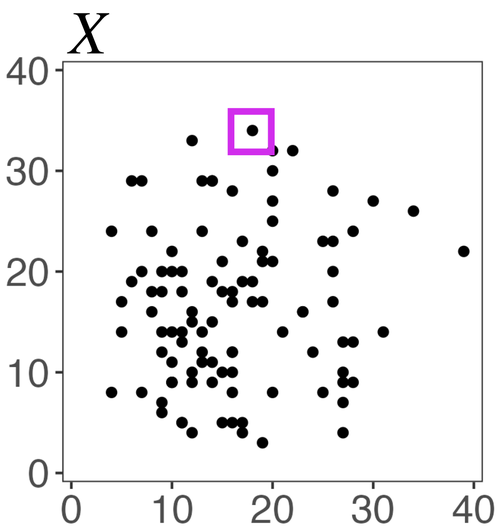


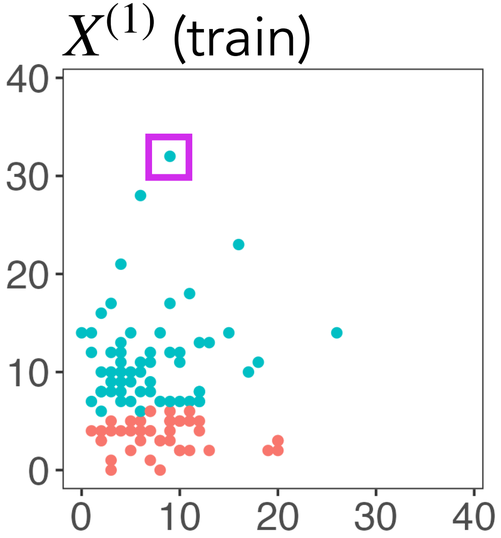


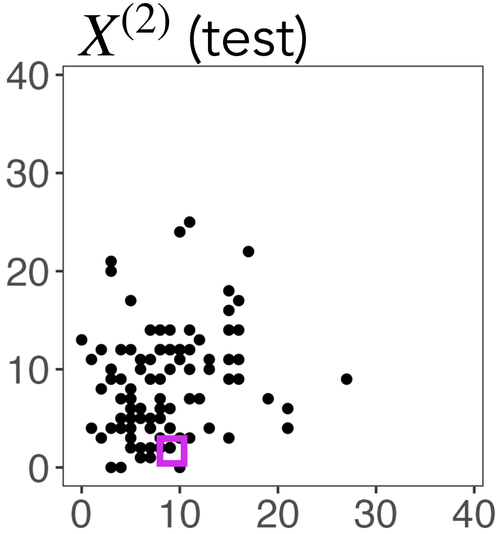
Step 1: thin observations into train/test.

Step 2: cluster the training set.

Thinning avoids the pitfall of sample splitting in Example 2



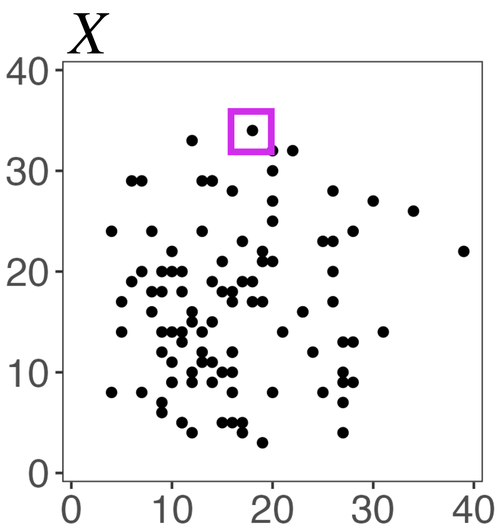


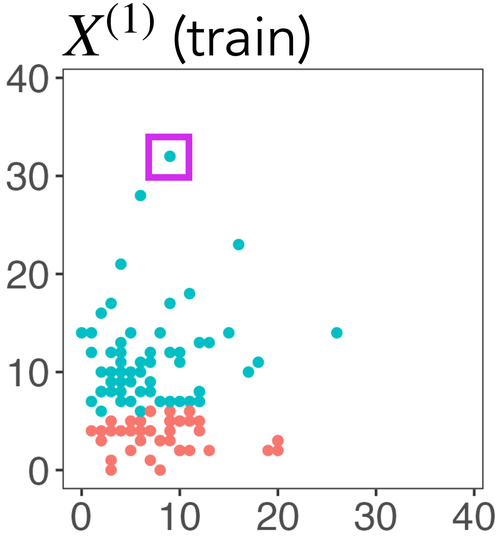


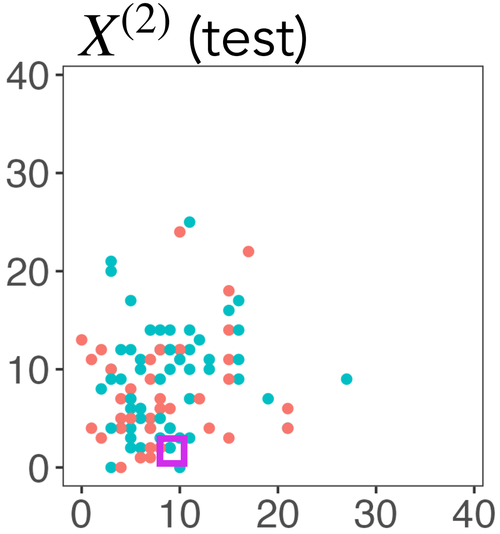
Step 1: thin observations into train/test.

Step 2: cluster the training set.

Thinning avoids the pitfall of sample splitting in Example 2



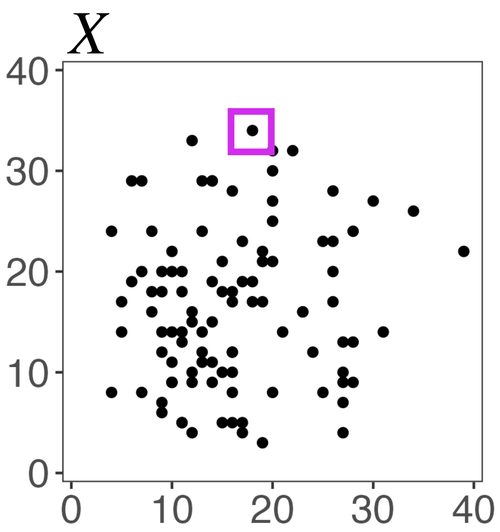


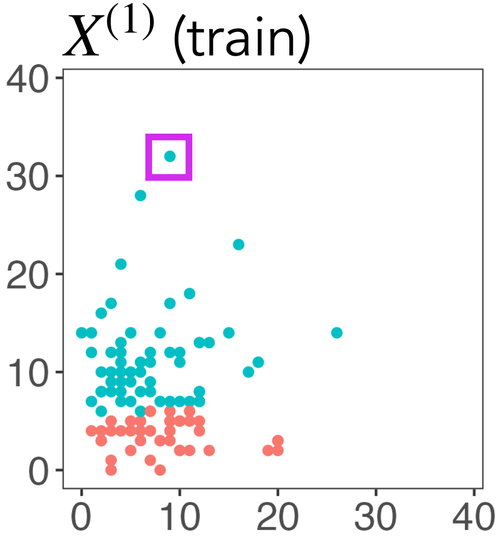


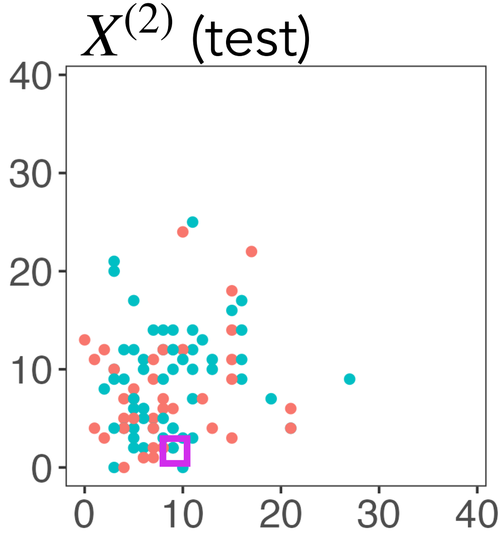
Step 1: thin observations into train/test.

Step 2: cluster the training set.

Thinning avoids the pitfall of sample splitting in Example 2





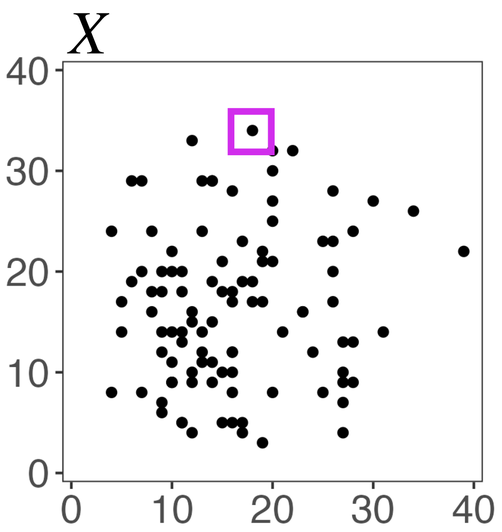


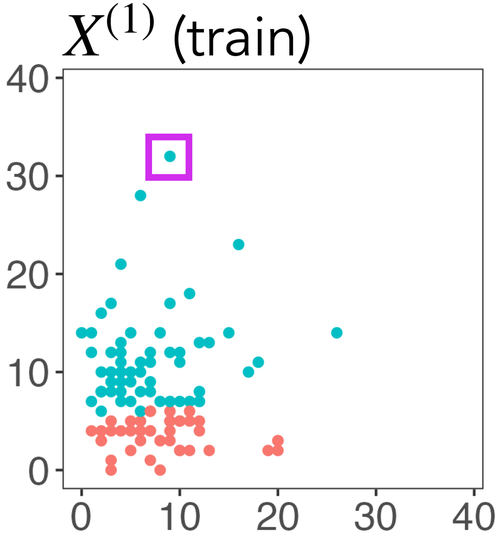
Step 1: thin observations into train/test.

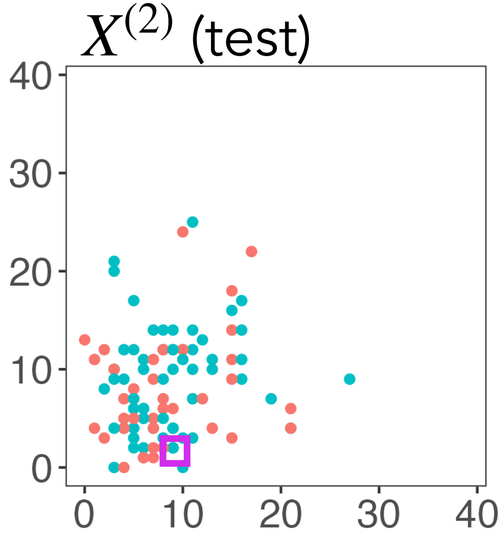
Step 2: cluster the training set.

Step 3: evaluate clusters on test set.

Thinning avoids the pitfall of sample splitting in Example 2



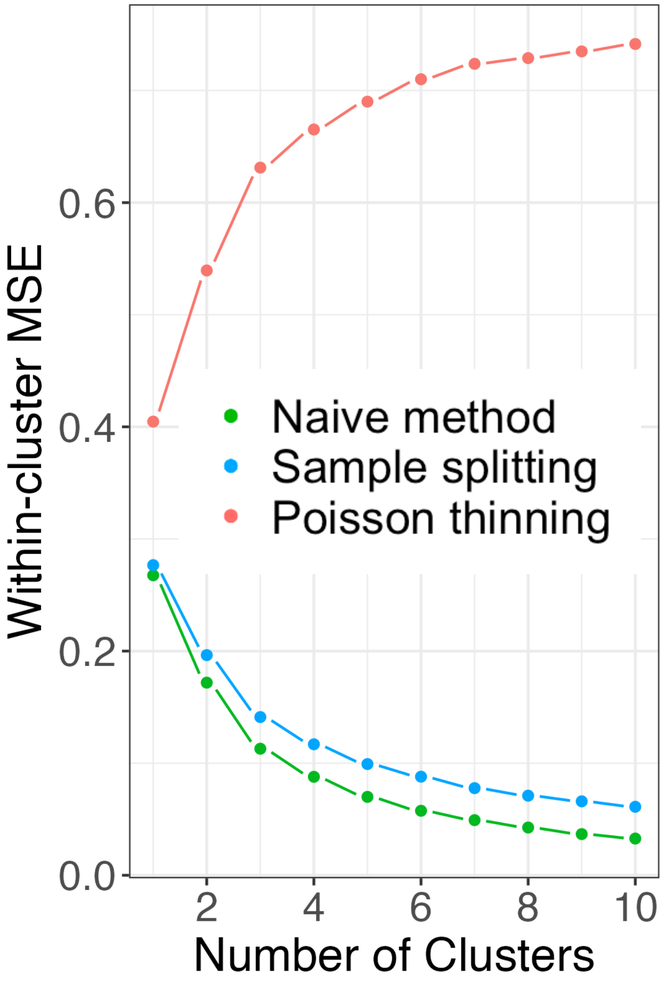




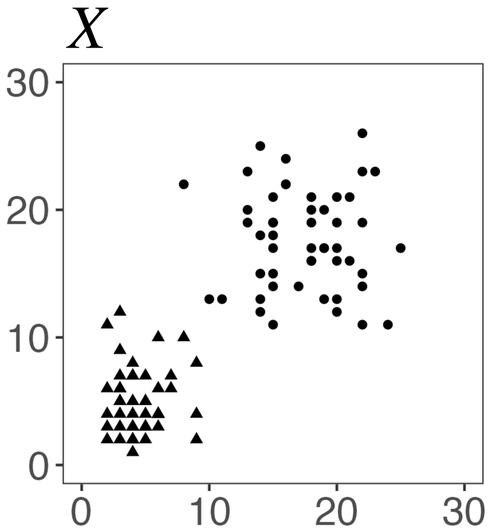
Step 1: thin observations into train/test.

Step 2: cluster the training set.

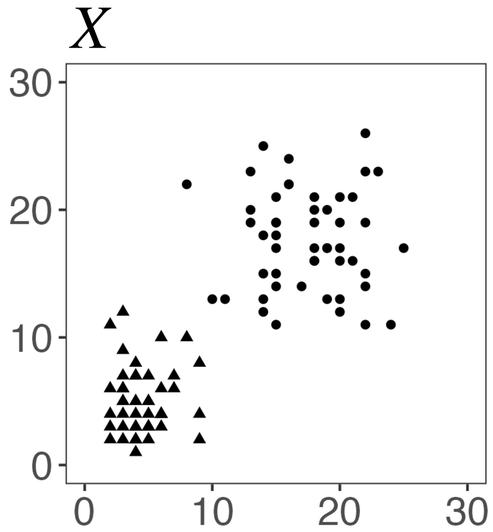
Step 3: evaluate clusters on test set.



Thinning avoids the pitfall of sample splitting in Example 2

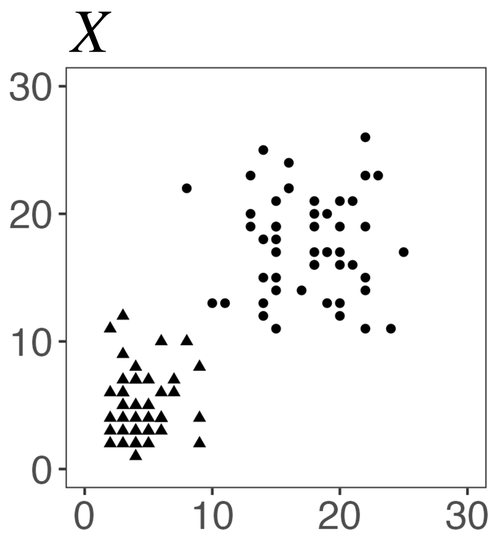


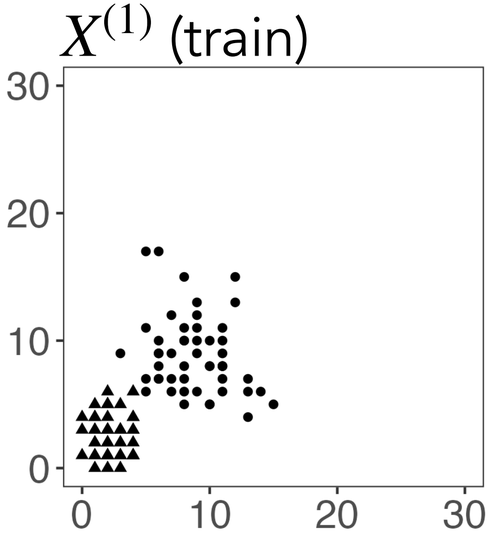
Thinning avoids the pitfall of sample splitting in Example 2

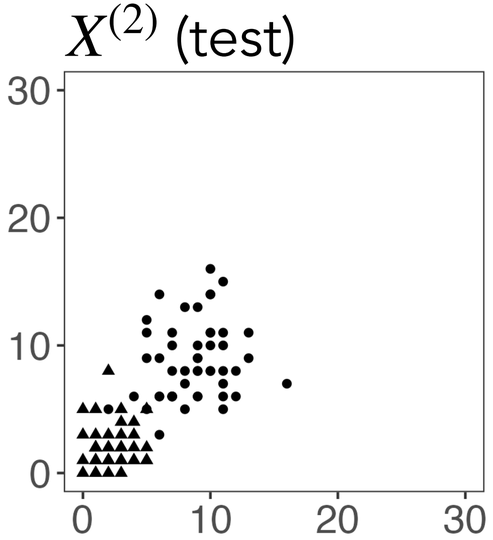


Step 1: thin observations into train/test.

Thinning avoids the pitfall of sample splitting in Example 2

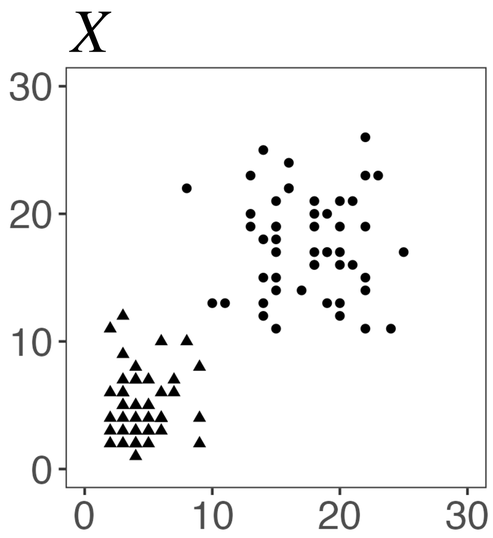


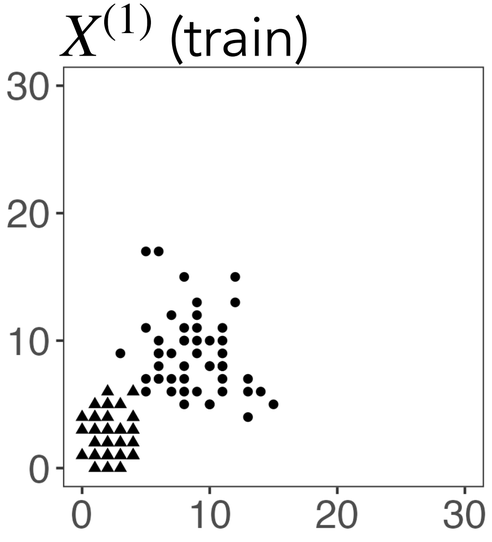


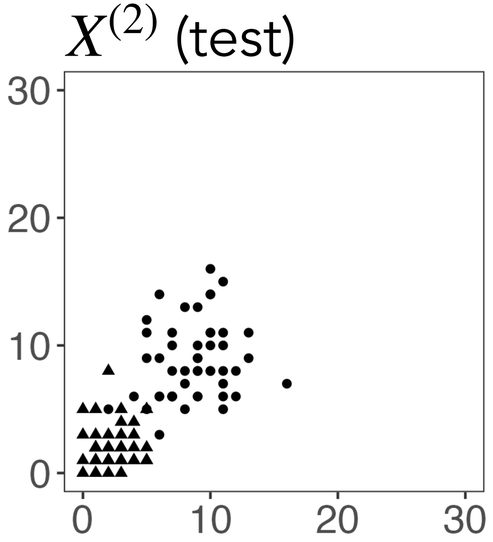


Step 1: thin observations into train/test.

Thinning avoids the pitfall of sample splitting in Example 2



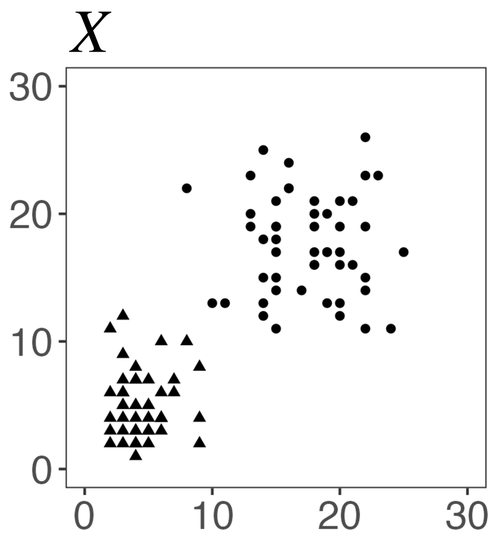


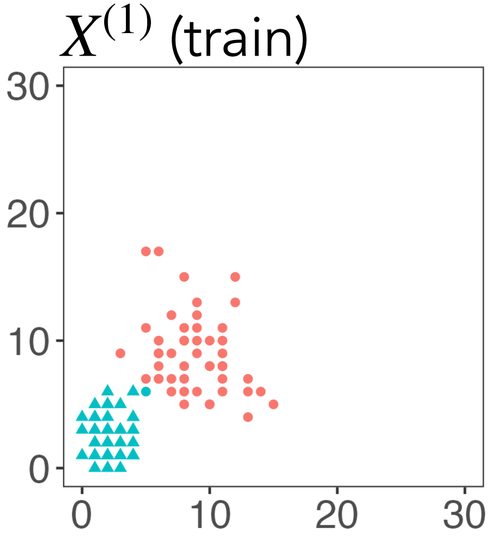


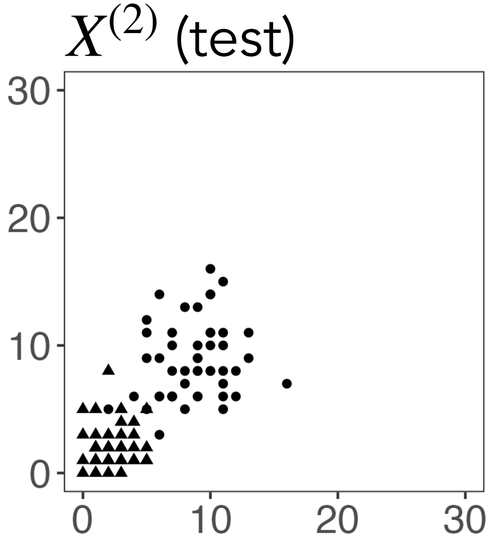
Step 1: thin observations into train/test.

Step 2: cluster the training set.

Thinning avoids the pitfall of sample splitting in Example 2



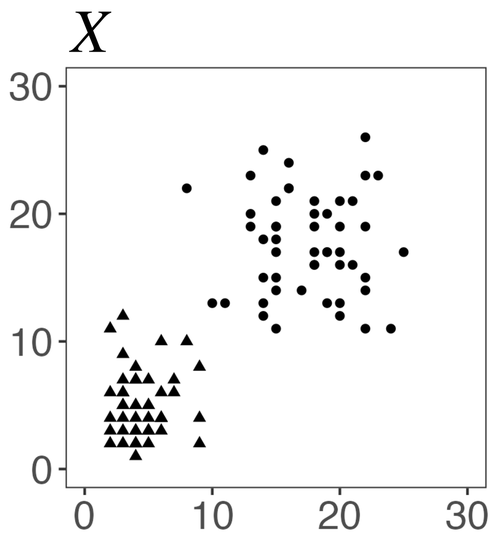


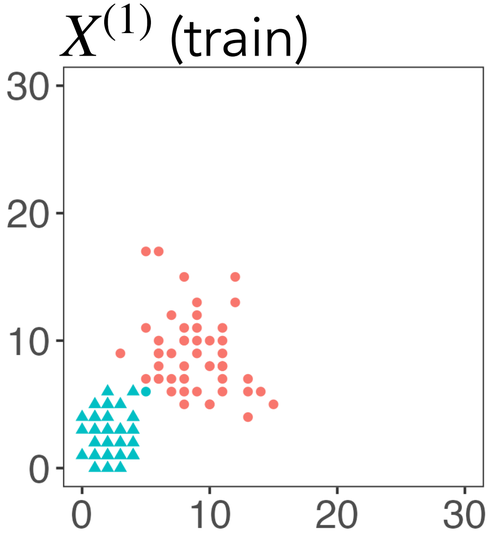


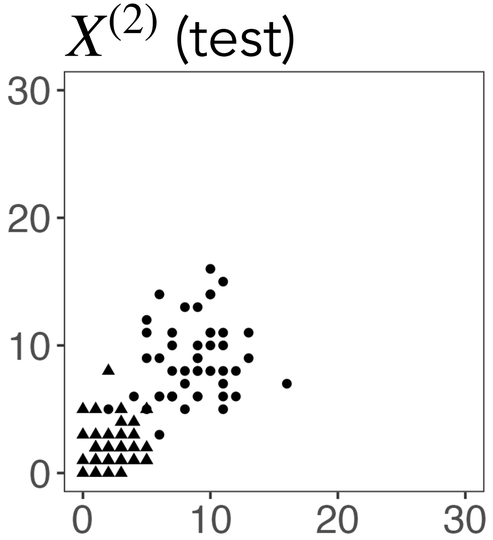
Step 1: thin observations into train/test.

Step 2: cluster the training set.

Thinning avoids the pitfall of sample splitting in Example 2





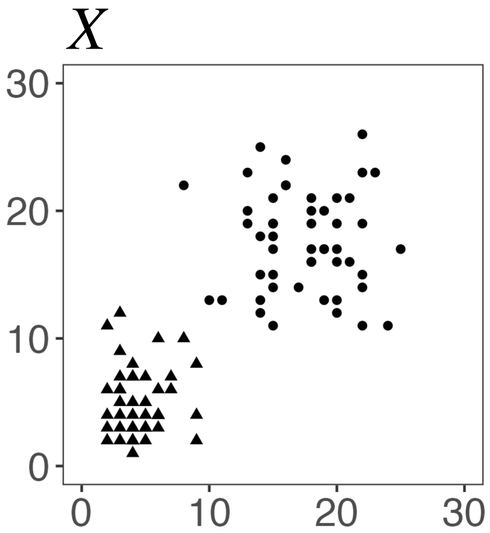


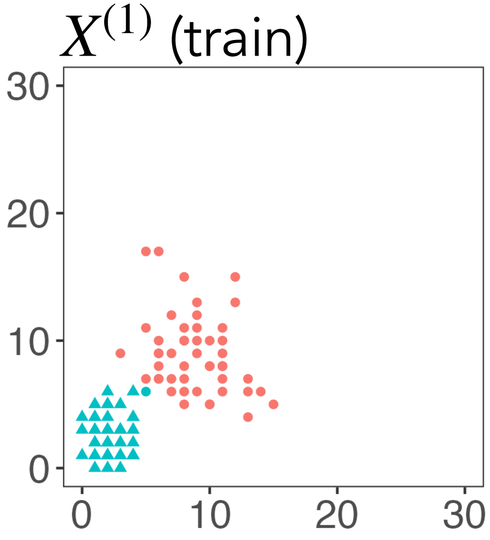
Step 1: thin observations into train/test.

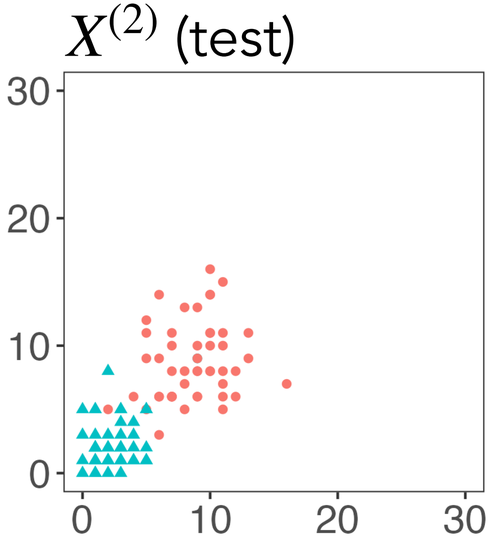
Step 2: cluster the training set.

Step 3: evaluate clusters on test set.

Thinning avoids the pitfall of sample splitting in Example 2





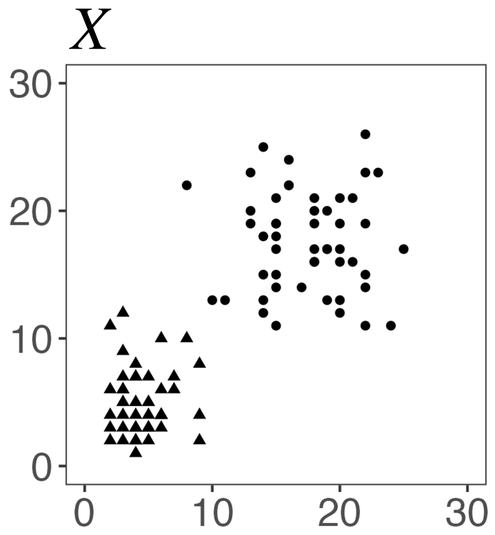


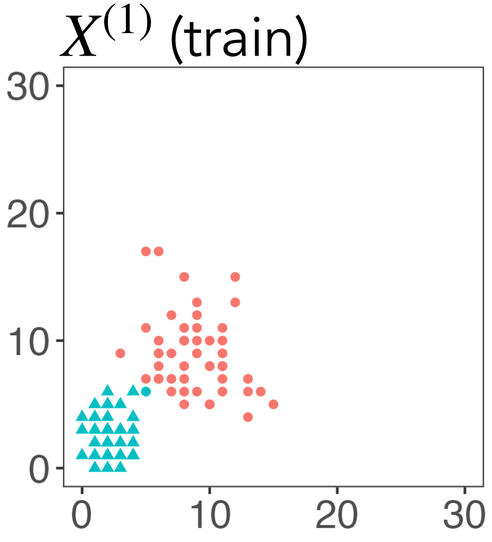
Step 1: thin observations into train/test.

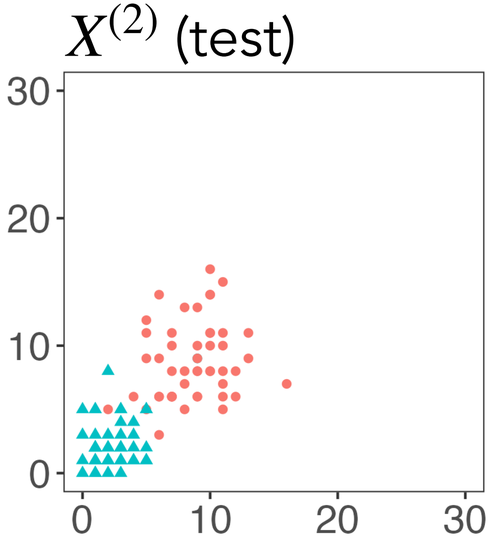
Step 2: cluster the training set.

Step 3: evaluate clusters on test set.

Thinning avoids the pitfall of sample splitting in Example 2



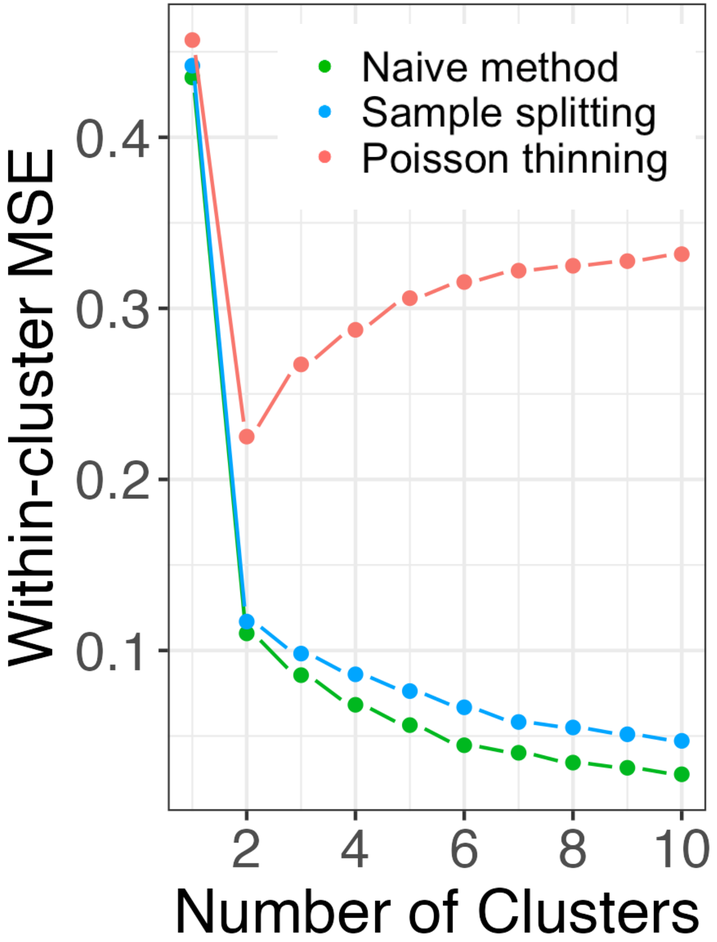




Step 1: thin observations into train/test.

Step 2: cluster the training set.

Step 3: evaluate clusters on test set.



Poisson thinning is useful in the analysis of scRNA sequencing data

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 Gurvev[35], Rens Holmer[36], Katharina Jahn[15,16], Thamar Jessurun Lobo[35], Emma M. Keizer[37], Tzu-Hao Kuo[3], Bou Tobias Marschall[47] Jeroen de Ridder[2] Fabian J. Theis[54], H Sohrab P. Shah[59] a

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).

Poisson thinning is useful in the analysis of scRNA sequencing data

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 Gurvev[35], Rens Holmer[36], Katharina Jahn[15,16], Thamar Jessurun Lobo[35], Emma M. Keizer[37], Tzu-Hao Kuo[3], Bou… Tobias Marschall[47], Jeroen de Ridder[2], Fabian J. Theis[54], H… Sohrab P. Shah[59] a…

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).

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

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.

Outline

1. Motivation: settings where sample splitting doesn’t work
2. Poisson thinning
3. **Data thinning**
4. Real data application
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?

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

Convolution-closed distributions

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

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

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

Joe, 1996, Journal of Applied Probability

Convolution-closed distributions

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

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

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

Distribution	Convolution-closed in:
$X \sim \text{Poisson}(\lambda)$	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.

Joe, 1996, Journal of Applied Probability

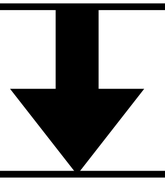
Data thinning for convolution-closed distributions

Data thinning for convolution-closed distributions

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

Data thinning for convolution-closed distributions

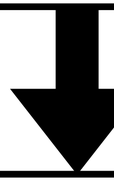
We know x could have arisen as $x' + x''$, where
$X' \sim F_{\epsilon\lambda}, \; X'' \sim F_{(1-\epsilon)\lambda}, \; X' \perp\!\!\!\perp X''.$



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

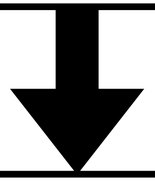


We observe realization x from $X \sim F_{\lambda}$.

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



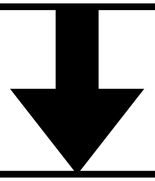
We observe realization x from $X \sim F_{\lambda}$.

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 $X \sim F_\lambda$.

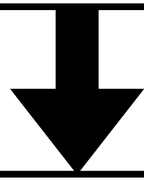
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

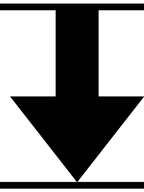
Let $G_{\epsilon,x}$ be the conditional distribution of $X' \mid X = x$.

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

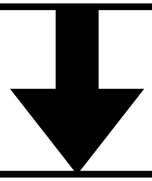
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

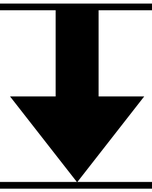
Let $G_{\epsilon,x}$ be the conditional distribution of $X' \mid X = x$.

Data thinning for convolution-closed distributions

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



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



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

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

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

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

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

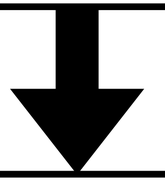
Data thinning for the Poisson distribution

Data thinning for the Poisson distribution

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

Data thinning for the Poisson distribution

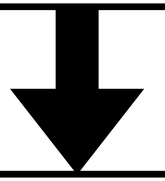
We know x could have arisen as $x' + x''$, where
$X' \sim \text{Pois}(\epsilon\lambda),\ \ X'' \sim \text{Pois}((1-\epsilon)\lambda),\ \ X' \perp\!\!\!\perp X''.$



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

Data thinning for the Poisson distribution

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

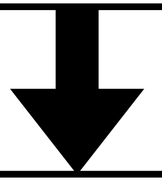


We observe realization x from $X \sim \text{Poisson}(\lambda)$.

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



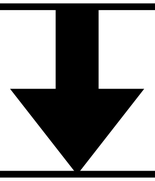
We observe realization x from $X \sim \text{Poisson}(\lambda)$.

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 $X \sim \text{Poisson}(\lambda)$.

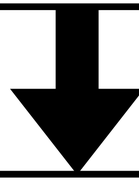
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

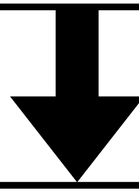
The conditional distribution of $X' \mid X = x$ is $\text{Binomial}(x, \epsilon)$.

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

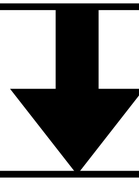
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

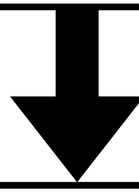
The conditional distribution of $X' \mid X = x$ is $\text{Binomial}(x, \epsilon)$.

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:

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

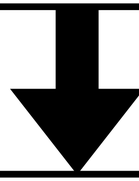
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

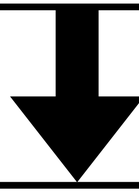
The conditional distribution of $X' \mid X = x$ is $\text{Binomial}(x, \epsilon)$.

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 Gaussian distribution

Data thinning for the Gaussian distribution

We observe realization x from $X \sim N(\mu, \sigma^2)$.

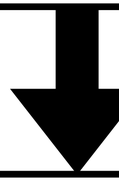
Data thinning for the Gaussian distribution

We know x could have arisen as $x' + x''$, where
$X' \sim N(\epsilon\mu, \epsilon\sigma^2), X'' \sim N((1-\epsilon)\mu, (1-\epsilon)\sigma^2), X' \perp\!\!\!\perp X''.$

We observe realization x from $X \sim N(\mu, \sigma^2)$.

Data thinning for the Gaussian distribution

We know x could have arisen as $x' + x''$, where
$X' \sim N(\epsilon\mu, \epsilon\sigma^2), X'' \sim N((1-\epsilon)\mu, (1-\epsilon)\sigma^2), X' \perp\!\!\!\perp X''.$

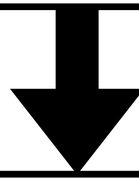


We observe realization x from $X \sim N(\mu, \sigma^2)$.

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

Data thinning for the Gaussian distribution

We know x could have arisen as $x' + x''$, where
$X' \sim N(\epsilon\mu, \epsilon\sigma^2), X'' \sim N((1-\epsilon)\mu, (1-\epsilon)\sigma^2), X' \perp\!\!\!\perp X''.$



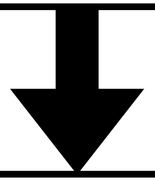
We observe realization x from $X \sim N(\mu, \sigma^2)$.

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''?

Data thinning for the Gaussian distribution

We know x could have arisen as $x' + x''$, where $X' \sim N(\epsilon\mu, \epsilon\sigma^2), X'' \sim N((1-\epsilon)\mu, (1-\epsilon)\sigma^2), X' \perp\!\!\!\perp X''$.



We observe realization x from $X \sim N(\mu, \sigma^2)$.

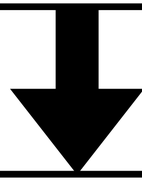
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

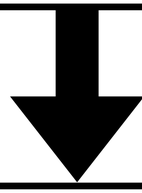
The conditional distribution of $X' \mid X = x$ is $\mathrm{N}(\epsilon x, \epsilon(1-\epsilon)\sigma^2)$.

Data thinning for the Gaussian distribution

We know x could have arisen as $x' + x''$, where $X' \sim N(\epsilon\mu, \epsilon\sigma^2), X'' \sim N((1-\epsilon)\mu, (1-\epsilon)\sigma^2), X' \perp\!\!\!\perp X''$.



We observe realization x from $X \sim N(\mu, \sigma^2)$.



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

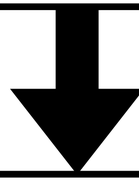
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

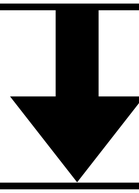
The conditional distribution of $X' \mid X = x$ is $\mathrm{N}(\epsilon x, \epsilon(1-\epsilon)\sigma^2)$.

Data thinning for the Gaussian distribution

We know x could have arisen as $x' + x''$, where $X' \sim N(\epsilon\mu, \epsilon\sigma^2), X'' \sim N((1-\epsilon)\mu, (1-\epsilon)\sigma^2), X' \perp\!\!\!\perp X''$.



We observe realization x from $X \sim N(\mu, \sigma^2)$.



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

Theorem:

$X^{(1)} \sim N(\epsilon\mu, \epsilon\sigma^2), X^{(2)} \sim N((1-\epsilon)\mu, (1-\epsilon)\sigma^2), X^{(1)} \perp\!\!\!\perp X^{(2)}$.

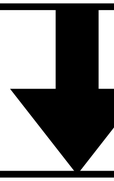
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

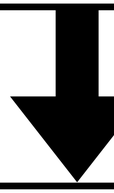
The conditional distribution of $X' \mid X = x$ is $\mathrm{N}(\epsilon x, \epsilon(1-\epsilon)\sigma^2)$.

Data thinning for the Gaussian distribution

We know x could have arisen as $x' + x''$, where
$X' \sim N(\epsilon\mu, \epsilon\sigma^2), X'' \sim N((1-\epsilon)\mu, (1-\epsilon)\sigma^2), X' \perp\!\!\!\perp X''$.



We observe realization x from $X \sim N(\mu, \sigma^2)$.



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

Theorem:
$X^{(1)} \sim N(\epsilon\mu, \epsilon\sigma^2), X^{(2)} \sim N((1-\epsilon)\mu, (1-\epsilon)\sigma^2), X^{(1)} \perp\!\!\!\perp X^{(2)}$.

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 $\mathrm{N}(\epsilon x, \epsilon(1-\epsilon)\sigma^2)$.

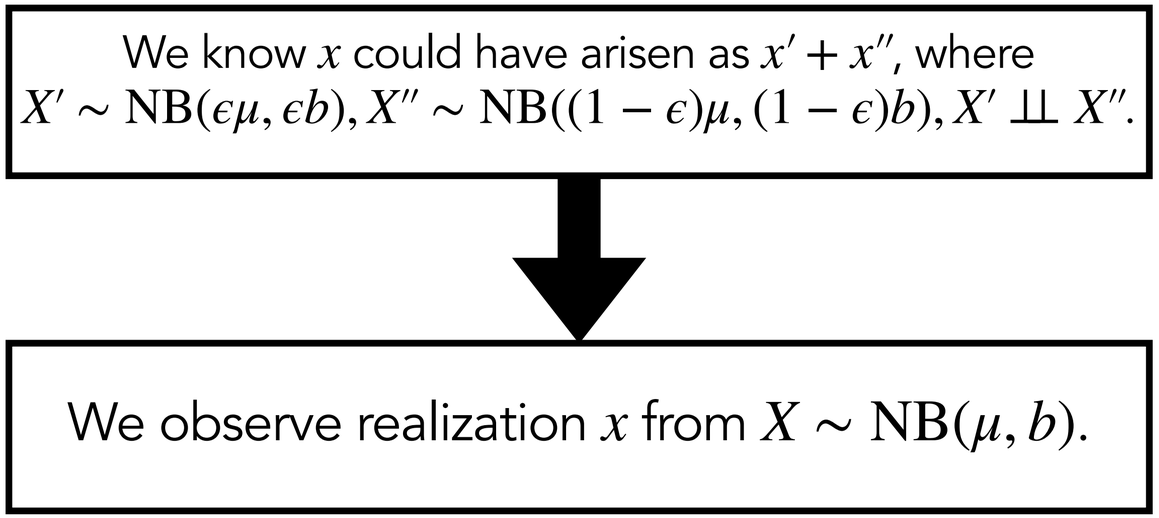
This is (similar to) a well-known result!

Data thinning recipe for the negative binomial distribution

Data thinning recipe for the negative binomial distribution

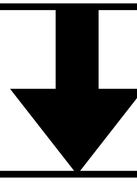
We observe realization x from $X \sim \text{NB}(\mu, b)$.

Data thinning recipe for the negative binomial distribution



Data thinning recipe for the negative binomial distribution

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

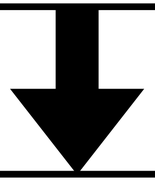


We observe realization x from $X \sim \text{NB}(\mu, b)$.

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

Data thinning recipe for the negative binomial distribution

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



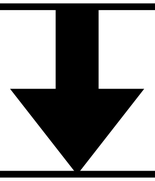
We observe realization x from $X \sim \text{NB}(\mu, b)$.

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 recipe 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 $X \sim \text{NB}(\mu, b)$.

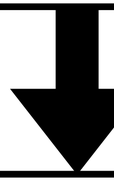
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

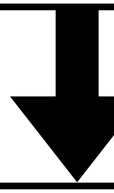
The conditional distribution of $X' \mid X = x$ is $\text{BetaBinomial}(x, \epsilon b, (1-\epsilon)b)$.

Data thinning recipe 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 $X^{(2)} := X - X^{(1)}$.

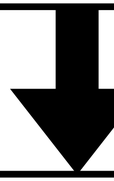
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

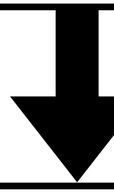
The conditional distribution of $X' \mid X = x$ is $\text{BetaBinomial}(x, \epsilon b, (1-\epsilon)b)$.

Data thinning recipe 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:

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

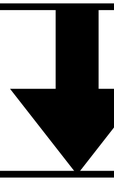
If we had observed x' and x'', we would have satisfied our goal of data thinning!

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

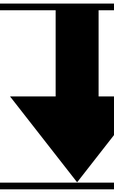
The conditional distribution of $X' \mid X = x$ is $\text{BetaBinomial}(x, \epsilon b, (1-\epsilon)b)$.

Data thinning recipe 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!

We can continue deriving recipes for more distributions

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

We can continue deriving recipes for more distributions

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., 2021, arXiv:2112.11079.
- Neufeld et al., 2022, Biostatistics.
- Oliveira, Lei, and Tibshirani, 2022, arXiv:2212.01943.

We can continue deriving recipes for more distributions

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

We can continue deriving recipes for more distributions

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., 2022, arXiv:2112.11079.
- Oliveira, Lei, and Tibshirani, 2022, arXiv:2111.09447.

We can continue deriving recipes for more distributions

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 do not know the value of a needed parameter?

Gaussian thinning algorithm

Suppose $X \sim \mathrm{N}\left(\mu, \sigma^2\right)$.

Draw
$X^{(1)} \sim \mathrm{N}(\epsilon x, \epsilon(1-\epsilon)\,\sigma^2)$ and
$X^{(2)} = X - X^{(1)}$.
Then:

1) $X^{(1)} \sim \mathrm{N}\left(\epsilon\mu, \epsilon\sigma^2\right)$
2) $X^{(2)} \sim \mathrm{N}\left((1-\epsilon)\mu, (1-\epsilon)\sigma^2\right)$
3) $X^{(1)} \perp\!\!\!\perp X^{(2)}$.

What if we do not know the value of a needed parameter?

Gaussian thinning algorithm

Suppose $X \sim \mathrm{N}\left(\mu, \sigma^2\right)$.

Draw

$X^{(1)} \sim \mathrm{N}(\epsilon x, \epsilon(1-\epsilon)\,\tilde{\sigma}^2)$ and

$X^{(2)} = X - X^{(1)}$.

Then:

1) $X^{(1)} \sim \mathrm{N}\left(\epsilon\mu, \epsilon\sigma^2\right)$
2) $X^{(2)} \sim \mathrm{N}\left((1-\epsilon)\mu, (1-\epsilon)\sigma^2\right)$
3) $X^{(1)} \perp\!\!\!\perp X^{(2)}$.

What if we do not know the value of a needed parameter?

Gaussian thinning algorithm

Suppose $X \sim \mathrm{N}\left(\mu, \sigma^2\right)$.

Draw

$X^{(1)} \sim \mathrm{N}(\epsilon x, \epsilon(1-\epsilon)\ \tilde{\sigma}^2)$ and

$X^{(2)} = X - X^{(1)}$.

Then:

~~1) $X^{(1)} \sim \mathrm{N}\left(\epsilon\mu, \epsilon\sigma^2\right)$~~

~~2) $X^{(2)} \sim \mathrm{N}\left((1-\epsilon)\mu, (1-\epsilon)\sigma^2\right)$~~

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

What if we do not know the value of a needed parameter?

Gaussian thinning algorithm

Suppose $X \sim \mathrm{N}\left(\mu, \sigma^2\right)$.

Draw

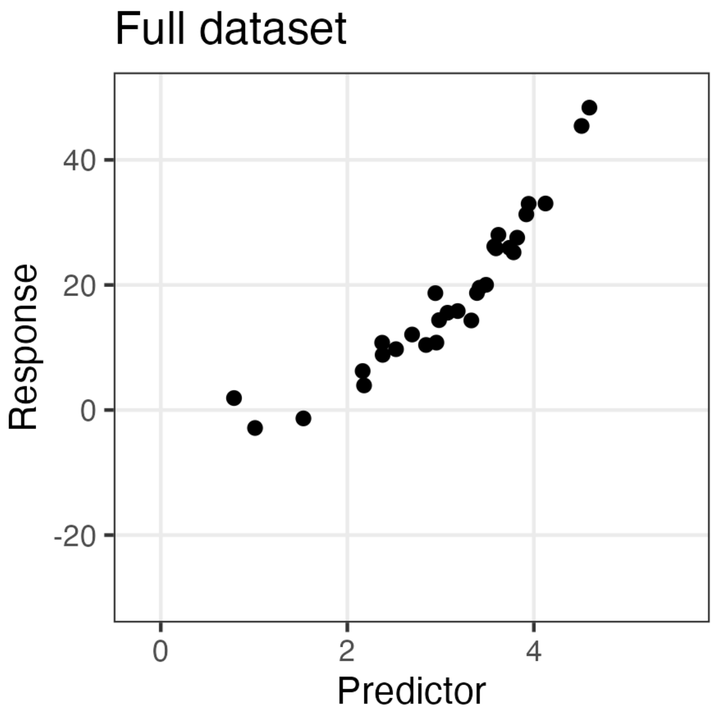
$X^{(1)} \sim \mathrm{N}(\epsilon x, \epsilon(1-\epsilon)\ \tilde{\sigma}^2)$ and

$X^{(2)} = X - X^{(1)}$.

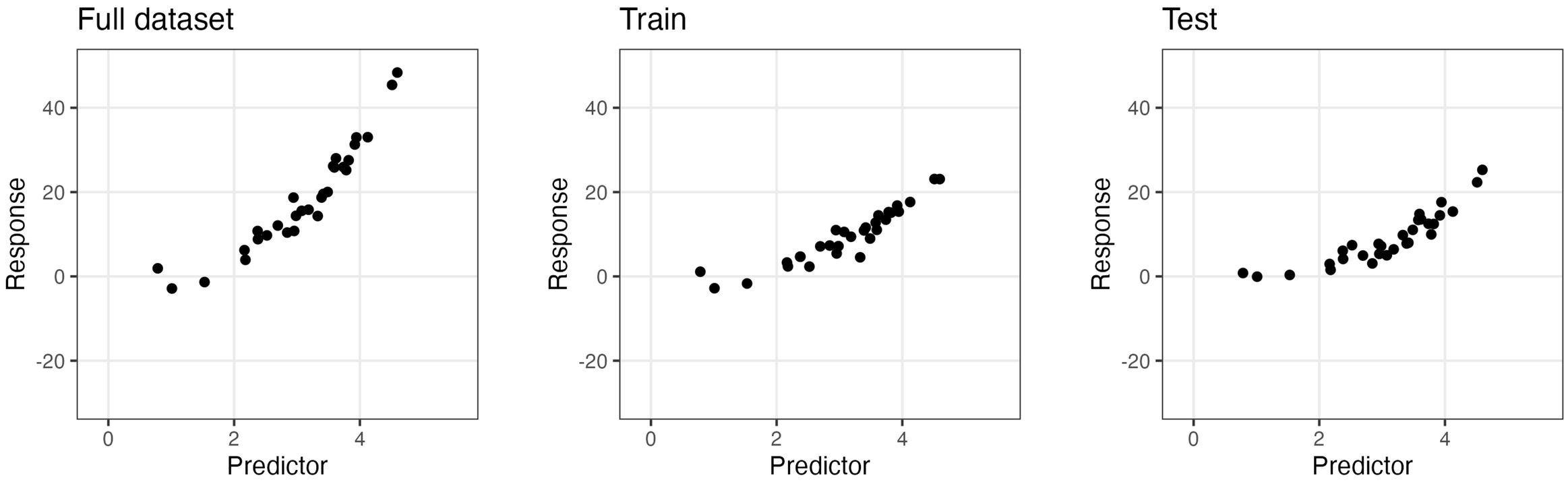
Then:

1) $X^{(1)} \sim \mathrm{N}\left(\epsilon\mu, \epsilon^2\sigma^2 + \epsilon(1-\epsilon)\tilde{\sigma}^2\right)$
2) $X^{(2)} \sim \mathrm{N}\left((1-\epsilon)\mu, (1-\epsilon)^2\sigma^2 + \epsilon(1-\epsilon)\tilde{\sigma}^2\right)$
3) $\mathrm{Cov}\left(X^{(1)}, X^{(2)}\right) = \epsilon(1-\epsilon)\left(\sigma^2 - \tilde{\sigma}^2\right)$.

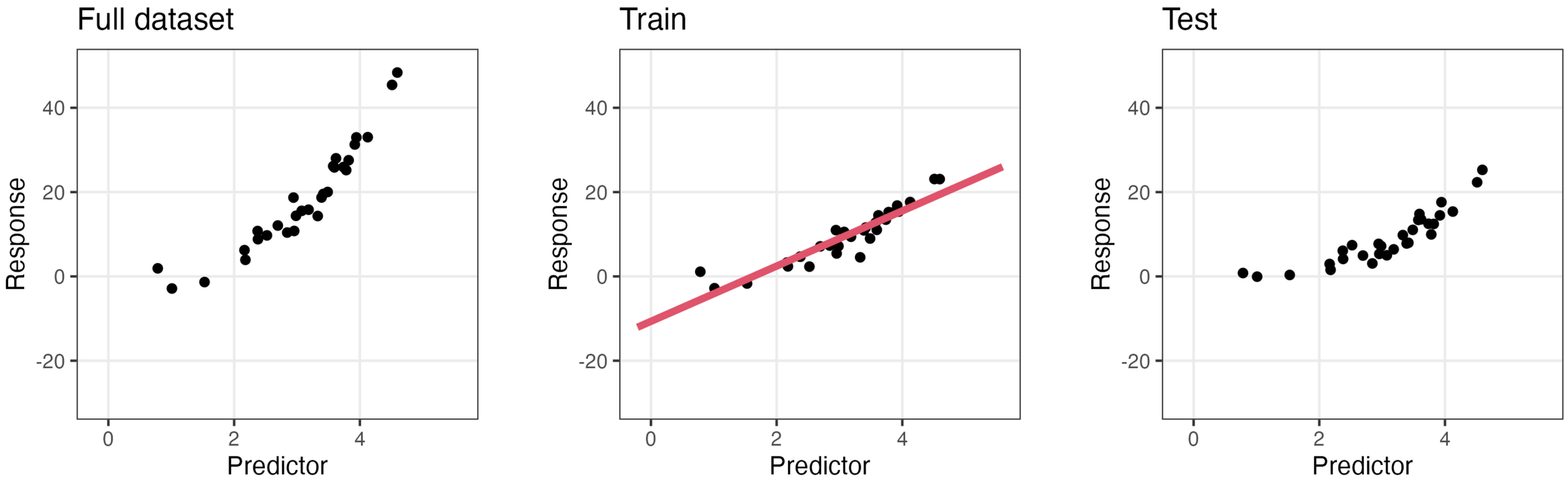
Data thinning can be applied in any setting where sample splitting can be applied



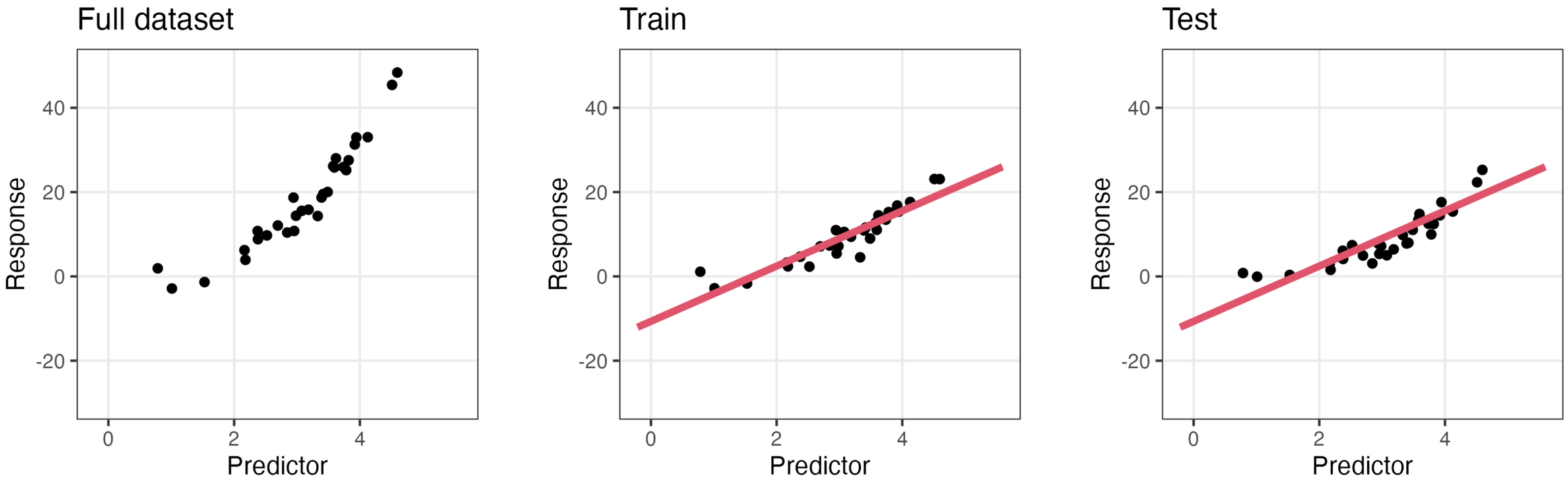
Data thinning can be applied in any setting where sample splitting can be applied



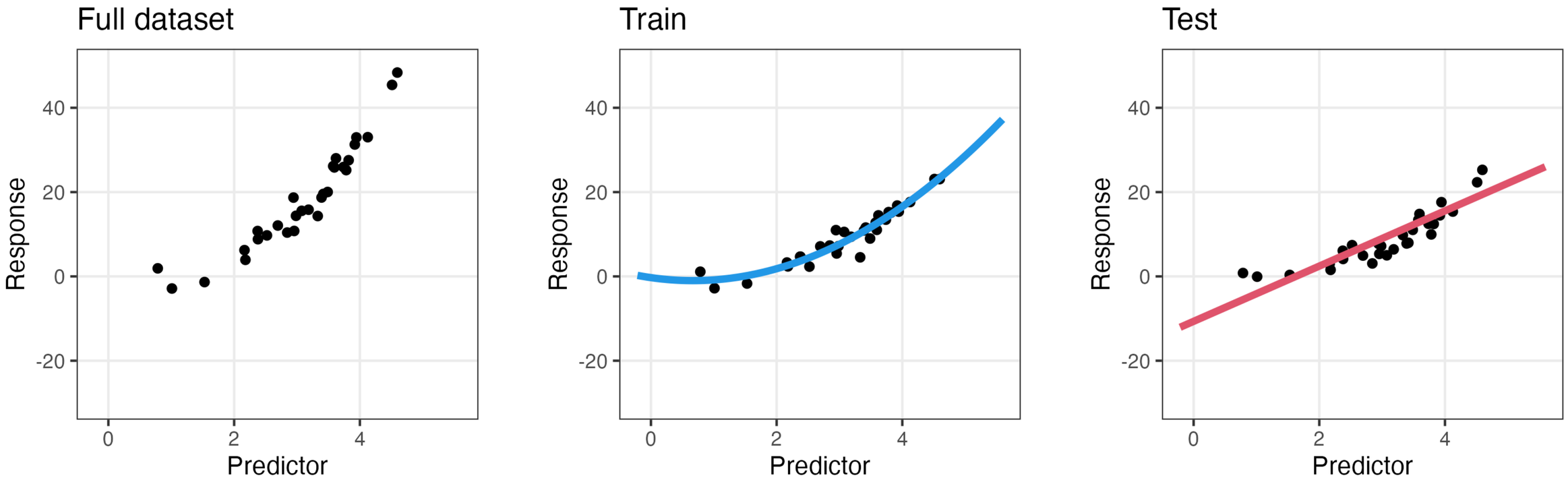
Data thinning can be applied in any setting where sample splitting can be applied



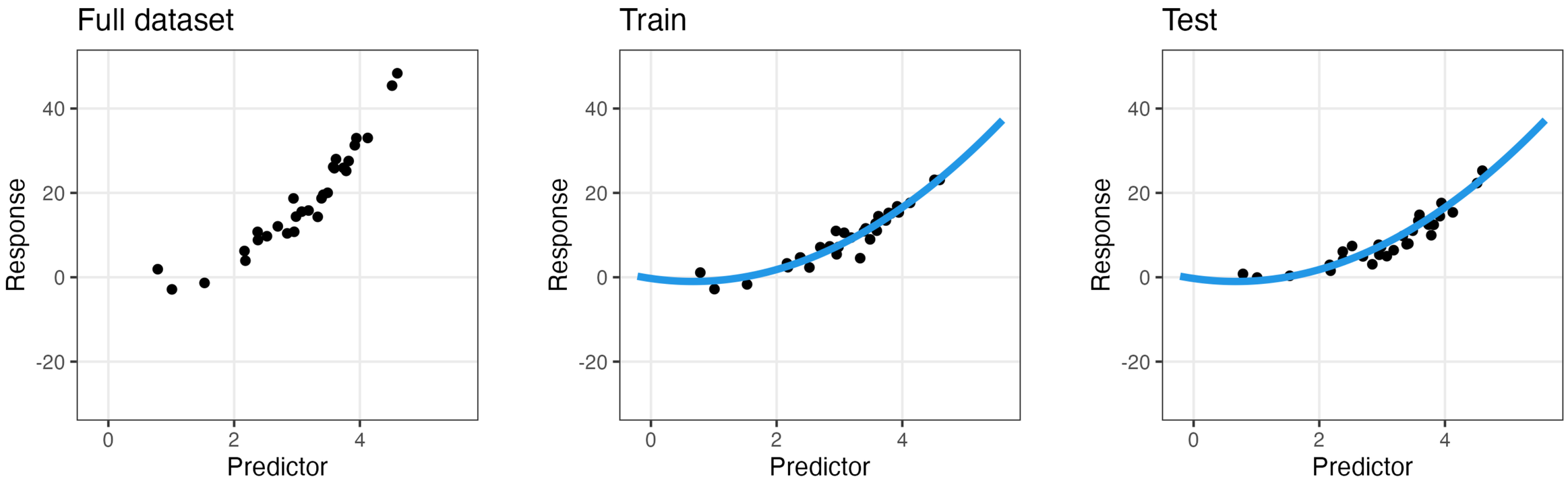
Data thinning can be applied in any setting where sample splitting can be applied



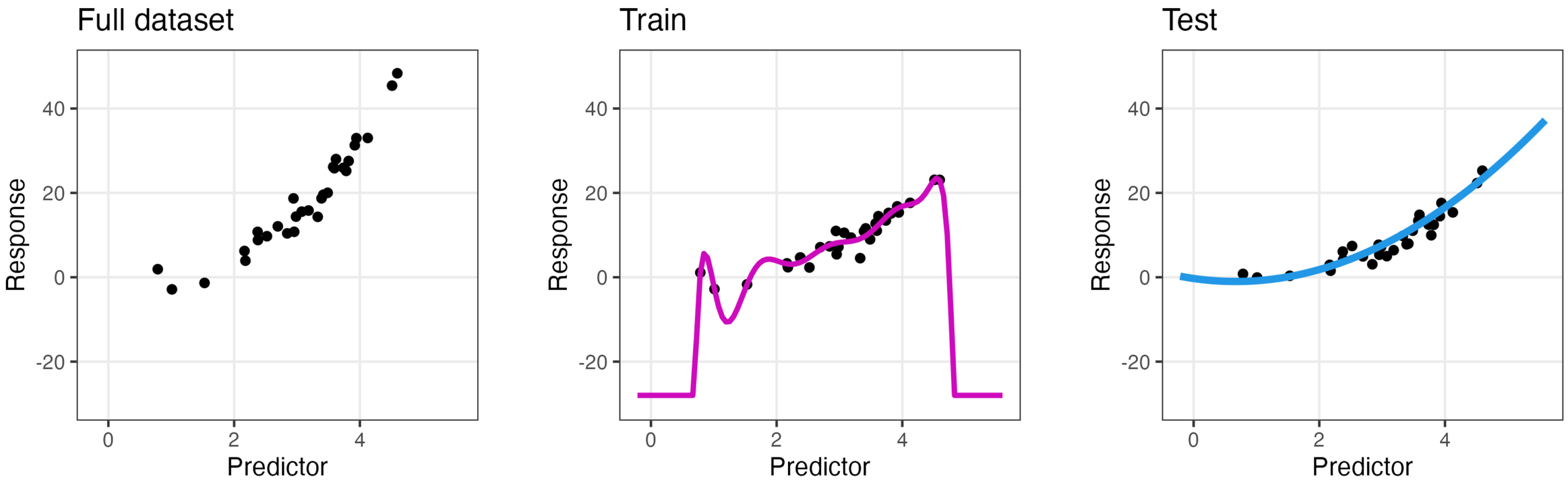
Data thinning can be applied in any setting where sample splitting can be applied



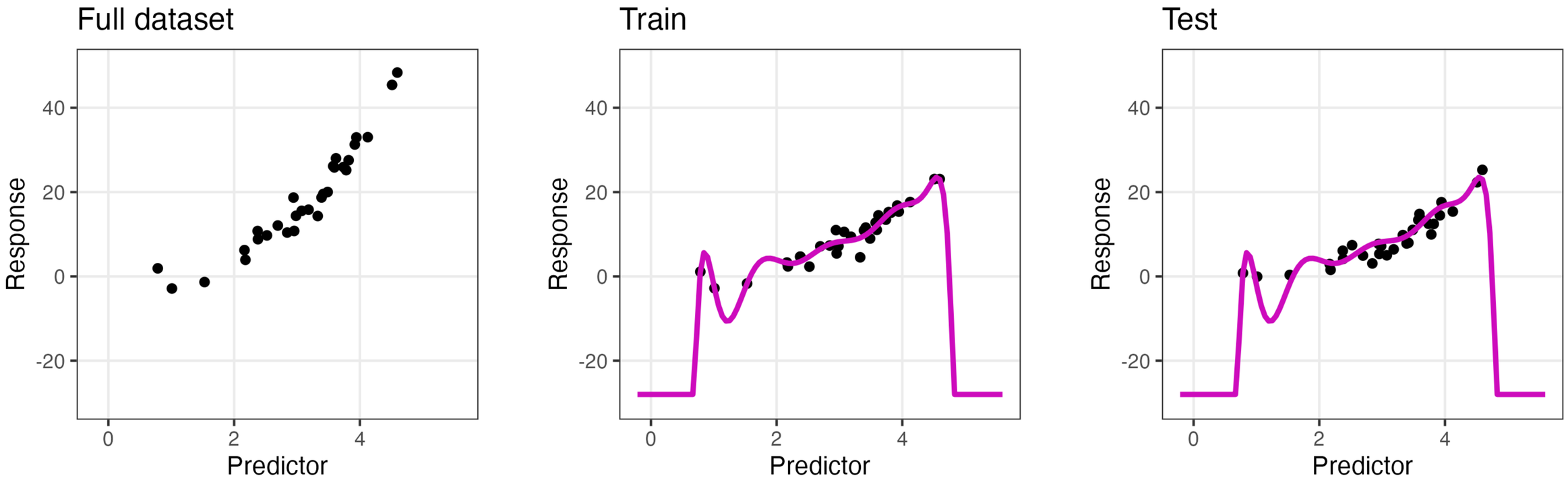
Data thinning can be applied in any setting where sample splitting can be applied



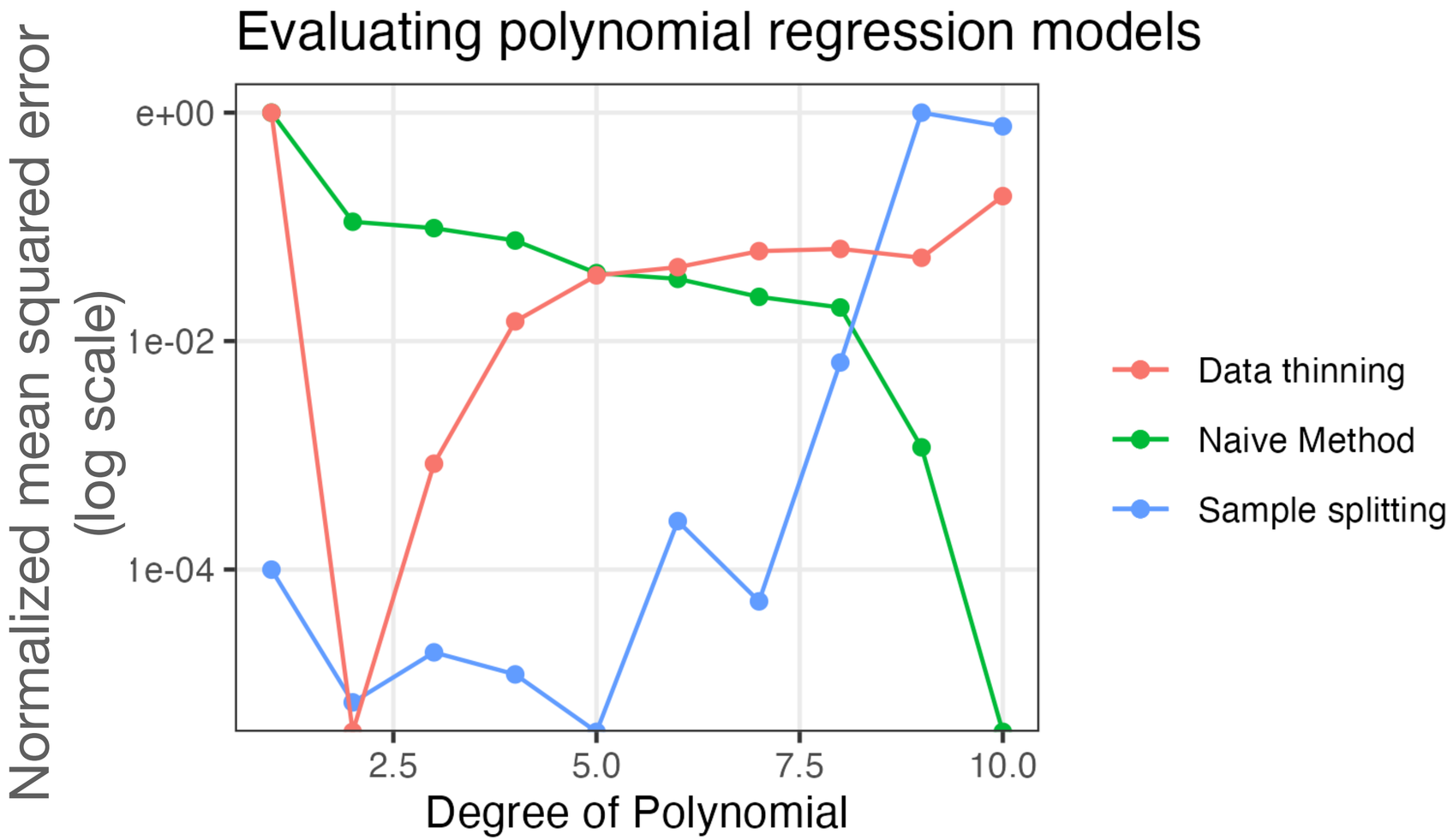
Data thinning can be applied in any setting where sample splitting can be applied



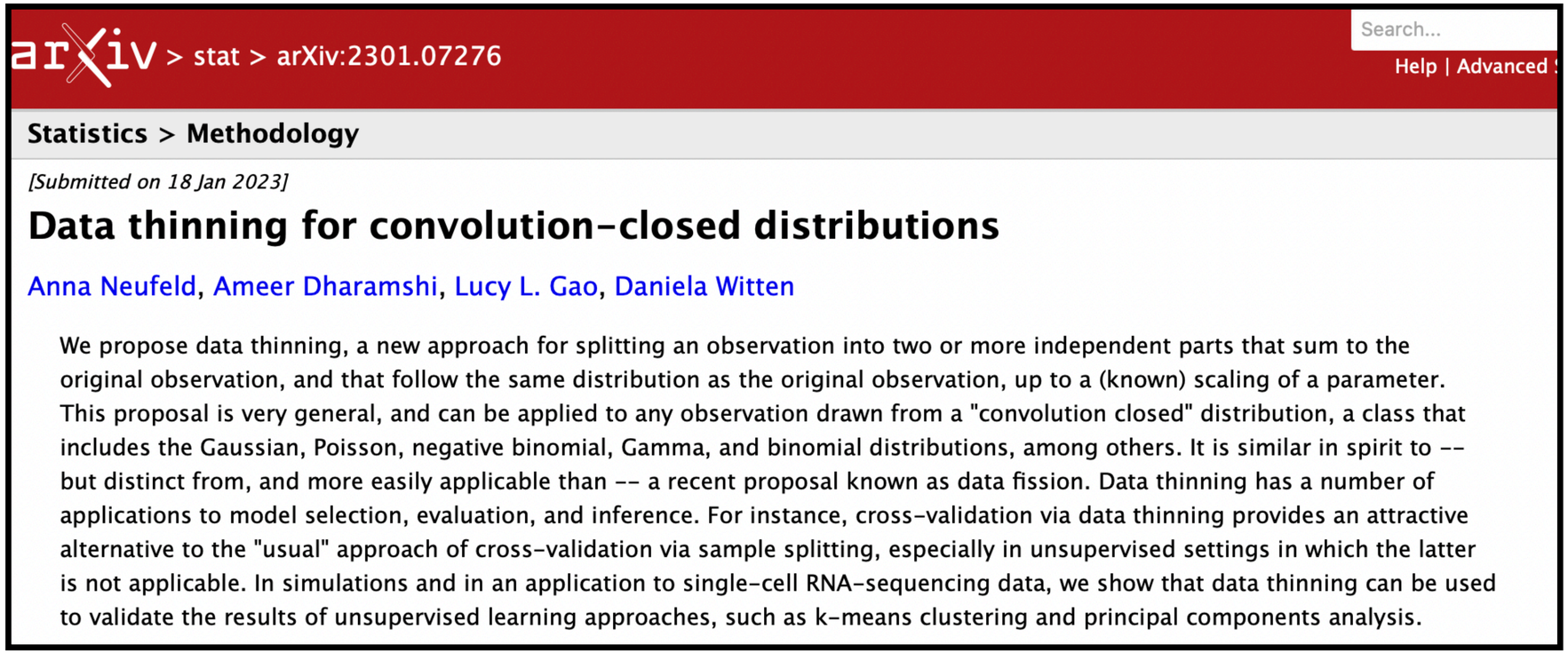
Data thinning can be applied in any setting where sample splitting can be applied



Data thinning can be applied in any setting where sample splitting can be applied



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. **Real data application**
5. Ongoing work

How can we validate the results of a clustering?

RESEARCH ARTICLE

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

HUMAN GENOMICS

A human cell atlas of fetal gene expression

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

How can we validate the results of a clustering?

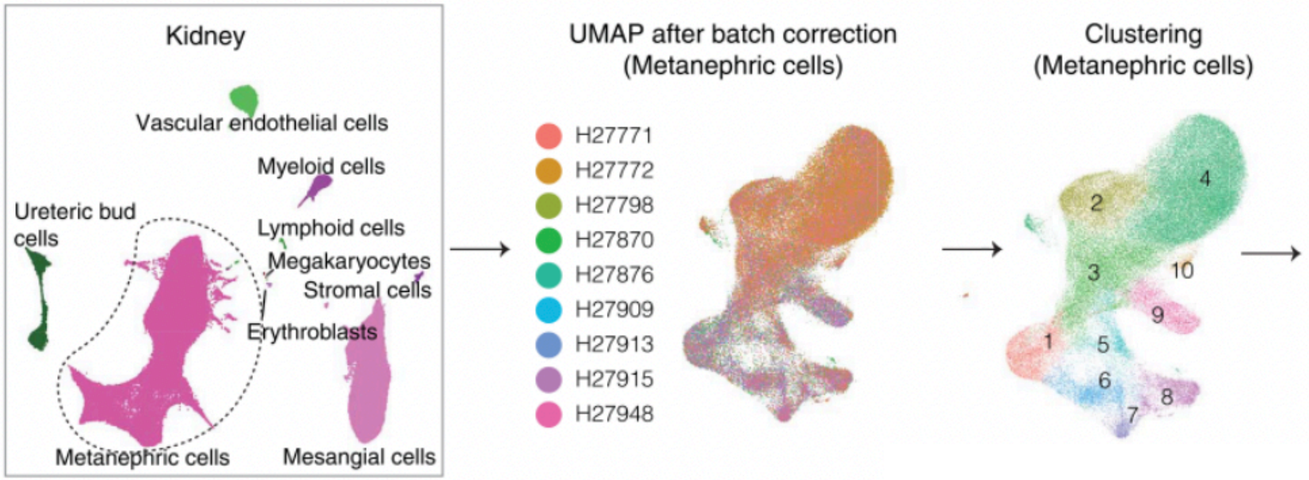
RESEARCH ARTICLE

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

HUMAN GENOMICS

A human cell atlas of fetal gene expression

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



How can we validate the results of a clustering?

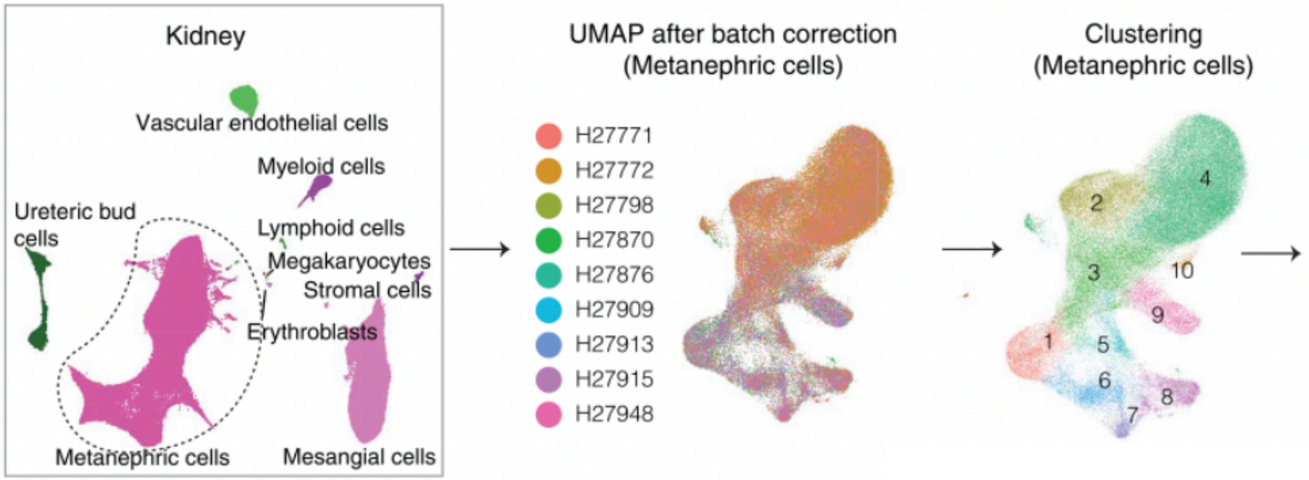
RESEARCH ARTICLE

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

HUMAN GENOMICS

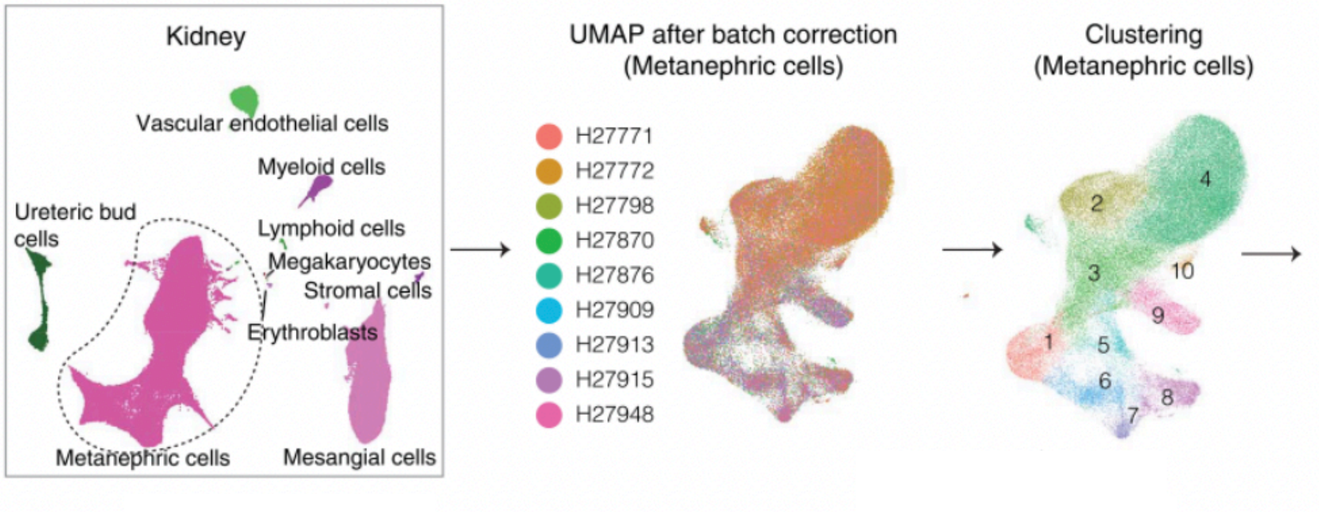
A human cell atlas of fetal gene expression

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

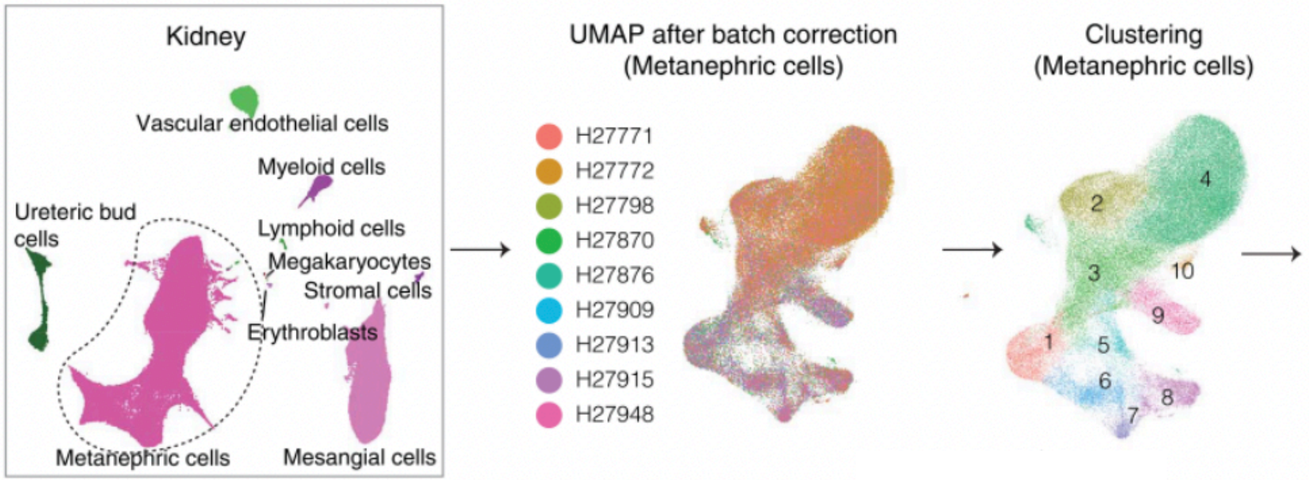


The authors ask: "are these clusters reproducible"?

Can the cluster labels be reliably reproduced?



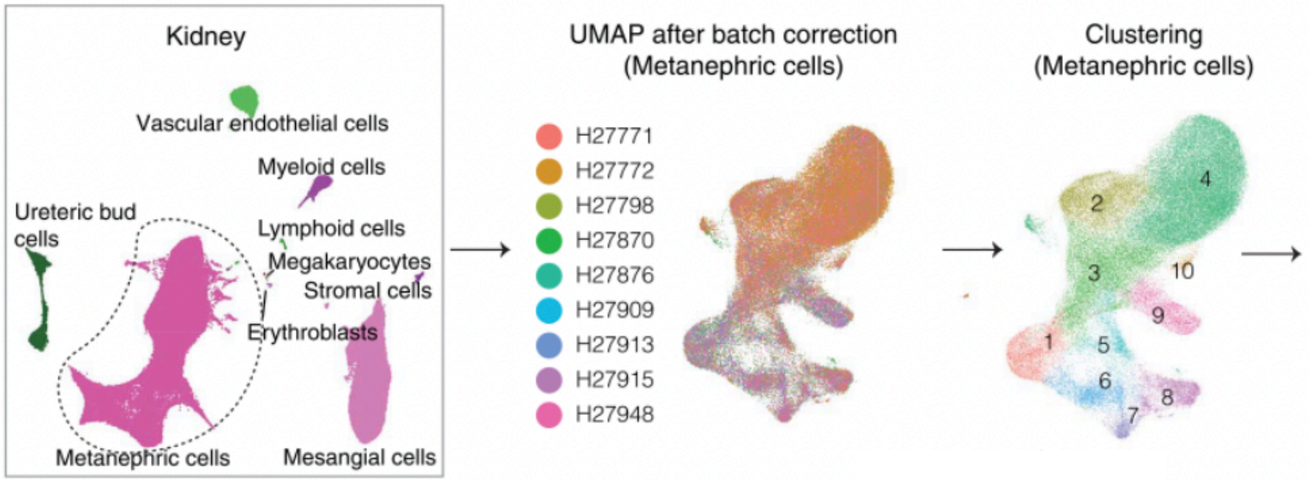
Can the cluster labels be reliably reproduced?



Intradataset cross validation (Cao et al.)

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

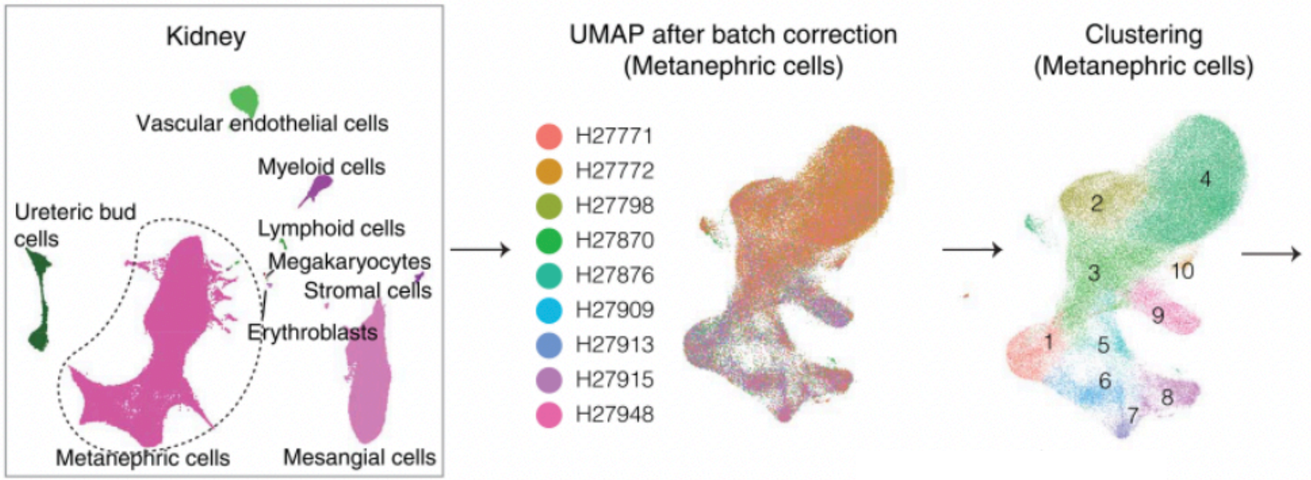
Can the cluster labels be reliably reproduced?



Intradataset cross validation (Cao et al.)

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

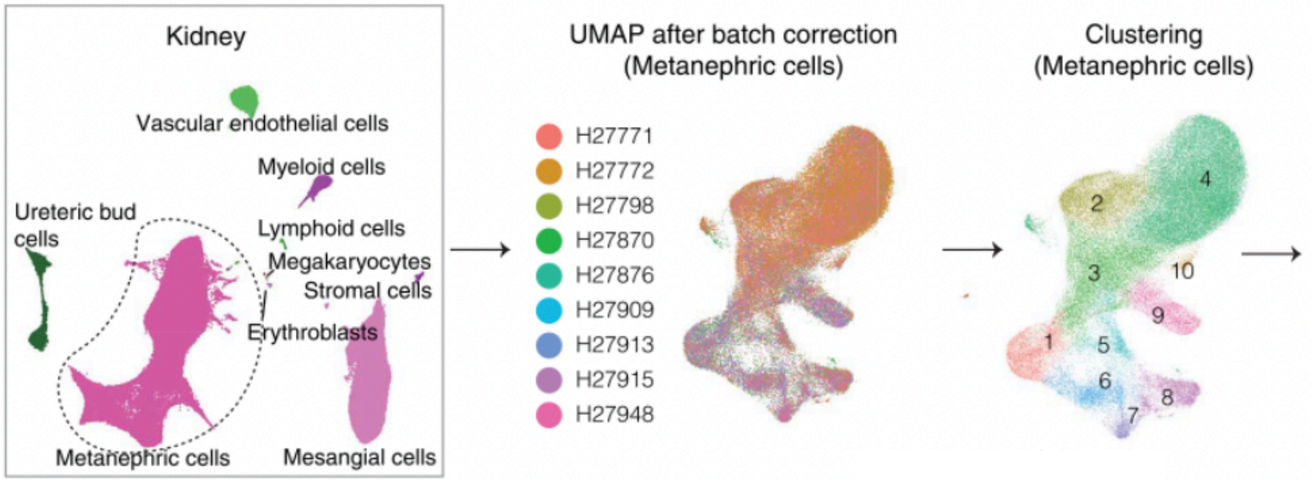
Can the cluster labels be reliably reproduced?



Intradataset cross validation (Cao et al.)

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

Can the cluster labels be reliably reproduced?

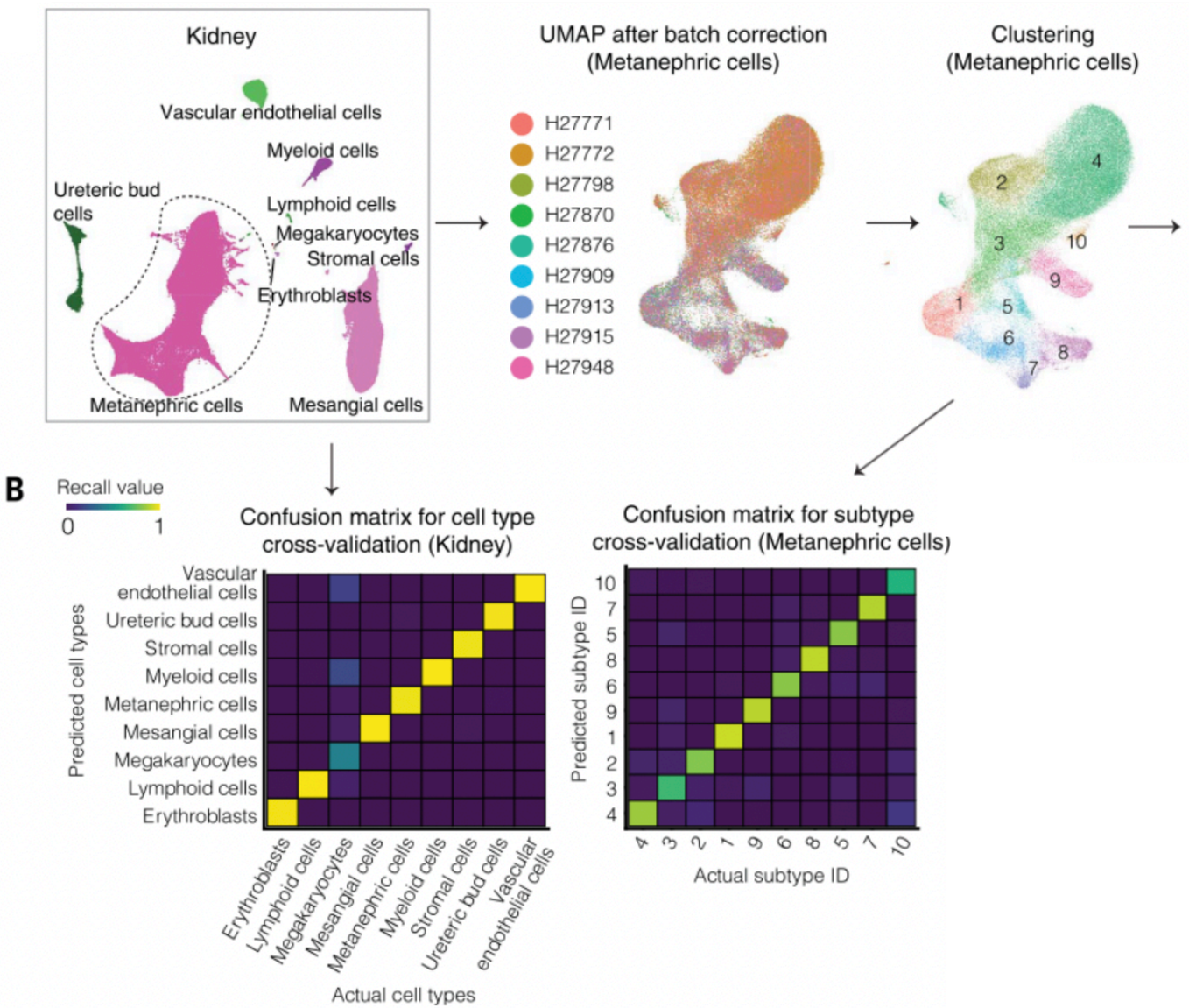


Intradataset cross validation (Cao et al.)

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

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

Can the cluster labels be reliably reproduced?

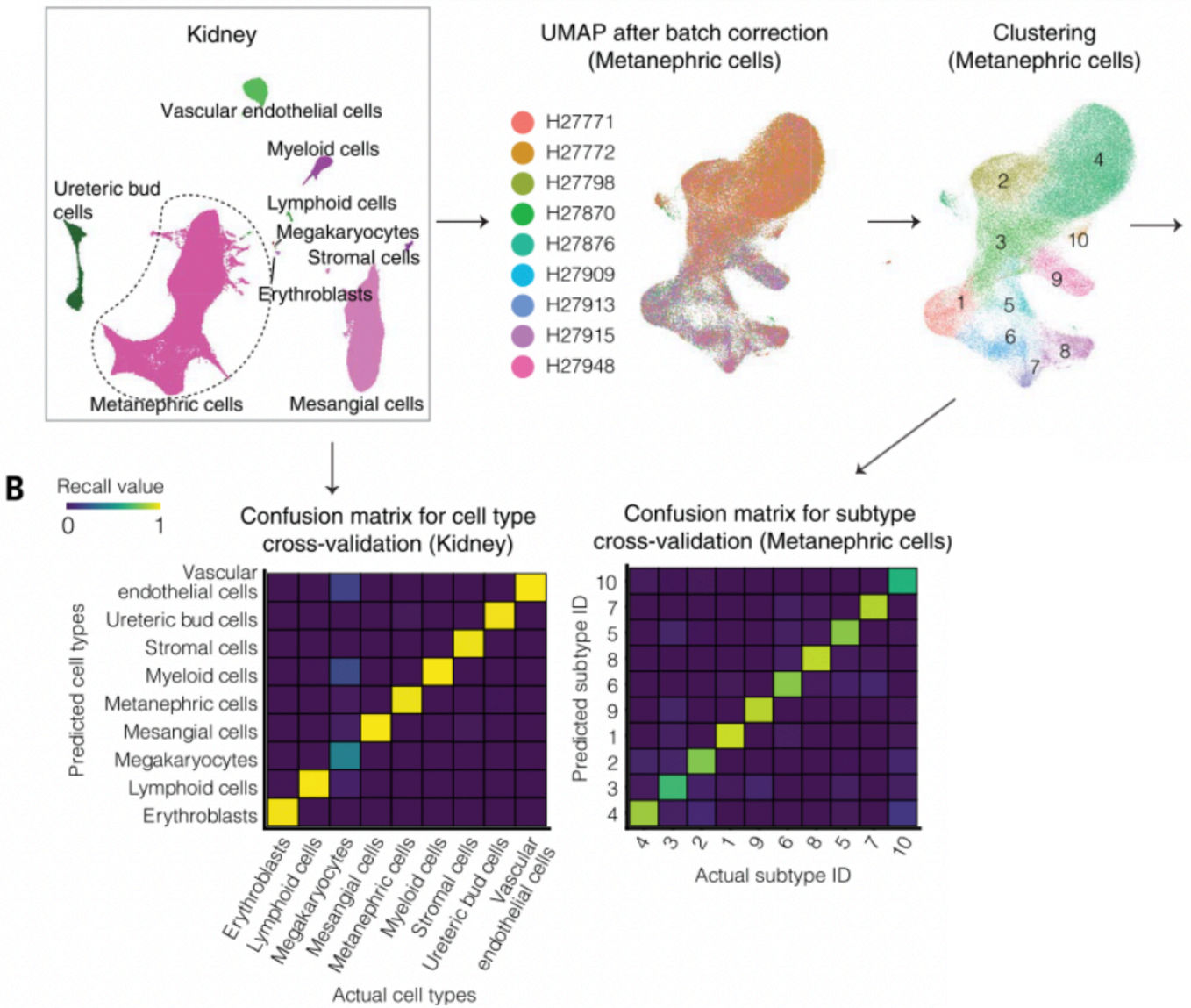


Intradataset cross validation (Cao et al.)

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

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

Can the cluster labels be reliably reproduced?



Intradataset cross validation (Cao et al.)

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

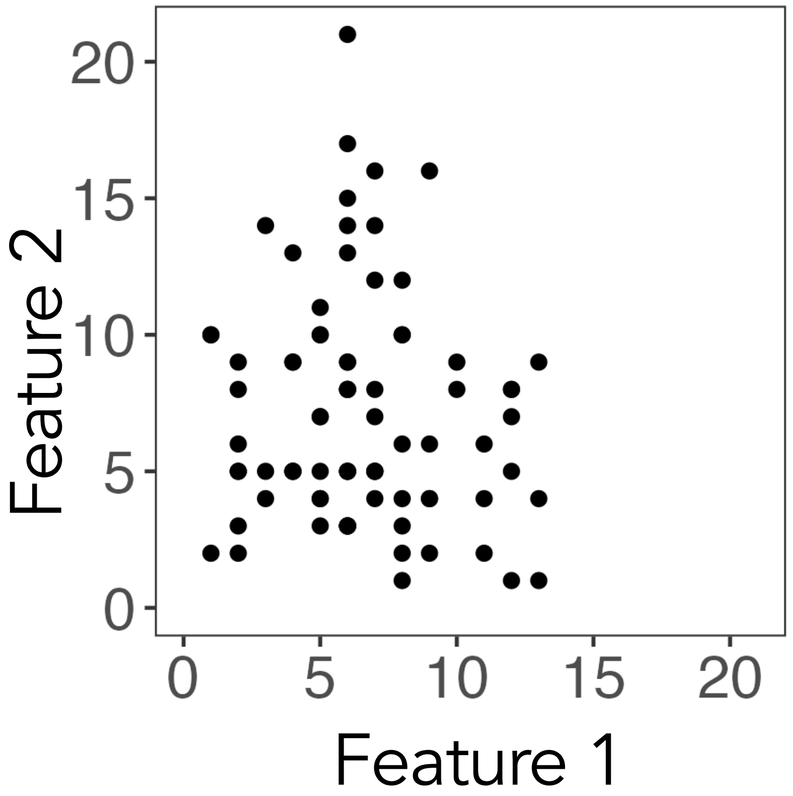
  But we already dipped in the data here!

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

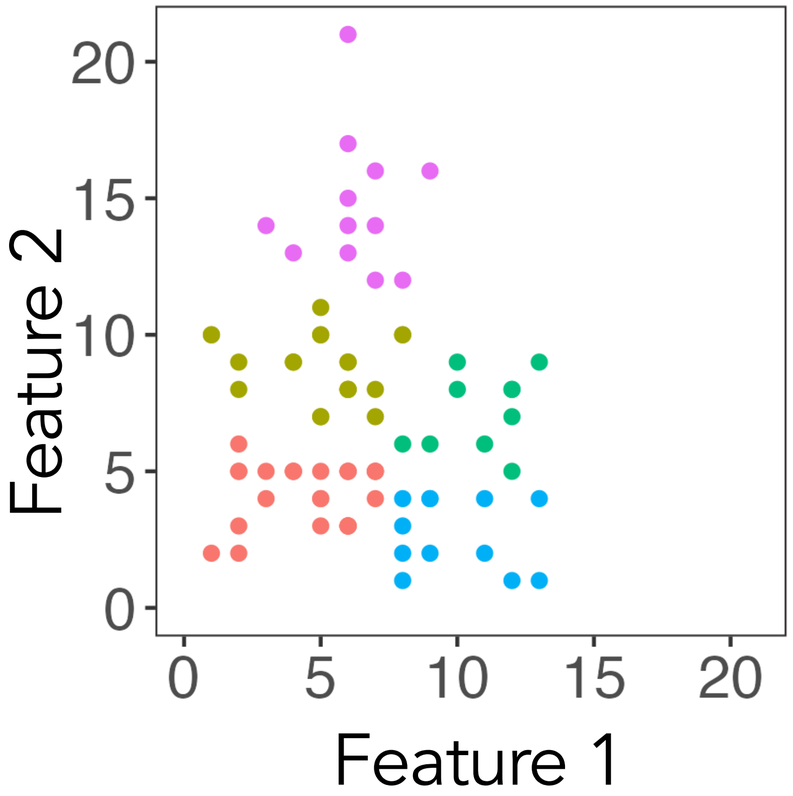
  Use cross validation to avoid double dipping between fitting and evaluating the classifier.

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

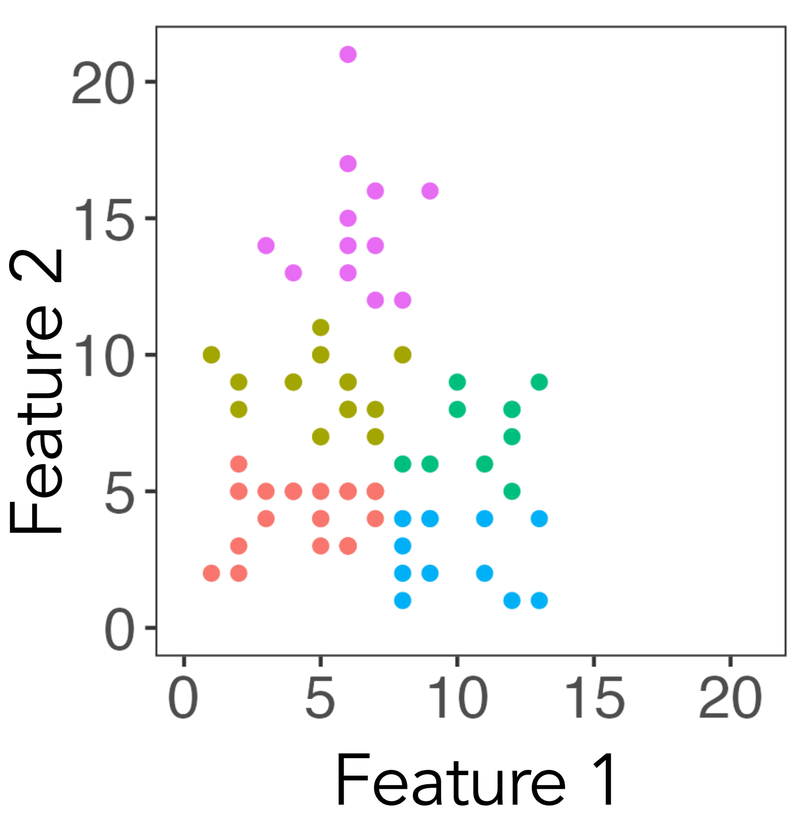
This cross validation procedure double dips

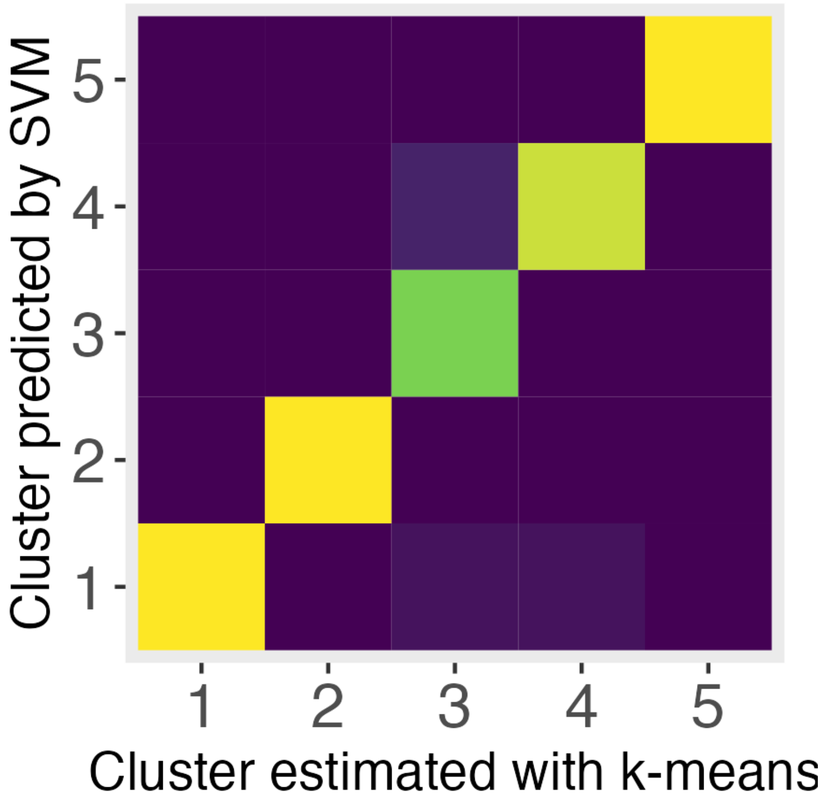


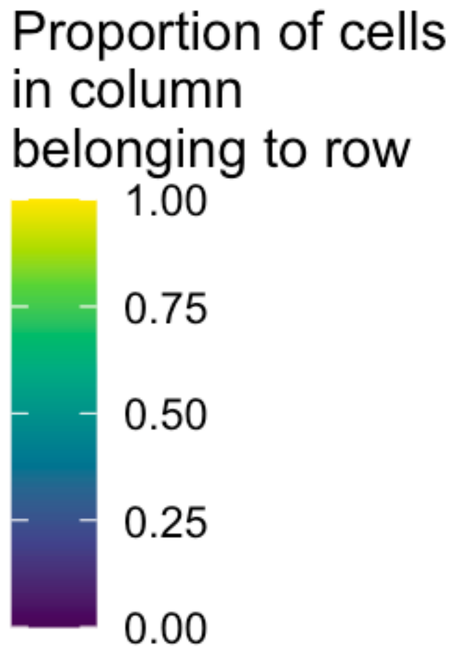
This cross validation procedure double dips



This cross validation procedure double dips

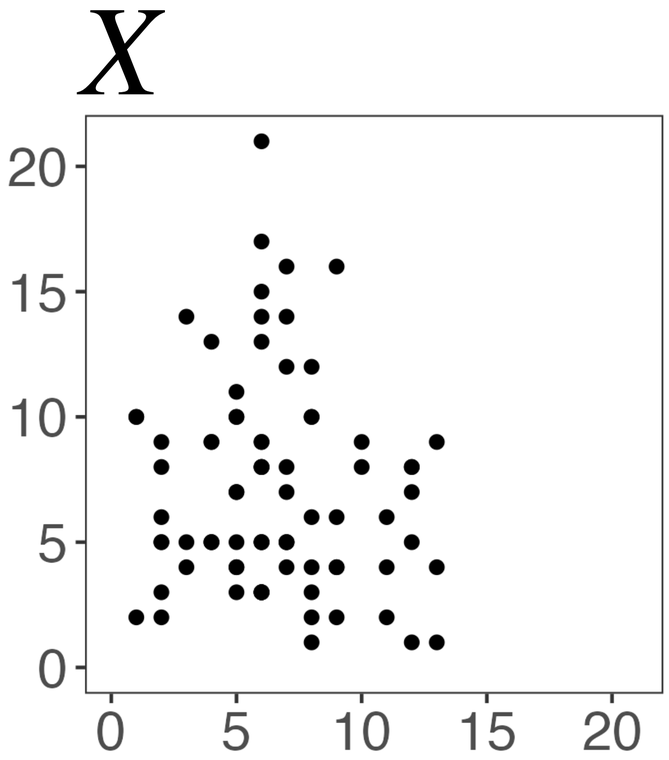




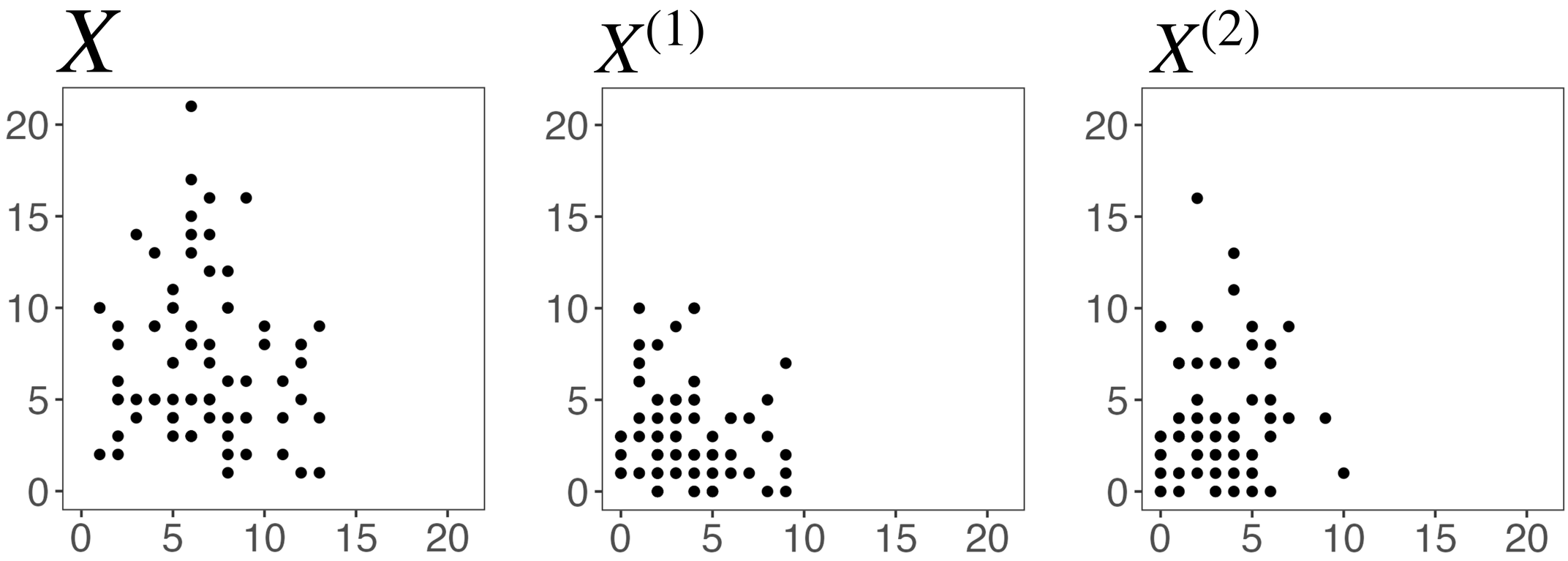


Classifier gets 96% accuracy to predict the five clusters, despite the fact that the five clusters are just random noise.

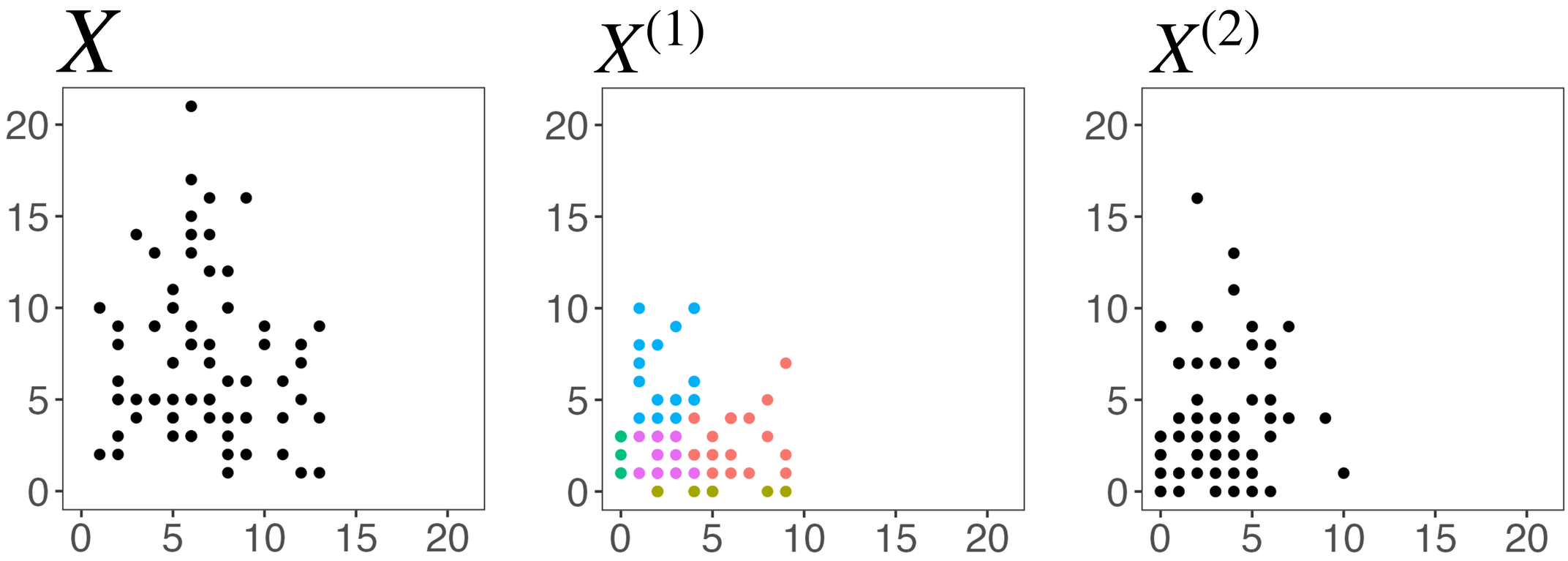
Data thinning provides a simple alternative



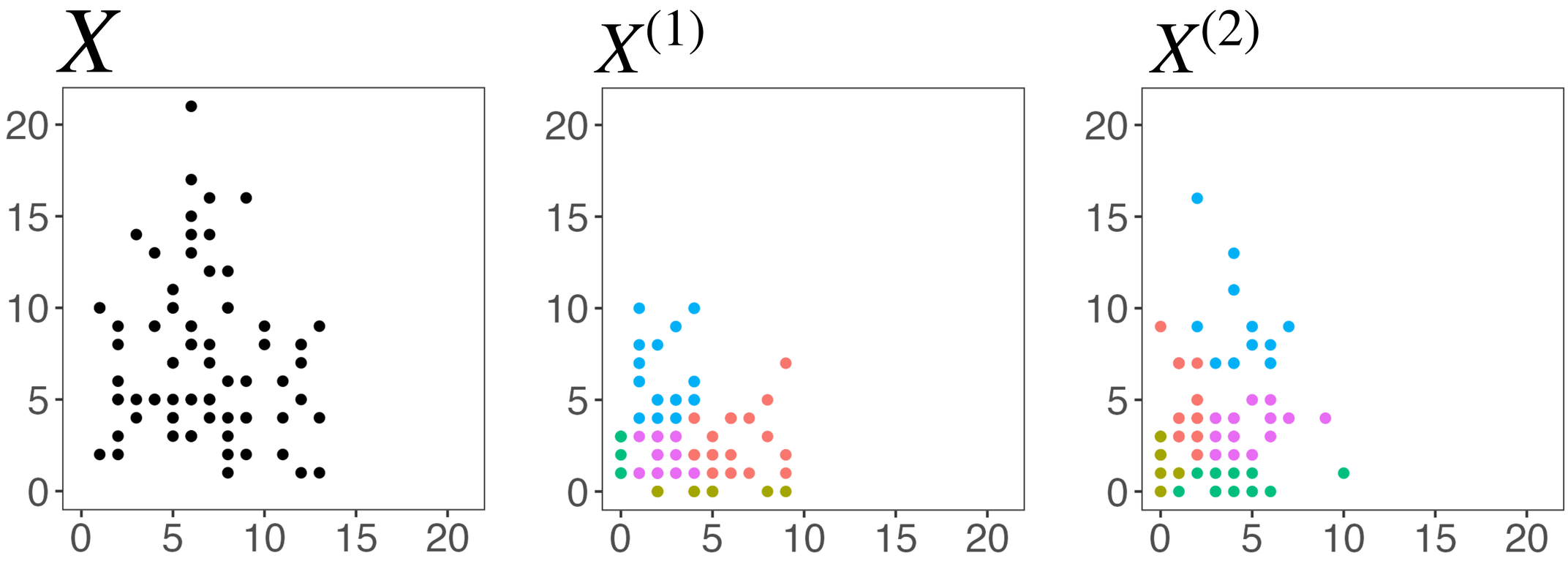
Data thinning provides a simple alternative



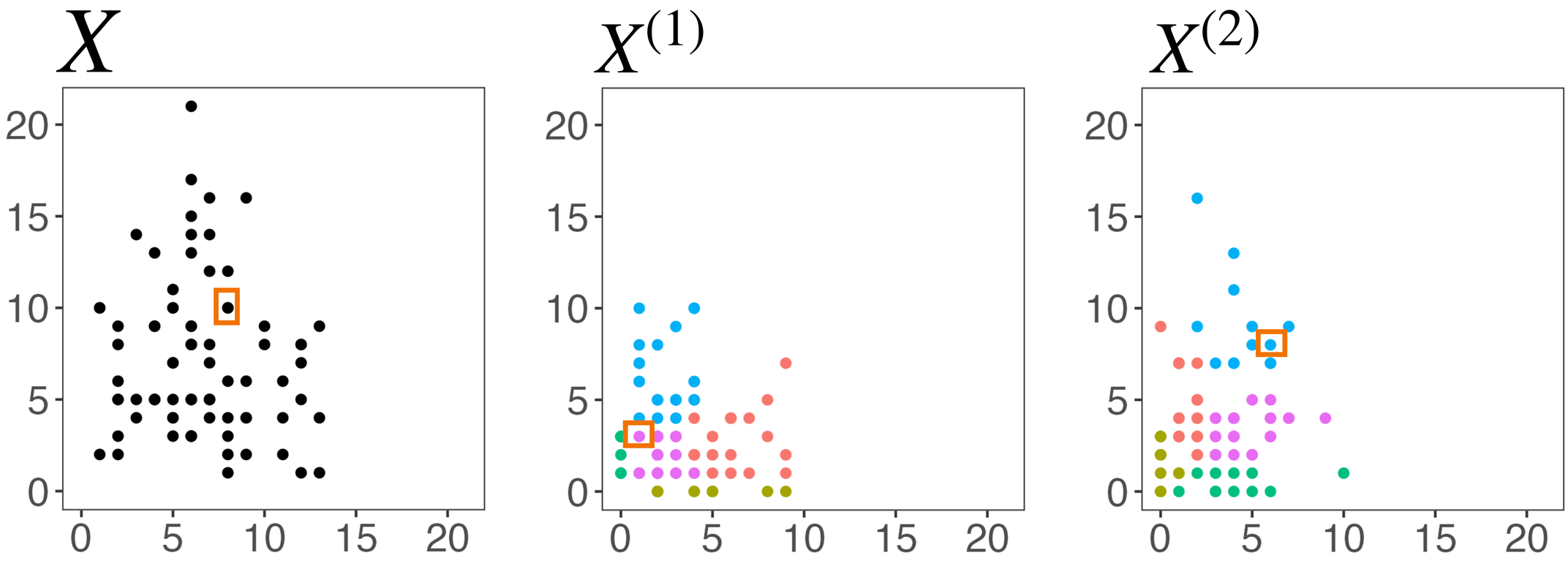
Data thinning provides a simple alternative



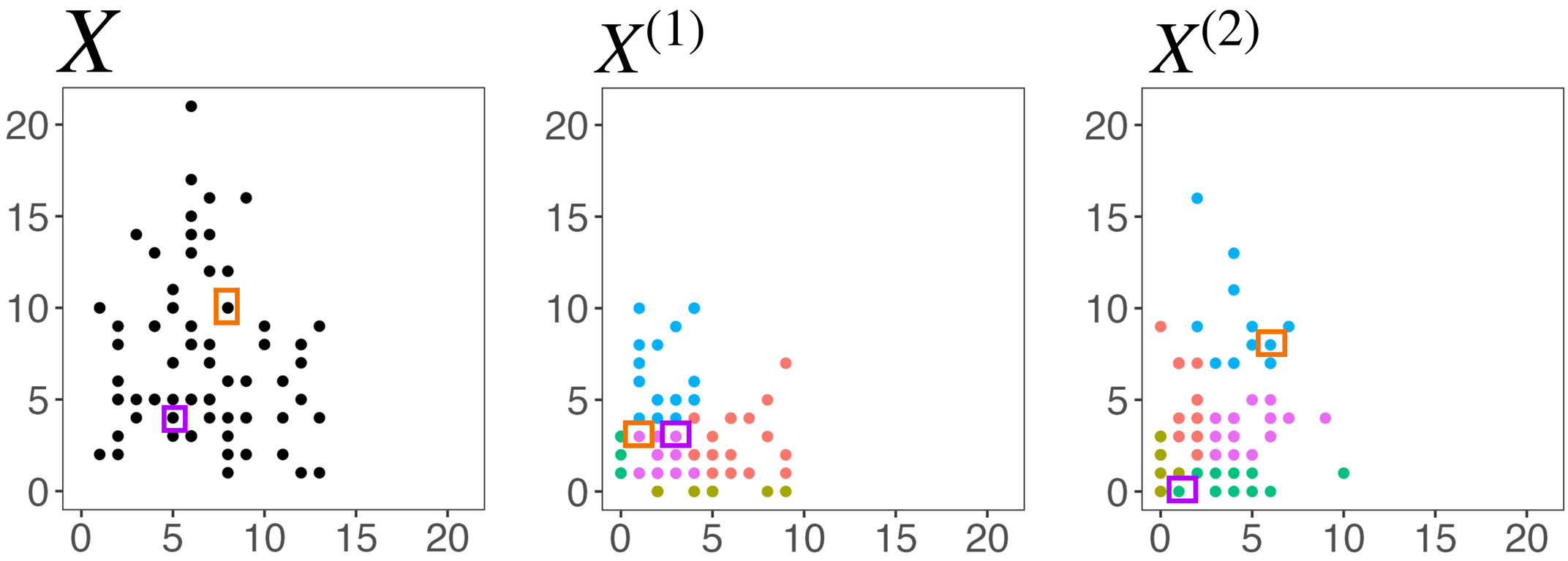
Data thinning provides a simple alternative



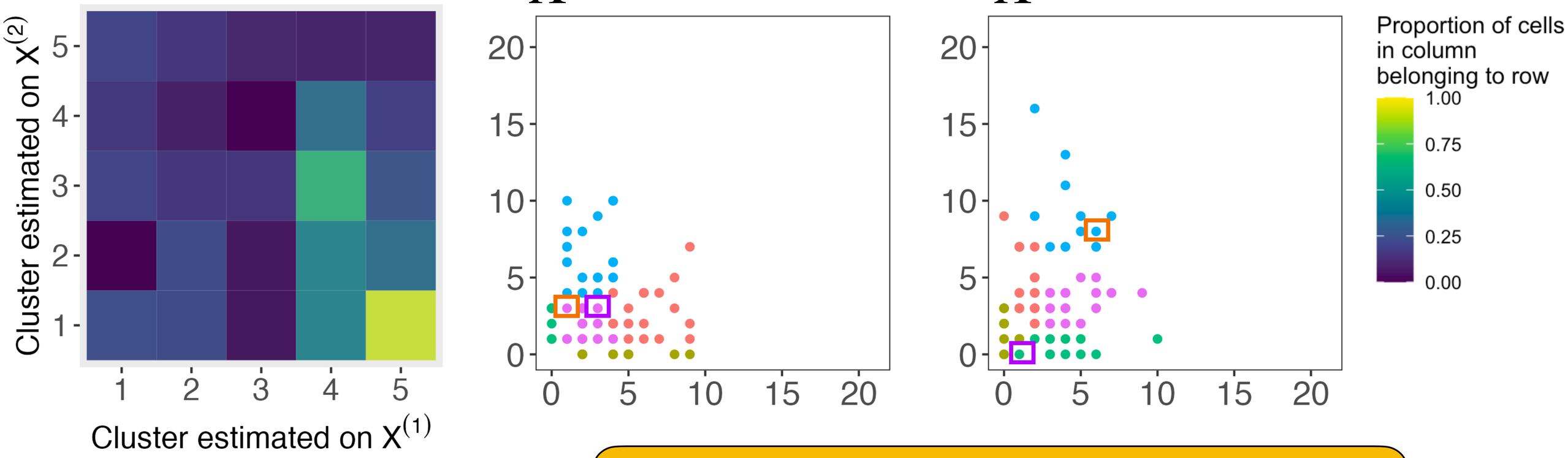
Data thinning provides a simple alternative



Data thinning provides a simple alternative

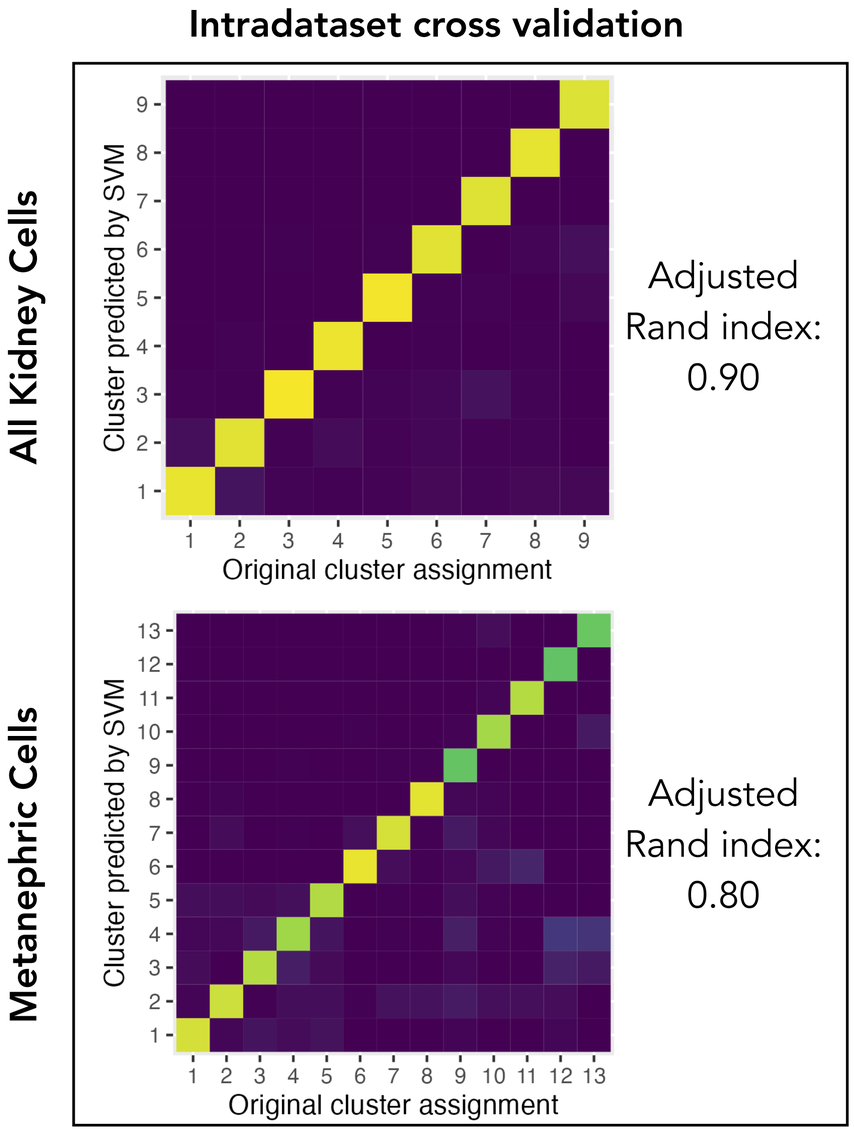


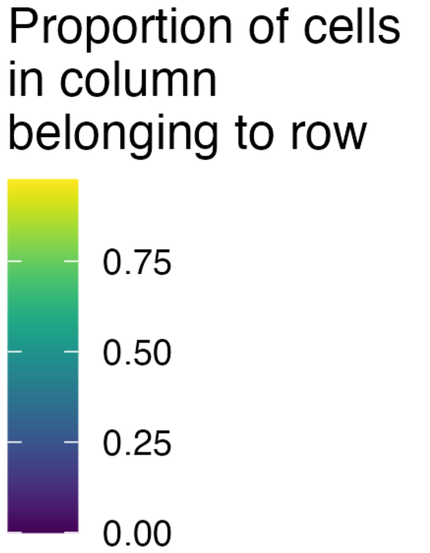
Data thinning provides a simple alternative



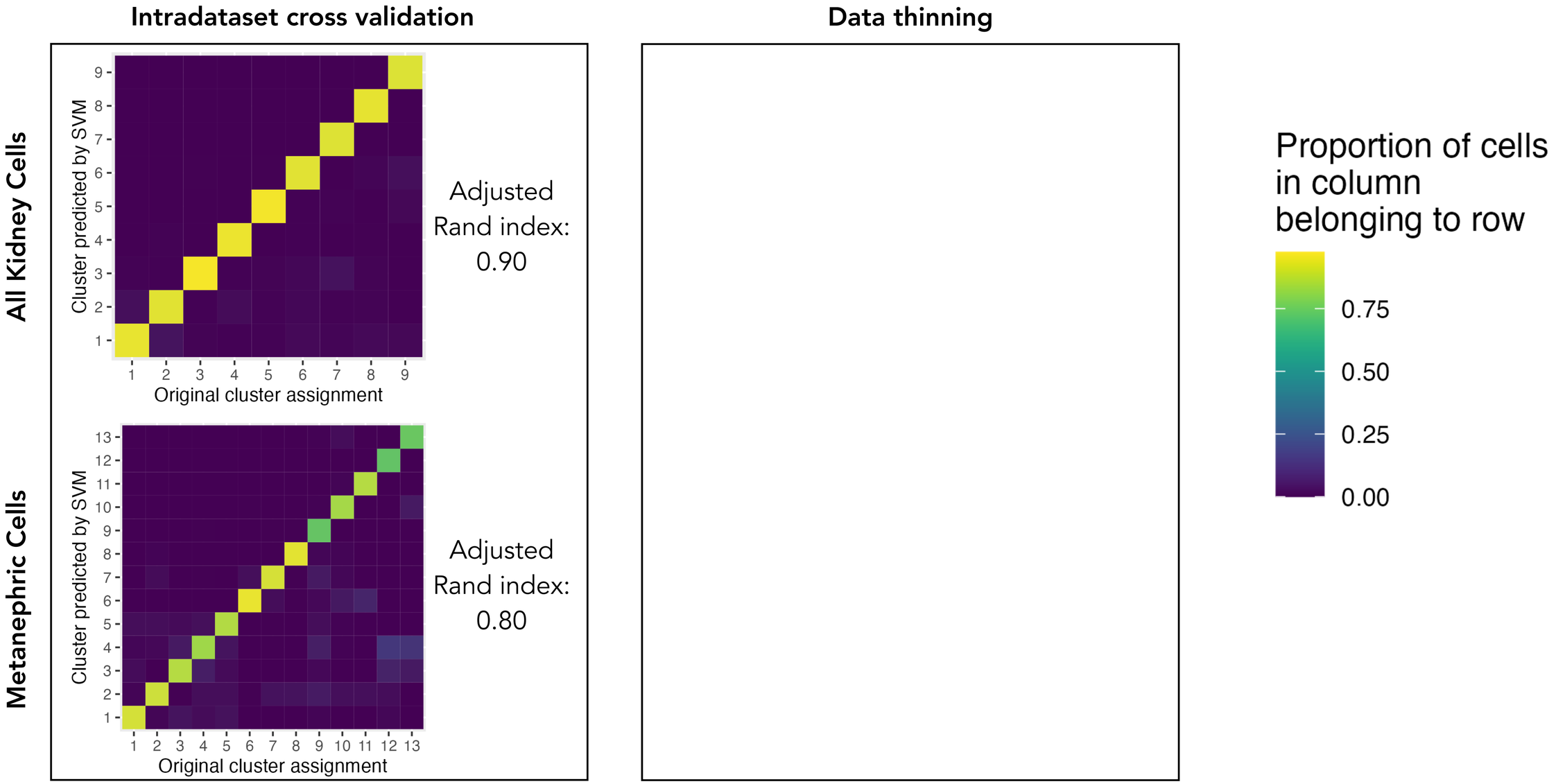
Adjusted Rand Index $= 0.01$

Re-analysis of Kidney cell data from fetal cell atlas

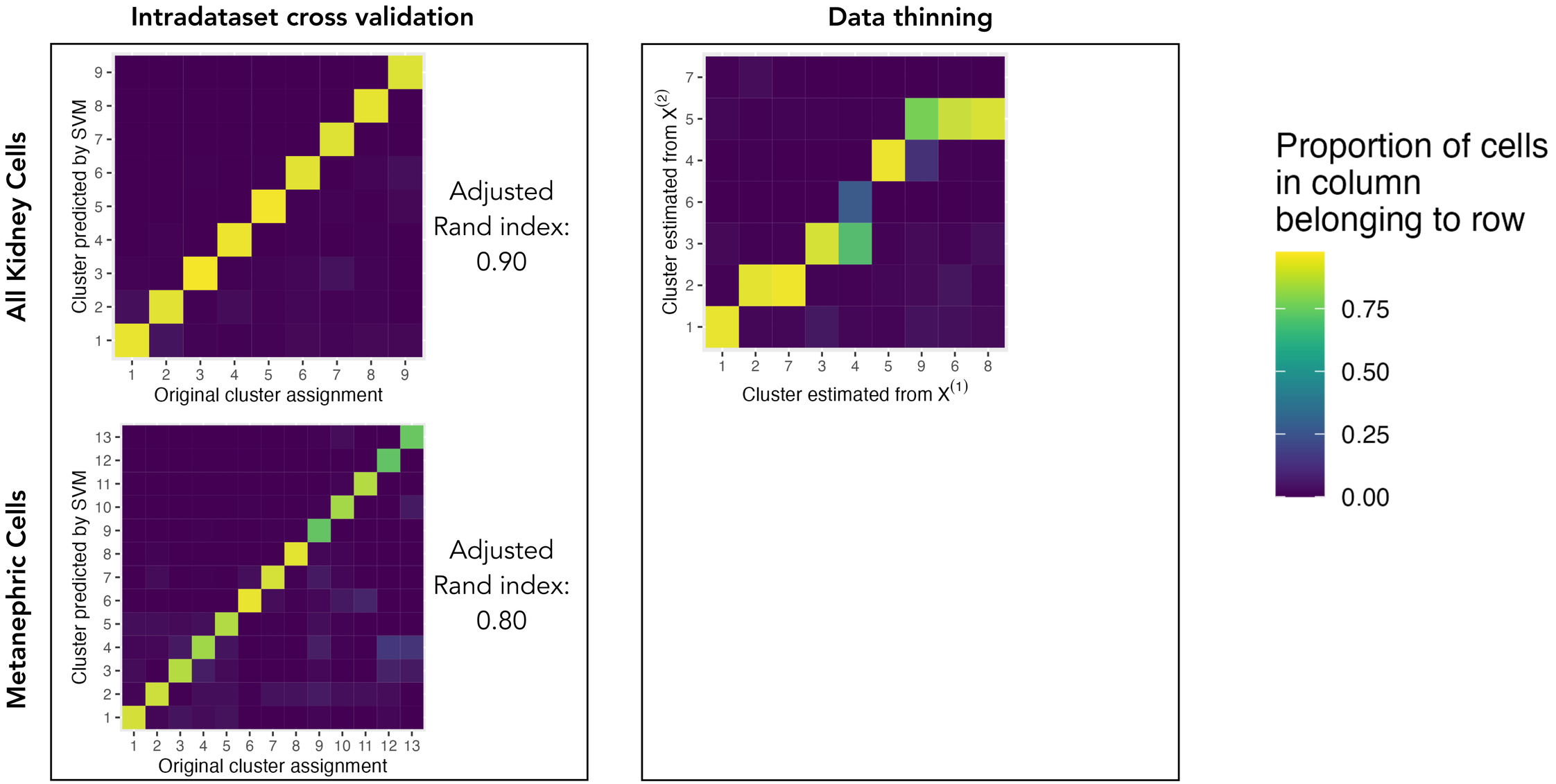




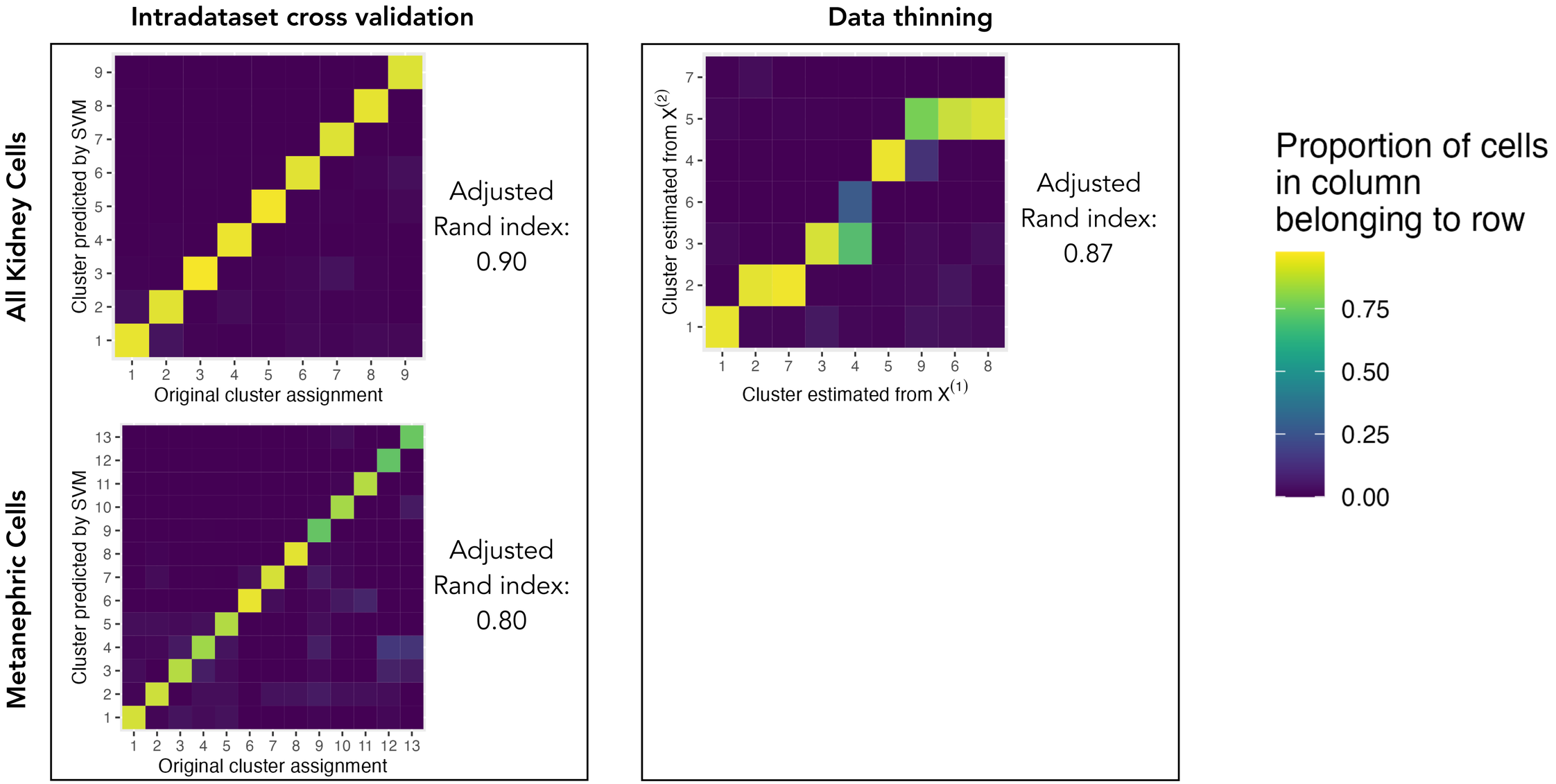
Re-analysis of Kidney cell data from fetal cell atlas



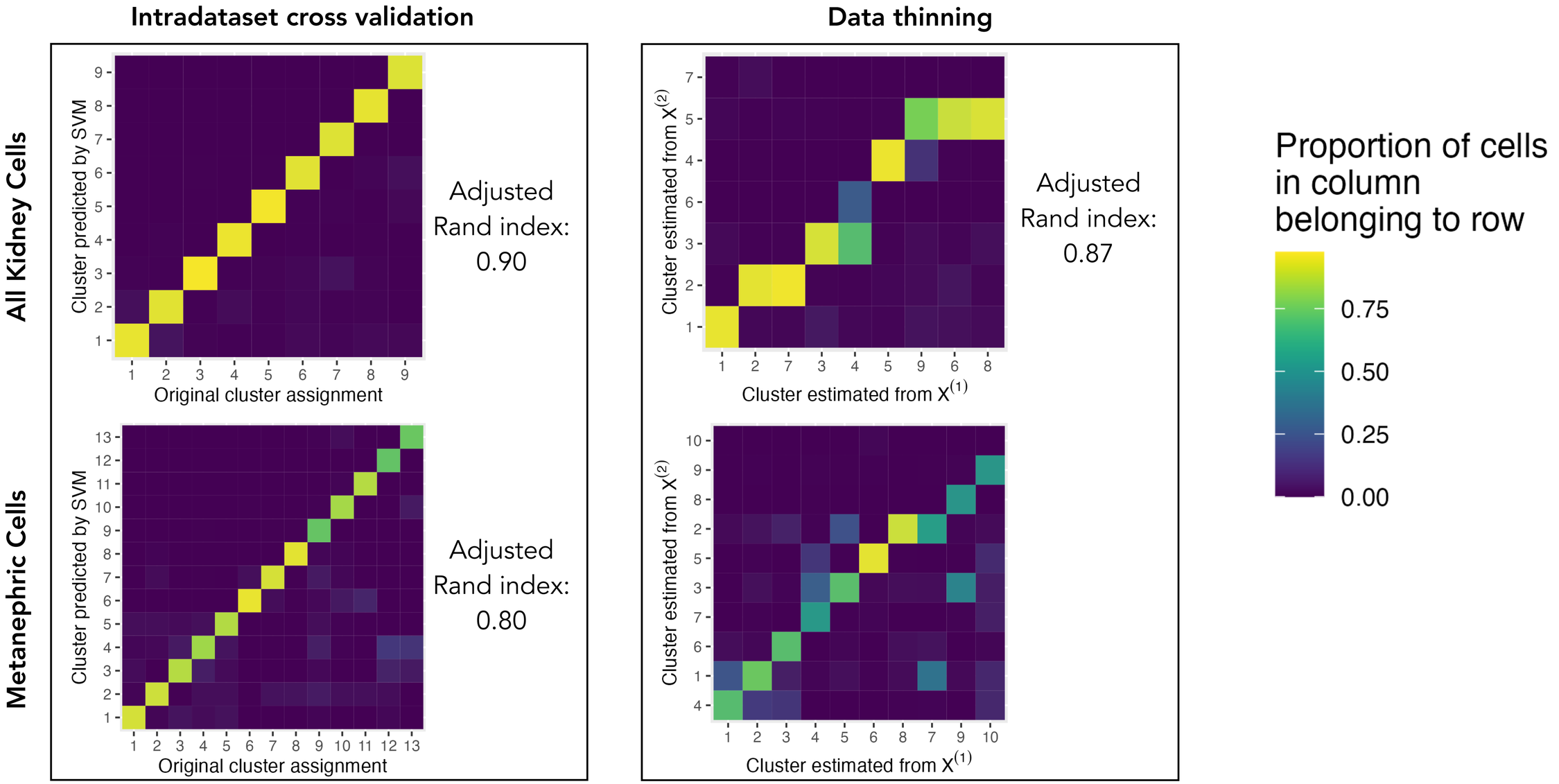
Re-analysis of Kidney cell data from fetal cell atlas



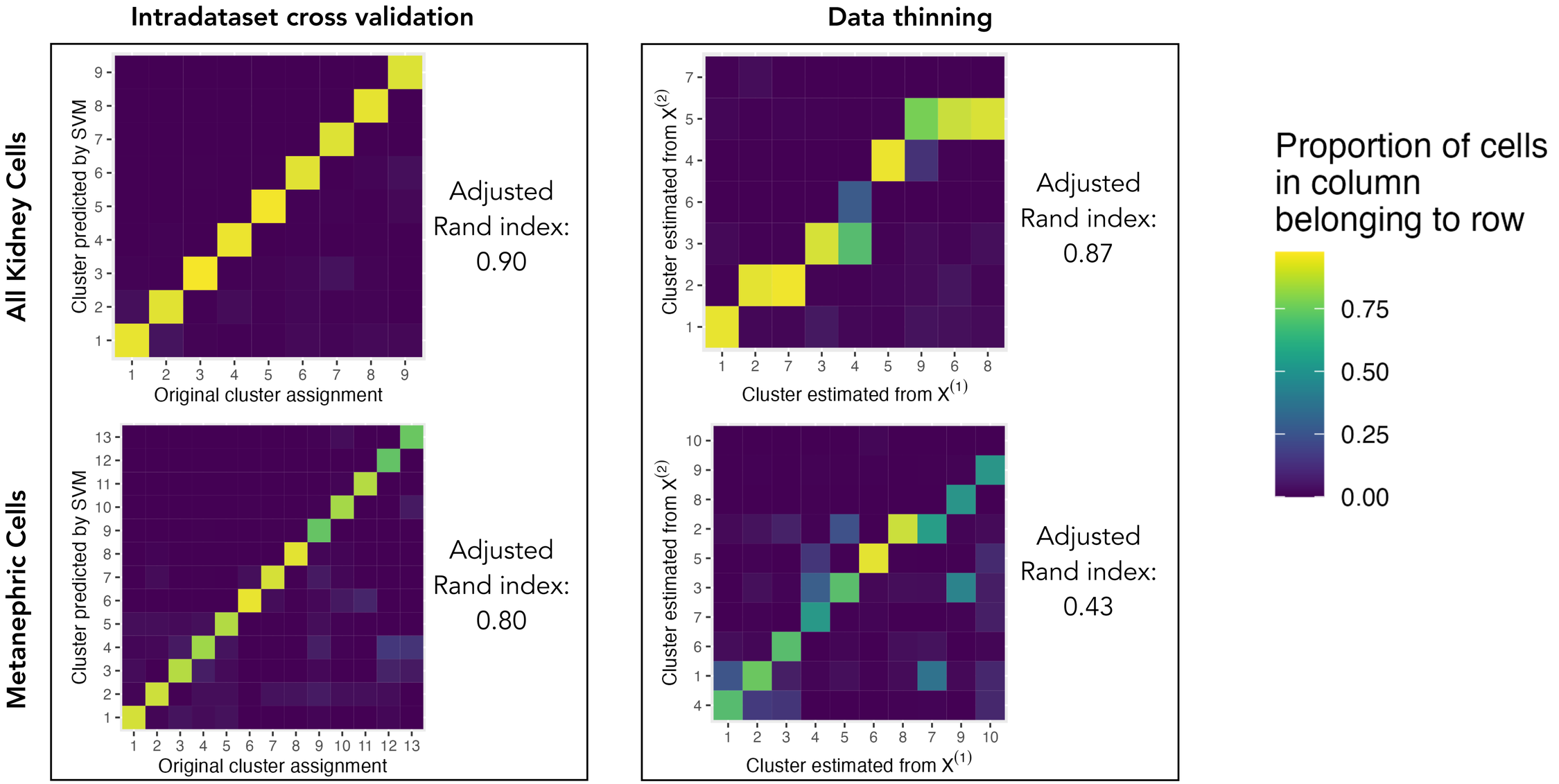
Re-analysis of Kidney cell data from fetal cell atlas



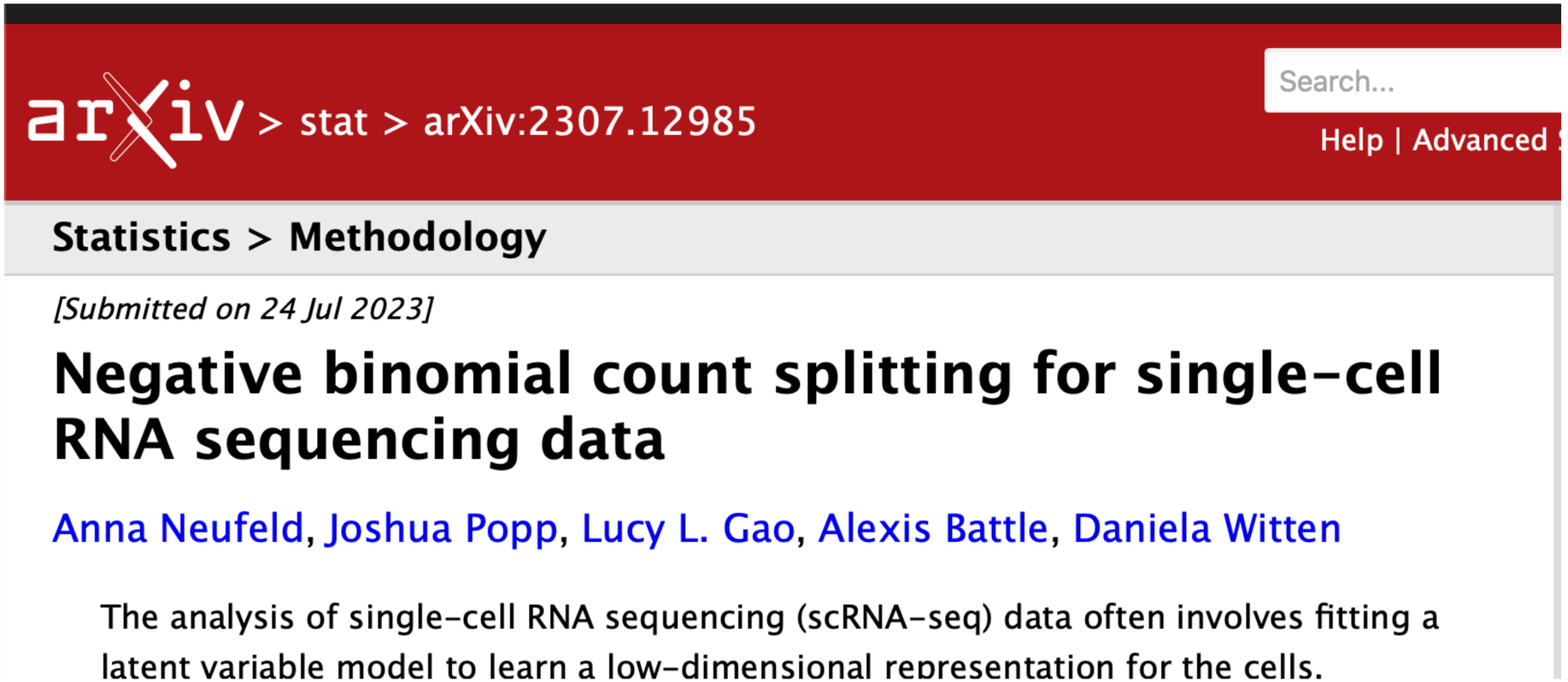
Re-analysis of Kidney cell data from fetal cell atlas



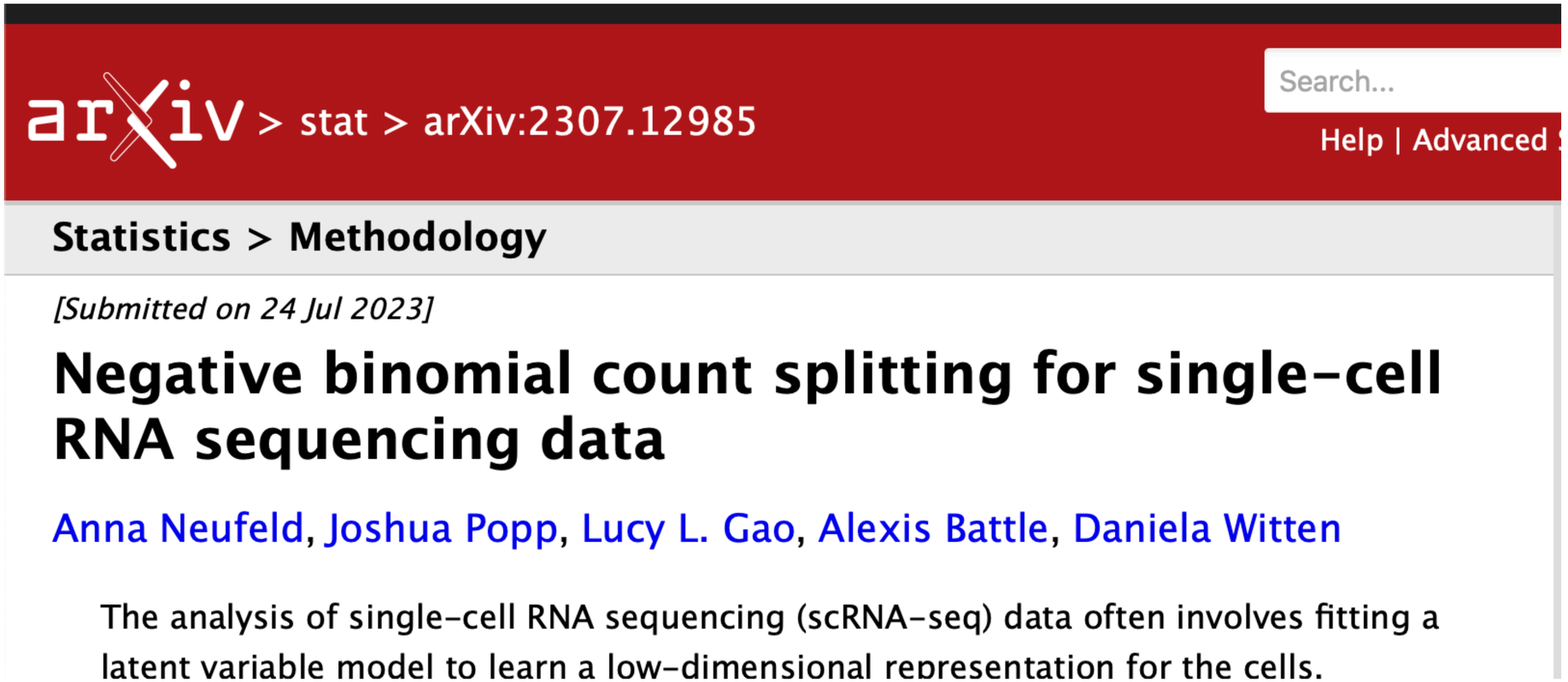
Re-analysis of Kidney cell data from fetal cell atlas



Negative binomial thinning is useful for scRNA-seq data



Negative binomial thinning is useful for scRNA-seq data



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

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**

Ways to avoid double dipping

1. Sample splitting.

2. Data thinning.

Ways to avoid double dipping

1. Sample splitting.

   Super flexible! No distributional assumptions needed.

2. Data thinning.

Ways to avoid double dipping

1. Sample splitting.

   Super flexible! No distributional assumptions needed.

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

2. Data thinning.

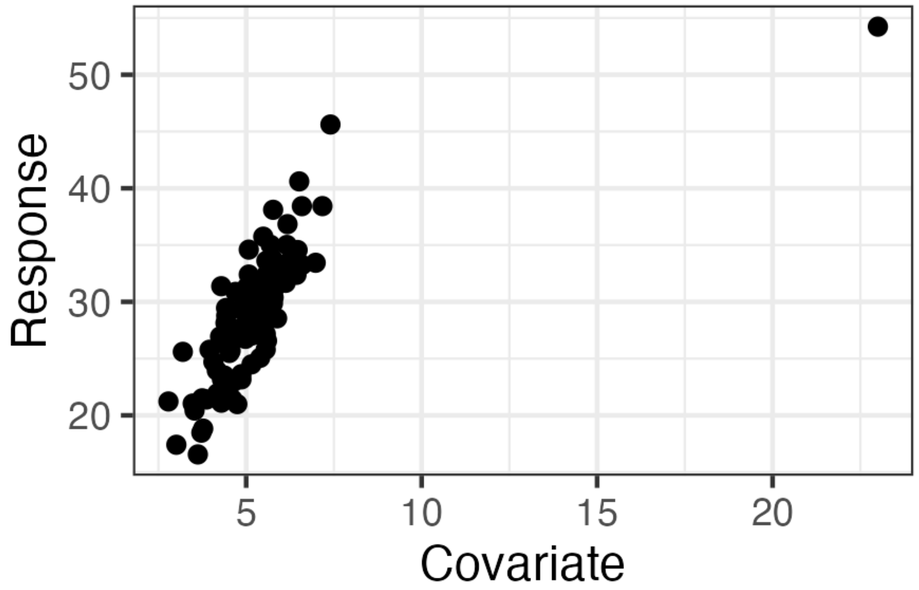
Ways to avoid double dipping

1. Sample splitting.

   Super flexible! No distributional assumptions needed.

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

2. Data thinning.



Ways to avoid double dipping

1. Sample splitting.

   Super flexible! No distributional assumptions needed.

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

2. Data thinning.

Ways to avoid double dipping

1. Sample splitting.

   Super flexible! No distributional assumptions needed.

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

2. Data thinning.

   Works in supervised and unsupervised settings.

Ways to avoid double dipping

1. Sample splitting.

   Super flexible! No distributional assumptions needed.

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

2. Data thinning.

   Works in supervised and unsupervised settings.

   Requires distributional assumptions and knowledge of parameters.

Ways to avoid double dipping

1. Sample splitting.

   Super flexible! No distributional assumptions needed.

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

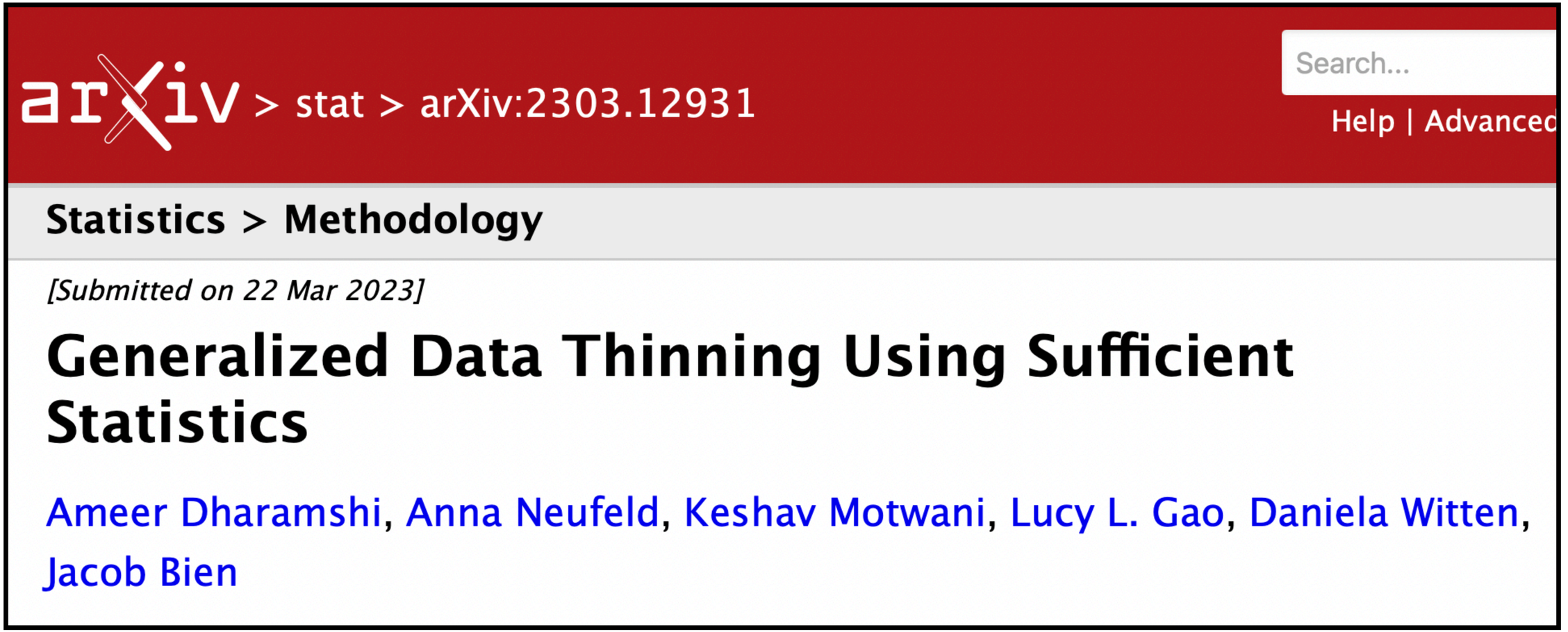
2. Data thinning.

   Works in supervised and unsupervised settings.

   Requires distributional assumptions and knowledge of parameters.

   Limited to convolution-closed distributions?

We have expanded the set of distributions that we can thin, and are working on additional extensions.



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?

The parameter ϵ governs an information tradeoff

Gaussian thinning algorithm

Suppose $X \sim \mathrm{N}\left(\mu, \sigma^2\right)$.

Draw
$X^{(1)} \sim \mathrm{N}(\epsilon x,\ \epsilon(1-\epsilon)\sigma^2)$ and
$X^{(2)} = X - X^{(1)}$.
Then:

1) $X^{(1)} \sim \mathrm{N}\left(\epsilon\mu, \epsilon\sigma^2\right)$
2) $X^{(2)} \sim \mathrm{N}\left((1-\epsilon)\mu, (1-\epsilon)\sigma^2\right)$
3) $X^{(1)} \perp\!\!\!\perp X^{(2)}$.

The parameter ϵ governs an information tradeoff

Gaussian thinning algorithm

Suppose $X \sim \mathrm{N}\left(\mu, \sigma^2\right)$.

Draw
$X^{(1)} \sim \mathrm{N}(\epsilon x,\ \epsilon(1-\epsilon)\sigma^2)$ and
$X^{(2)} = X - X^{(1)}$.
Then:

1) $X^{(1)} \sim \mathrm{N}\left(\epsilon\mu, \epsilon\sigma^2\right)$
2) $X^{(2)} \sim \mathrm{N}\left((1-\epsilon)\mu, (1-\epsilon)\sigma^2\right)$
3) $X^{(1)} \perp\!\!\!\perp X^{(2)}$.

Theorem: If we data thin with parameter ϵ, the Fisher information in X about μ is divided between $X^{(1)}$ and $X^{(2)}$ with proportions ϵ and $1-\epsilon$.

The parameter ϵ governs an information tradeoff

Gaussian thinning algorithm

Suppose $X \sim \mathrm{N}\left(\mu, \sigma^2\right)$.

Draw
$X^{(1)} \sim \mathrm{N}(\epsilon x,\ \epsilon(1-\epsilon)\sigma^2)$ and
$X^{(2)} = X - X^{(1)}$.
Then:

1) $X^{(1)} \sim \mathrm{N}\left(\epsilon\mu, \epsilon\sigma^2\right)$
2) $X^{(2)} \sim \mathrm{N}\left((1-\epsilon)\mu, (1-\epsilon)\sigma^2\right)$
3) $X^{(1)} \perp\!\!\!\perp X^{(2)}$.

Theorem: If we data thin with parameter ϵ, the Fisher information in X about μ is divided between $X^{(1)}$ and $X^{(2)}$ with proportions ϵ and $1-\epsilon$.

Similar results can be derived for other decompositions.

Our recipe extends naturally to splitting into M>2 folds

Our recipe extends naturally to splitting into M>2 folds

Goal: split a single observation X into $\left(X^{(1)}, \ldots, X^{(M)}\right)$ such that:

(1) Each $X^{(m)}$ has the same distribution as X, up to a parameter scaling.

(2) The $X^{(m)}$ are mutually independent.

Our recipe extends naturally to splitting into M>2 folds

Distribution of X	Draw $\left(X^{(1)}, \ldots, X^{(M)}\right) \mid X = x$ from:	Distribution of $X^{(m)}$
$\text{Poisson}(\lambda)$	$\text{Multinomial}(x, \epsilon_1, \ldots, \epsilon_M)$	$\text{Poisson}(\epsilon_m \lambda)$

Goal: split a single observation X into $\left(X^{(1)}, \ldots, X^{(M)}\right)$ such that:

(1) Each $X^{(m)}$ has the same distribution as X, up to a parameter scaling.

(2) The $X^{(m)}$ are mutually independent.

Our recipe extends naturally to splitting into M>2 folds

Distribution of X	Draw $\left(X^{(1)}, \ldots, X^{(M)}\right) \mid X = x$ from:	Distribution of $X^{(m)}$
$\text{Poisson}(\lambda)$	$\text{Multinomial}(x, \epsilon_1, \ldots, \epsilon_M)$	$\text{Poisson}(\epsilon_m \lambda)$

Our recipe extends naturally to splitting into M>2 folds

Distribution of X	Draw $\left(X^{(1)}, \ldots, X^{(M)}\right) \mid X = x$ from:	Distribution of $X^{(m)}$
$\text{Poisson}(\lambda)$	$\text{Multinomial}(x, \epsilon_1, \ldots, \epsilon_M)$	$\text{Poisson}(\epsilon_m \lambda)$
$\text{N}(\mu, \sigma^2)$	$\text{N}_M(\epsilon\mu, \sigma^2 \text{diag}(\epsilon) - \sigma^2 \epsilon\epsilon^T)$.	$\text{N}(\epsilon_m \mu, \epsilon_m \sigma^2)$

Our recipe extends naturally to splitting into M>2 folds

Distribution of X	Draw $\left(X^{(1)}, \ldots, X^{(M)}\right) \mid X = x$ from:	Distribution of $X^{(m)}$
$\text{Poisson}(\lambda)$	$\text{Multinomial}(x, \epsilon_1, \ldots, \epsilon_M)$	$\text{Poisson}(\epsilon_m \lambda)$
$\text{N}(\mu, \sigma^2)$	$\text{N}_M(\epsilon\mu, \sigma^2 \text{diag}(\epsilon) - \sigma^2 \epsilon\epsilon^T)$.	$\text{N}(\epsilon_m \mu, \epsilon_m \sigma^2)$
$\text{NegativeBinomial}(\mu, b)$	$\text{DirichletMultinomial}(x, \epsilon_1 b, \ldots, \epsilon_M b)$.	$\text{NegativeBinomial}(\epsilon_m \mu, \epsilon_m b)$

Our recipe extends naturally to splitting into M>2 folds

Distribution of X	Draw $\left(X^{(1)}, \ldots, X^{(M)}\right) \mid X = x$ from:	Distribution of $X^{(m)}$
$\text{Poisson}(\lambda)$	$\text{Multinomial}(x, \epsilon_1, \ldots, \epsilon_M)$	$\text{Poisson}(\epsilon_m \lambda)$
$\text{N}(\mu, \sigma^2)$	$\text{N}_M(\epsilon\mu, \sigma^2 \text{diag}(\epsilon) - \sigma^2 \epsilon\epsilon^T)$.	$\text{N}(\epsilon_m \mu, \epsilon_m \sigma^2)$
$\text{NegativeBinomial}(\mu, b)$	$\text{DirichletMultinomial}(x, \epsilon_1 b, \ldots, \epsilon_M b)$.	$\text{NegativeBinomial}(\epsilon_m \mu, \epsilon_m b)$
$\text{Gamma}(\alpha, \beta)$	$x \cdot \text{Dirichlet}\left(\epsilon_1 \alpha, \ldots, \epsilon_M \alpha\right)$	$\text{Gamma}(\epsilon_m \alpha, \beta)$
$\text{Exponential}(\lambda)$	$x \cdot \text{Dirichlet}\left(\epsilon_1, \ldots, \epsilon_M\right)$	$\text{Gamma}(\epsilon_m, \lambda)$
$\text{Binomial}(r, p)$	$\text{MultivariateHypergeometric}(\epsilon_1 r, \ldots, \epsilon_M r, x)$.	$\text{Binomial}(\epsilon_m r, p)$

Re-analysis of Kidney cell data from fetal cell atlas