

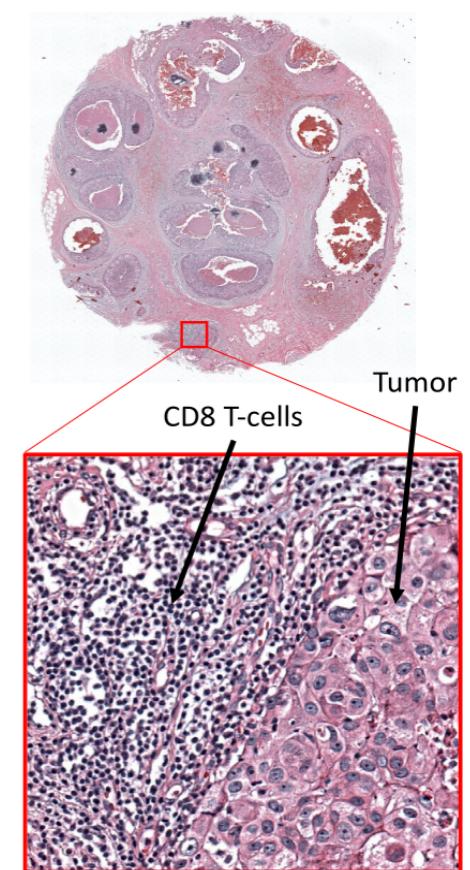
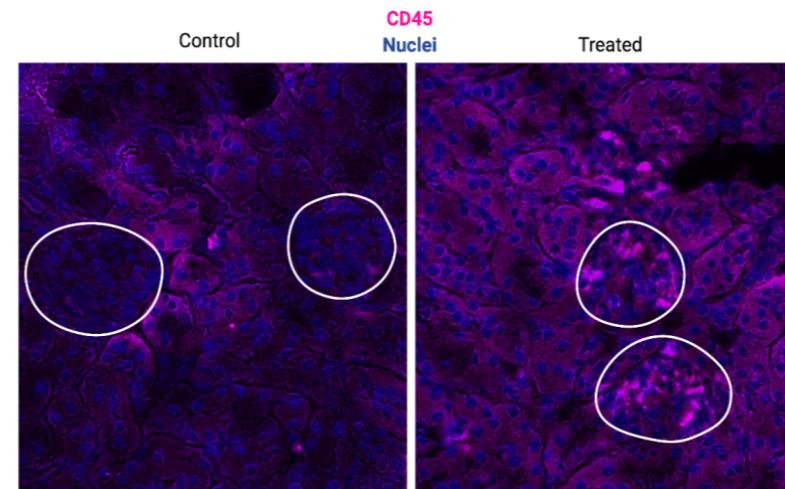
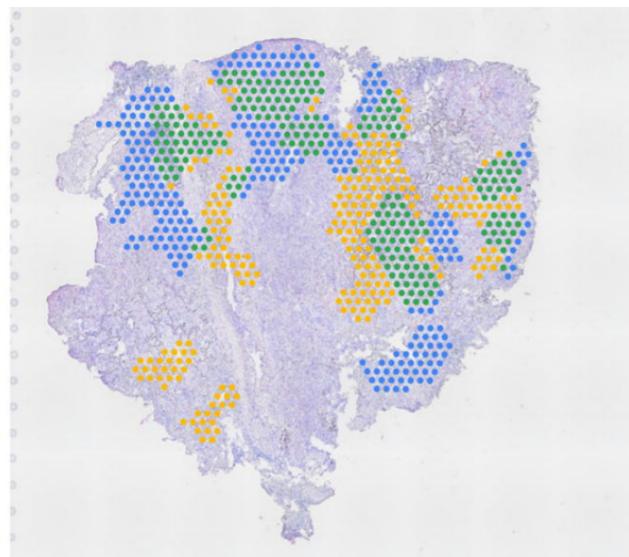
Topological Data Analysis and Spatial Transcriptomics

Instructor:

Mathieu Carrière

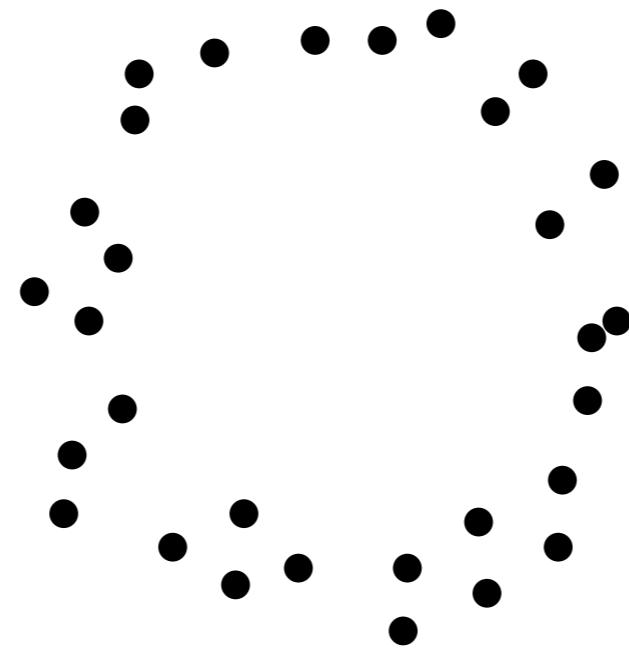
Centre Inria d'Université Côte d'Azur

firstname.lastname@inria.fr



Introduction: what is TDA?

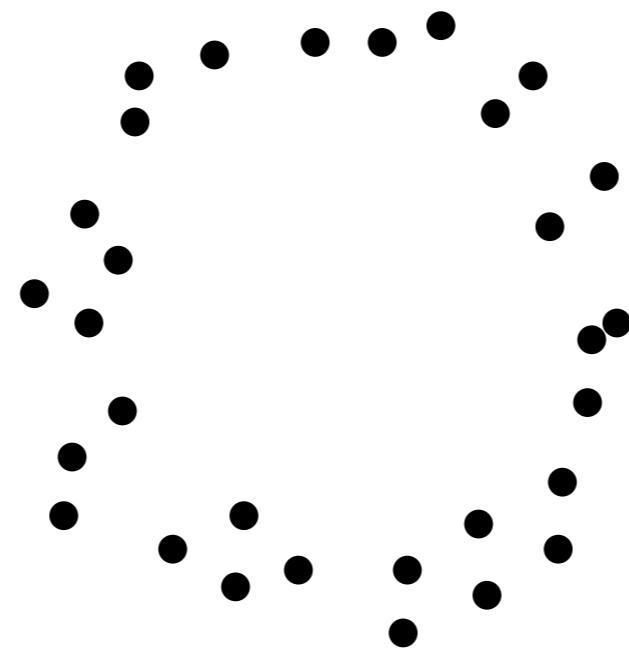
Roughly speaking, TDA allows to build features and descriptors from data sets using **topology** (i.e., the presence of holes in arbitrary dimensions)...



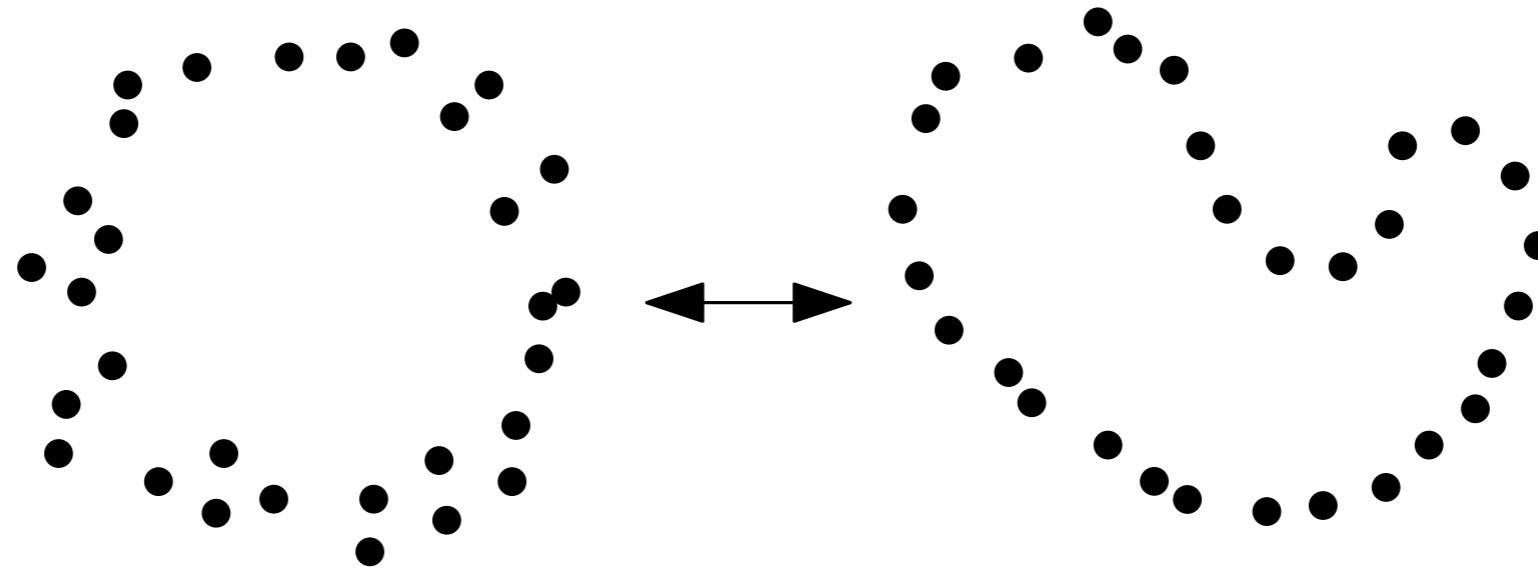
Introduction: what is TDA?

Roughly speaking, TDA allows to build features and descriptors from data sets using **topology** (i.e., the presence of holes in arbitrary dimensions)...

...but why is that interesting?

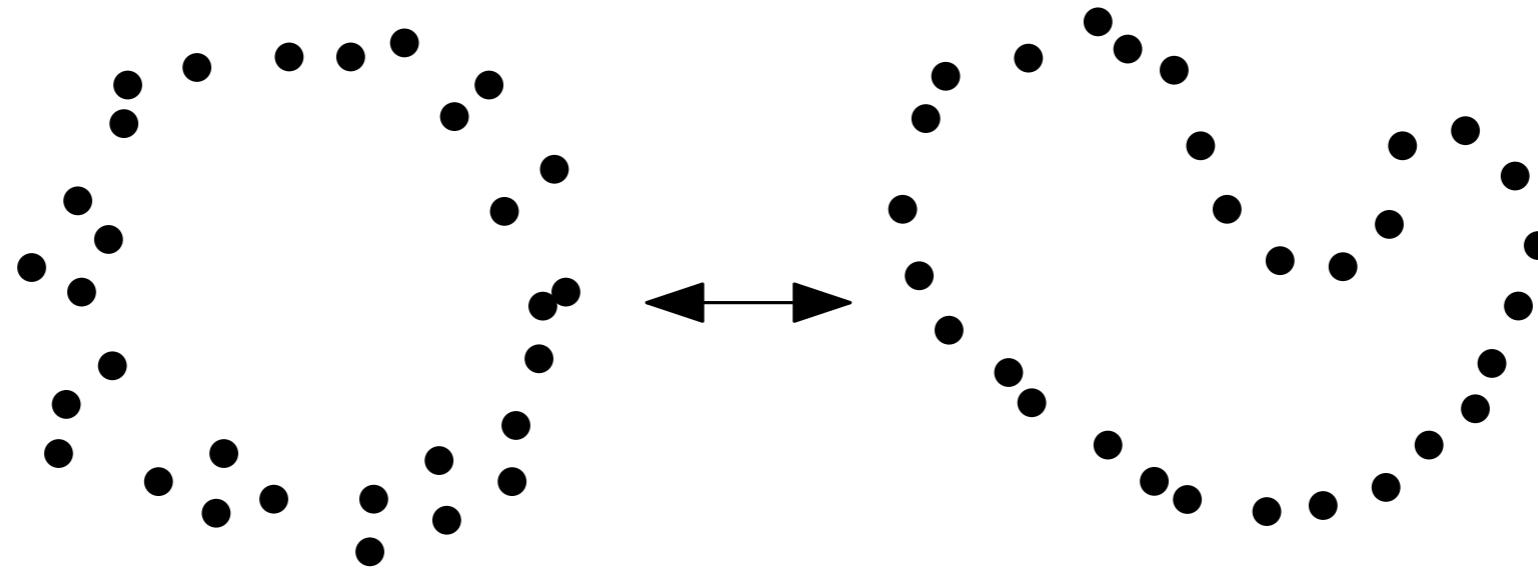


Introduction: what is TDA?



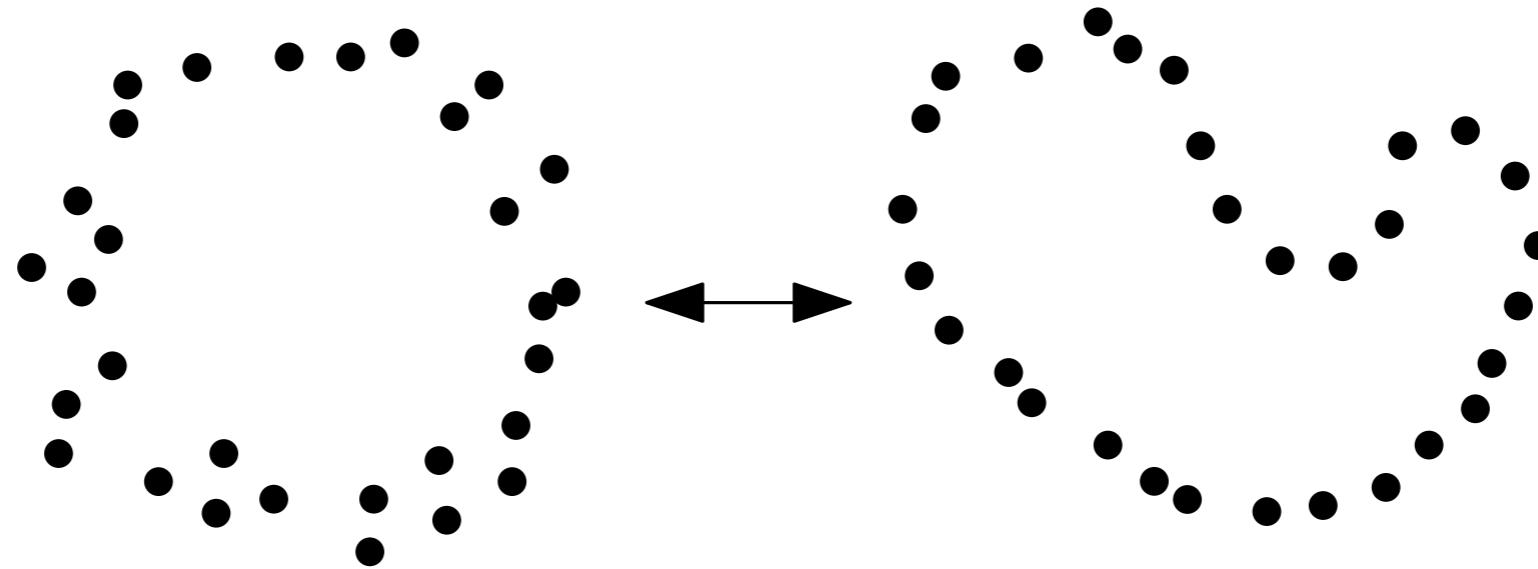
- **Coordinate invariance:** topological features/invariants do not rely on any coordinate system so no need to have data with coordinates, or to embed data in spaces with coordinates... but the metric (distance/similarity between data points) is important.

Introduction: what is TDA?



- **Coordinate invariance:** topological features/invariants do not rely on any coordinate system so no need to have data with coordinates, or to embed data in spaces with coordinates... but the metric (distance/similarity between data points) is important.
- **Deformation invariance:** topological features are invariant under homeomorphism and reparameterization.

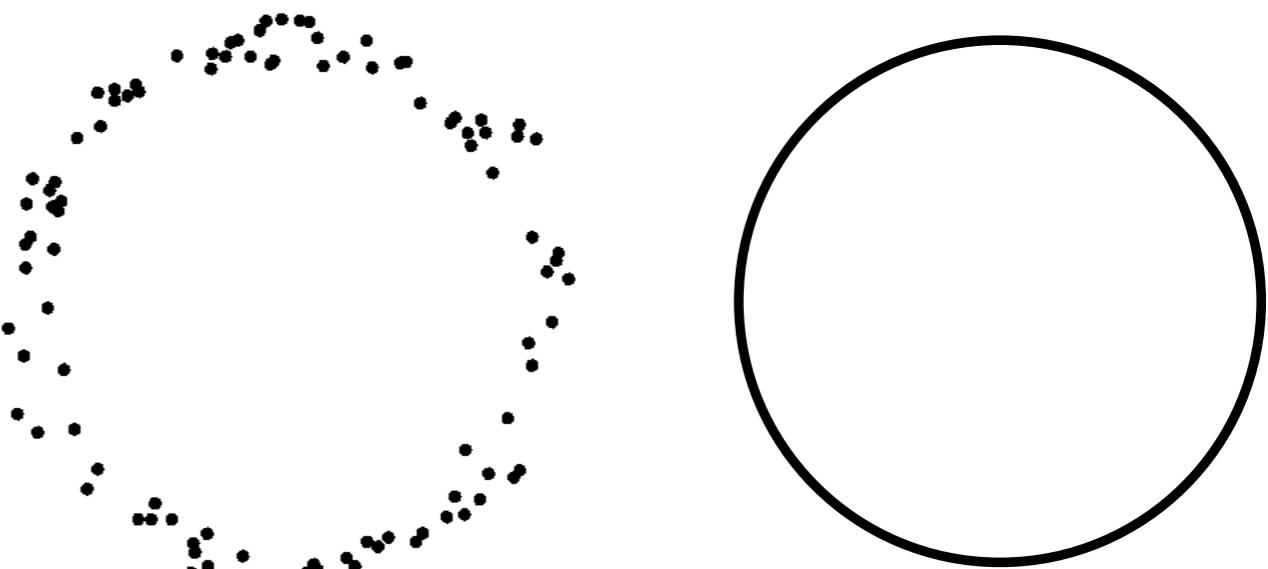
Introduction: what is TDA?



- **Coordinate invariance:** topological features/invariants do not rely on any coordinate system so no need to have data with coordinates, or to embed data in spaces with coordinates... but the metric (distance/similarity between data points) is important.
- **Deformation invariance:** topological features are invariant under homeomorphism and reparameterization.
- **Compressed representation:** topology offers a set of tools to summarize the data in compact ways while preserving its topological structure.

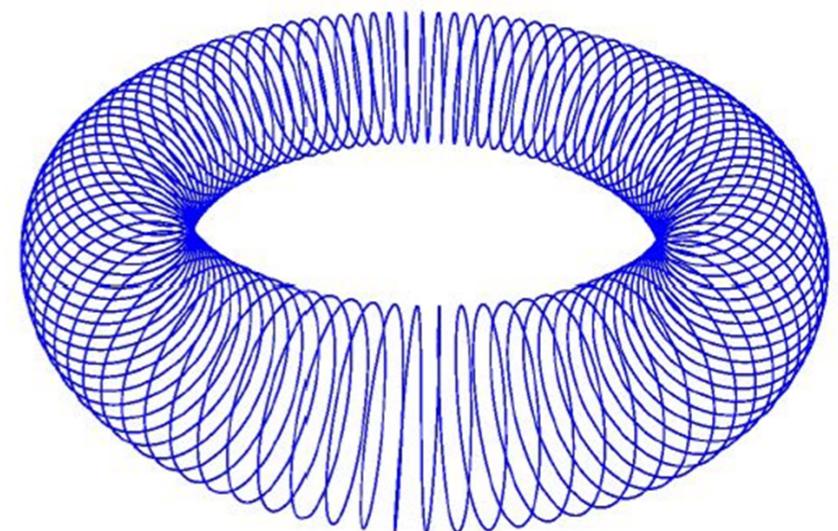
Introduction: what is TDA?

Problem: how to define the *topology* of a data set?



Cons of topology:

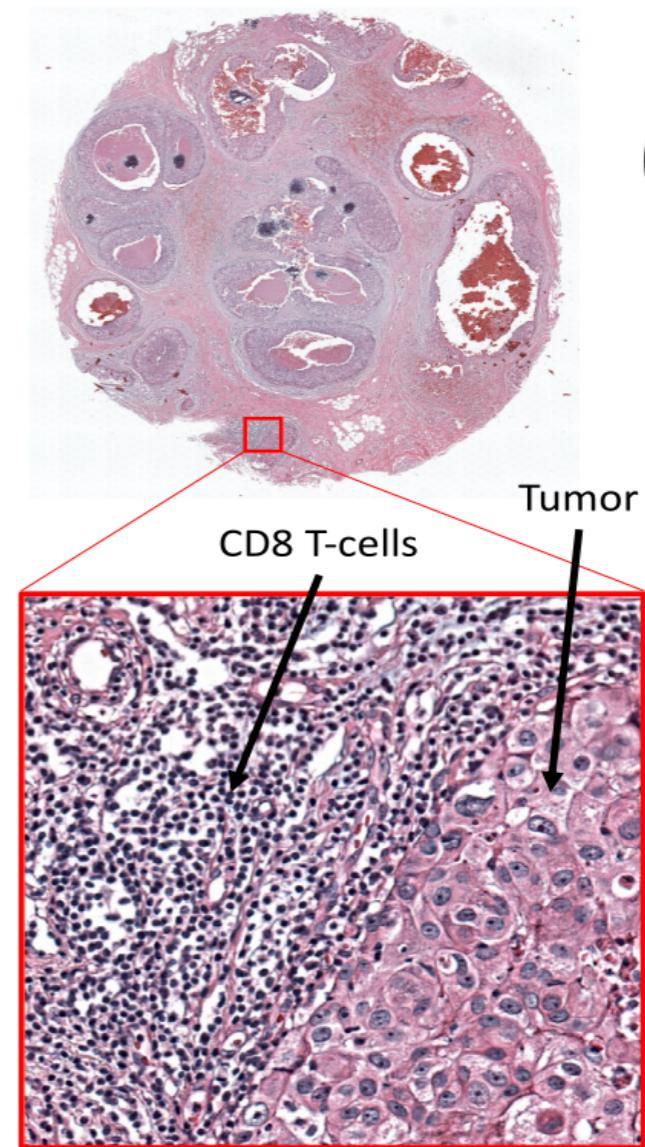
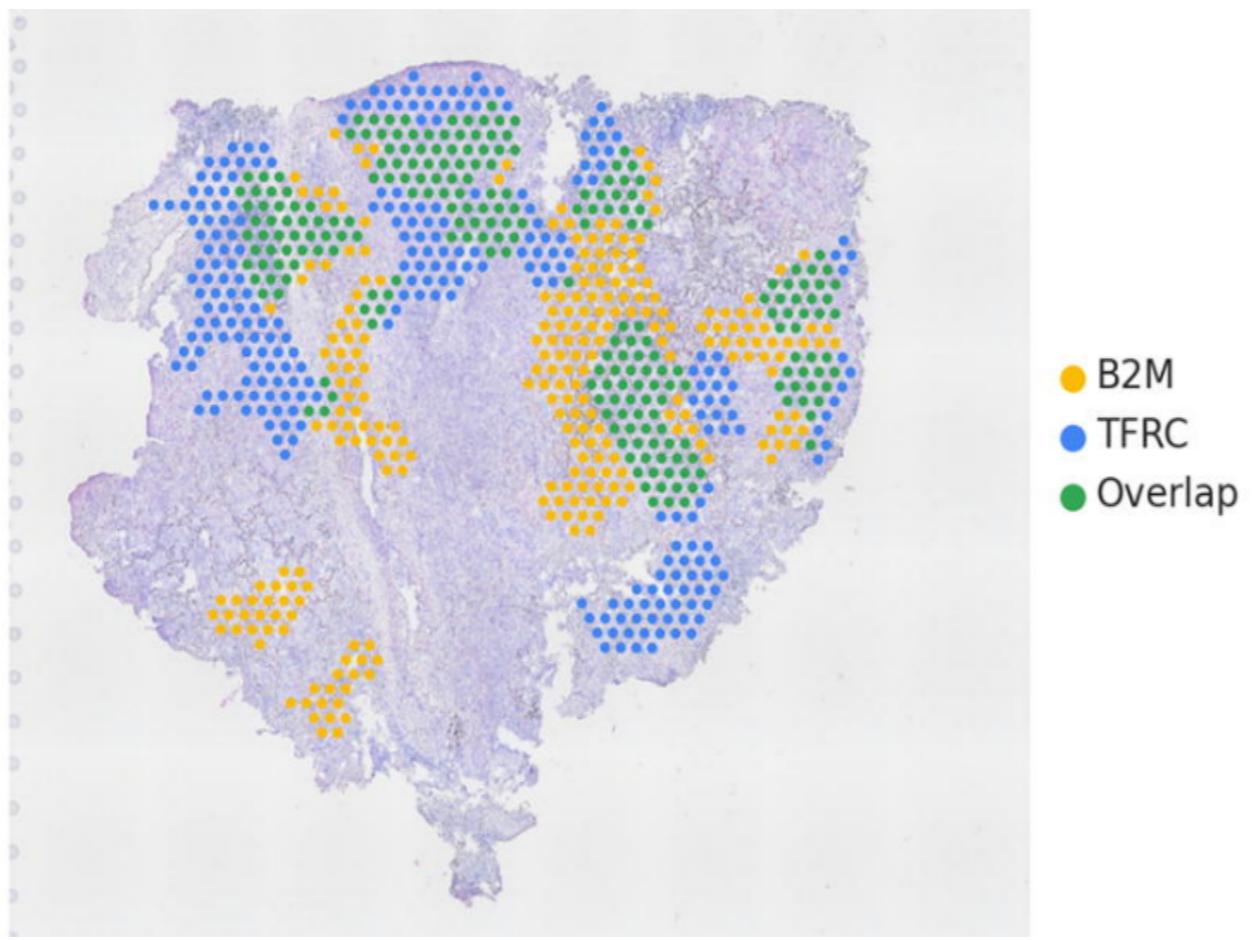
- No direct access to topological/geometric information: need of intermediate constructions built on top of the data.
- Distinguish topological “signal” from noise.
- Topological information may be multiscale.
- Statistical analysis of topological information.



Introduction: spatial transcriptomics data

Spatial transcriptomics data measures two things:

- the **position** (x and y coordinates) of each cell in a tissue,
- the **expression** of every gene of each cell in a tissue.



Plan of the course

1. ToMATo for colocalizing cell types
2. Rips persistence for marker gene correlations
3. Multi-persistence for immune cell arrangements
4. Future research directions

Plan of the course

1. ToMATo for colocalizing cell types

[Bae et al. - 2022 - *STopover captures spatial colocalization and interaction in the tumor microenvironment using topological analysis in spatial transcriptomics data*]

2. Rips persistence for marker gene correlations

[Alsaleh et al. - 2022 - *Spatial transcriptomic analysis reveals associations between genes and cellular topology in breast and prostate cancers*]

3. Multi-persistence for immune cell arrangements

[Vipond et al. - 2021 - *Multiparameter persistent homology landscapes identify immune cell spatial patterns in tumors*]

4. Future research directions

Plan of the talk

1. ToMATo for colocalizing cell types

[Bae et al. - 2022 - *STopover captures spatial colocalization and interaction in the tumor microenvironment using topological analysis in spatial transcriptomics data*]

2. Rips persistence for marker gene correlations

[Alsaleh et al. - 2022 - *Spatial transcriptomic analysis reveals associations between genes and cellular topology in breast and prostate cancers*]

3. Multi-persistence for immune cell arrangements

[Vipond et al. - 2021 - *Multiparameter persistent homology landscapes identify immune cell spatial patterns in tumors*]

4. Future research directions

1. ToMATo for colocalizing cell types

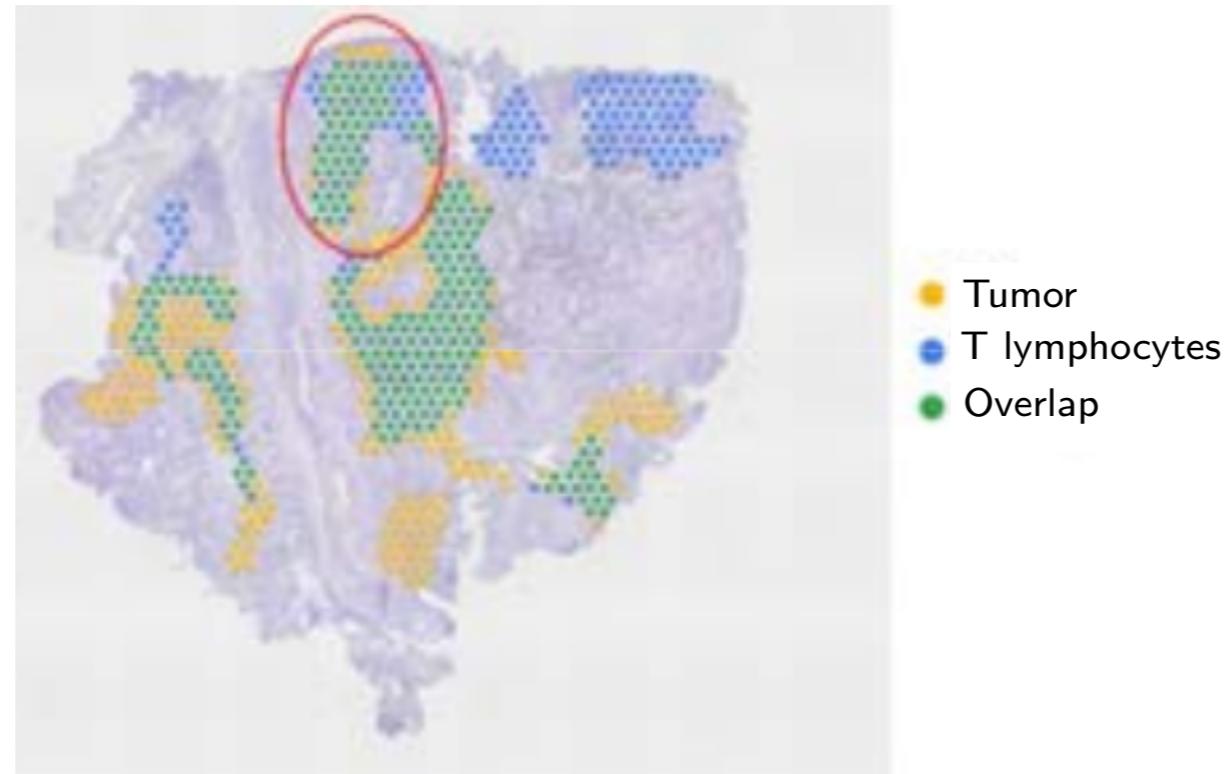
1. ToMATo for colocalizing cell types

Q: How to characterize and encode the interactions between cell types and markers *using their spatial locations*, i.e., their *colocalizations*?



1. ToMATo for colocalizing cell types

Q: How to characterize and encode the interactions between cell types and markers *using their spatial locations*, i.e., their *colocalizations*?



A: Compute the *Jaccard similarity* between *spatial clusters* computed from marker gene expression as a *colocalization quantifier*.

1. ToMATo for colocalizing cell types

Q: How to characterize and encode the interactions between cell types and markers *using their spatial locations*, i.e., their *colocalizations*?



A: Compute the *Jaccard similarity* between **stable spatial clusters** computed from **marker gene expression** as a *colocalization quantifier*.

→ 0-dimensional persistent homology with **ToMATo**

Motivation: the (in)stability of dendograms

Input: A set $X_n = \{x_1, \dots, x_n\}$ in a metric space (X, d) (or just a matrix of pairwise dissimilarities $((d_{i,j}))_{i,j}$).

Given two clusters $C, C' \subseteq X_n$ let $d(C, C') = \inf_{x \in C, x' \in C'} d(x, x')$.

Motivation: the (in)stability of dendograms

Input: A set $X_n = \{x_1, \dots, x_n\}$ in a metric space (X, d) (or just a matrix of pairwise dissimilarities $((d_{i,j}))_{i,j}$).

Given two clusters $C, C' \subseteq X_n$ let $d(C, C') = \inf_{x \in C, x' \in C'} d(x, x')$.

Agglomerative (bottom-up)

1. Start with a clustering where each x_i is a cluster.
2. At each step, merge the two closest clusters until it remains a single cluster (containing all data points).

Output: the resulting dendrogram.

Motivation: the (in)stability of dendograms

Input: A set $X_n = \{x_1, \dots, x_n\}$ in a metric space (X, d) (or just a matrix of pairwise dissimilarities $((d_{i,j}))_{i,j}$).

Given two clusters $C, C' \subseteq X_n$ let $d(C, C') = \inf_{x \in C, x' \in C'} d(x, x')$.

sup: complete linkage

Agglomerative (bottom-up)

$\frac{1}{|C| \cdot |C'|} \sum$: average linkage

1. Start with a clustering where each x_i is a cluster.
2. At each step, merge the two closest clusters until it remains a single cluster (containing all data points).

Output: the resulting dendrogram.

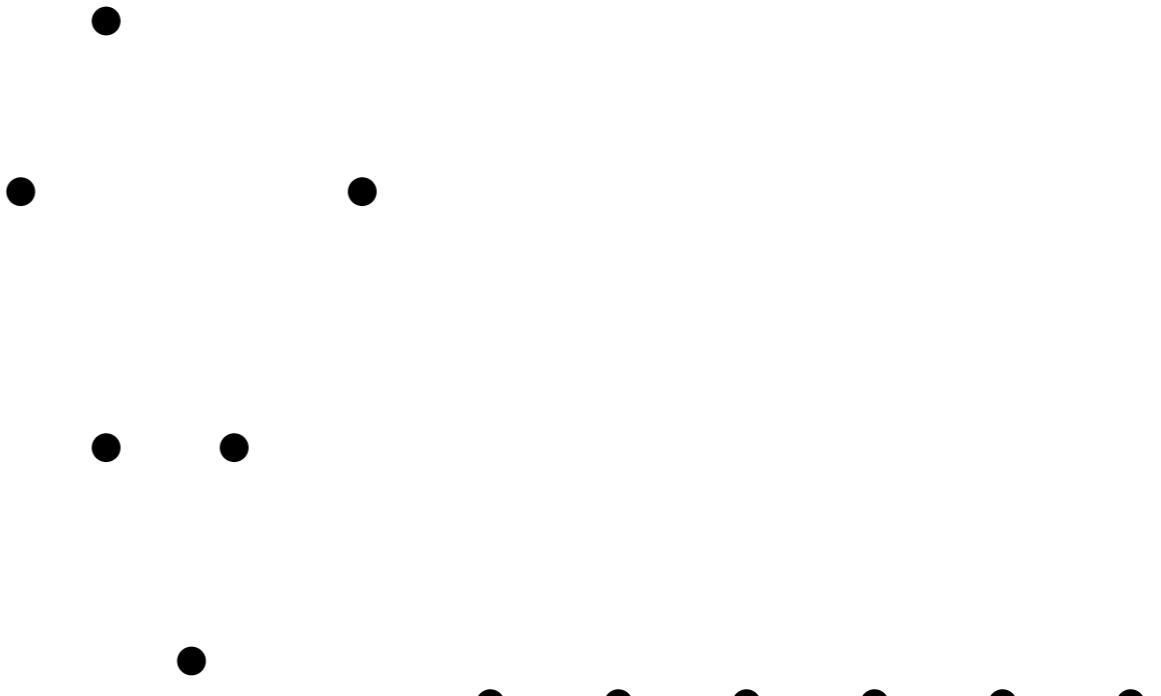
Motivation: the (in)stability of dendograms

Input: A set $X_n = \{x_1, \dots, x_n\}$ in a metric space (X, d) (or just a matrix of pairwise dissimilarities $((d_{i,j}))_{i,j}$).

Given two clusters $C, C' \subseteq X_n$ let $d(C, C') = \inf_{x \in C, x' \in C'} d(x, x')$.

Agglomerative (bottom-up)

1. Start with a clustering where each x_i is a cluster.
2. At each step, merge the two closest clusters until it remains a single cluster (containing all data points).



Output: the resulting dendrogram.

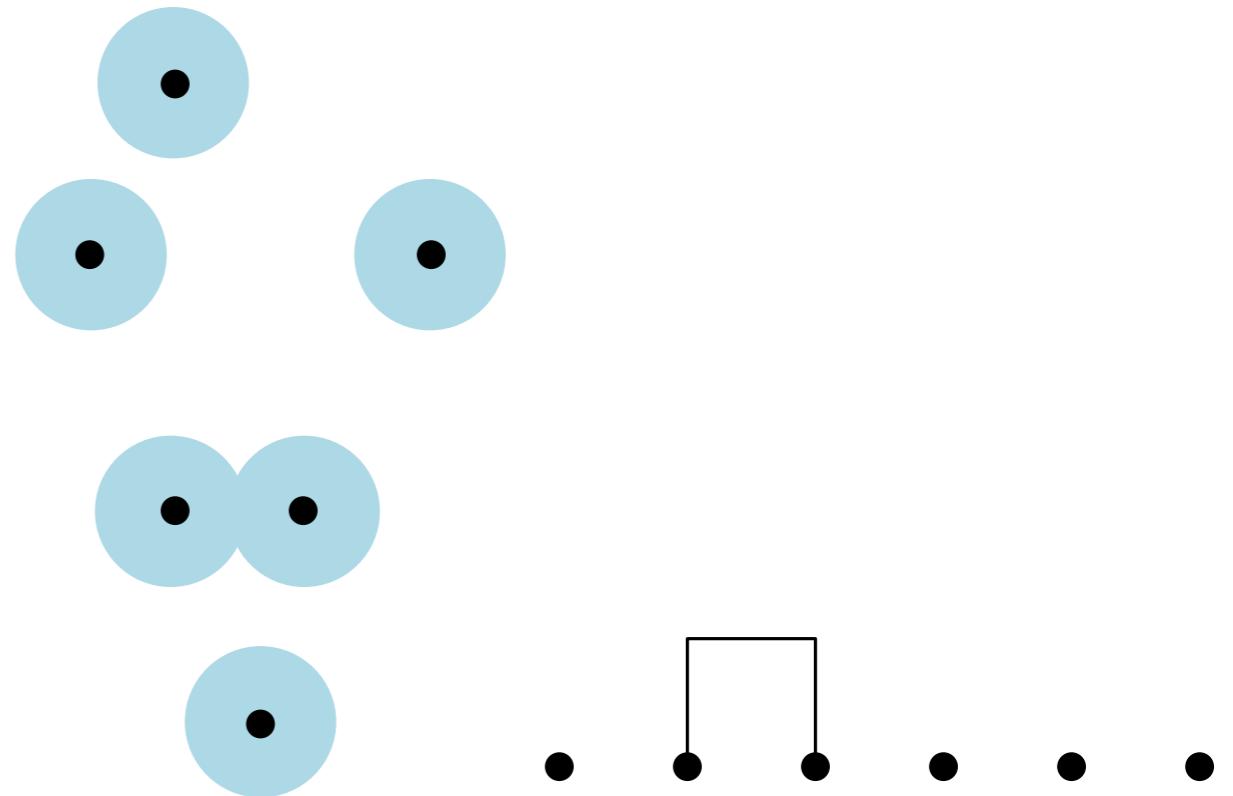
Motivation: the (in)stability of dendograms

Input: A set $X_n = \{x_1, \dots, x_n\}$ in a metric space (X, d) (or just a matrix of pairwise dissimilarities $((d_{i,j}))_{i,j}$).

Given two clusters $C, C' \subseteq X_n$ let $d(C, C') = \inf_{x \in C, x' \in C'} d(x, x')$.

Agglomerative (bottom-up)

1. Start with a clustering where each x_i is a cluster.
2. At each step, merge the two closest clusters until it remains a single cluster (containing all data points).



Output: the resulting dendrogram.

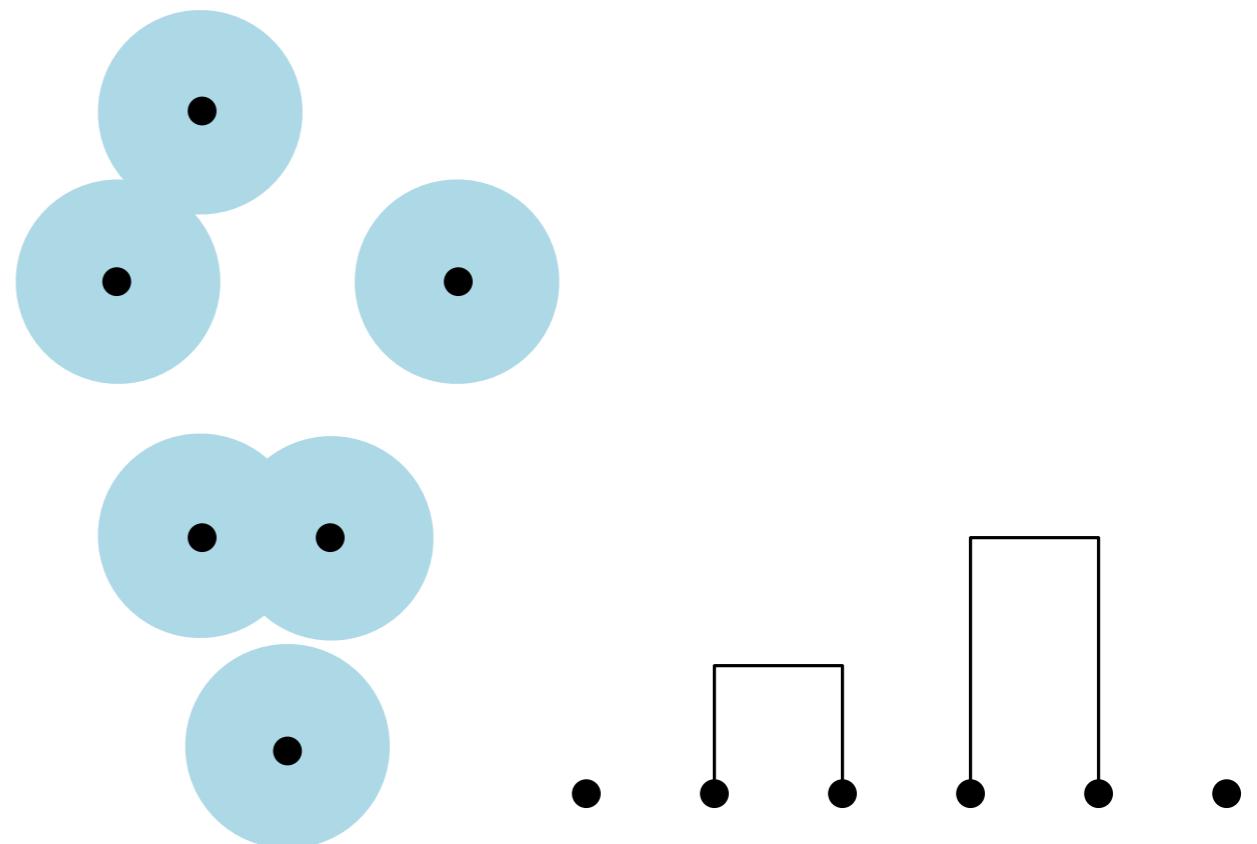
Motivation: the (in)stability of dendrograms

Input: A set $X_n = \{x_1, \dots, x_n\}$ in a metric space (X, d) (or just a matrix of pairwise dissimilarities $((d_{i,j}))_{i,j}$).

Given two clusters $C, C' \subseteq X_n$ let $d(C, C') = \inf_{x \in C, x' \in C'} d(x, x')$.

Agglomerative (bottom-up)

1. Start with a clustering where each x_i is a cluster.
2. At each step, merge the two closest clusters until it remains a single cluster (containing all data points).



Output: the resulting dendrogram.

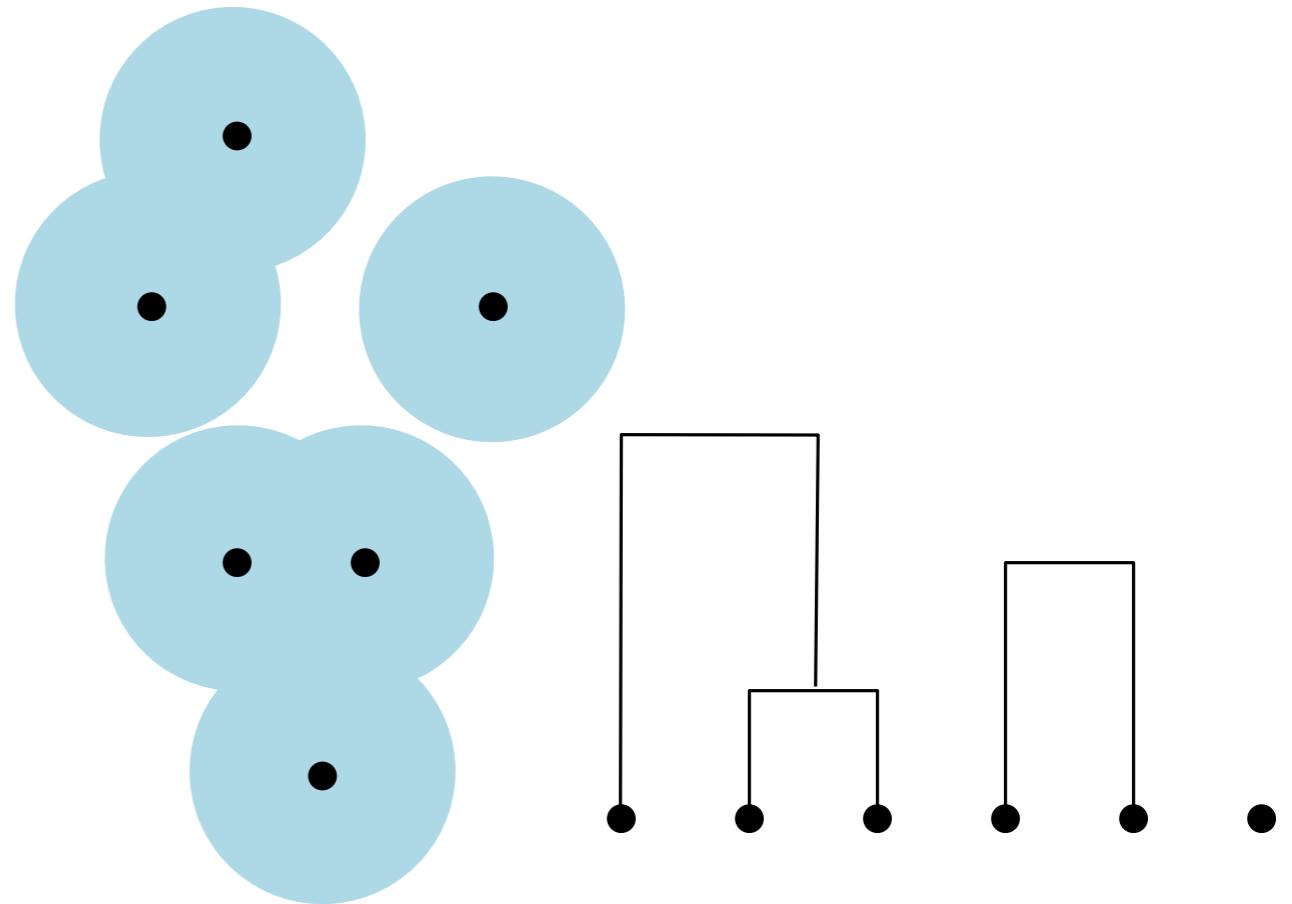
Motivation: the (in)stability of dendograms

Input: A set $X_n = \{x_1, \dots, x_n\}$ in a metric space (X, d) (or just a matrix of pairwise dissimilarities $((d_{i,j}))_{i,j}$).

Given two clusters $C, C' \subseteq X_n$ let $d(C, C') = \inf_{x \in C, x' \in C'} d(x, x')$.

Agglomerative (bottom-up)

1. Start with a clustering where each x_i is a cluster.
2. At each step, merge the two closest clusters until it remains a single cluster (containing all data points).



Output: the resulting dendrogram.

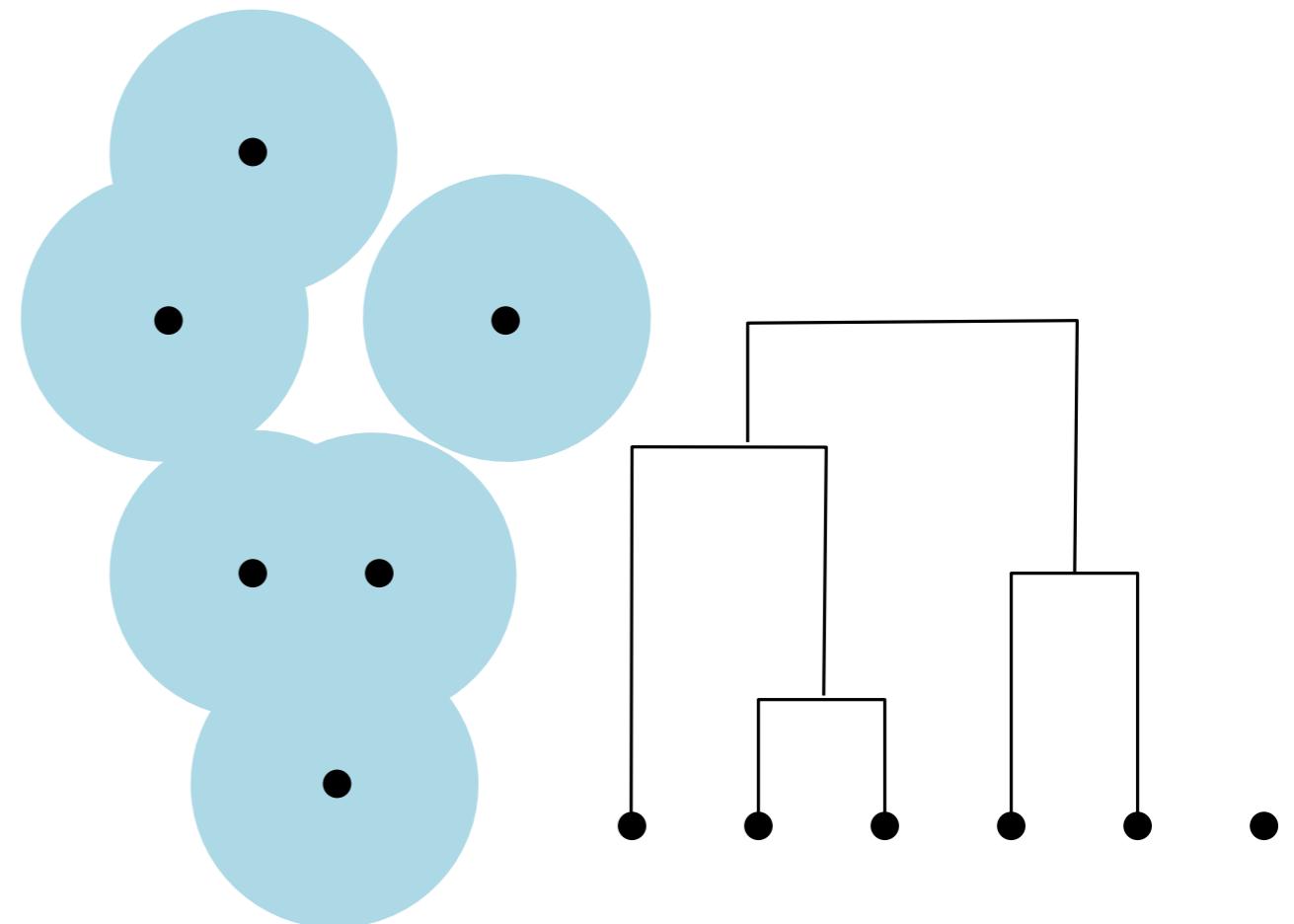
Motivation: the (in)stability of dendograms

Input: A set $X_n = \{x_1, \dots, x_n\}$ in a metric space (X, d) (or just a matrix of pairwise dissimilarities $((d_{i,j}))_{i,j}$).

Given two clusters $C, C' \subseteq X_n$ let $d(C, C') = \inf_{x \in C, x' \in C'} d(x, x')$.

Agglomerative (bottom-up)

1. Start with a clustering where each x_i is a cluster.
2. At each step, merge the two closest clusters until it remains a single cluster (containing all data points).



Output: the resulting dendrogram.

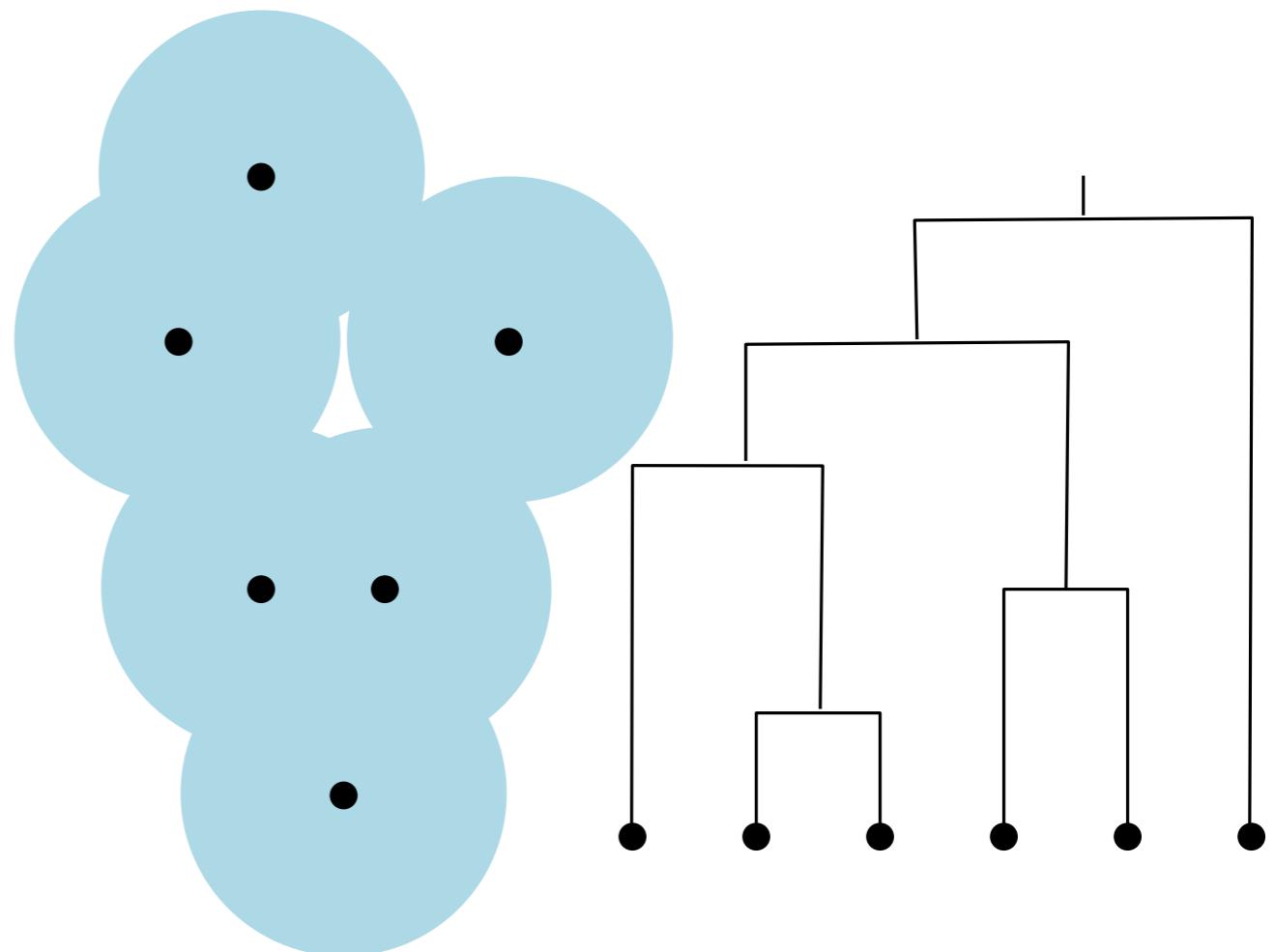
Motivation: the (in)stability of dendrograms

Input: A set $X_n = \{x_1, \dots, x_n\}$ in a metric space (X, d) (or just a matrix of pairwise dissimilarities $((d_{i,j}))_{i,j}$).

Given two clusters $C, C' \subseteq X_n$ let $d(C, C') = \inf_{x \in C, x' \in C'} d(x, x')$.

Agglomerative (bottom-up)

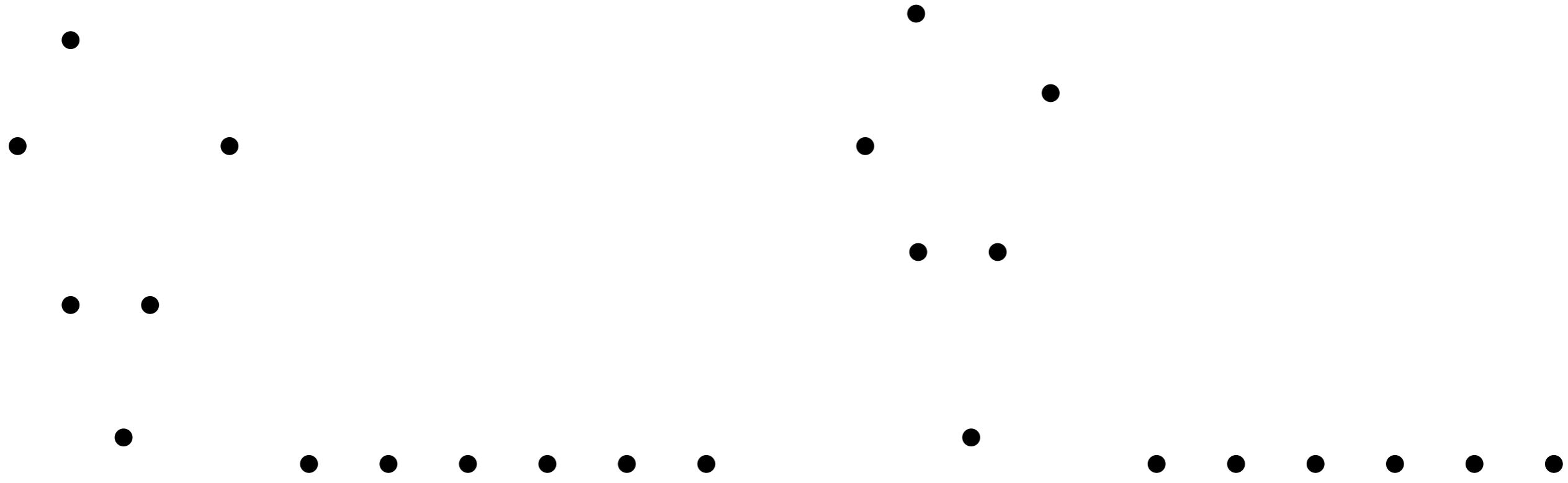
1. Start with a clustering where each x_i is a cluster.
2. At each step, merge the two closest clusters until it remains a single cluster (containing all data points).



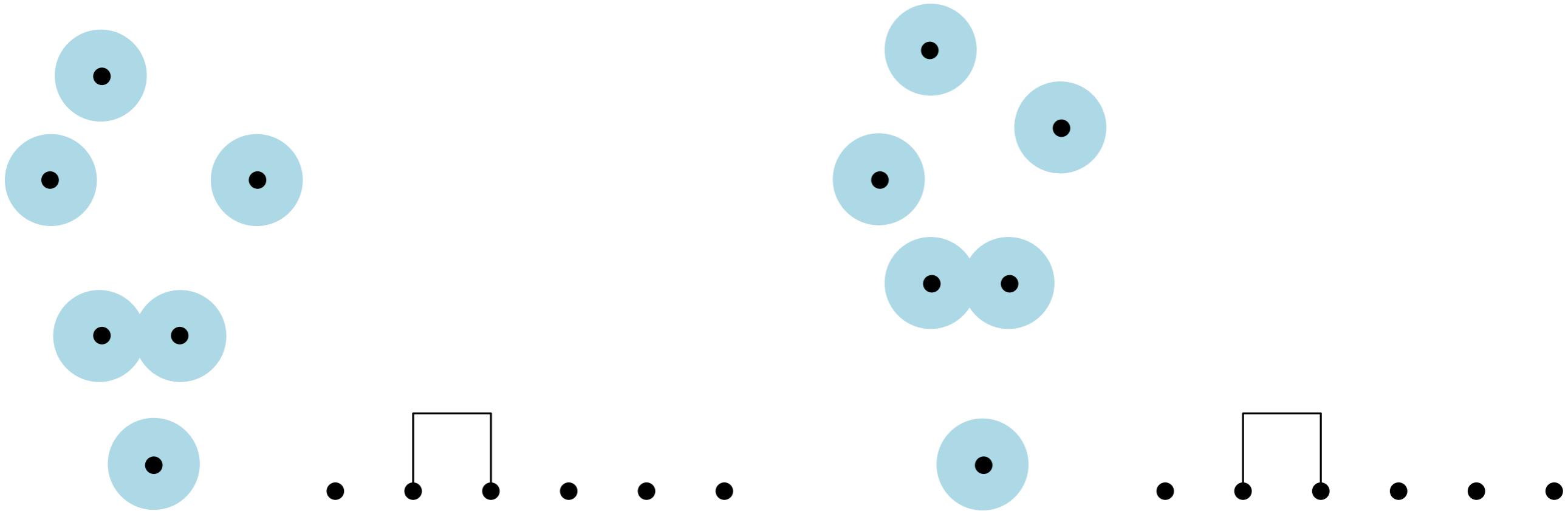
Output: the resulting dendrogram.

Motivation: the (in)stability of dendrograms

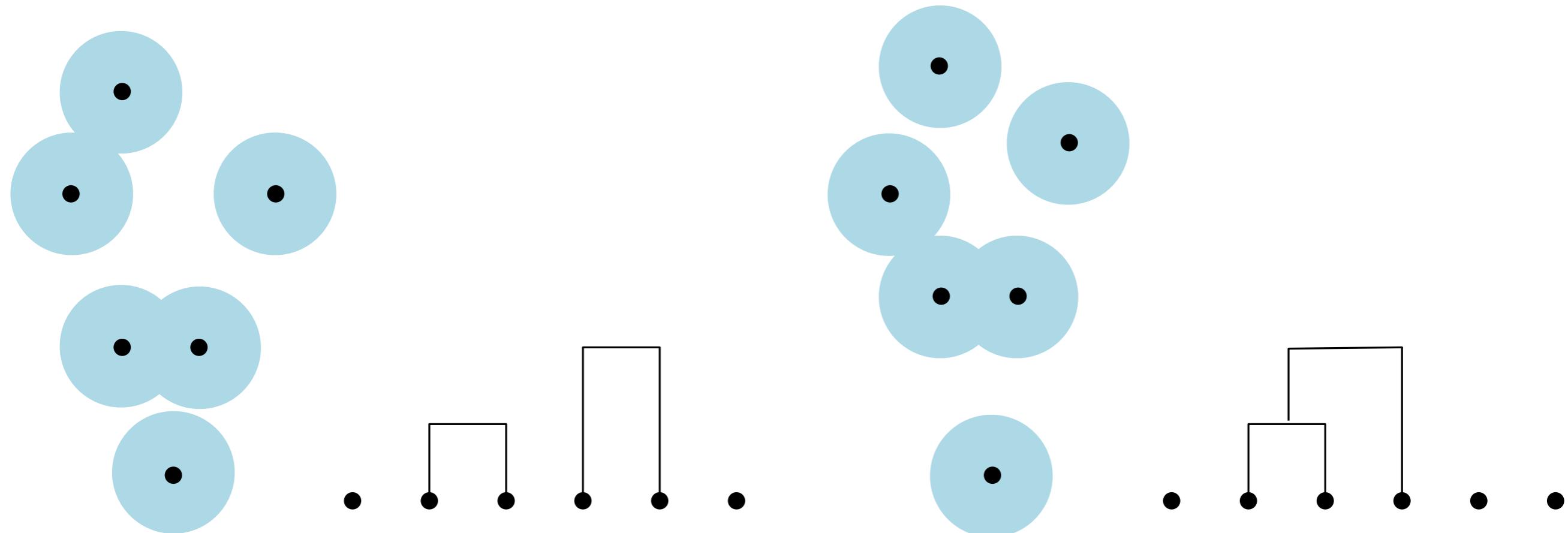
Motivation: the (in)stability of dendrograms



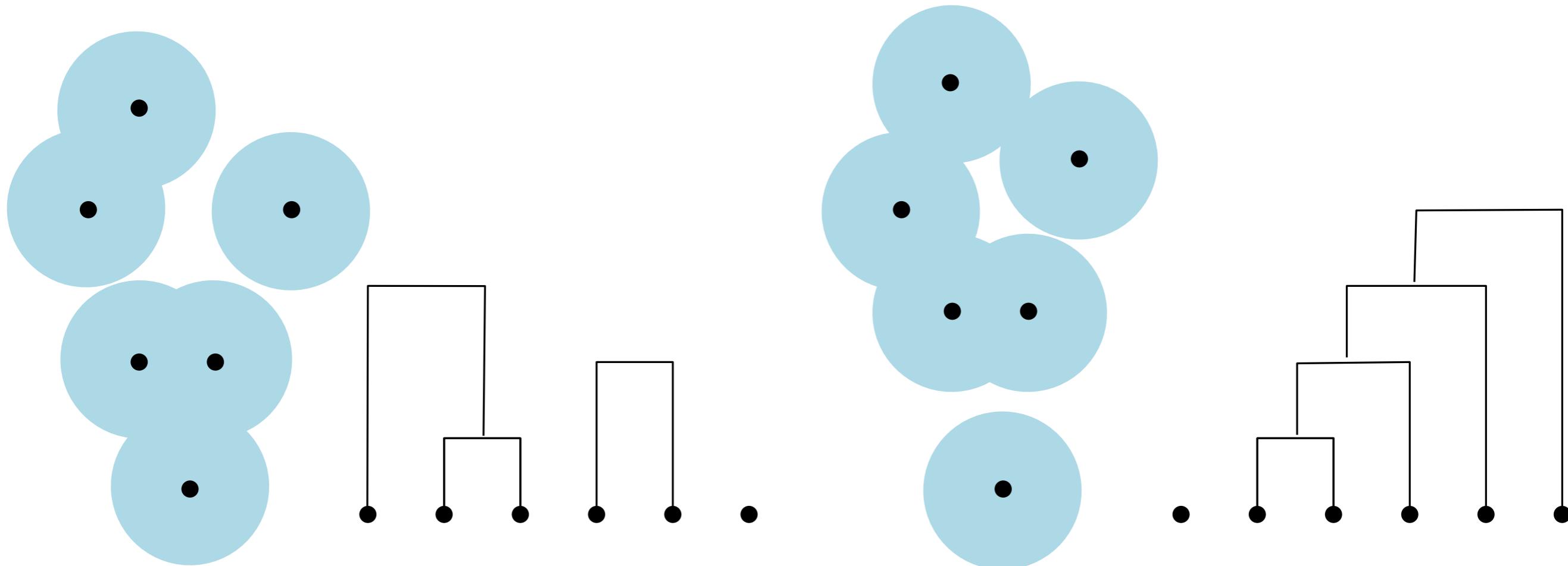
Motivation: the (in)stability of dendograms



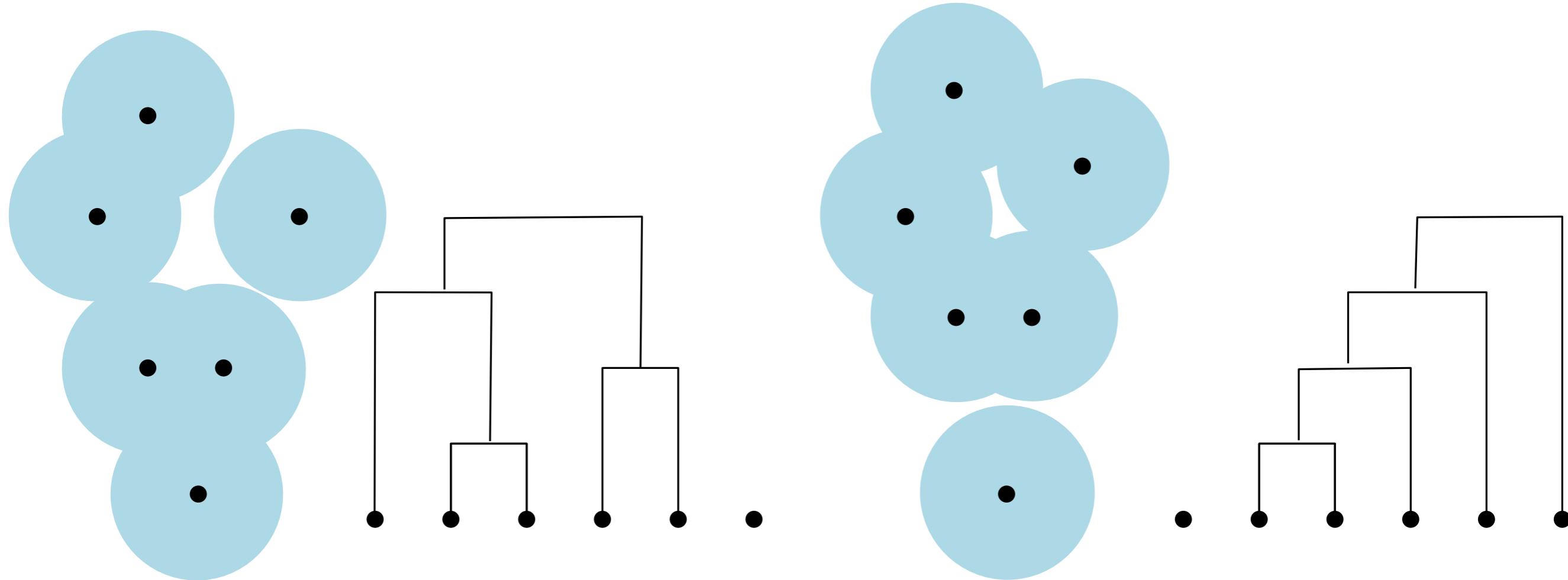
Motivation: the (in)stability of dendograms



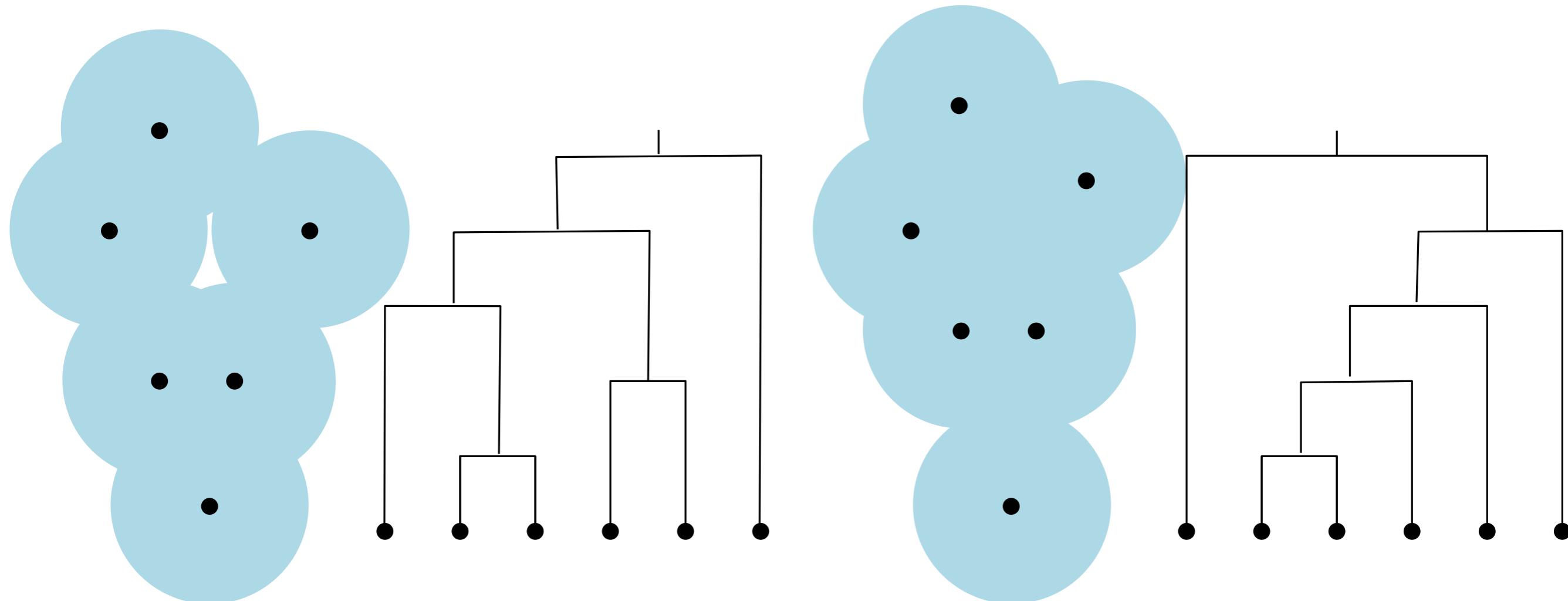
Motivation: the (in)stability of dendograms



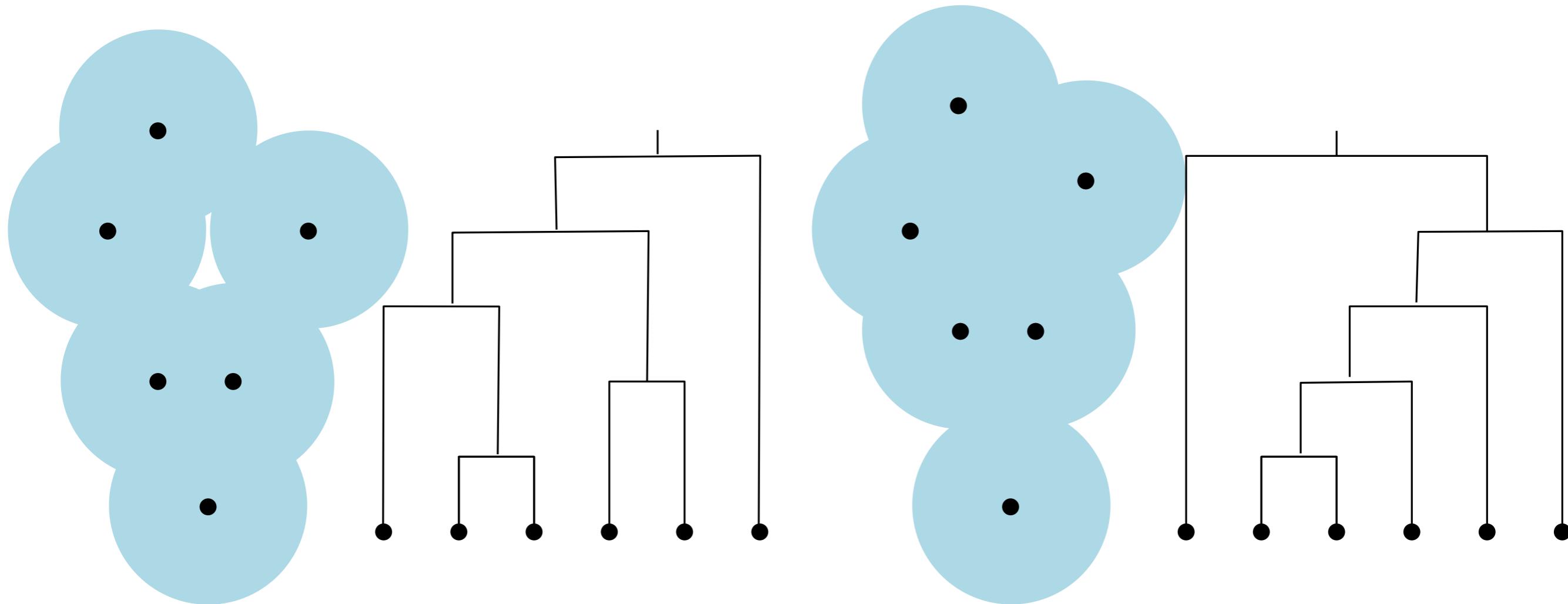
Motivation: the (in)stability of dendograms



Motivation: the (in)stability of dendograms

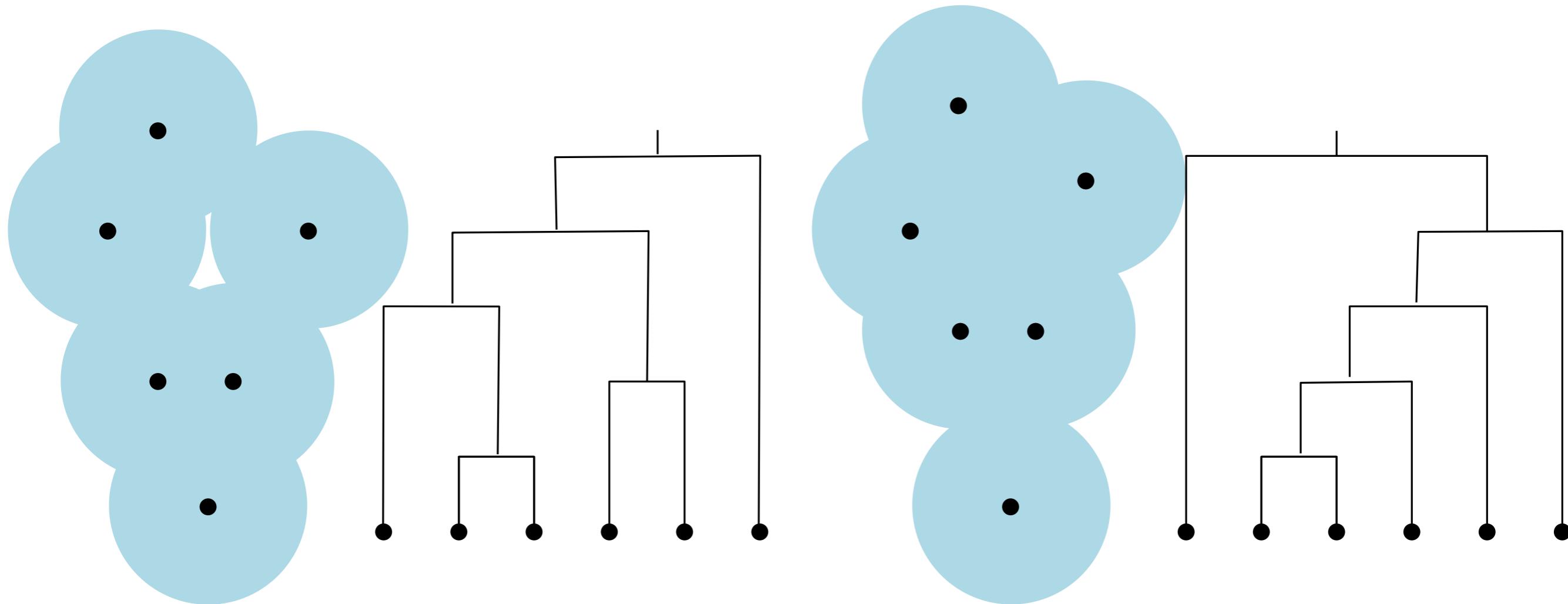


Motivation: the (in)stability of dendrograms



Small perturbations on the input data can induce wide changes in the structure of the output dendograms. However, the merging times (height of dendrogram nodes) remain stable.

Motivation: the (in)stability of dendograms



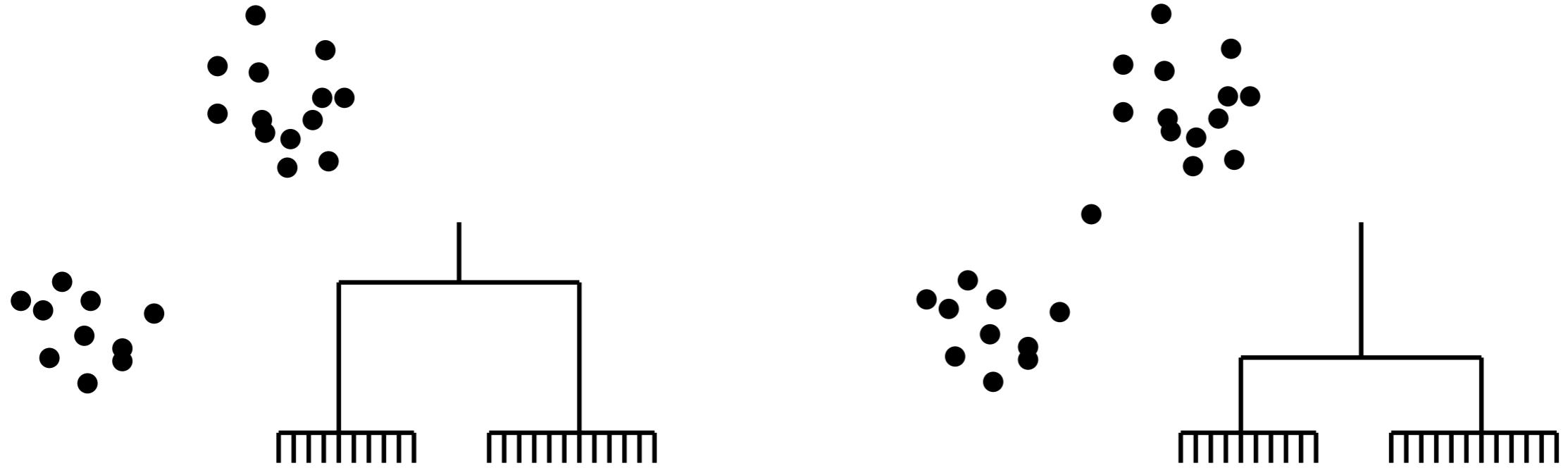
Small perturbations on the input data can induce wide changes in the structure of the output dendograms. However, the merging times (height of dendrogram nodes) remain stable.

Moreover, single linkage clustering keeps track of the evolution of the connected components of the distance function to the data (for Euclidean data).

Motivation: the (in)stability of dendograms

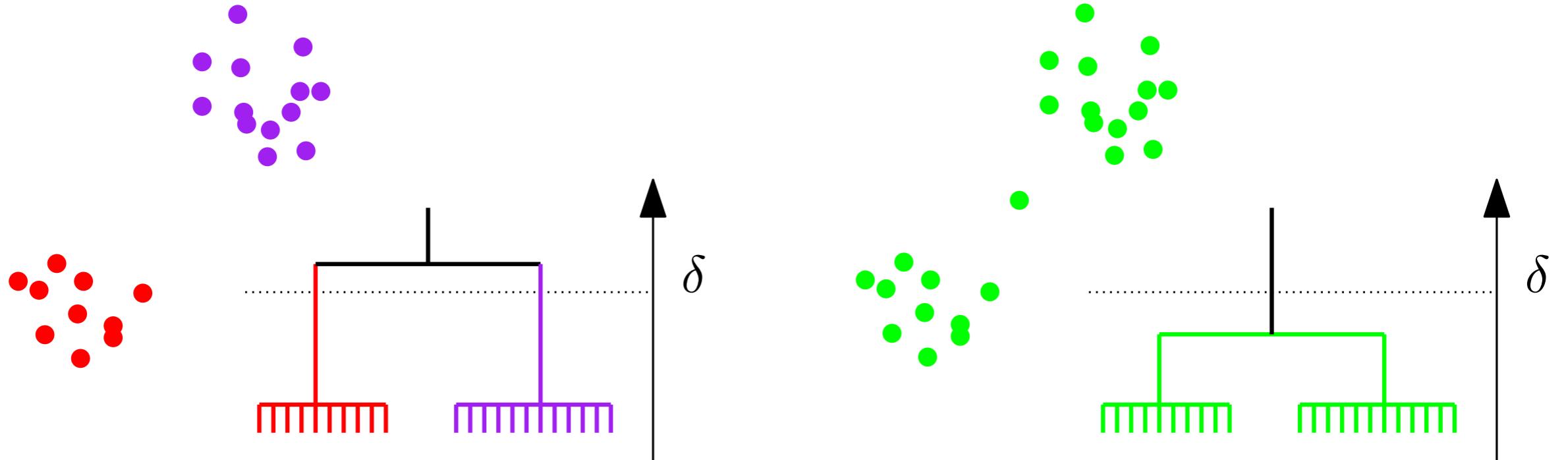
However, building a hierarchy based on spatial proximity is still not a great idea when there are outliers, since there is no stability of merging times anymore.

Motivation: the (in)stability of dendrograms



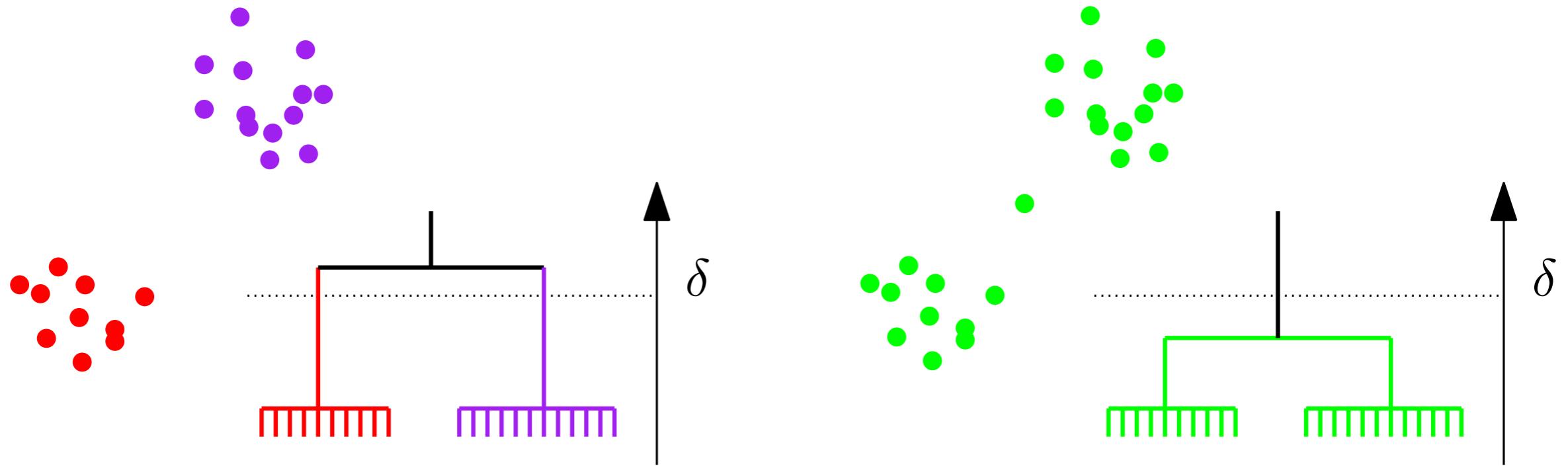
However, building a hierarchy based on spatial proximity is still not a great idea when there are outliers, since there is no stability of merging times anymore.

Motivation: the (in)stability of dendrograms



However, building a hierarchy based on spatial proximity is still not a great idea when there are outliers, since there is no stability of merging times anymore.

Motivation: the (in)stability of dendrograms

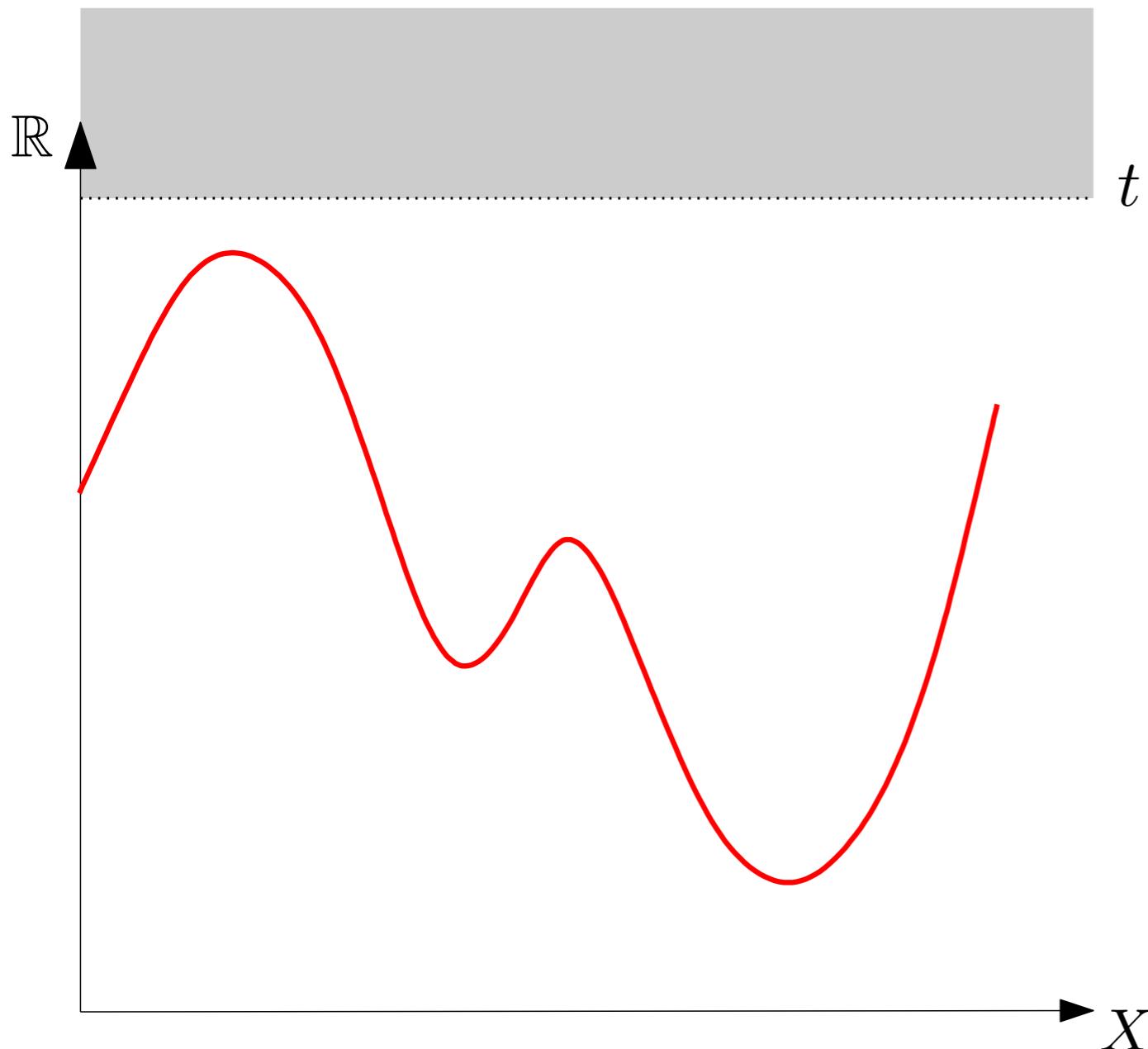


However, building a hierarchy based on spatial proximity is still not a great idea when there are outliers, since there is no stability of merging times anymore.

Another way to build a hierarchy is with the sublevel sets of a **filter function**. For instance, using density as filter is at the core of mode-seeking algorithms.

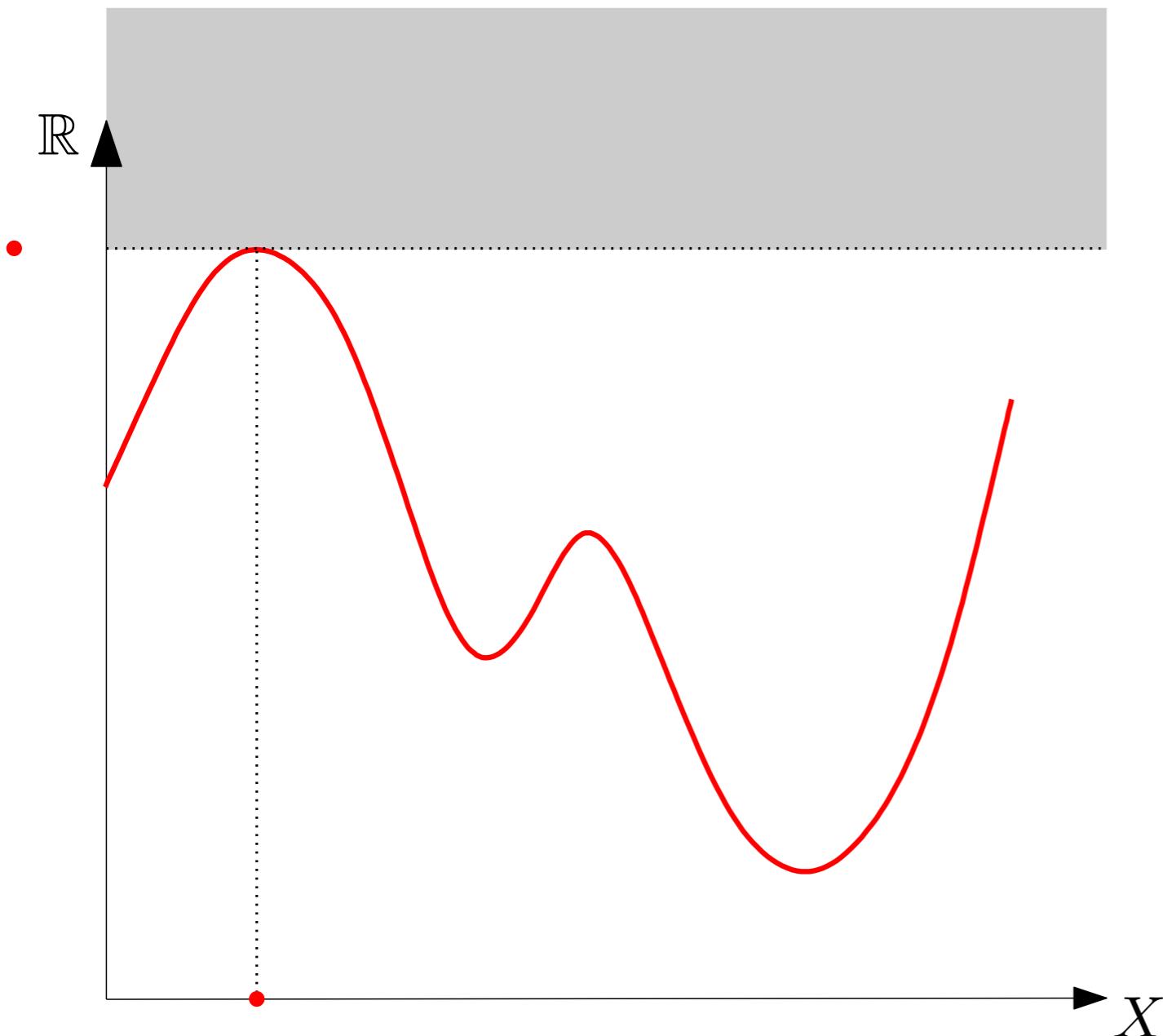
0-dimensional PH of function

Given a continuous function $f : X \rightarrow \mathbb{R}$, we will consider the *superlevel-set filtration* $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, and track the **appearance** and **merging** of connected components / clusters in the process.



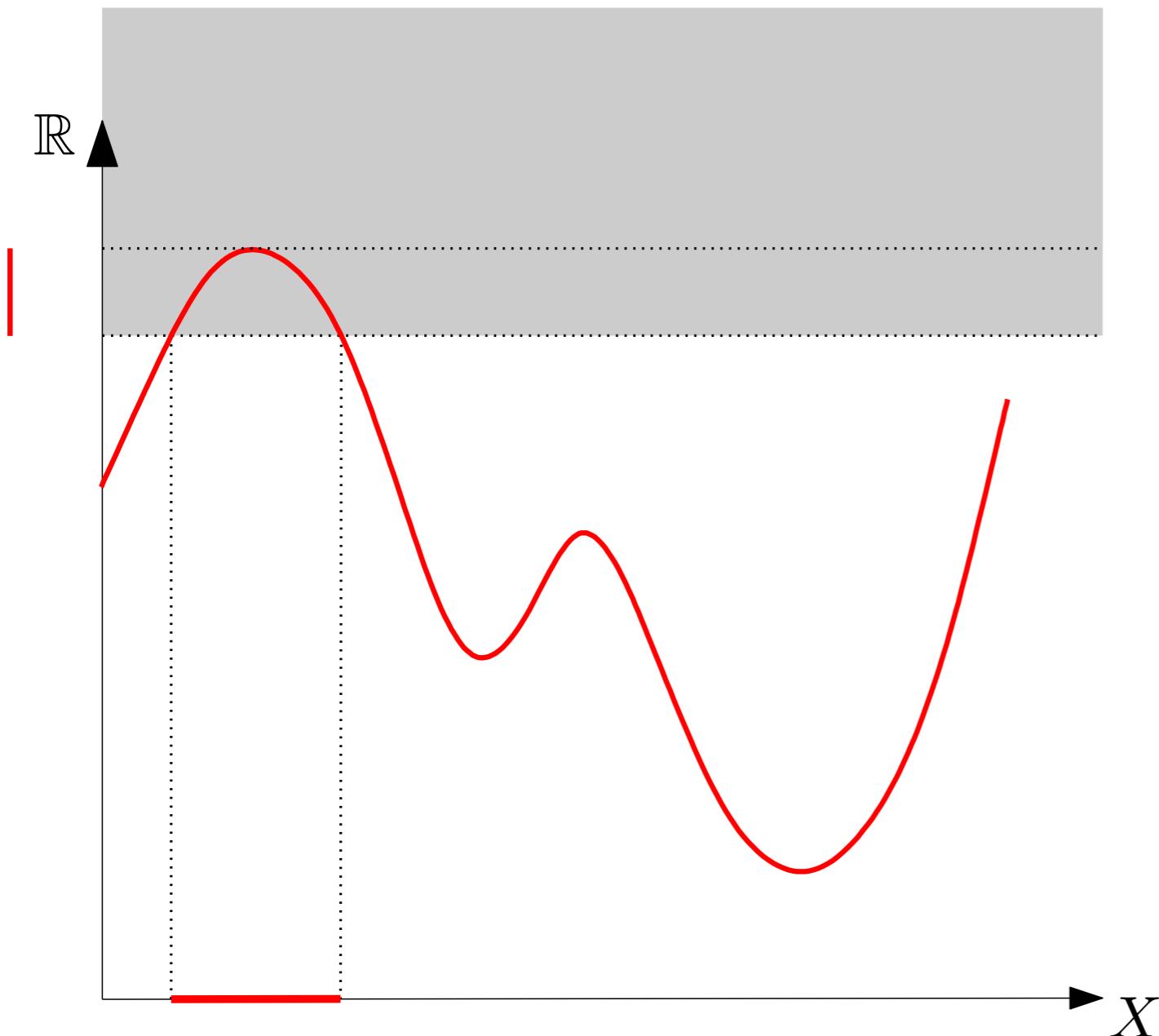
0-dimensional PH of function

Given a continuous function $f : X \rightarrow \mathbb{R}$, we will consider the *superlevel-set filtration* $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, and track the **appearance** and **merging** of connected components / clusters in the process.



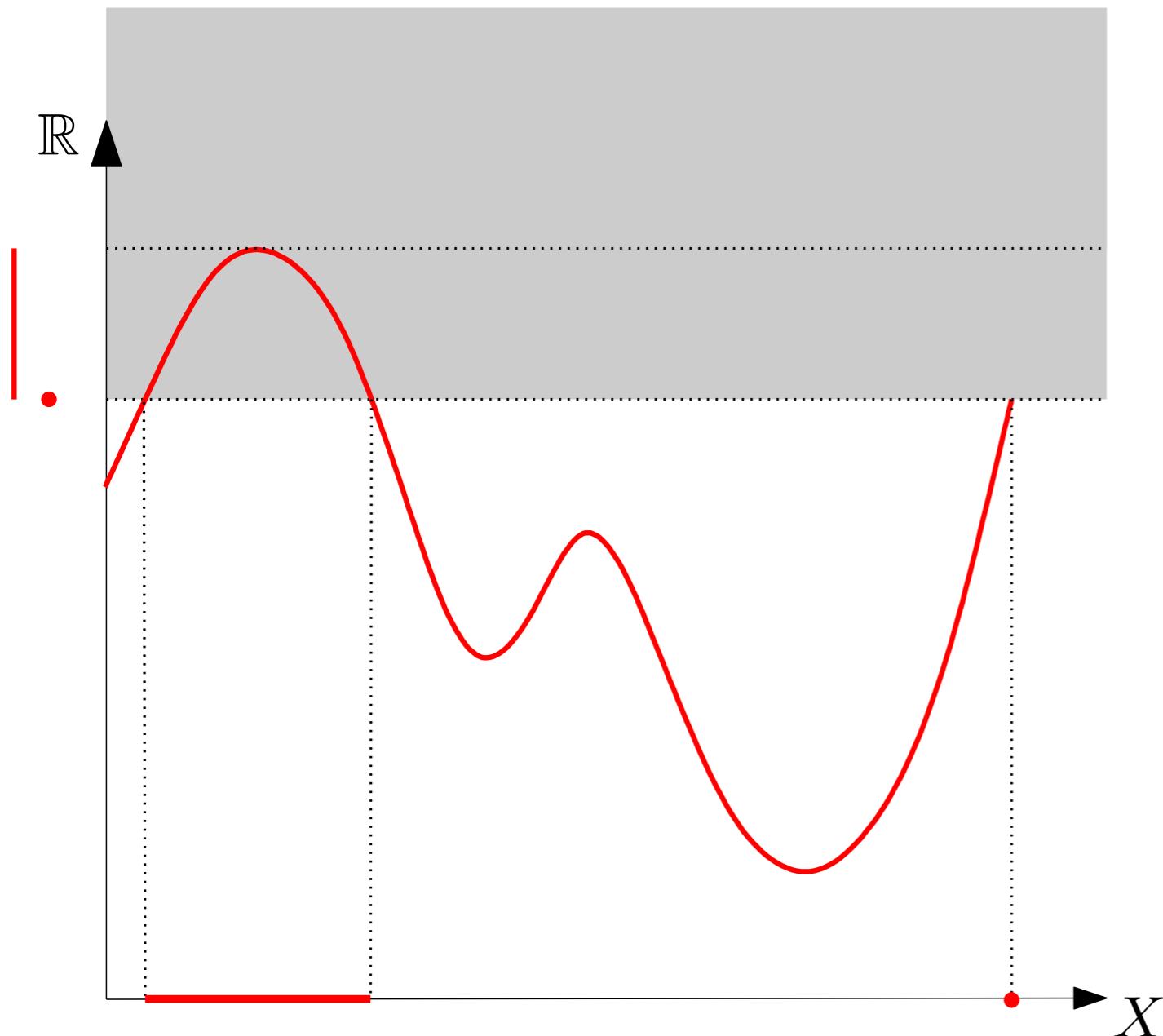
0-dimensional PH of function

Given a continuous function $f : X \rightarrow \mathbb{R}$, we will consider the *superlevel-set filtration* $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, and track the **appearance** and **merging** of connected components / clusters in the process.



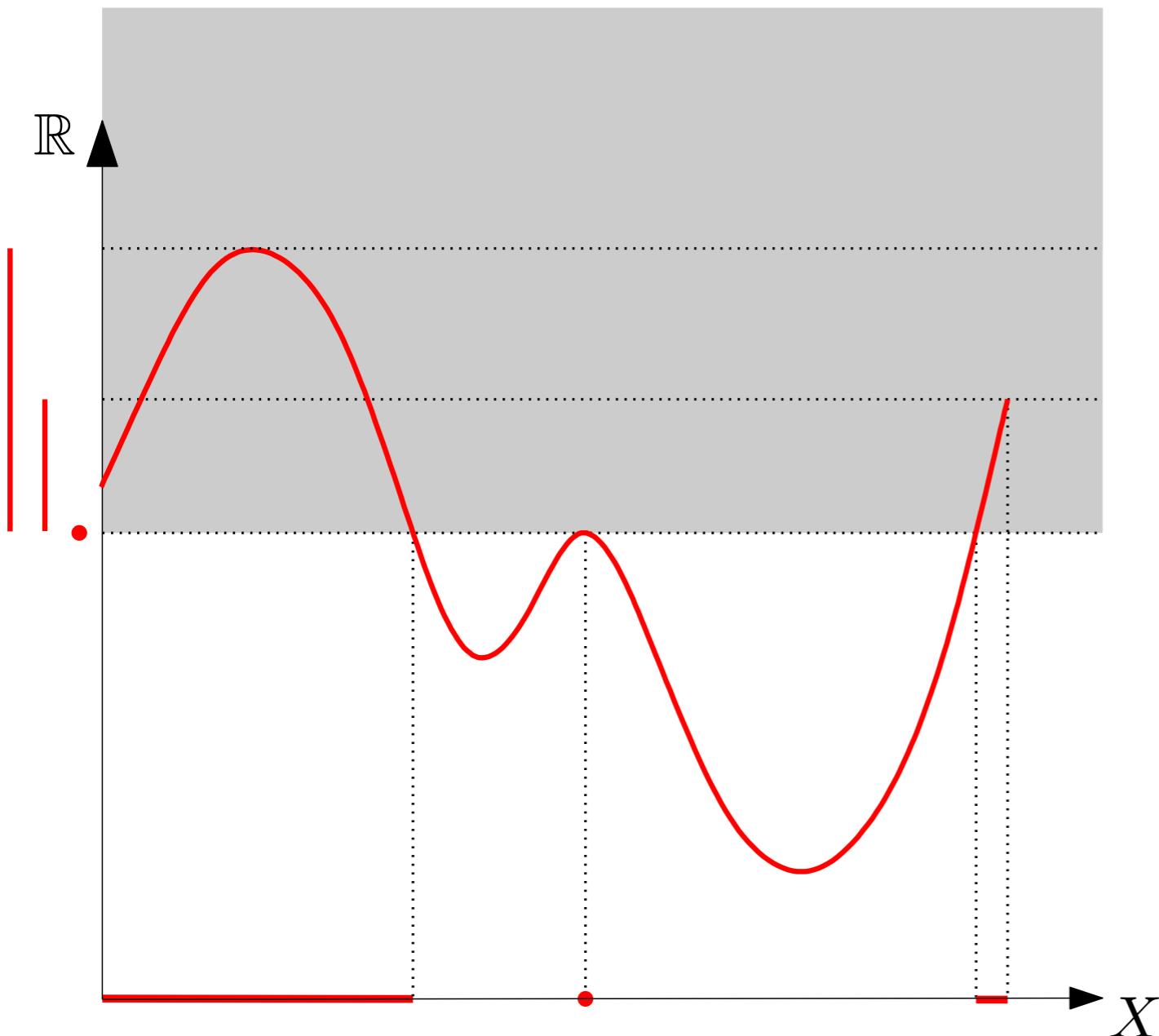
0-dimensional PH of function

Given a continuous function $f : X \rightarrow \mathbb{R}$, we will consider the *superlevel-set filtration* $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, and track the **appearance** and **merging** of connected components / clusters in the process.



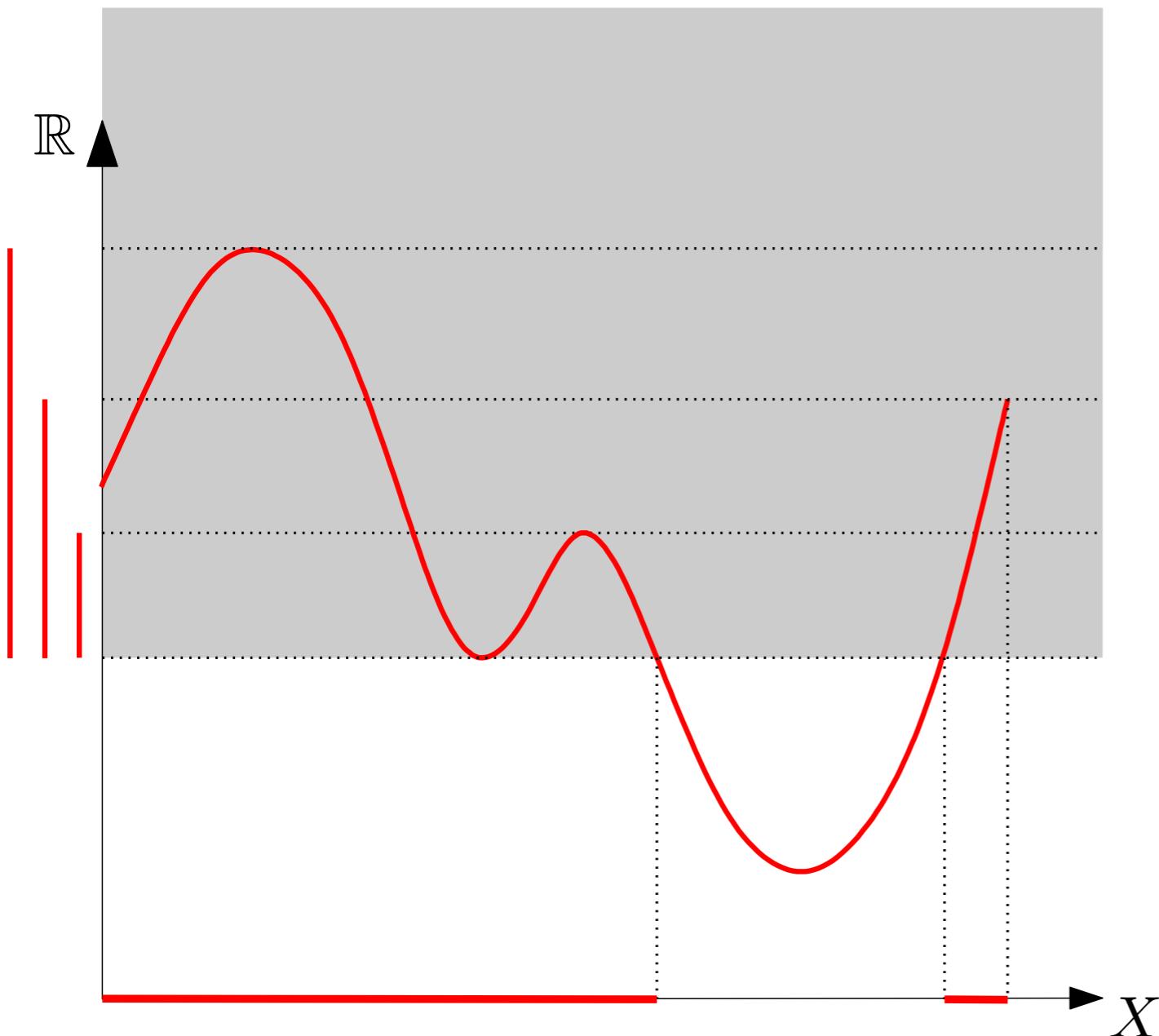
0-dimensional PH of function

Given a continuous function $f : X \rightarrow \mathbb{R}$, we will consider the *superlevel-set filtration* $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, and track the **appearance** and **merging** of connected components / clusters in the process.



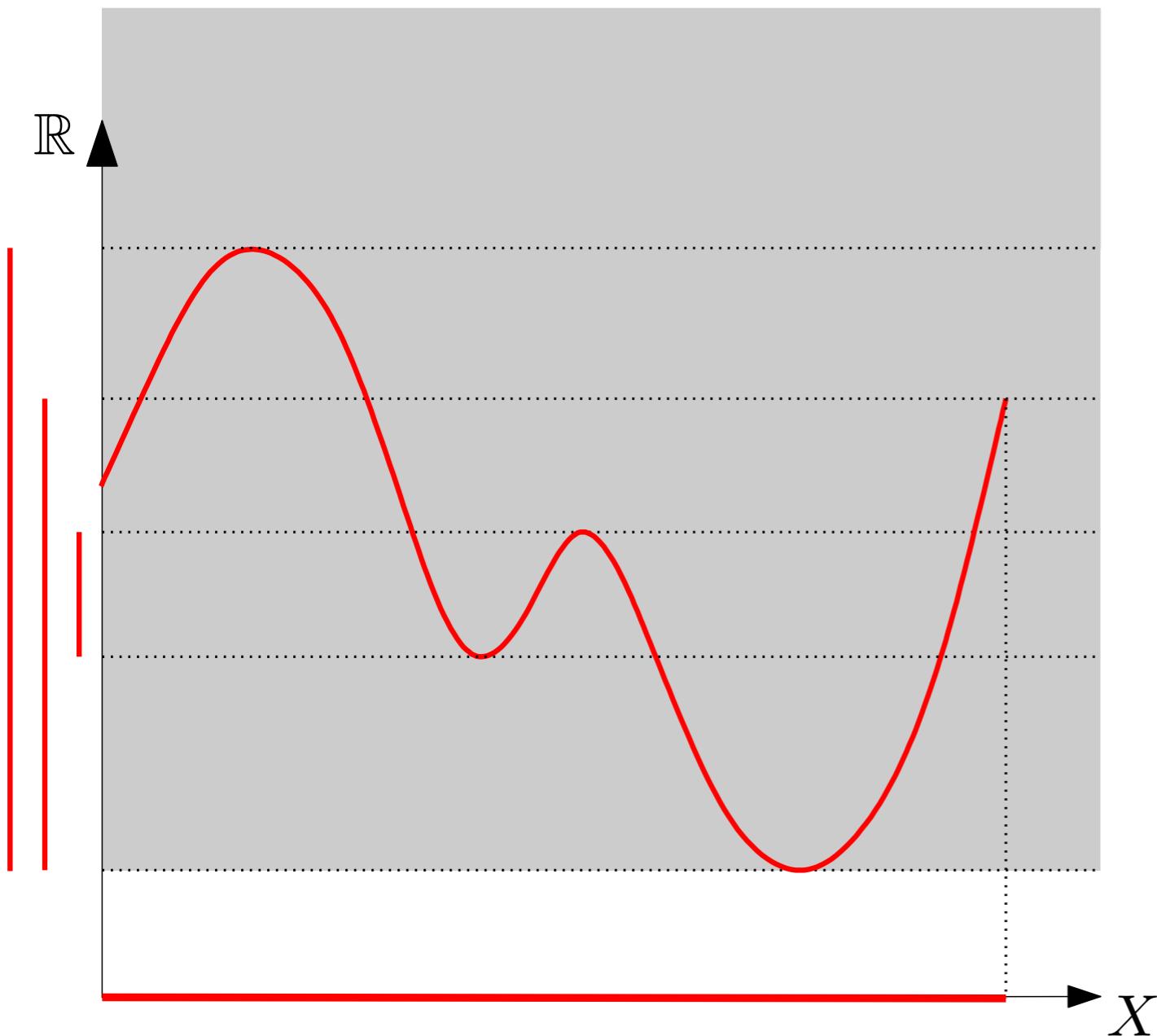
0-dimensional PH of function

Given a continuous function $f : X \rightarrow \mathbb{R}$, we will consider the *superlevel-set filtration* $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, and track the **appearance** and **merging** of connected components / clusters in the process.



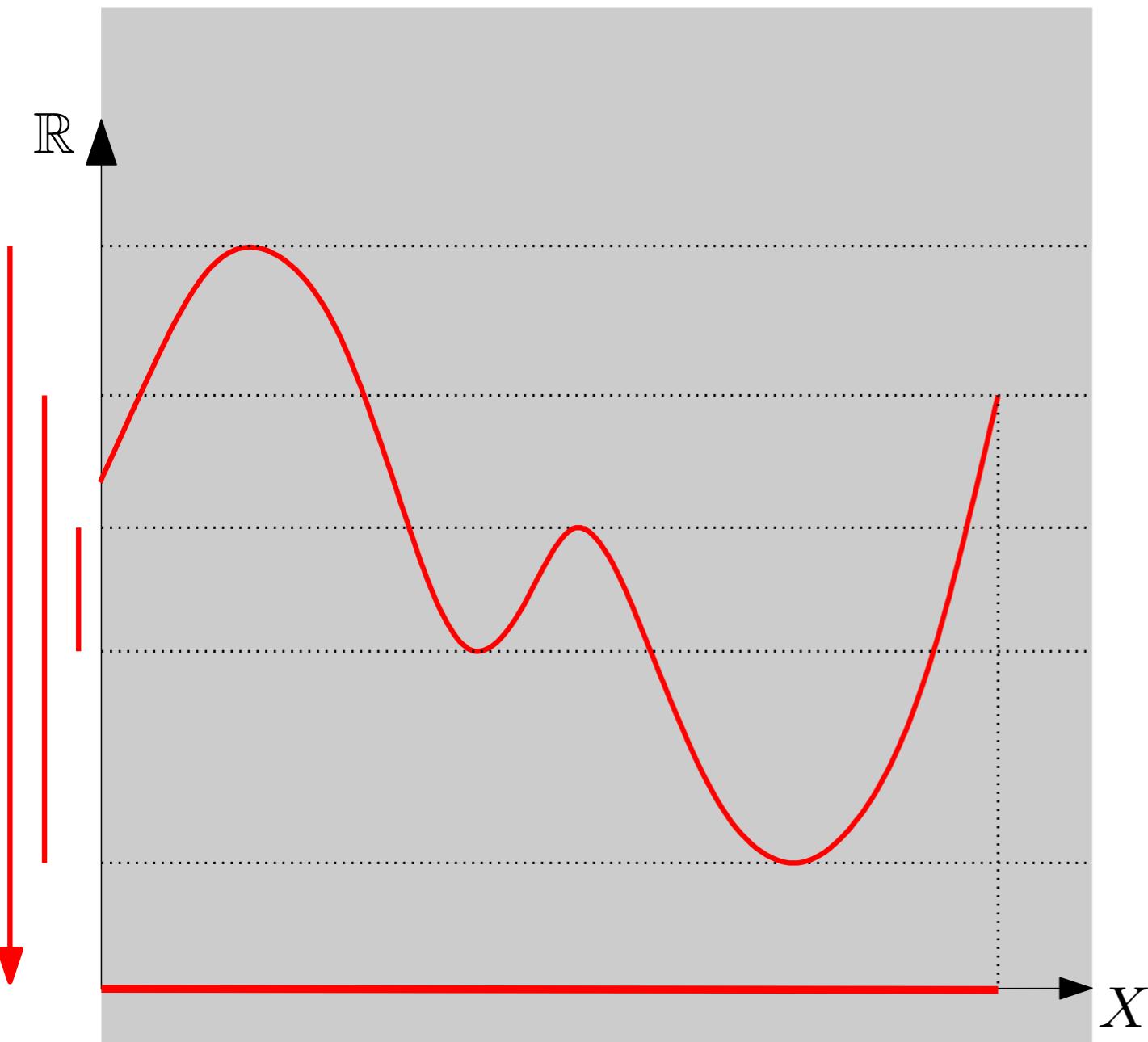
0-dimensional PH of function

Given a continuous function $f : X \rightarrow \mathbb{R}$, we will consider the *superlevel-set filtration* $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, and track the **appearance** and **merging** of connected components / clusters in the process.



0-dimensional PH of function

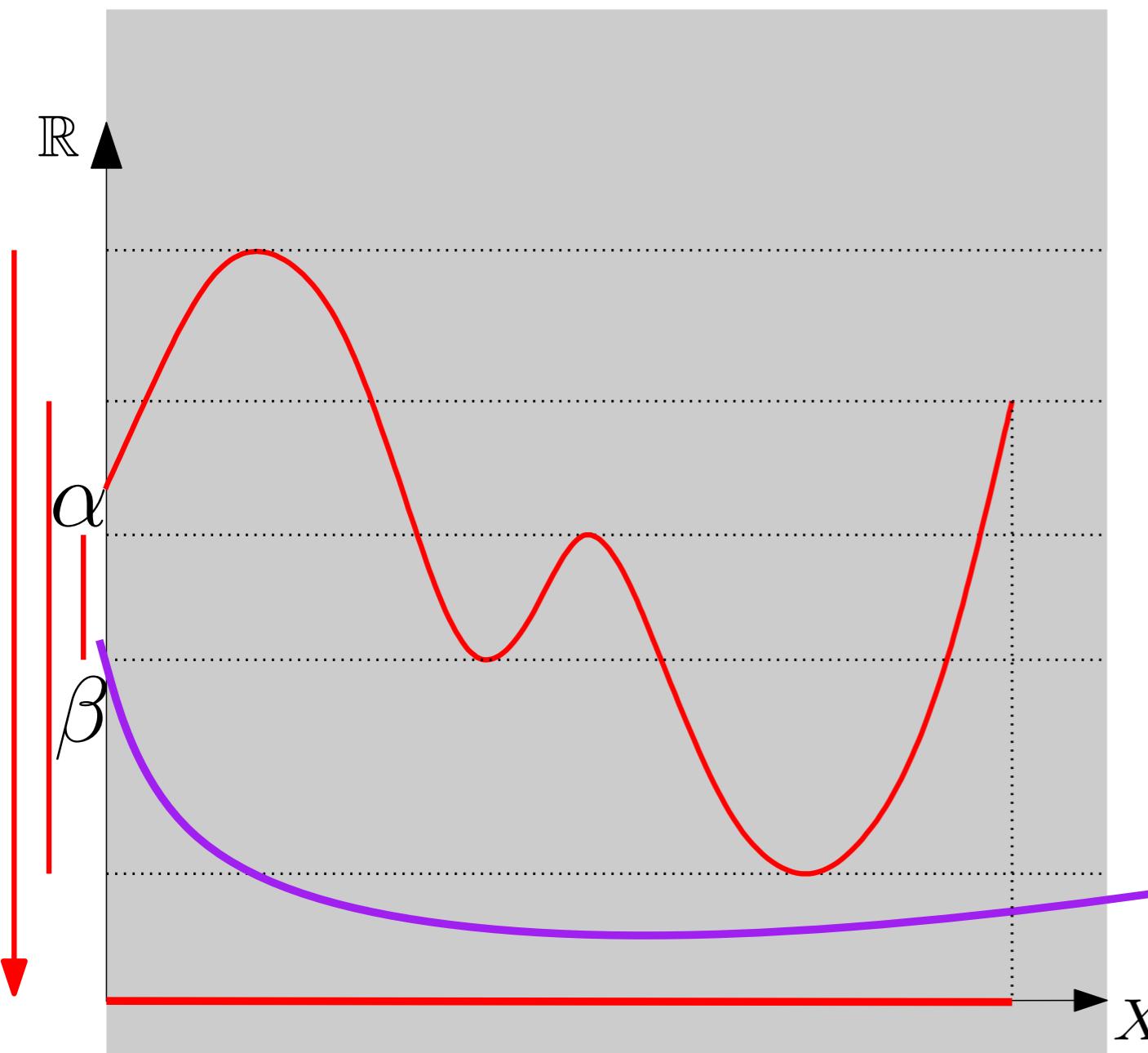
Given a continuous function $f : X \rightarrow \mathbb{R}$, we will consider the *superlevel-set filtration* $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, and track the **appearance** and **merging** of connected components / clusters in the process.



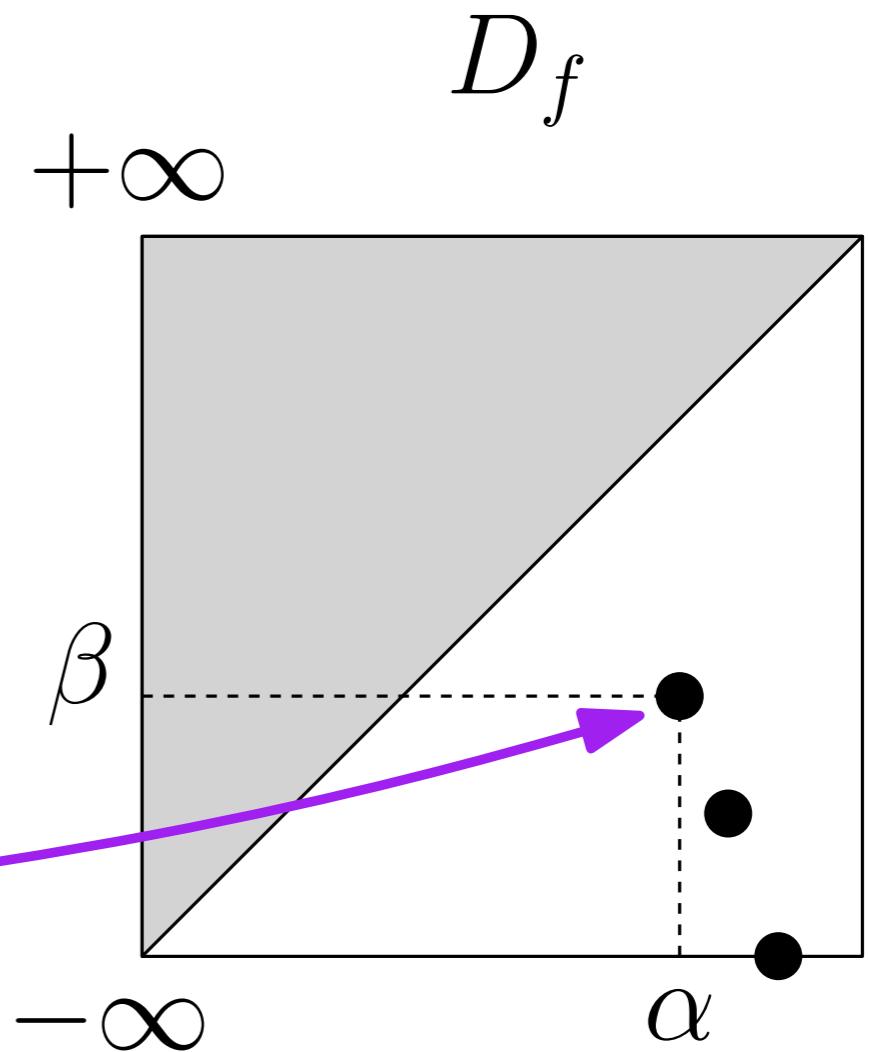
0-dimensional PH of function

Given a continuous function $f : X \rightarrow \mathbb{R}$, we will consider the *superlevel-set filtration* $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, and track the **appearance** and **merging** of connected components / clusters in the process.

Persistence barcode



Persistence diagram



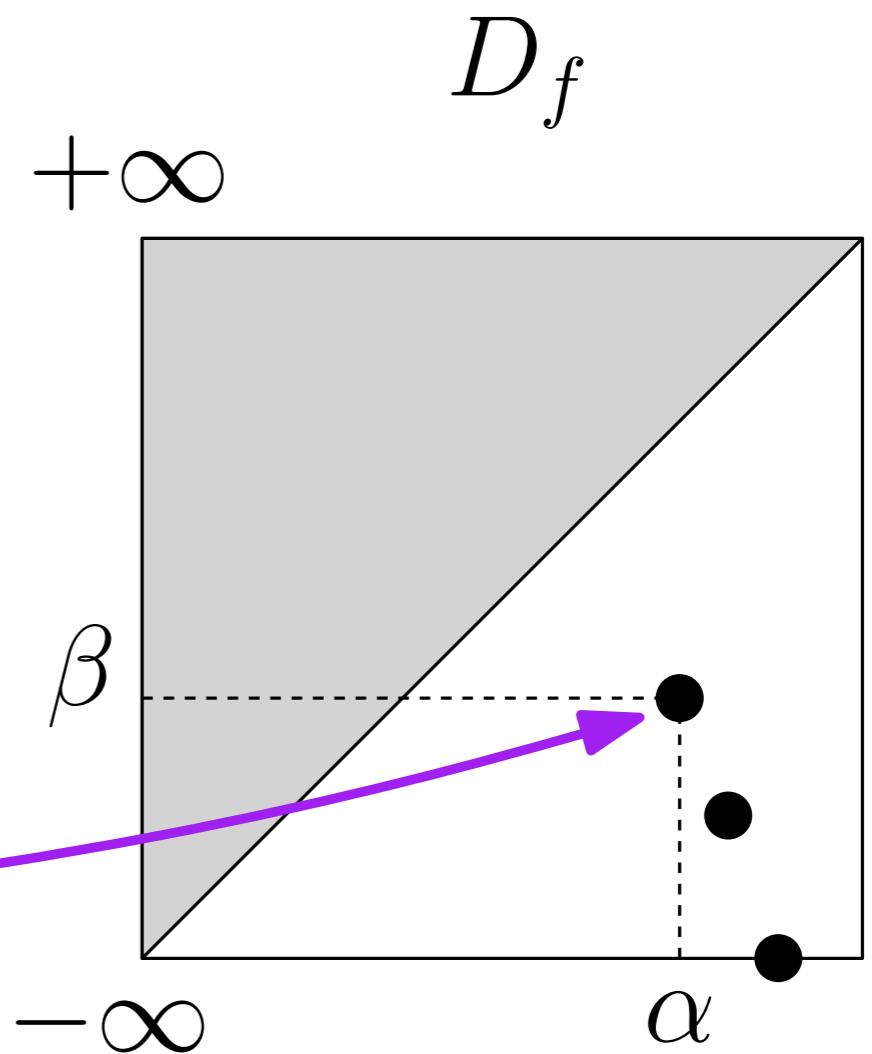
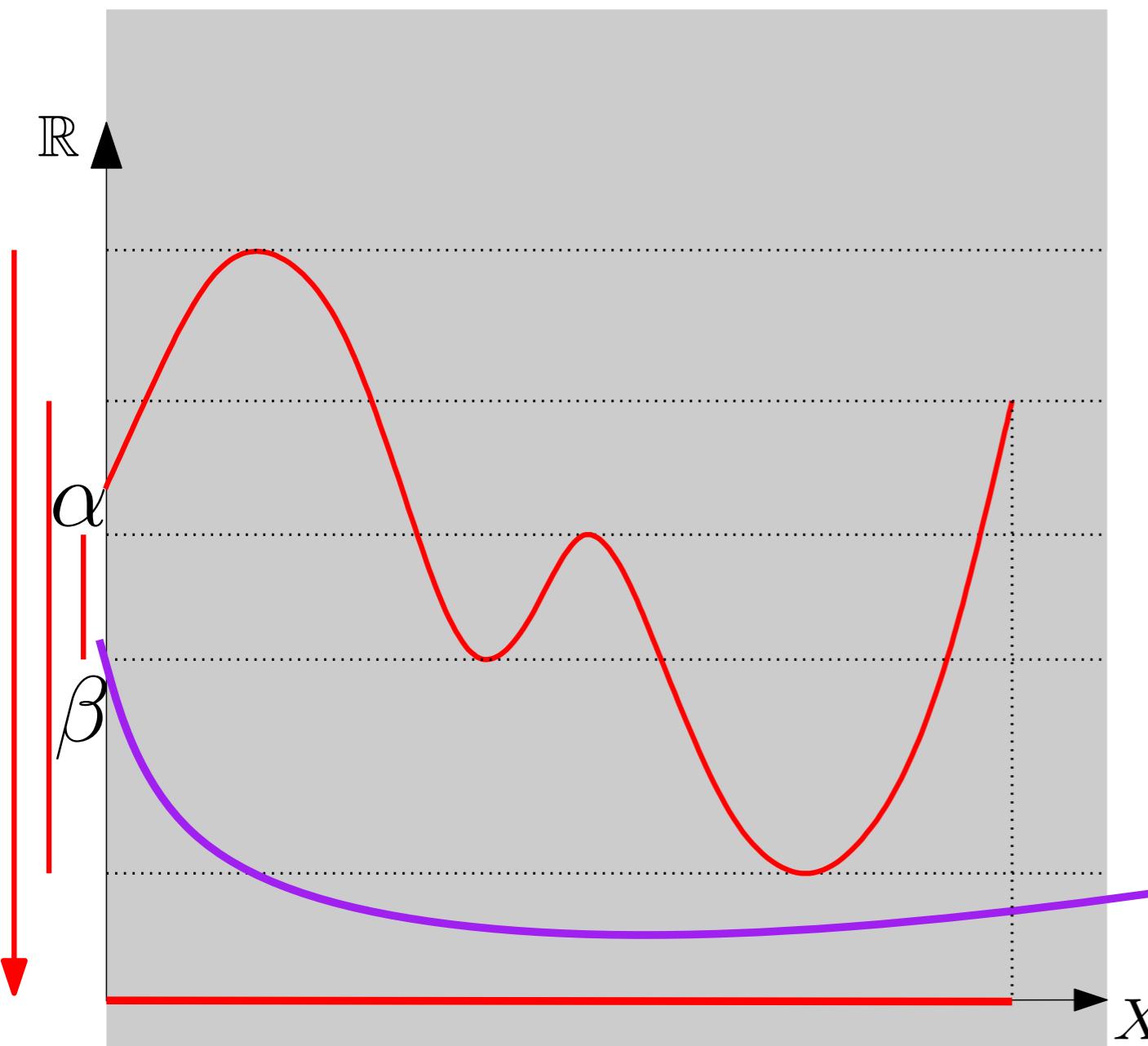
0-dimensional PH of function

Given a continuous function $f : X \rightarrow \mathbb{R}$, we will consider the *superlevel-set filtration* $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$, and track the **appearance** and **merging** of connected components / clusters in the process.

Persistence barcode

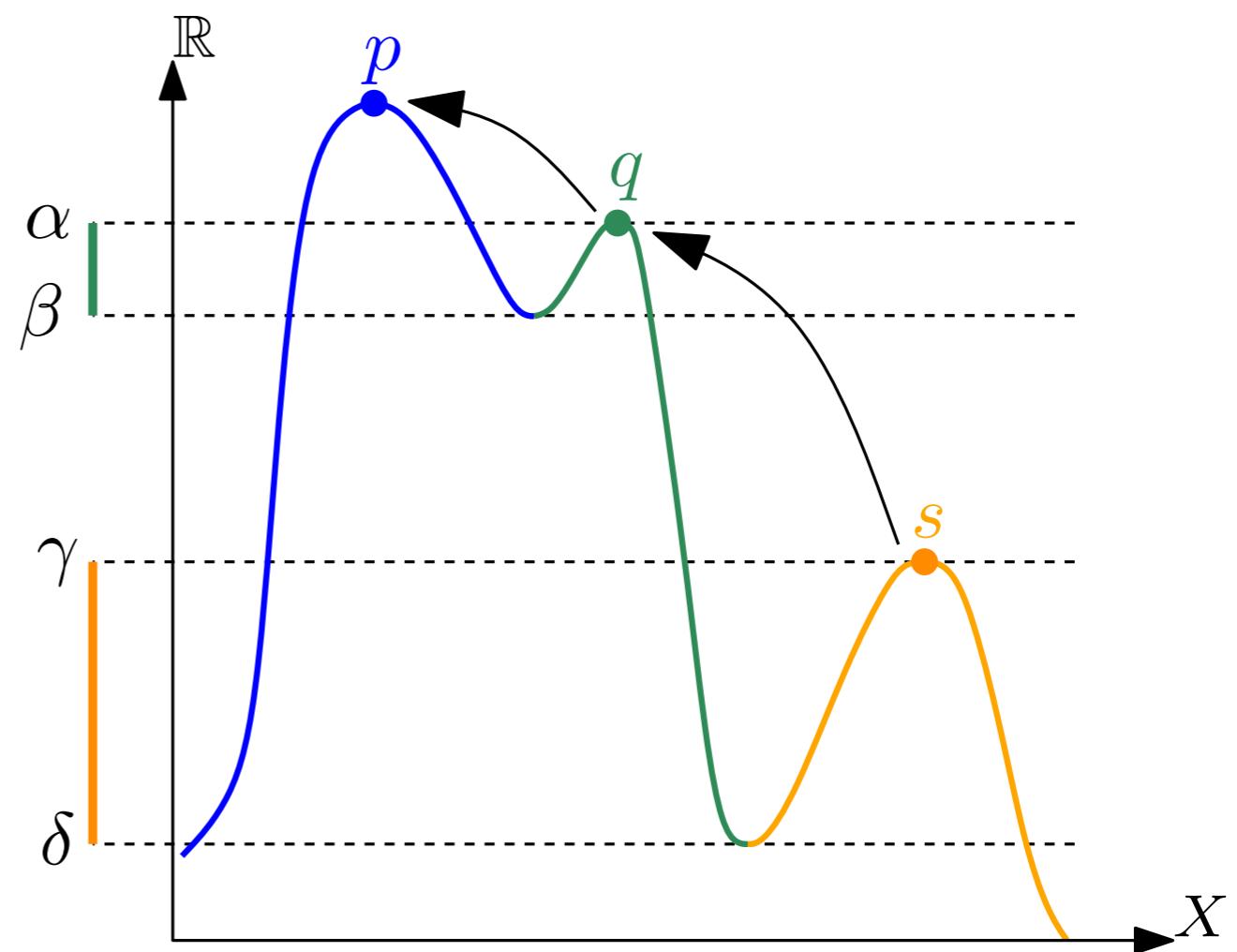
requires some sort of **connectivity**,
e.g., neighborhood graph

Persistence diagram



Building a hierarchy of clusters

Moreover, 0-dimensional PH also remembers the connected components / clusters that were merged together during the filtration process and builds a hierarchy out of this information.



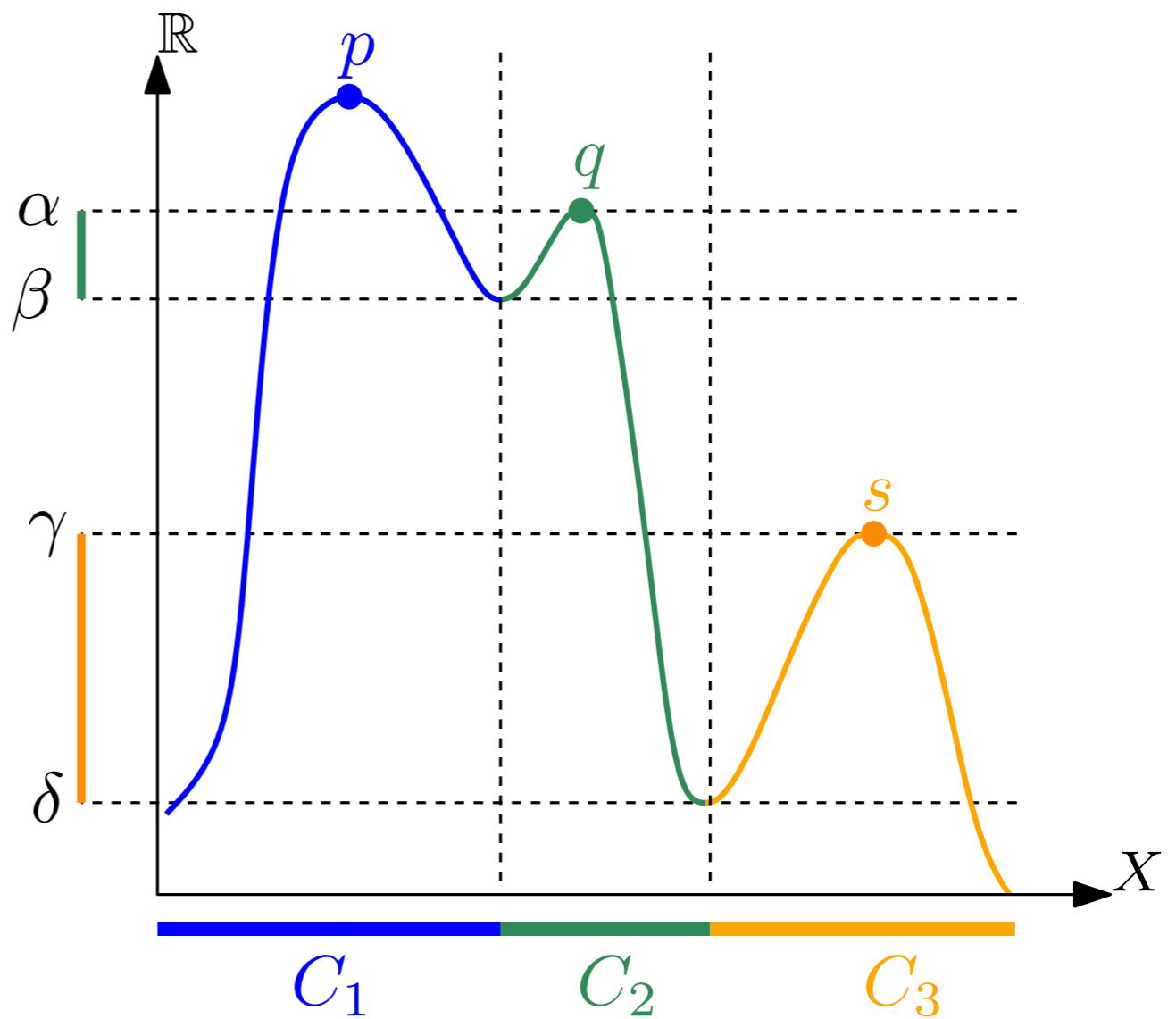
Building a hierarchy of clusters

Moreover, 0-dimensional PH also remembers the connected components / clusters that were merged together during the filtration process and builds a hierarchy out of this information.

This means that, given a fixed threshold $\tau \geq 0$, one can even retrieve the connected components / clusters associated to all the bars of length $> \tau$!

Topological Mode Analysis Tool

$$0 \leq \tau \leq \alpha - \beta$$



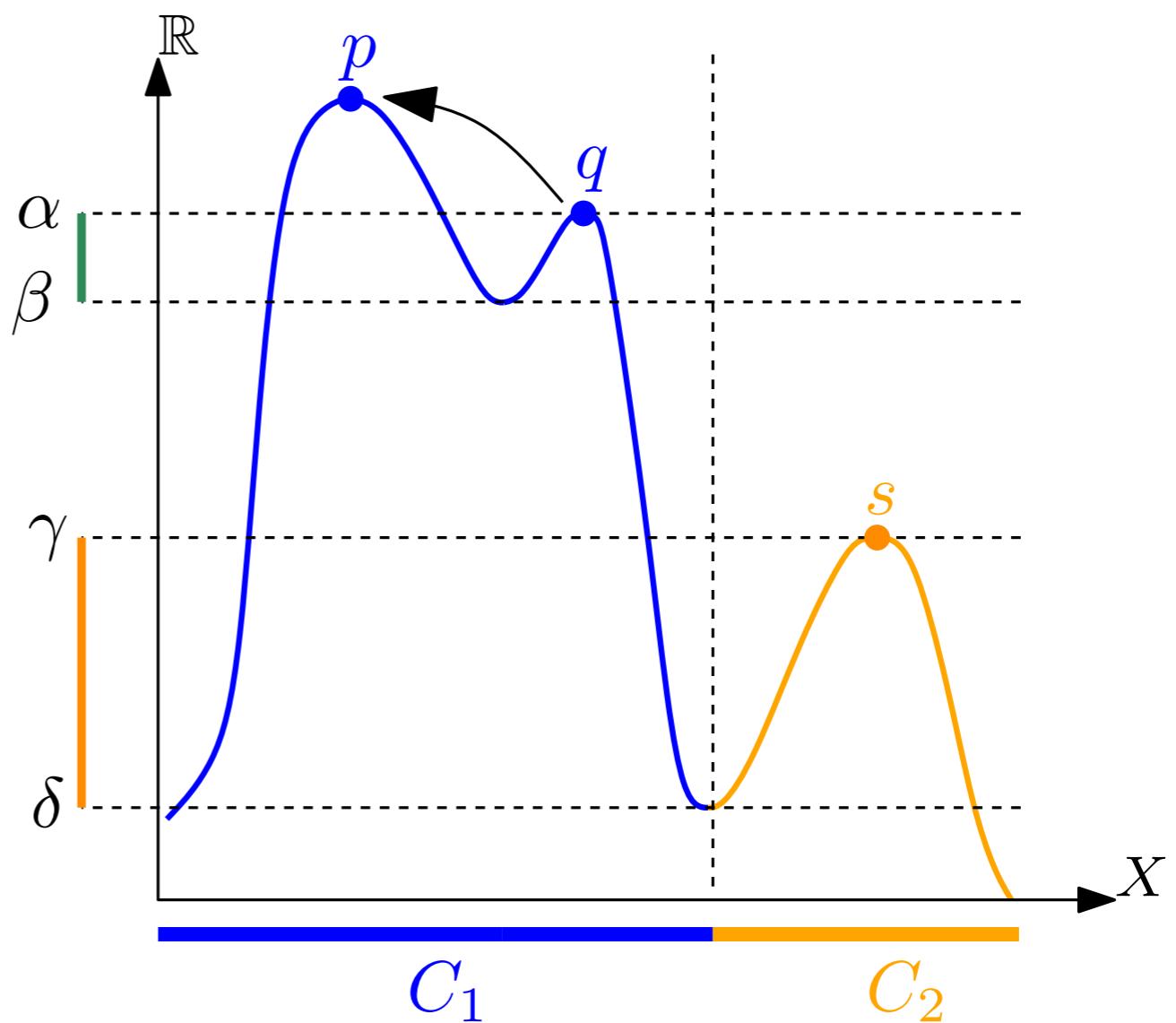
Building a hierarchy of clusters

Moreover, 0-dimensional PH also remembers the connected components / clusters that were merged together during the filtration process and builds a hierarchy out of this information.

This means that, given a fixed threshold $\tau \geq 0$, one can even retrieve the connected components / clusters associated to all the bars of length $> \tau$!

Topological Mode Analysis Tool

$$\alpha - \beta < \tau \leq \gamma - \delta$$



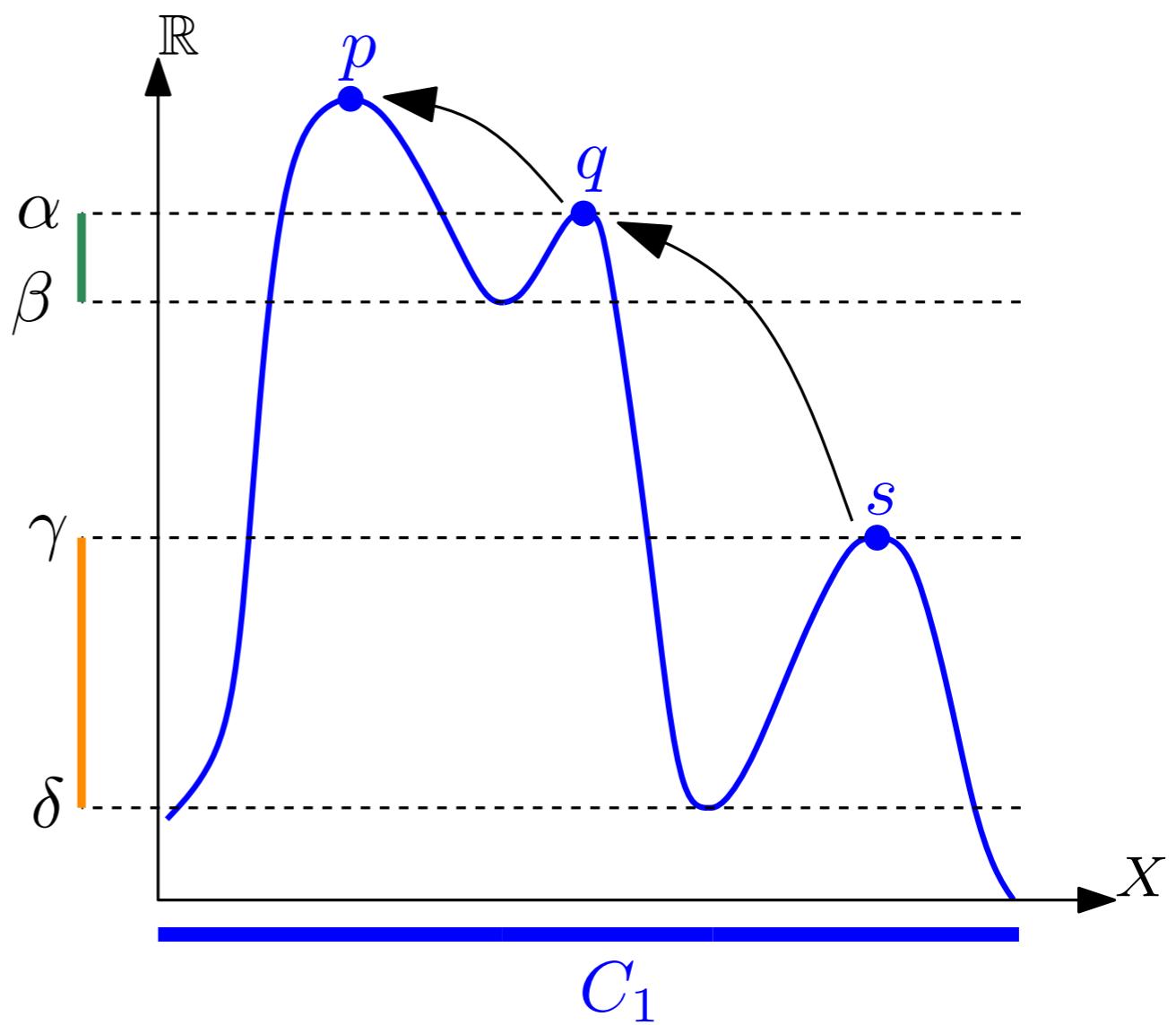
Building a hierarchy of clusters

Moreover, 0-dimensional PH also remembers the connected components / clusters that were merged together during the filtration process and builds a hierarchy out of this information.

This means that, given a fixed threshold $\tau \geq 0$, one can even retrieve the connected components / clusters associated to all the bars of length $> \tau$!

Topological Mode Analysis Tool

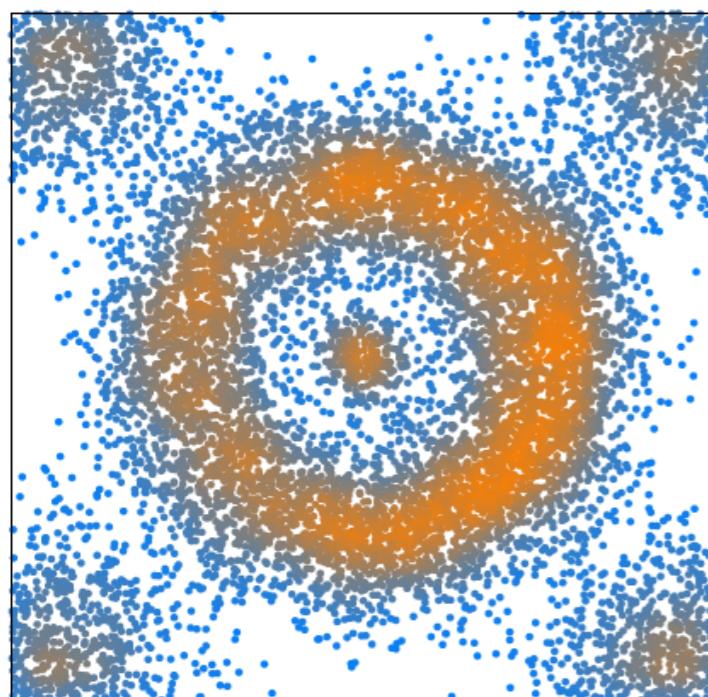
$$\gamma - \delta < \tau \leq +\infty$$



ToMATo: Topological Mode Analysis Tool

[*Persistence-Based Clustering in Riemannian Manifolds*, Chazal, Oudot, Skraba, Guibas, J. ACM, 2013]

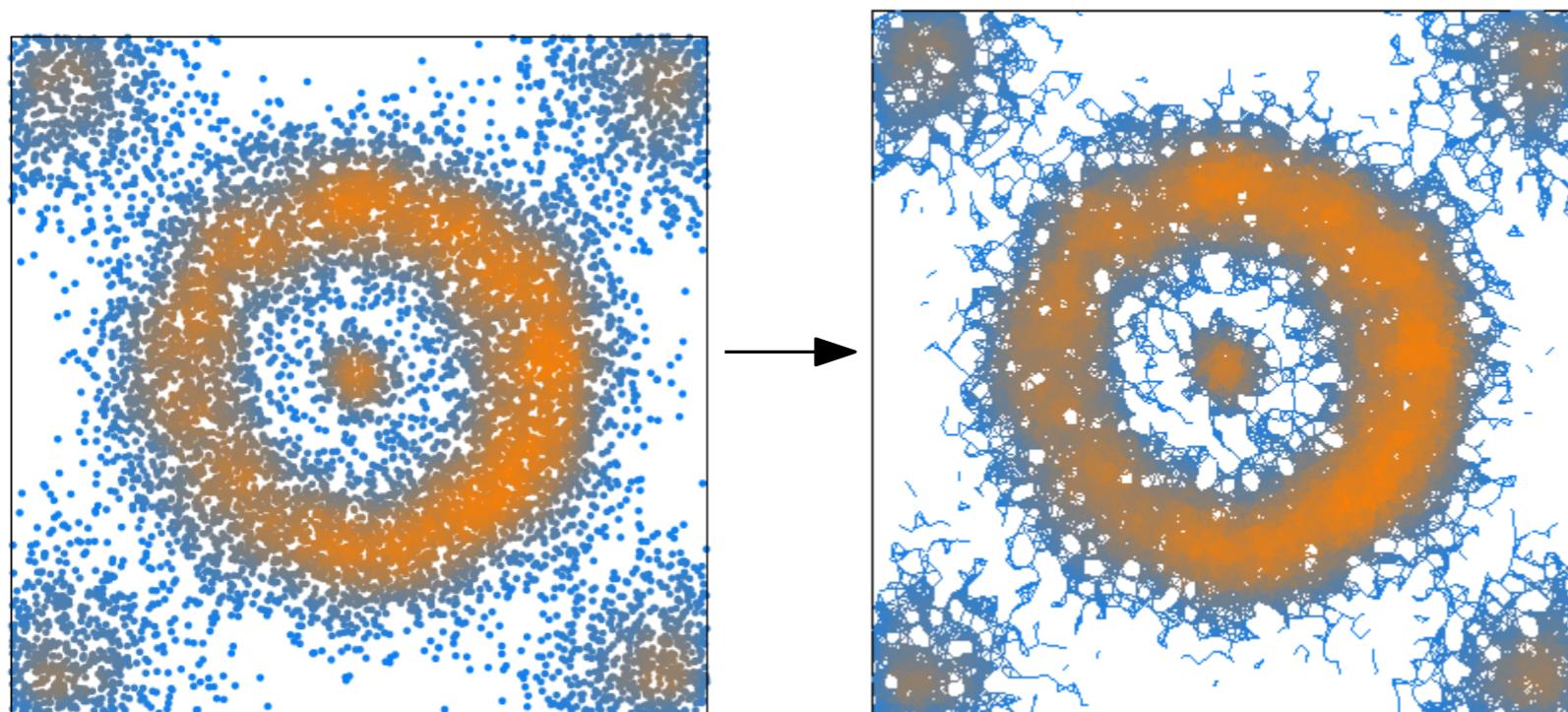
1. Define an order on the point cloud with filter \hat{f} (e.g., density estimator).
(sort data points by **decreasing** estimated filter values)



ToMATo: Topological Mode Analysis Tool

[Persistence-Based Clustering
in Riemannian Manifolds,
Chazal, Oudot, Skraba,
Guibas, J. ACM, 2013]

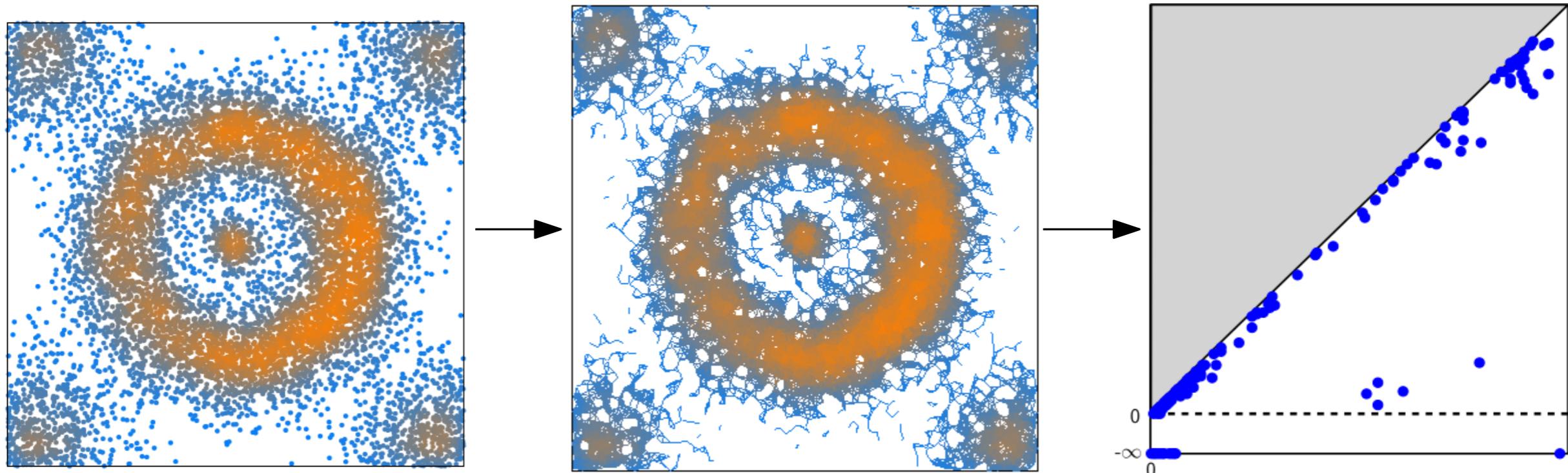
1. Define an order on the point cloud with filter \hat{f} (e.g., density estimator).
(sort data points by **decreasing** estimated filter values)
2. Extend order to the graph edges (i.e., compute the *upper-star filtration*).
 $(\hat{f}([u, v]) = \min\{\hat{f}(u), \hat{f}(v)\})$



ToMATo: Topological Mode Analysis Tool

[Persistence-Based Clustering in Riemannian Manifolds, Chazal, Oudot, Skraba, Guibas, J. ACM, 2013]

1. Define an order on the point cloud with filter \hat{f} (e.g., density estimator).
(sort data points by **decreasing** estimated filter values)
2. Extend order to the graph edges (i.e., compute the *upper-star filtration*).
 $(\hat{f}([u, v]) = \min\{\hat{f}(u), \hat{f}(v)\})$
3. Compute the 0-dimensional persistence diagram of this filtration.
(apply 0-dimensional persistence algorithm → union-find data structure)



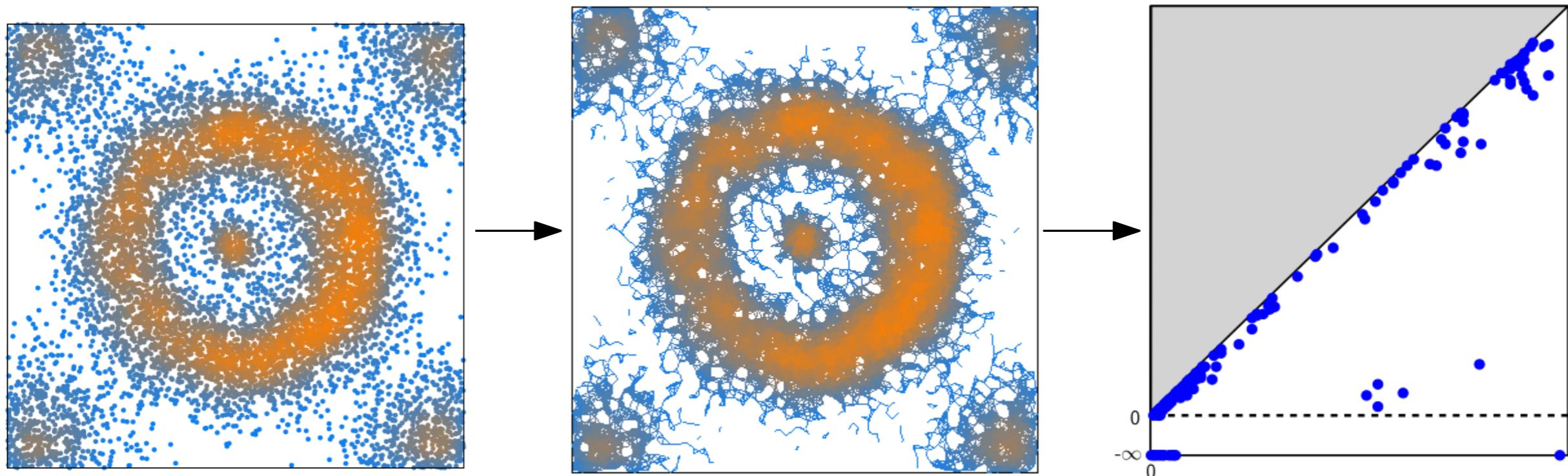
ToMATo: Topological Mode Analysis Tool

[Persistence-Based Clustering
in Riemannian Manifolds,
Chazal, Oudot, Skraba,
Guibas, J. ACM, 2013]

Given a neighborhood graph with n vertices and m edges:

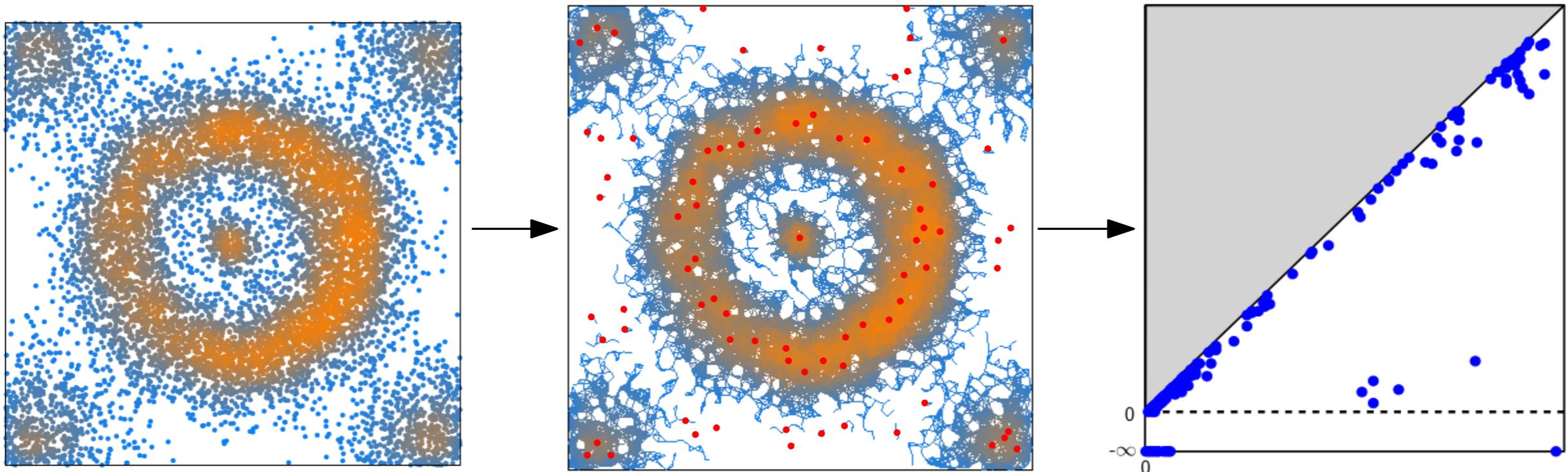
1. the algorithm sorts the vertices by decreasing density values,
2. and then makes a single pass through the vertex set, merging clusters on the fly using a union-find data structure.

→ Running time: $O(n \log n + (n + m)\alpha(n))$
→ Space complexity: $O(n + m)$
→ Main memory usage: $O(n)$



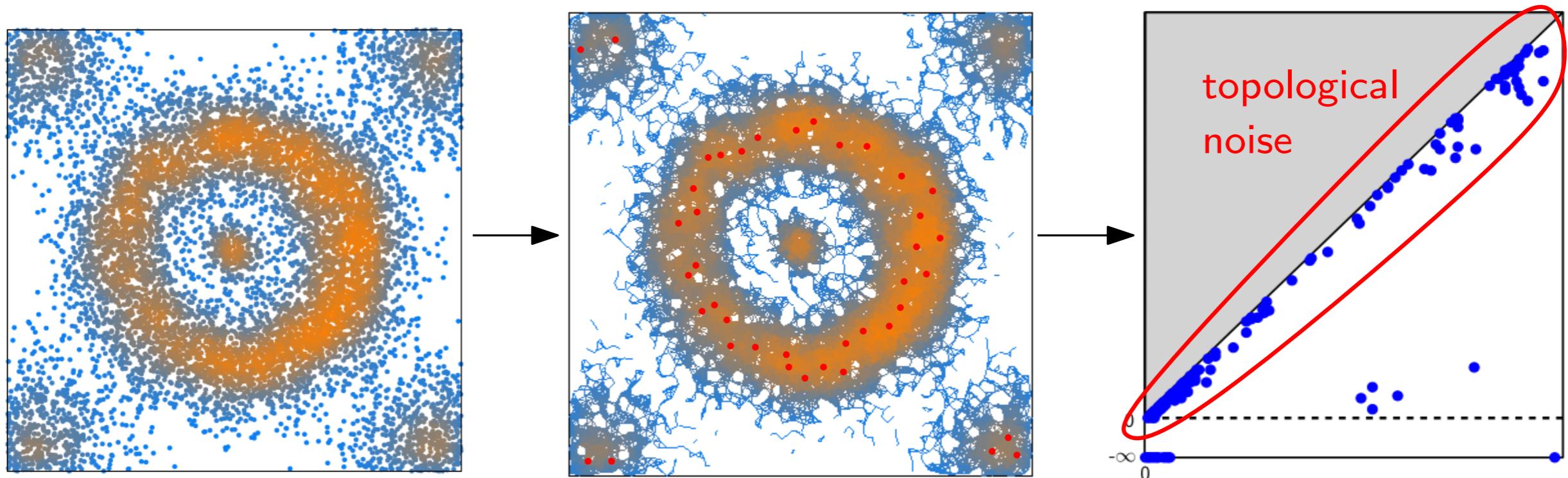
Estimating the correct number of clusters

1. Define an order on the point cloud with a density estimator \hat{f} .
(sort data points by **decreasing** estimated density values)
2. Extend order to the graph edges (i.e., compute the *upper-star filtration*).
 $(\hat{f}([u, v]) = \min\{\hat{f}(u), \hat{f}(v)\})$
3. Compute the 0-dimensional persistence diagram of this filtration.
(apply 0-dimensional persistence algorithm → union-find data structure)



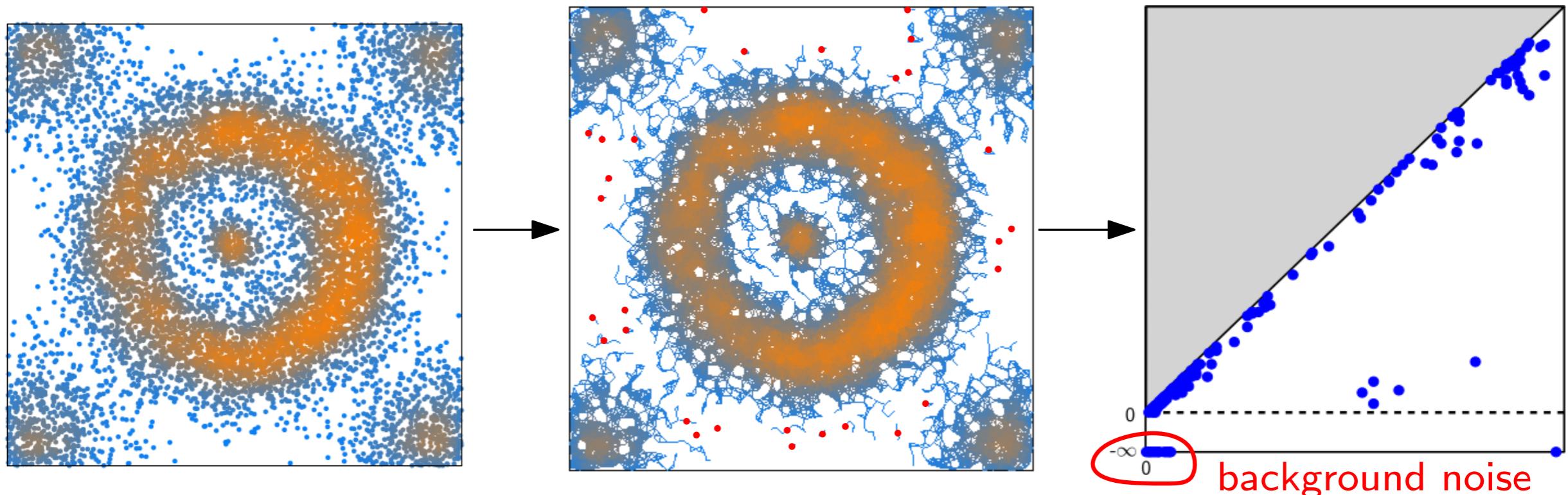
Estimating the correct number of clusters

1. Define an order on the point cloud with a density estimator \hat{f} .
(sort data points by **decreasing** estimated density values)
2. Extend order to the graph edges (i.e., compute the *upper-star filtration*).
 $(\hat{f}([u, v]) = \min\{\hat{f}(u), \hat{f}(v)\})$
3. Compute the 0-dimensional persistence diagram of this filtration.
(apply 0-dimensional persistence algorithm → union-find data structure)



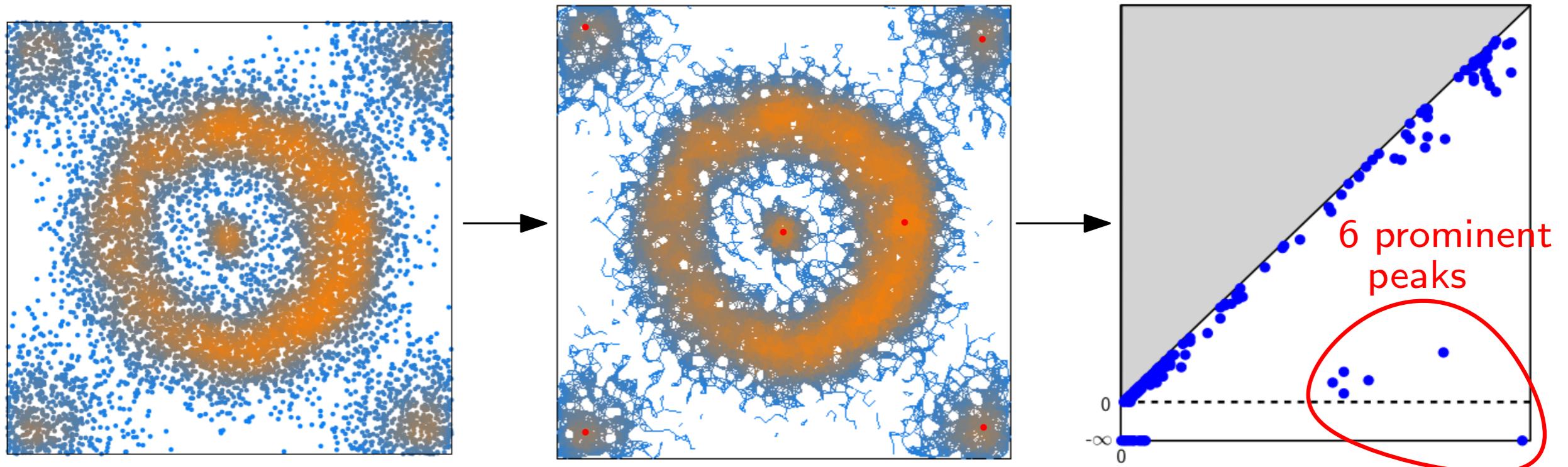
Estimating the correct number of clusters

1. Define an order on the point cloud with a density estimator \hat{f} .
(sort data points by **decreasing** estimated density values)
2. Extend order to the graph edges (i.e., compute the *upper-star filtration*).
 $(\hat{f}([u, v]) = \min\{\hat{f}(u), \hat{f}(v)\})$
3. Compute the 0-dimensional persistence diagram of this filtration.
(apply 0-dimensional persistence algorithm → union-find data structure)



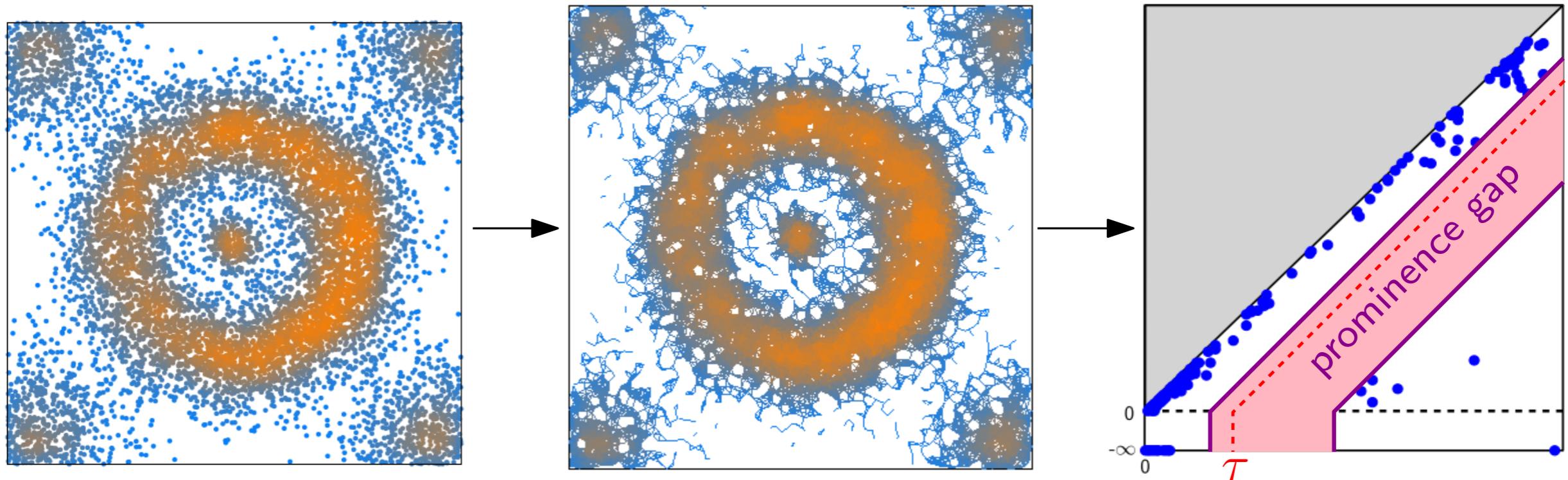
Estimating the correct number of clusters

1. Define an order on the point cloud with a density estimator \hat{f} .
(sort data points by **decreasing** estimated density values)
2. Extend order to the graph edges (i.e., compute the *upper-star filtration*).
 $(\hat{f}([u, v]) = \min\{\hat{f}(u), \hat{f}(v)\})$
3. Compute the 0-dimensional persistence diagram of this filtration.
(apply 0-dimensional persistence algorithm → union-find data structure)



Estimating the correct number of clusters

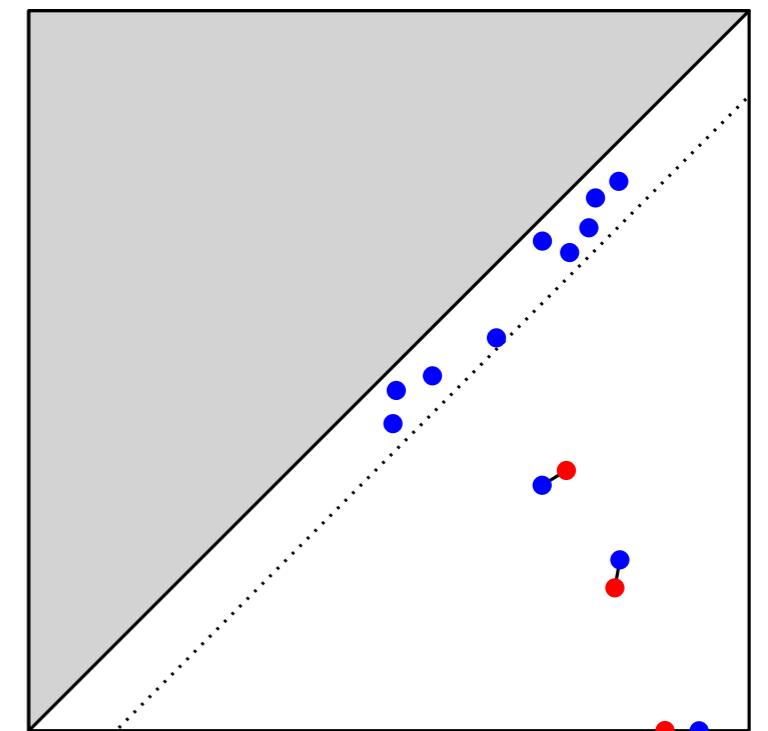
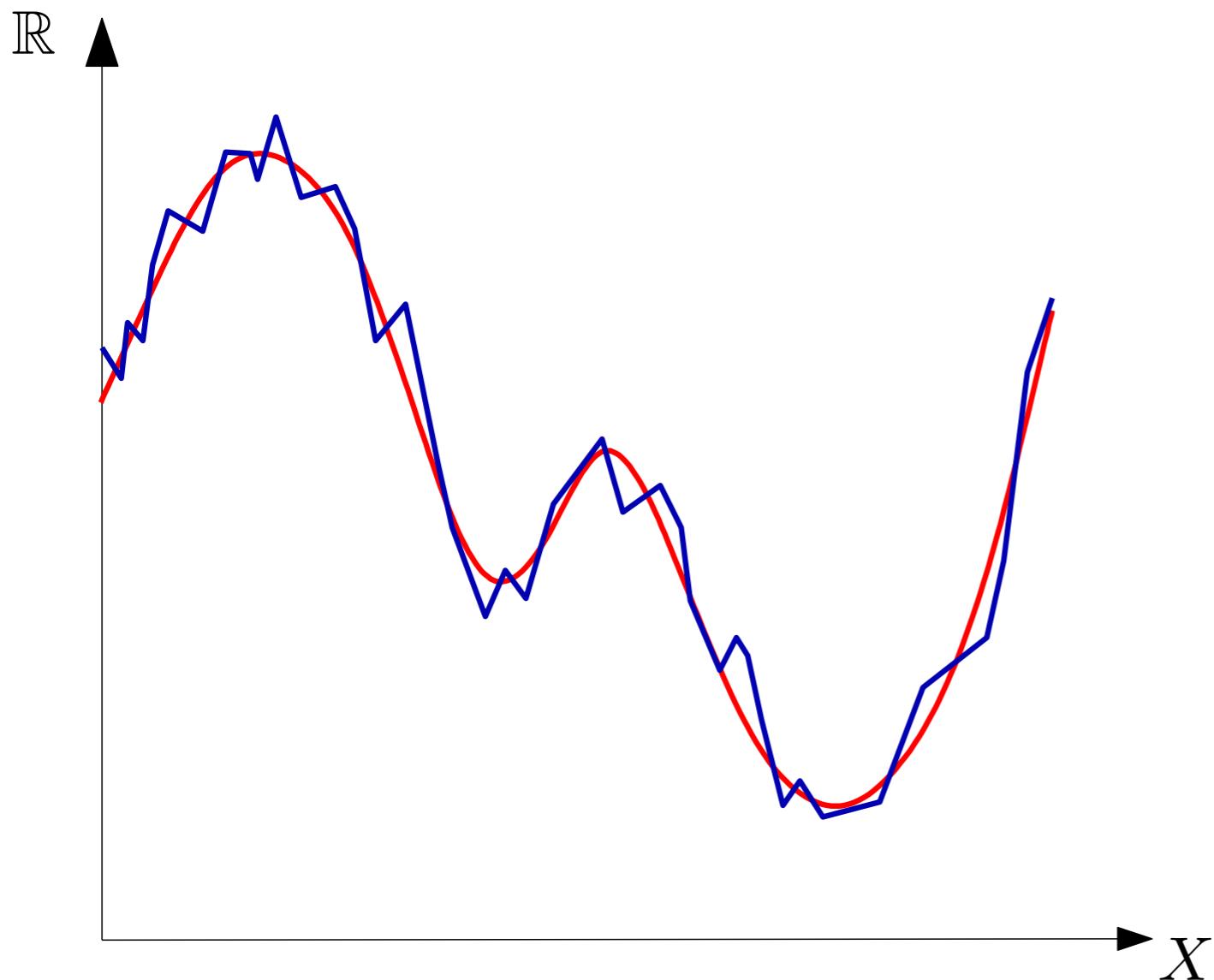
1. Define an order on the point cloud with a density estimator \hat{f} .
(sort data points by **decreasing** estimated density values)
2. Extend order to the graph edges (i.e., compute the *upper-star filtration*).
 $(\hat{f}([u, v]) = \min\{\hat{f}(u), \hat{f}(v)\})$
3. Compute the 0-dimensional persistence diagram of this filtration.
(apply 0-dimensional persistence algorithm → union-find data structure)



The Stability Theorem

This seminal TDA result ensures that, given an underlying ground-truth function $f : X \rightarrow \mathbb{R}$, and an estimator $\hat{f} : X \rightarrow \mathbb{R}$ of it, one has:

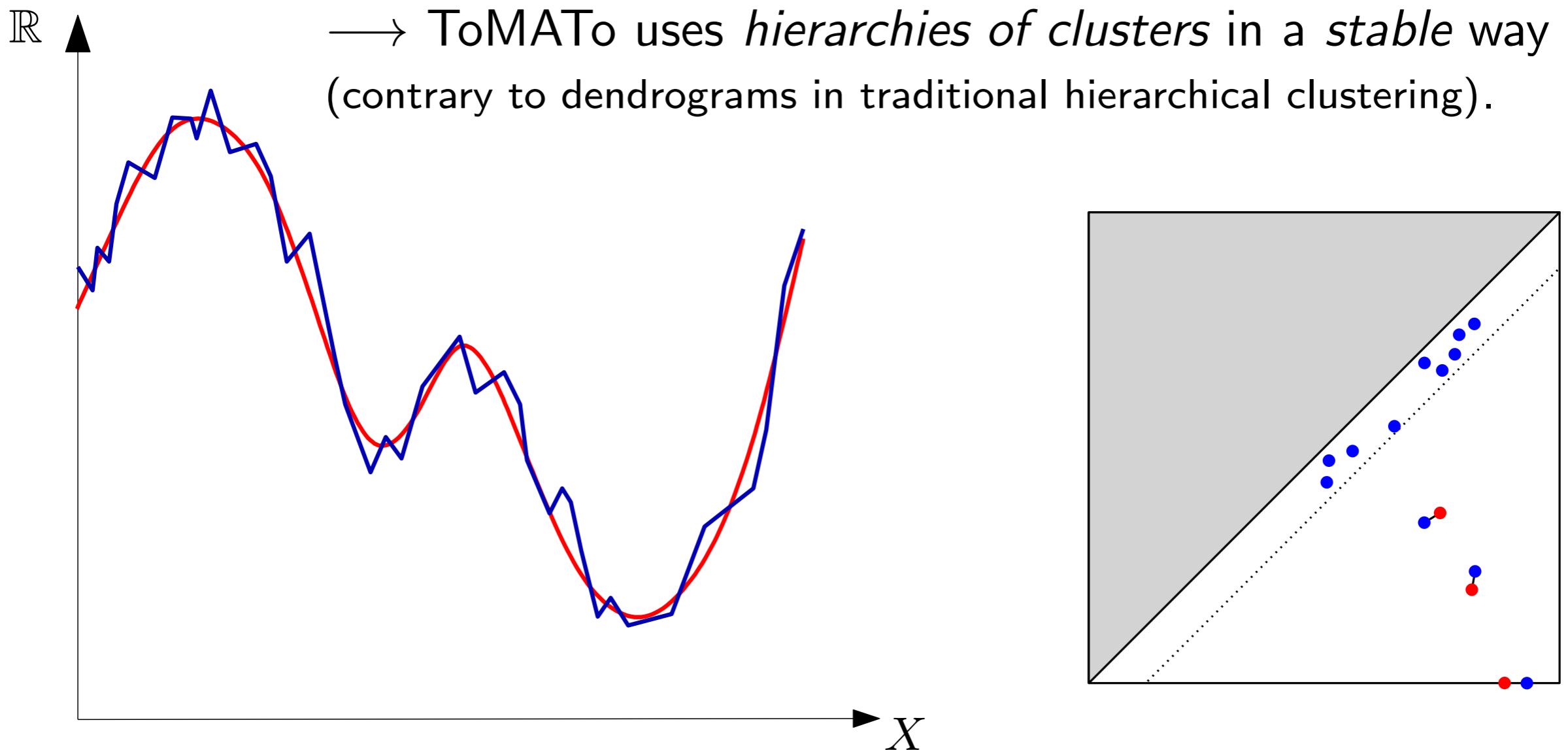
$$d_b(D_f, D_{\hat{f}}) \leq \|f - \hat{f}\|_\infty.$$



The Stability Theorem

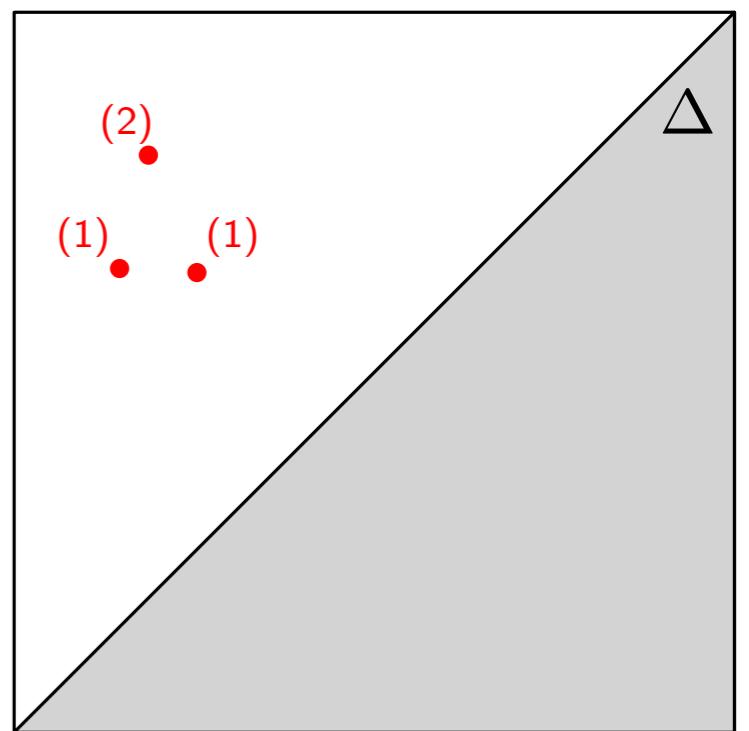
This seminal TDA result ensures that, given an underlying ground-truth function $f : X \rightarrow \mathbb{R}$, and an estimator $\hat{f} : X \rightarrow \mathbb{R}$ of it, one has:

$$d_b(D_f, D_{\hat{f}}) \leq \|f - \hat{f}\|_\infty.$$



Distance between persistence diagrams

Persistence diagram \equiv **finite** multiset in the open half-plane $\Delta \times \mathbb{R}_{>0}$.



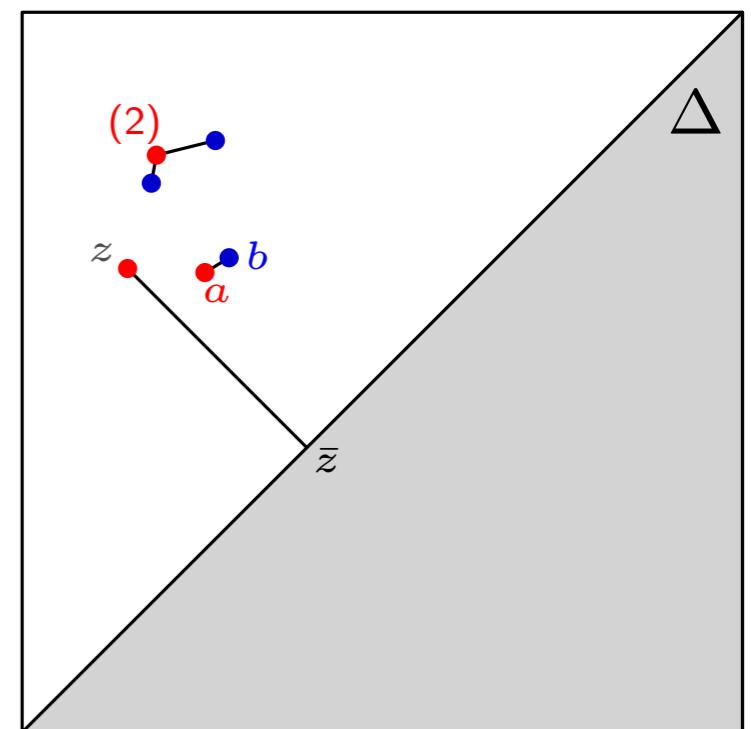
Distance between persistence diagrams

Persistence diagram \equiv **finite multiset** in the open half-plane $\Delta \times \mathbb{R}_{>0}$.

Given a **partial matching** $M : D \leftrightarrow D'$:

- cost of a matched pair $(a, b) \in M$: $c_p(a, b) := \|a - b\|_\infty^p$,
- cost of an unmatched point $c \in D \sqcup D'$: $c_p(c) := \|c - \bar{c}\|_\infty^p$,
- **cost of M** :

$$c_p(M) := \left(\sum_{(a, b) \text{ matched}} c_p(a, b) + \sum_{c \text{ unmatched}} c_p(c) \right)^{1/p}$$



Distance between persistence diagrams

Persistence diagram \equiv **finite multiset** in the open half-plane $\Delta \times \mathbb{R}_{>0}$.

Given a **partial matching** $M : D \leftrightarrow D'$:

- cost of a matched pair $(a, b) \in M$: $c_p(a, b) := \|a - b\|_\infty^p$,
- cost of an unmatched point $c \in D \sqcup D'$: $c_p(c) := \|c - \bar{c}\|_\infty^p$,
- **cost of M** :

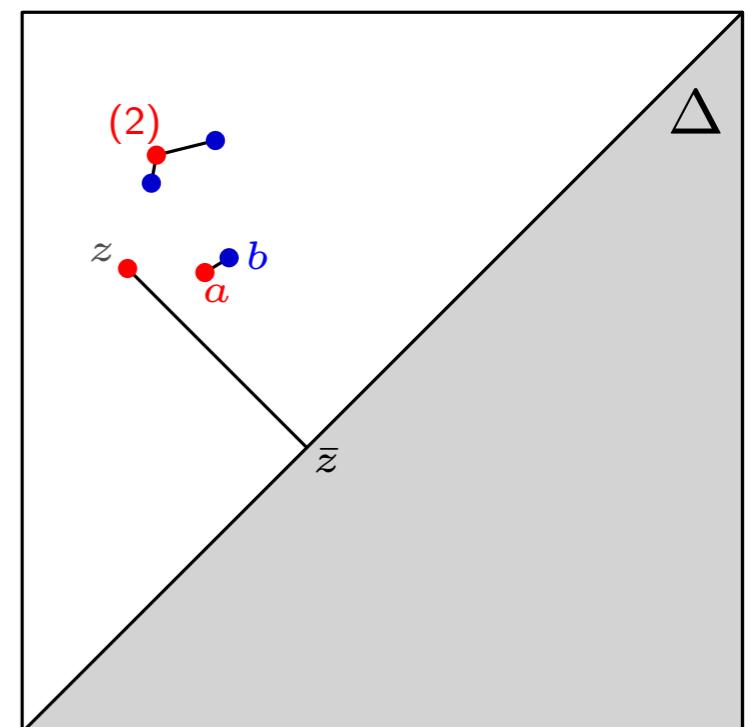
$$c_p(M) := \left(\sum_{(a, b) \text{ matched}} c_p(a, b) + \sum_{c \text{ unmatched}} c_p(c) \right)^{1/p}$$

Def: p -th diagram distance (extended metric):

$$d_p(D, D') := \inf_{M: D \leftrightarrow D'} c_p(M)$$

Def: bottleneck distance:

$$d_b(D, D') = d_\infty(D, D') := \lim_{p \rightarrow \infty} d_p(D, D')$$

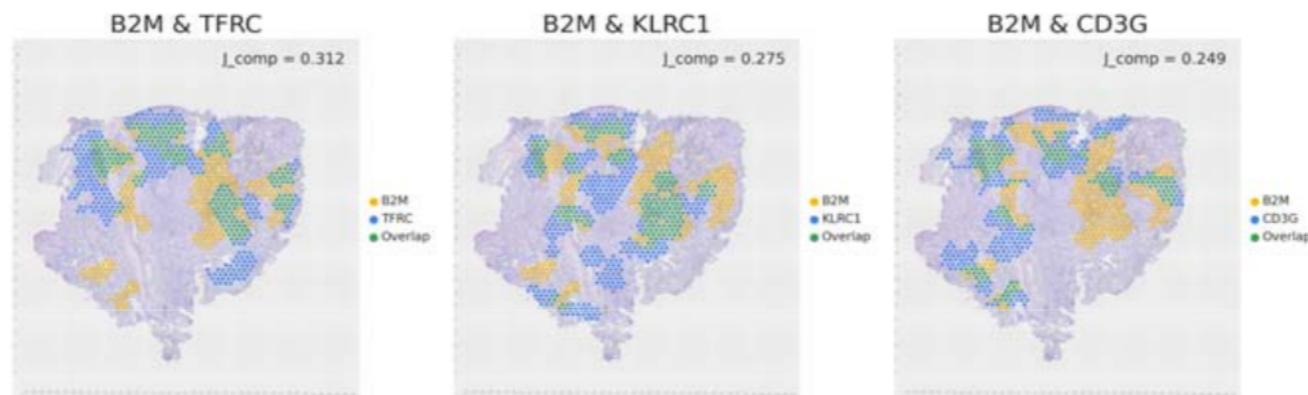


Application to colocalization

Method:

1. Compute clusters associated to several gene markers with ToMATo and compute pairwise Jaccard similarities:

$$0 \leq J(C, C') := \frac{\#\{C \cap C'\}}{\#\{C \cup C'\}} \leq 1$$



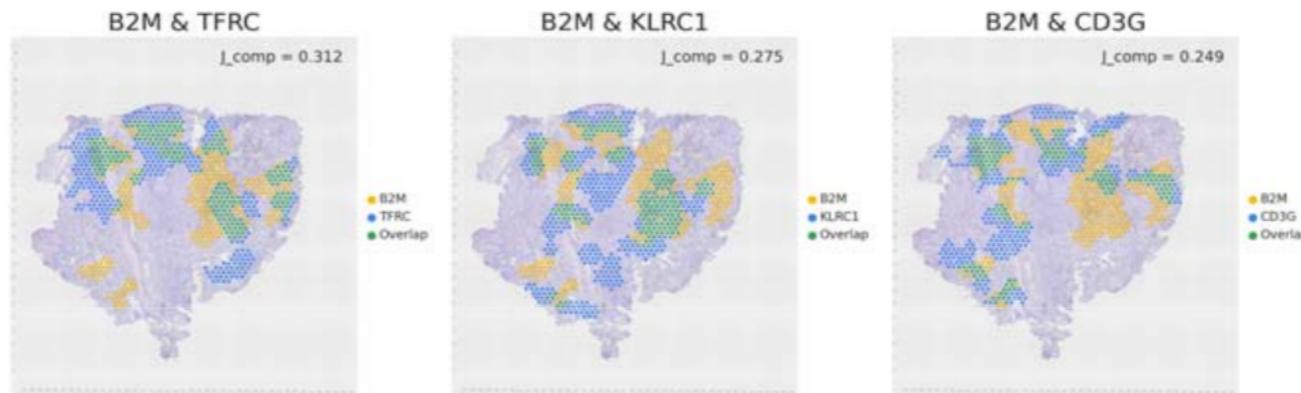
Application to colocalization

Method:

1. Compute clusters associated to several gene markers with ToMATo and compute pairwise Jaccard similarities:

$$0 \leq J(C, C') := \frac{\#\{C \cap C'\}}{\#\{C \cup C'\}} \leq 1$$

connectivity given by adjacency between spots
 τ chosen with persistence gap



Application to colocalization

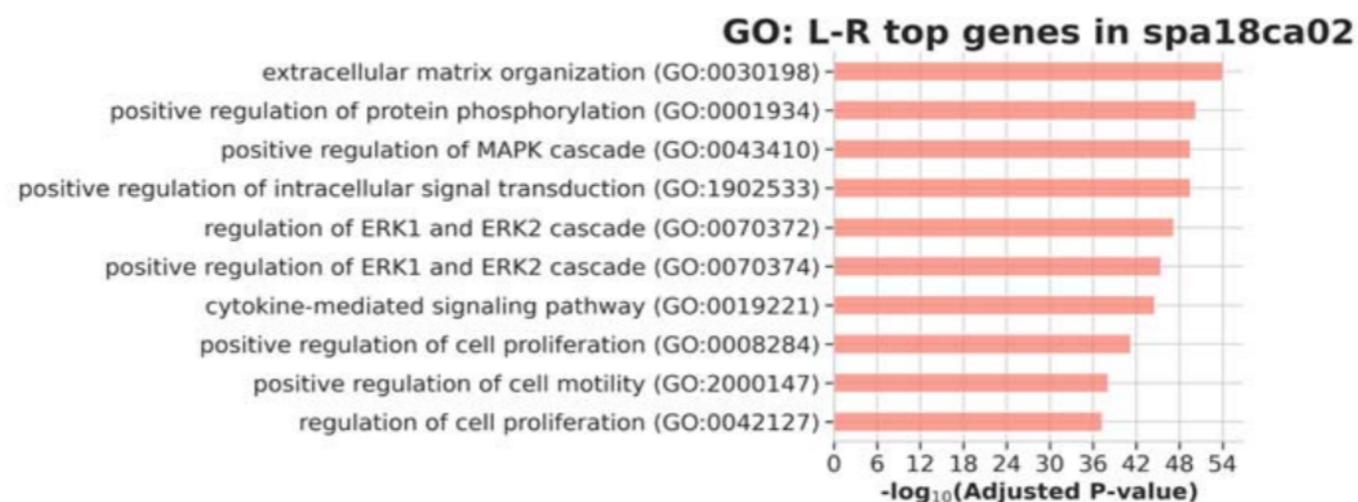
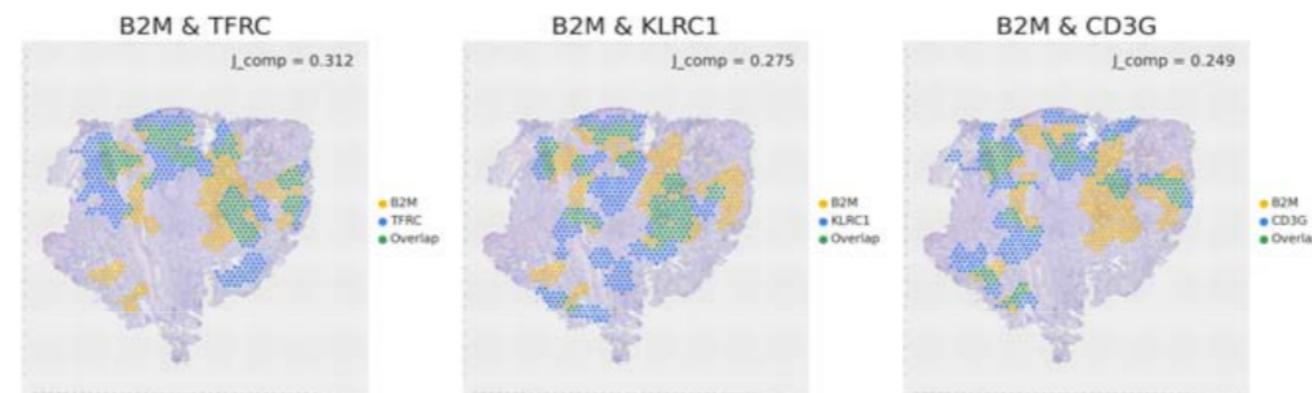
Method:

1. Compute clusters associated to several gene markers with ToMATo and compute pairwise Jaccard similarities:

$$0 \leq J(C, C') := \frac{\#\{C \cap C'\}}{\#\{C \cup C'\}} \leq 1$$

connectivity given by adjacency between spots
 τ chosen with persistence gap

2. Retrieve the marker genes / ligand receptor pairs with highest Jaccard similarities and match against gene ontology.



Application to colocalization

Method:

1. Compute clusters associated to several gene markers with ToMATo and compute pairwise Jaccard similarities:

$$0 \leq J(C, C') := \frac{\#\{C \cap C'\}}{\#\{C \cup C'\}} \leq 1$$

connectivity given by
adjacency between spots

τ chosen with
persistence gap

2. Retrieve the marker genes / ligand receptor pairs with highest Jaccard similarities and match against gene ontology.

One can also play the same game by using *higher-dimensional* homology, and then predict phenotypes solely from the corresponding persistence diagrams.

[Aukerman et al. - 2022 - *Persistent homology based characterization of the breast cancer immune microenvironment: a feasibility study*]