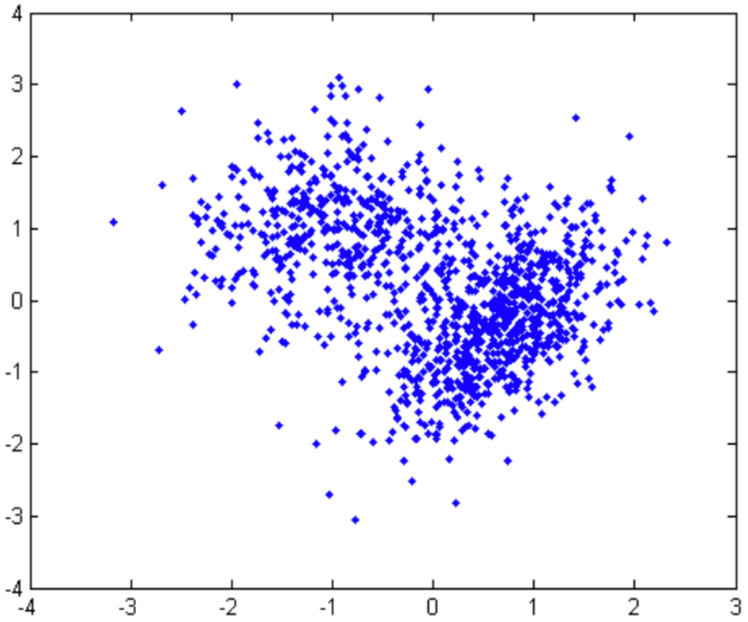
Persistent homology and clustering

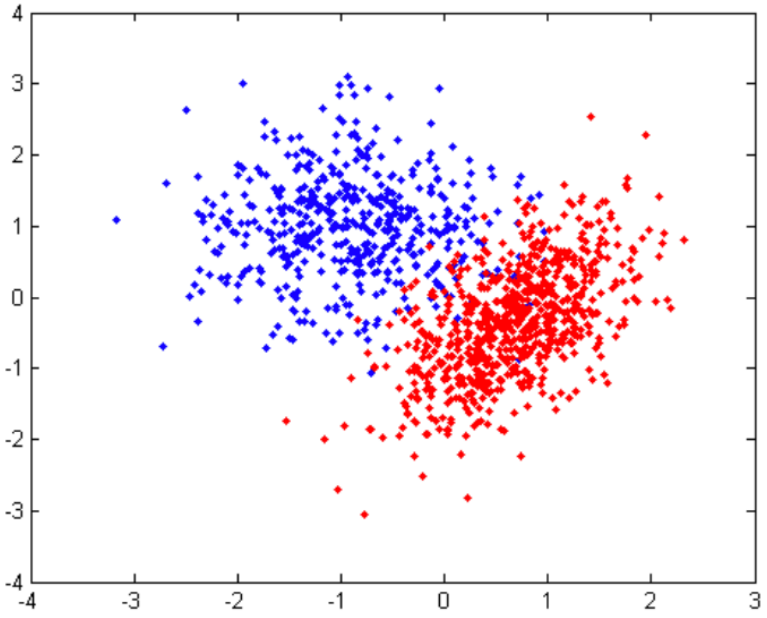
Based on lecture notes by Frédéric Chazal and Steve Oudot
(as well as various other sources, cited throughout)

Recall: Clustering

- Clustering: A partition of data into groups of similar observations. The observations in each group (cluster) are similar to each other and dissimilar to observations from other groups.

- Input: a set of points embedded in an Euclidean space (with coordinates) or a more general metric space (pairwise distance/ similarity) matrix.





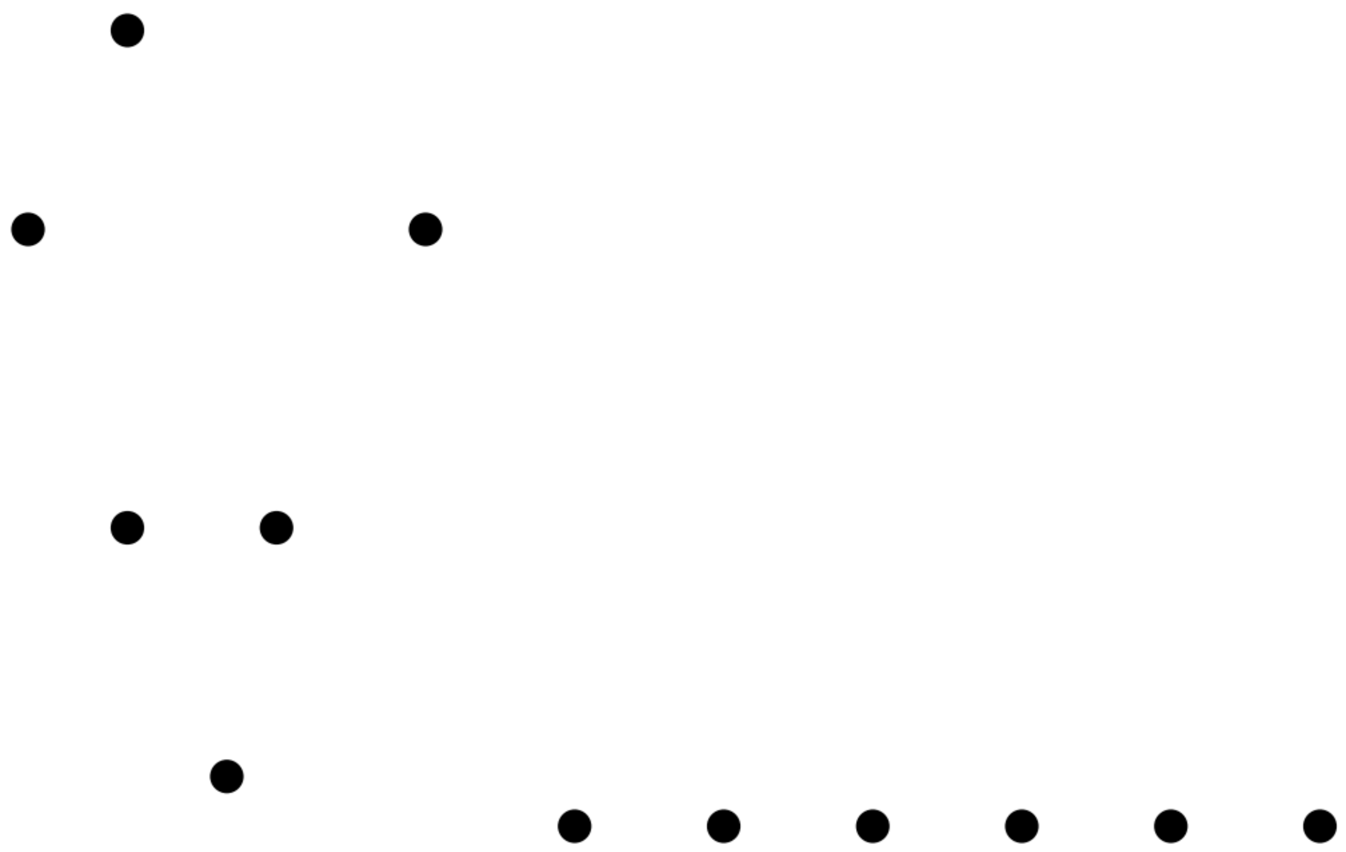
A motivation for persistent homology

- Recall our hierarchical clustering algorithm:
  - Make each point its own cluster
  - As long as you have more than 1 cluster:
    - Merge the two closest clusters C and C’, where distance between clusters is:

$$d(C, C') = \inf_{x \in C, x' \in C'} d(x, x')$$

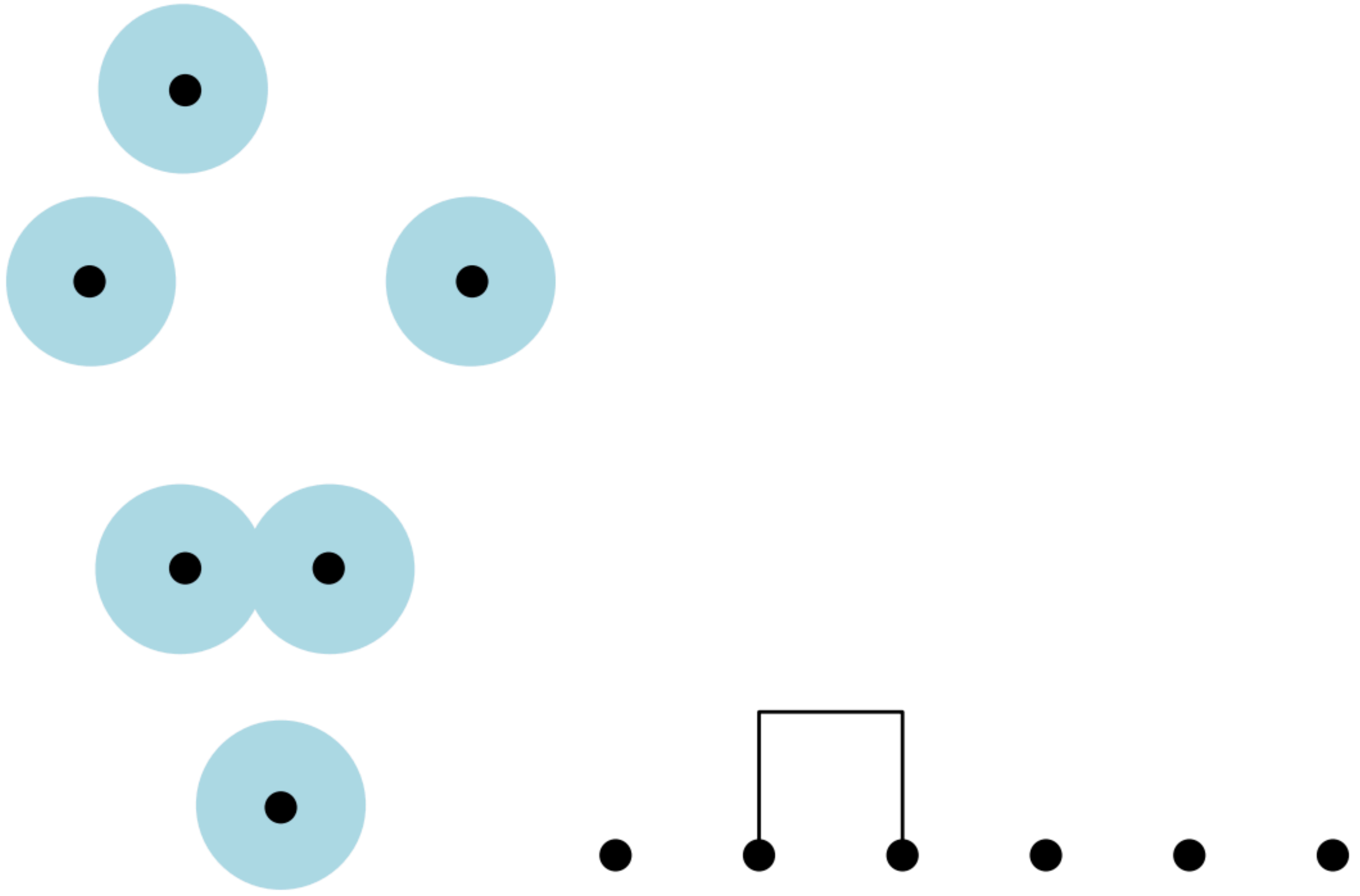
Hierarchical clustering

- Result: (input points to left, “tree” to the right)



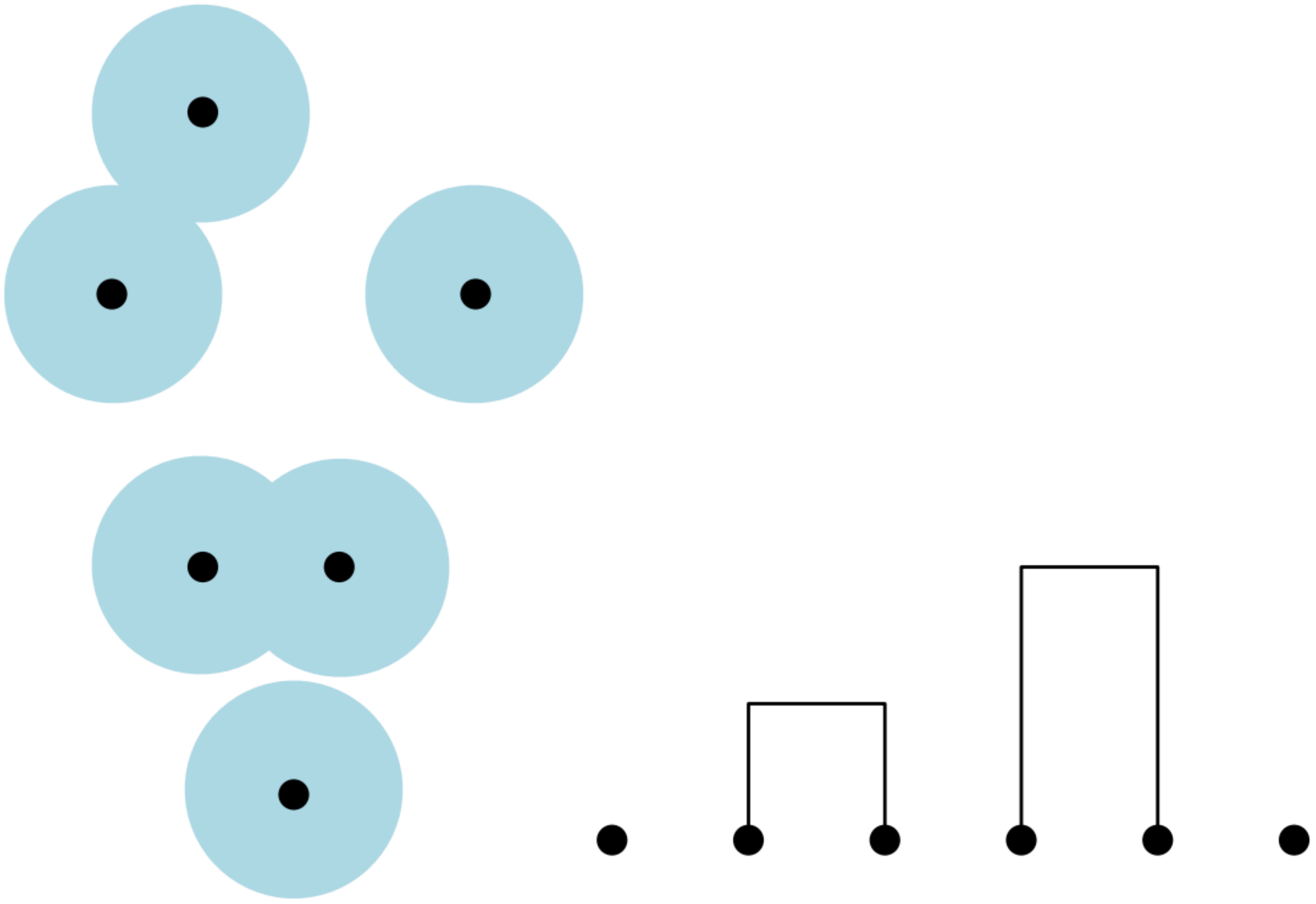
Hierarchical clustering

- Result: (input points to left, “tree” to the right)



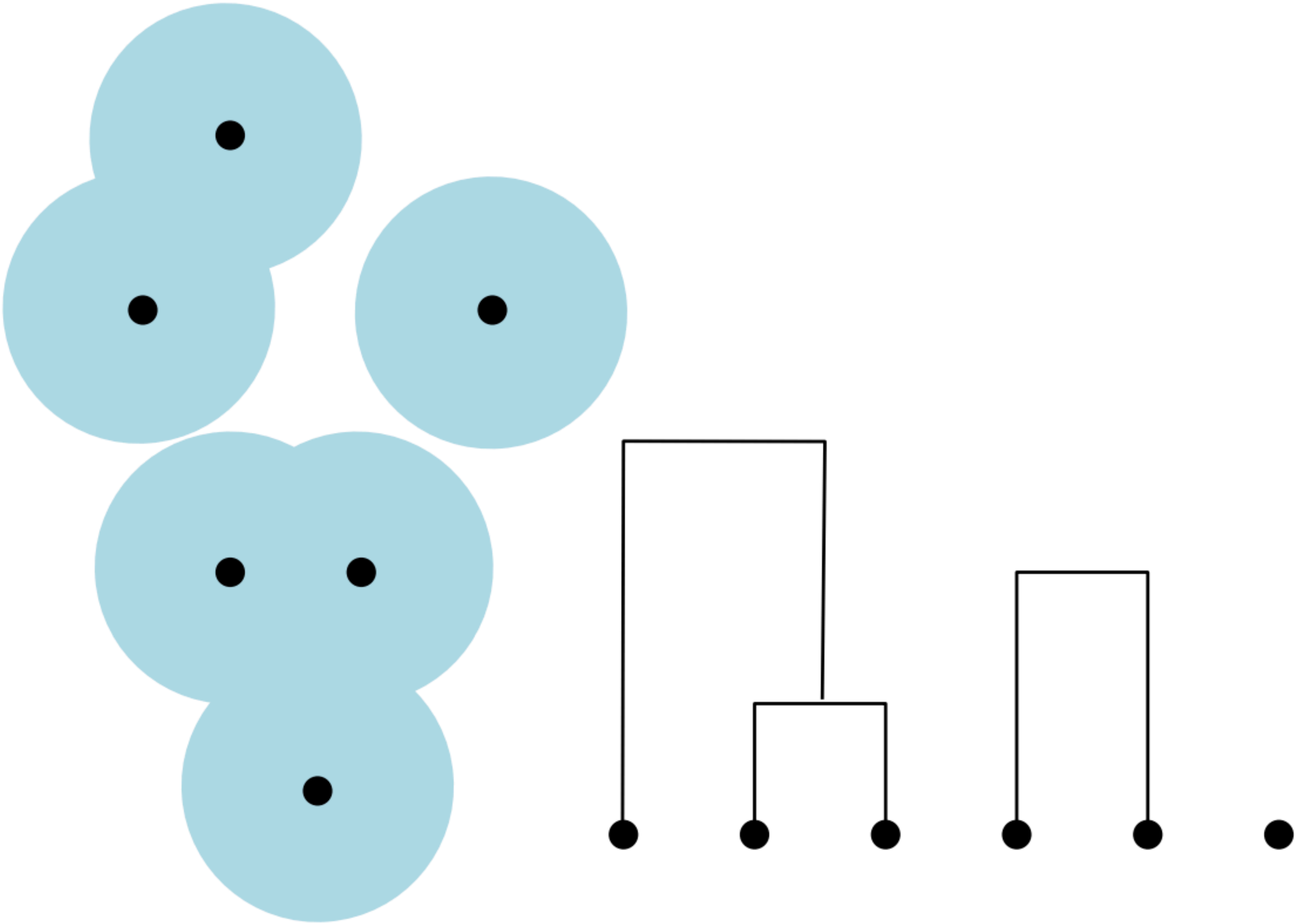
Hierarchical clustering

- Result: (input points to left, “tree” to the right)



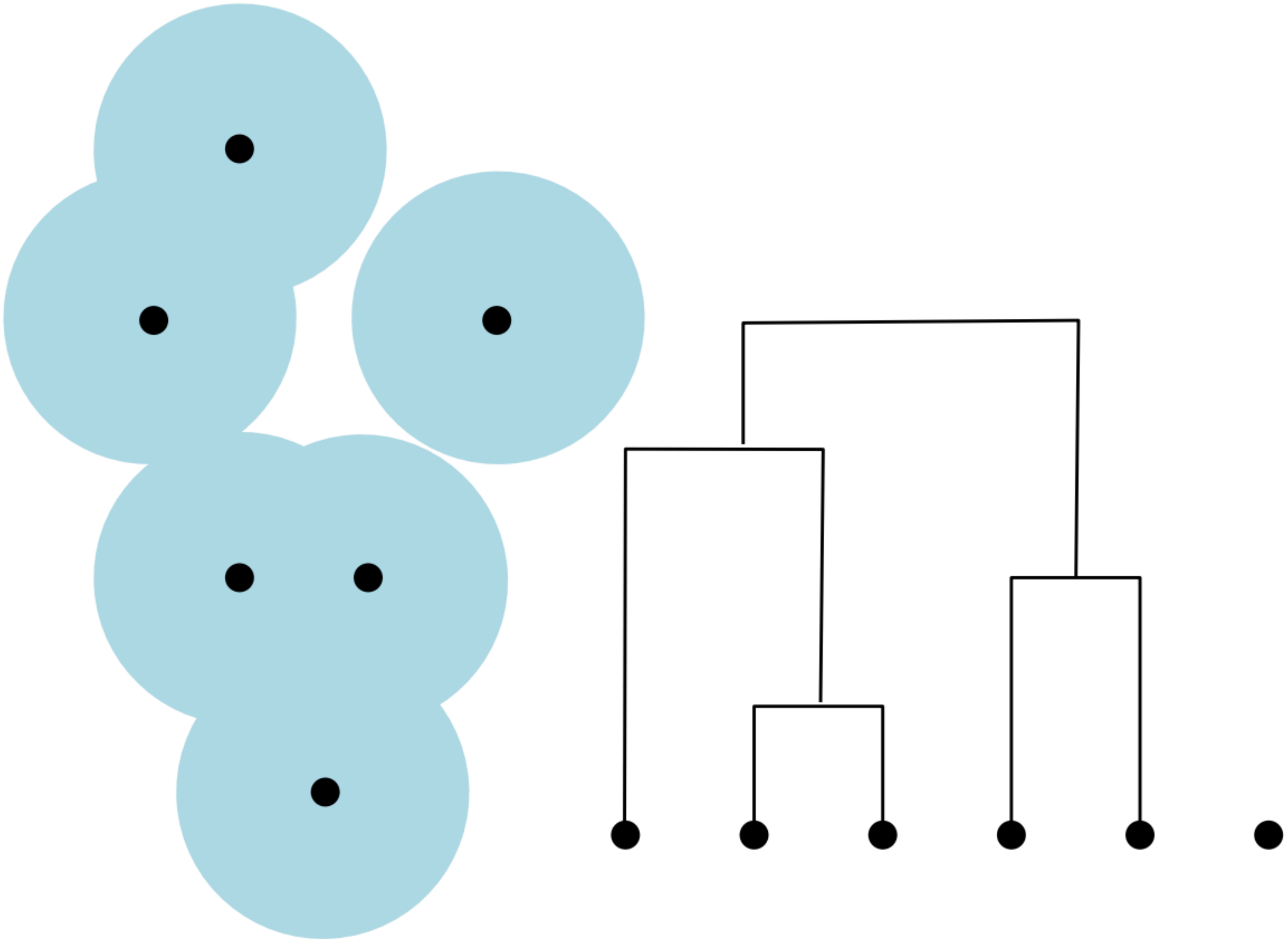
Hierarchical clustering

- Result: (input points to left, “tree” to the right)



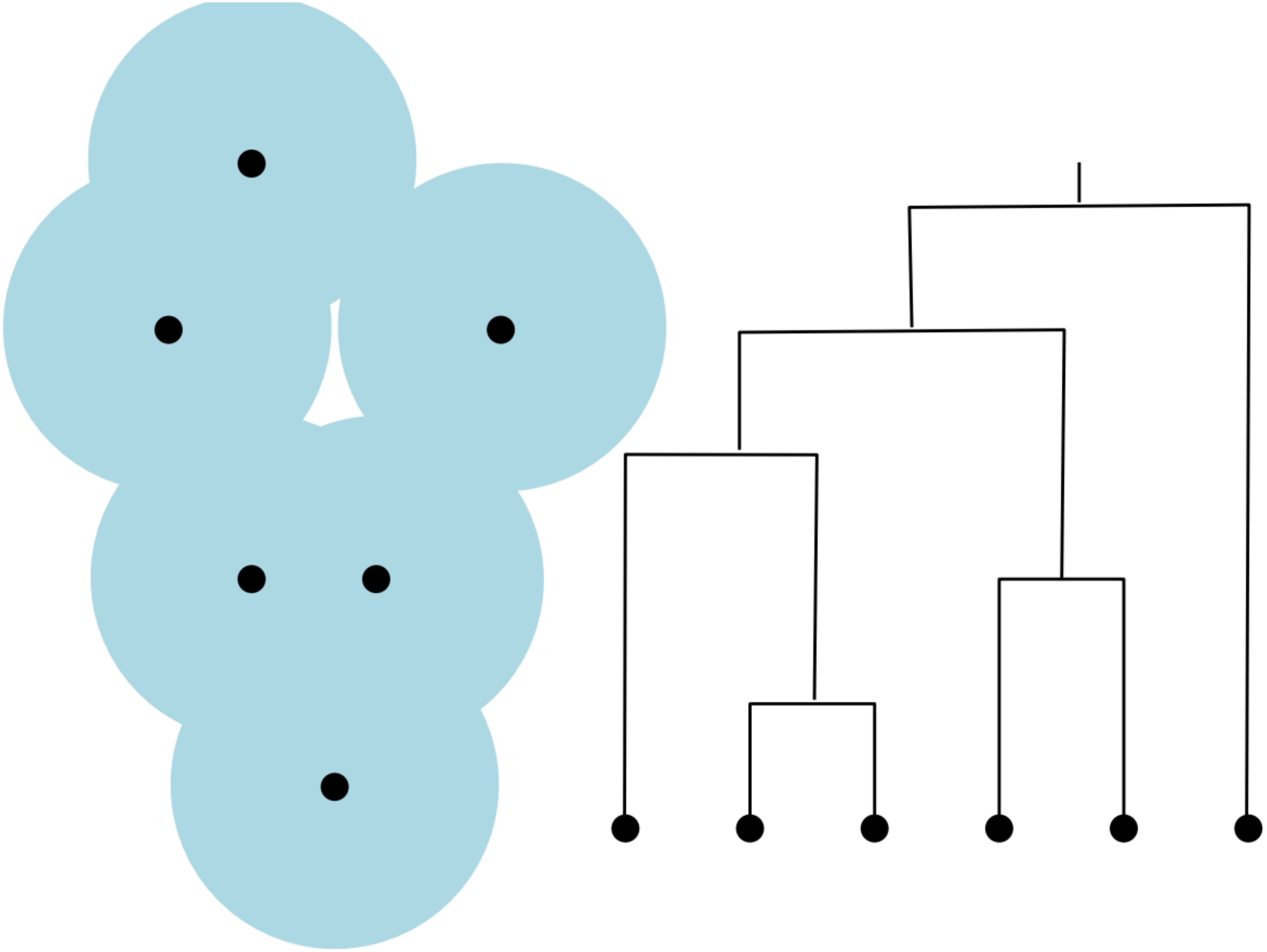
Hierarchical clustering

- Result: (input points to left, “tree” to the right)



Hierarchical clustering

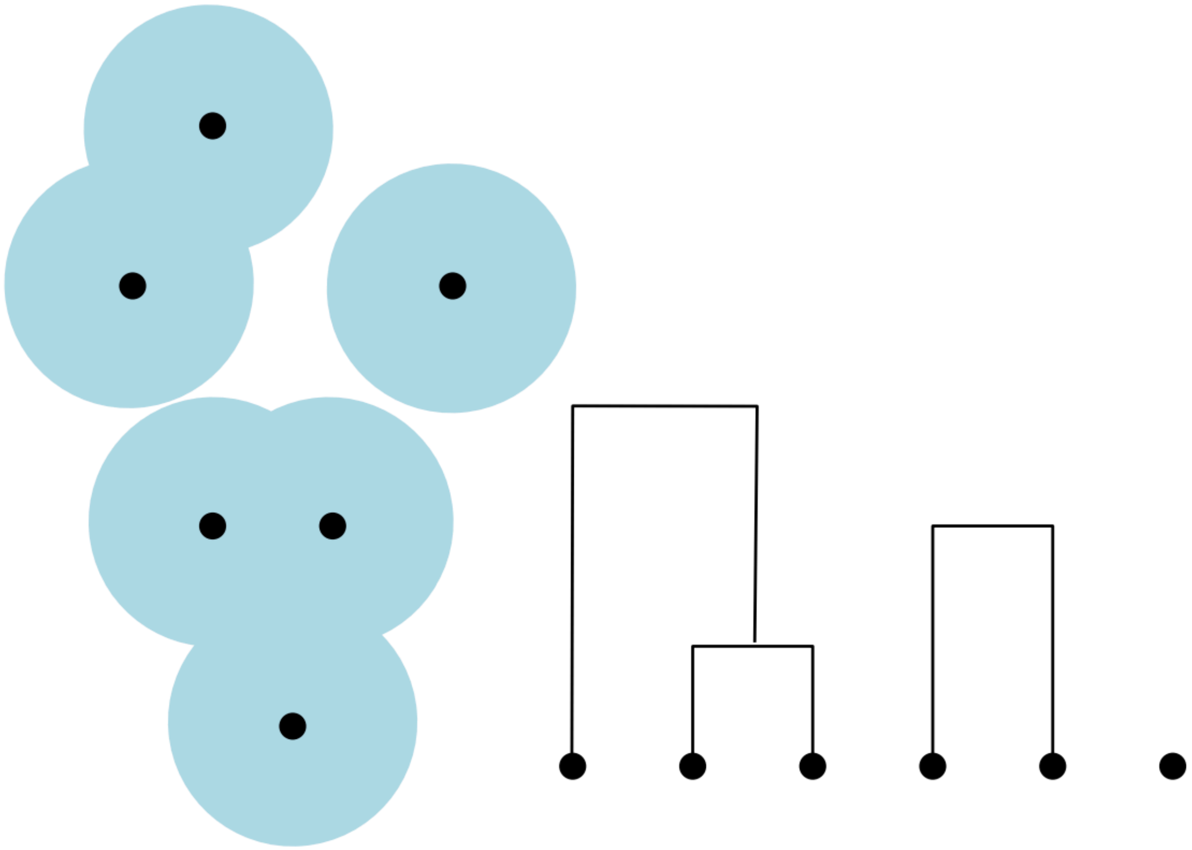
- Result: (input points to left, “tree” to the right)



Turning this into a k-cluster

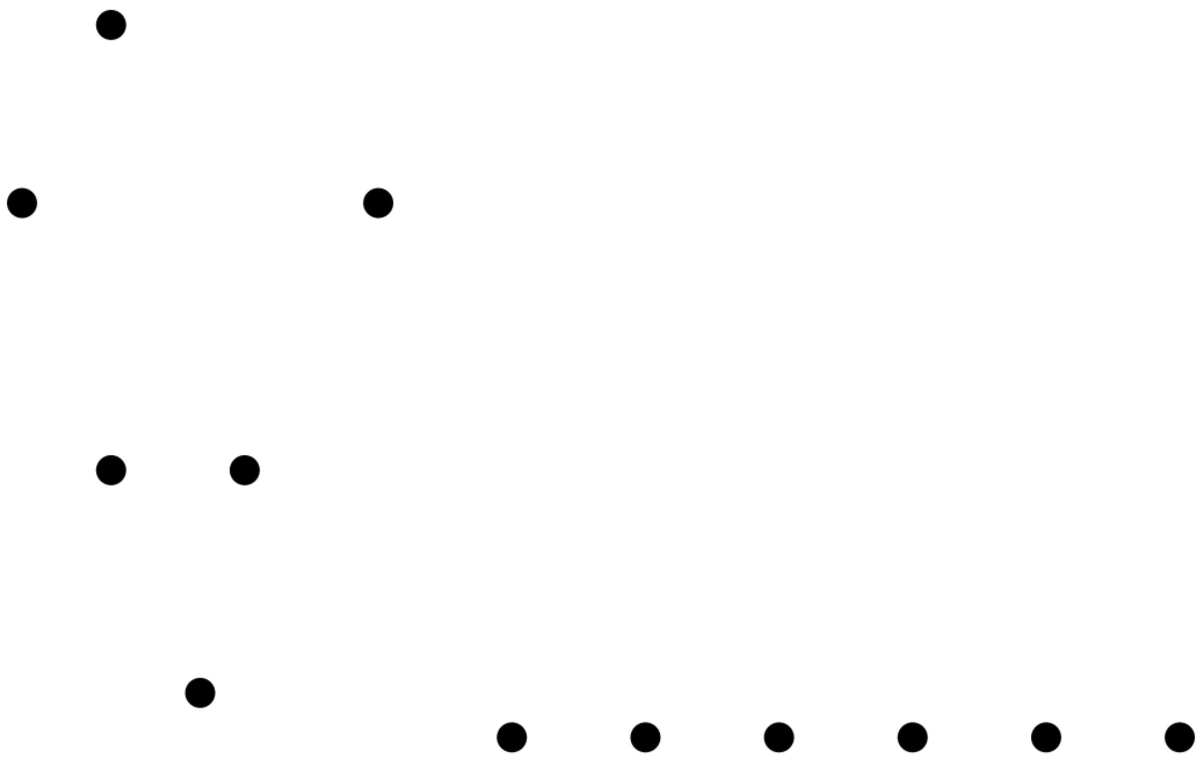
- The output of this is a hierarchical cluster; to get a smaller number of clusters, you can simply “halt” the process early, so that you have fewer clusters, each of which has nearby points

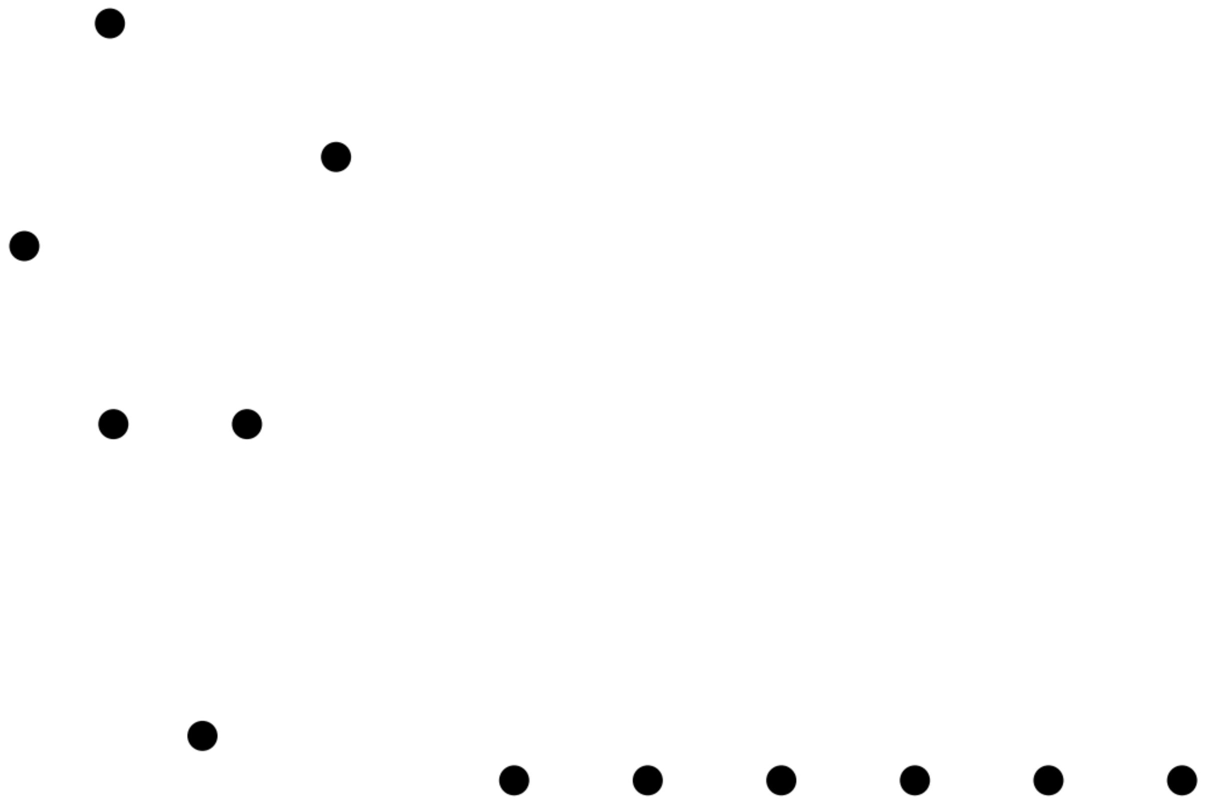
One downside:
Still more about similarity, and less about dissimilarity



Problem: instability

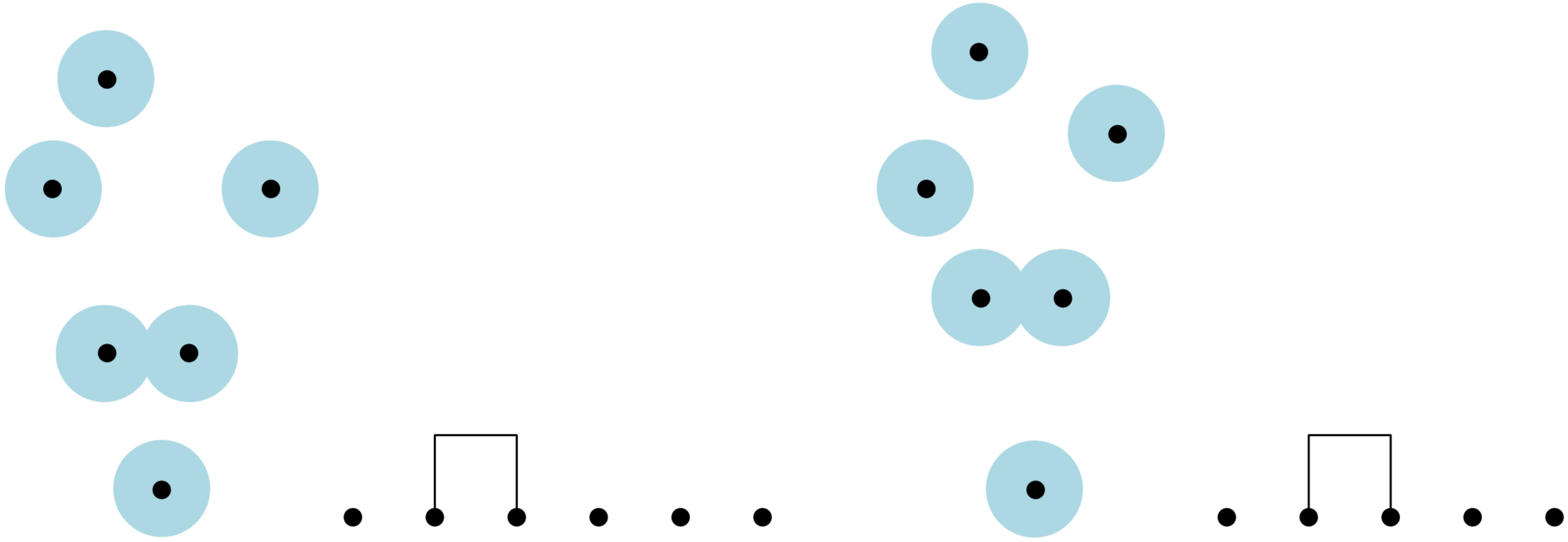
- Let’s consider two very similar inputs:





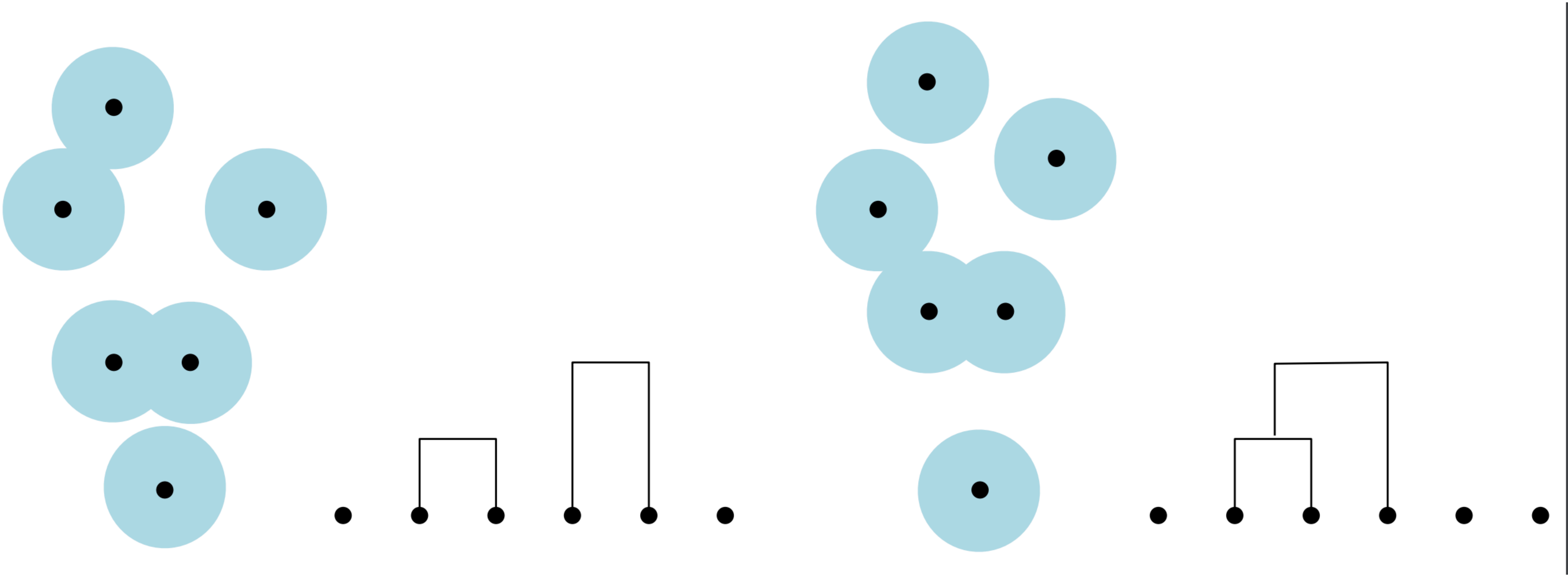
Problem: instability

- Let’s consider two very similar inputs:



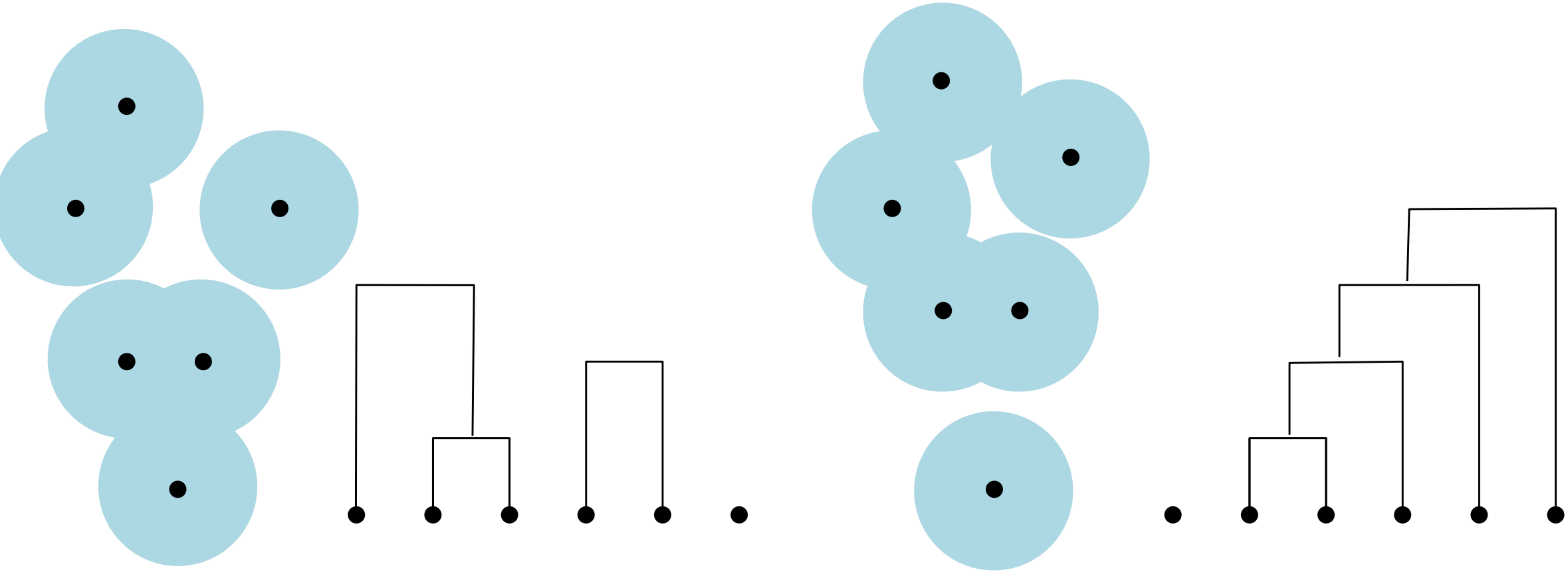
Problem: instability

- Let’s consider two very similar inputs:



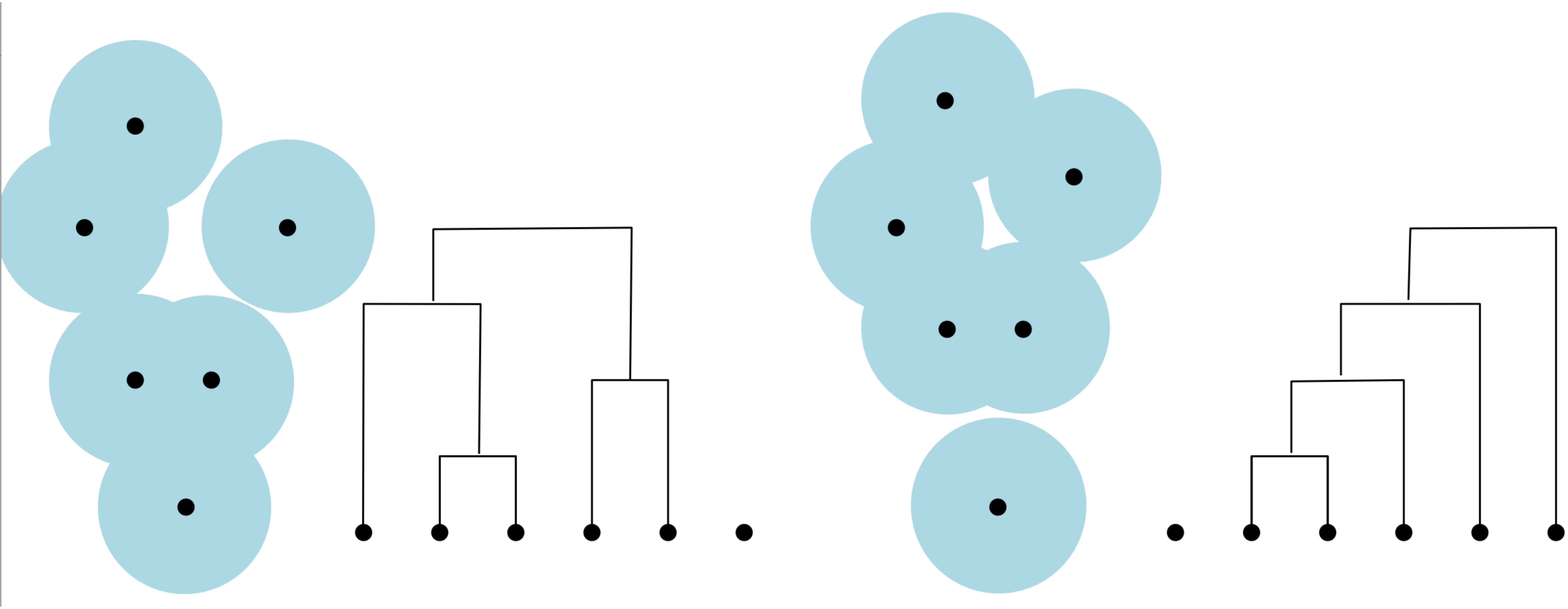
Problem: instability

- Let’s consider two very similar inputs:



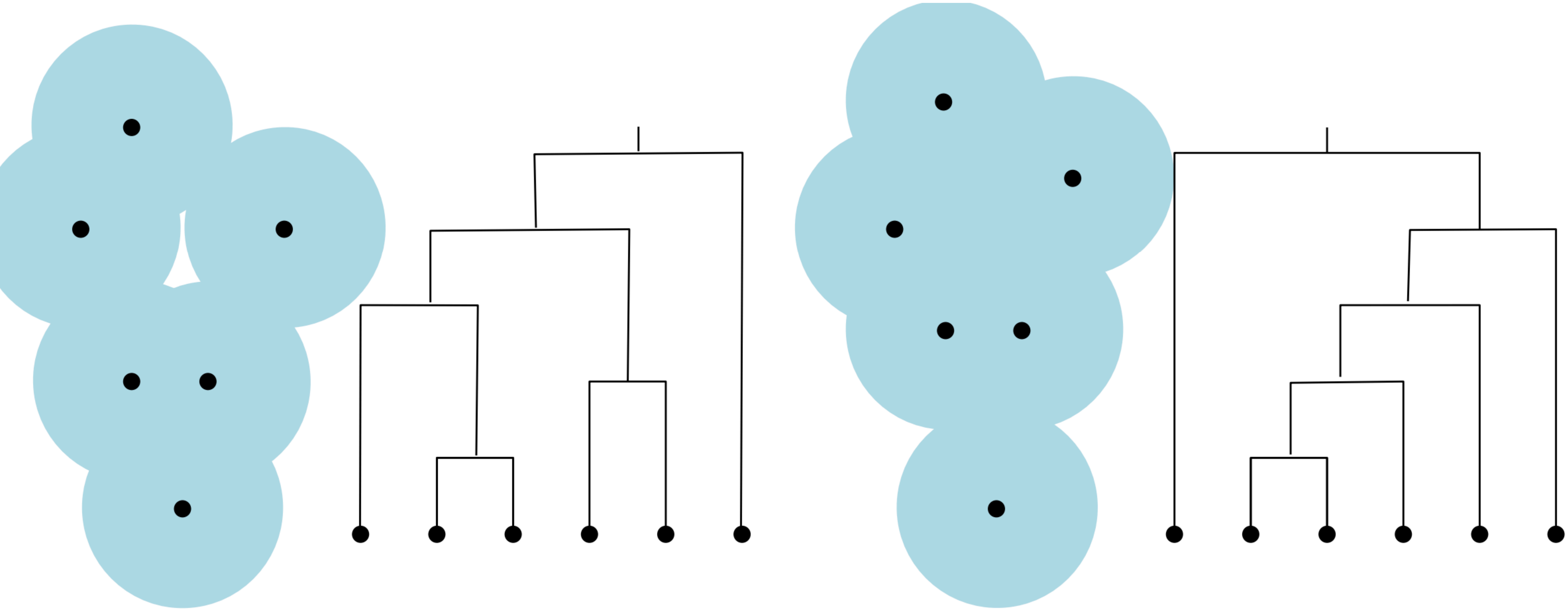
Problem: instability

- Let's consider two very similar inputs:



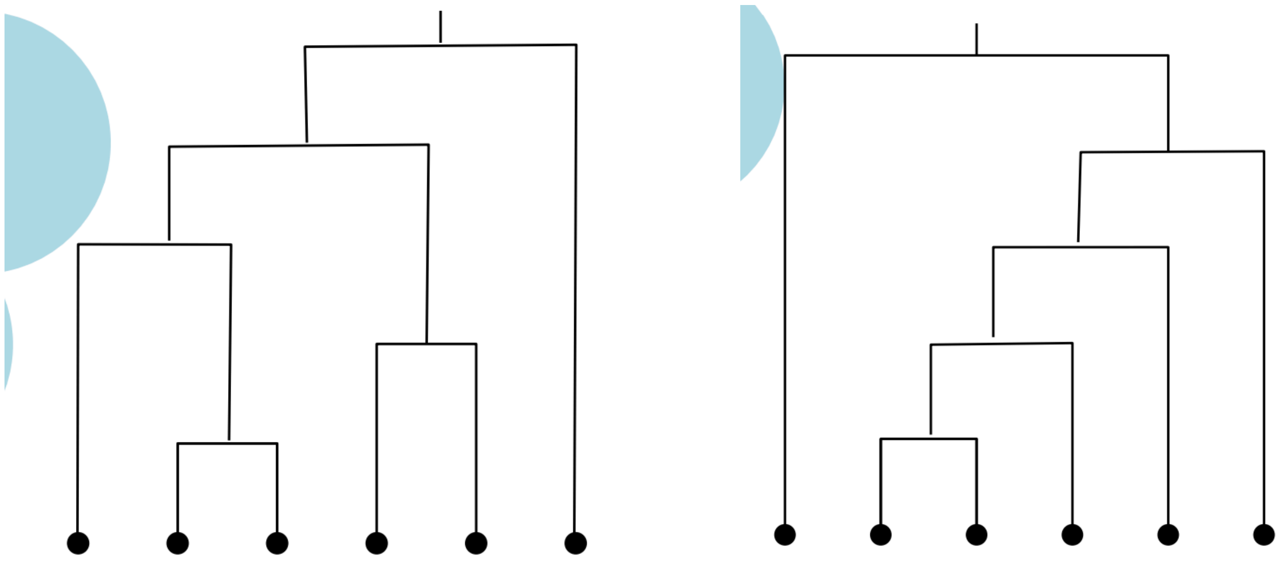
Problem: instability

- Unfortunately, this means this approach is not stable: similar point sets can yield very different clusters.

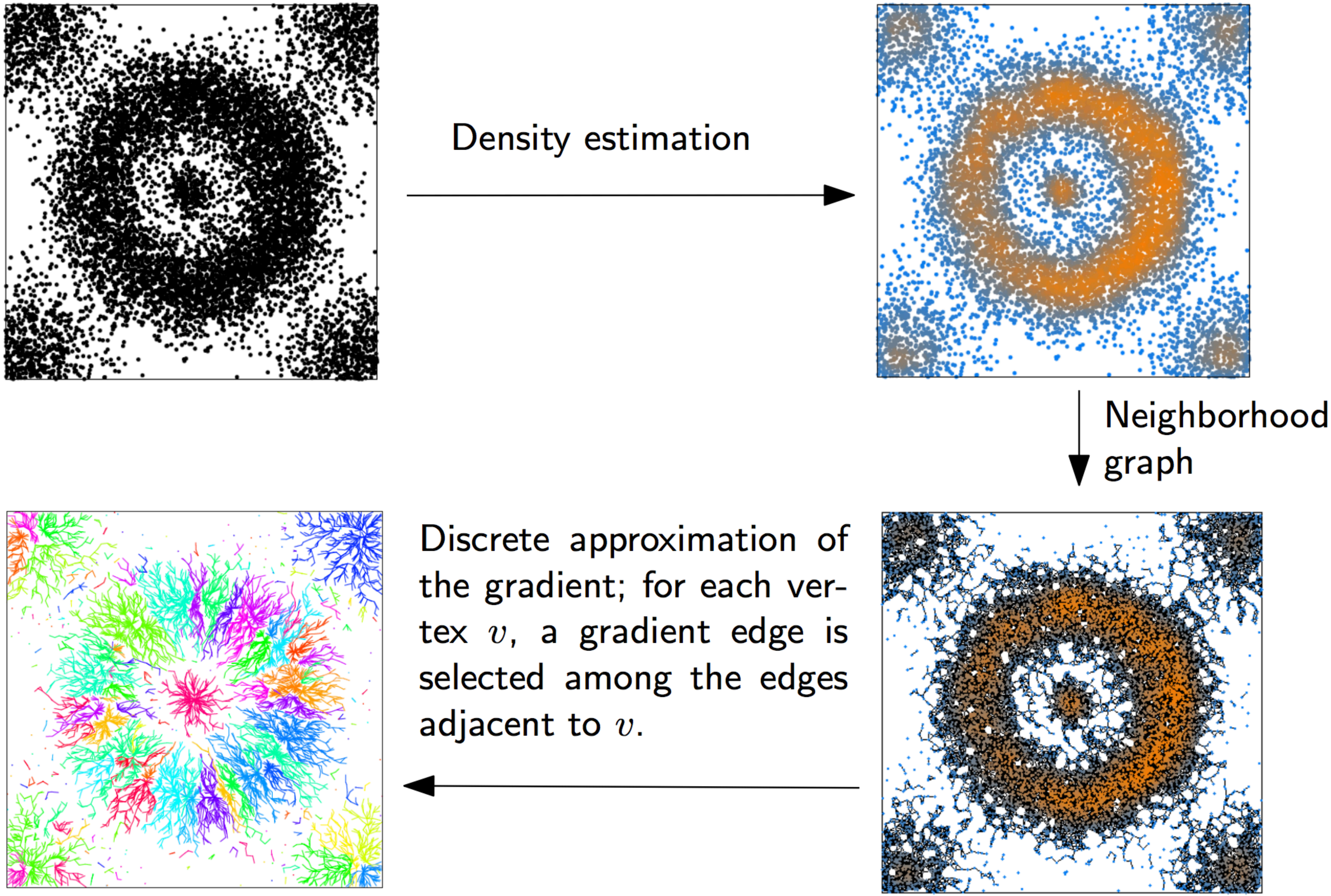


However…

- Small perturbations on the input data may lead to wide change in the structure of the trees.
- However, the “merging times” remain stable.
- Taking a close look: (At least for Euclidean data), the single linkage clustering keeps track of the evolution of the connected components of the distance function to the data.



Recall: Koonz, Narendra, and Fukunaga algorithm (from 1976)



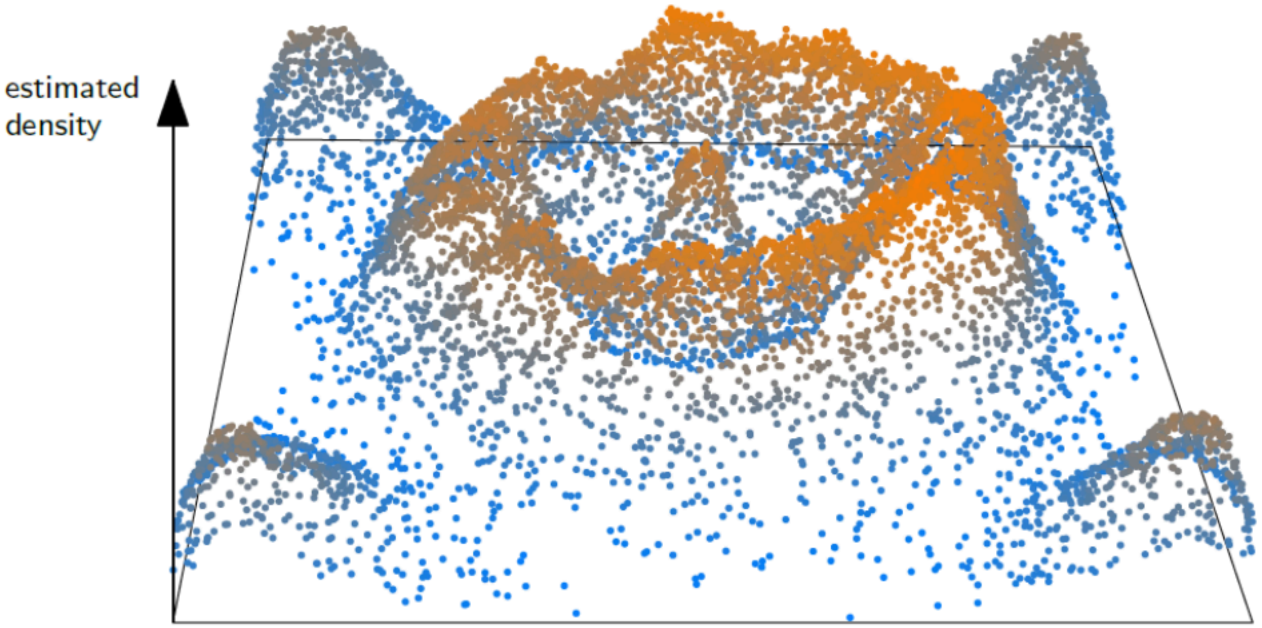
Density estimation

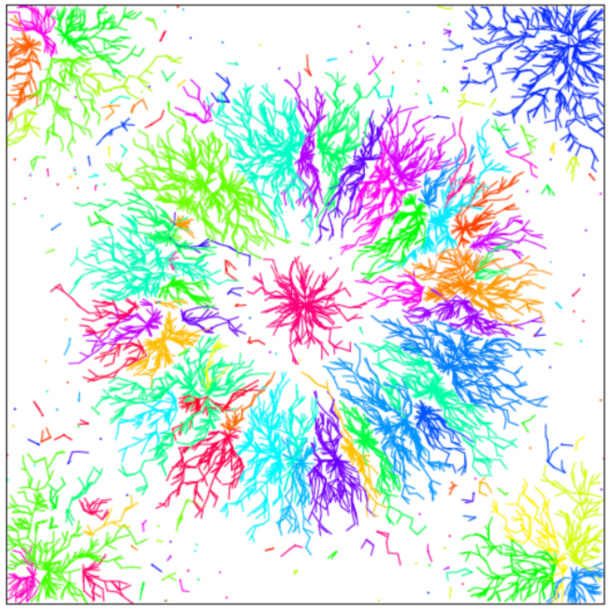
Neighborhood graph

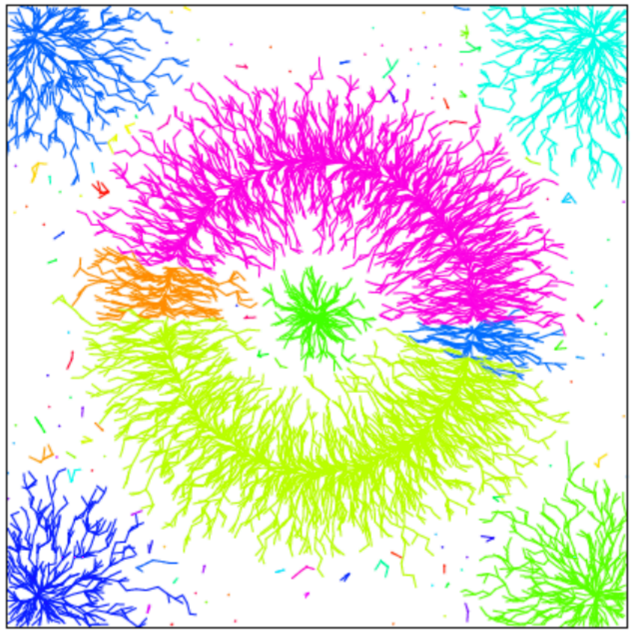
Discrete approximation of the gradient; for each vertex v, a gradient edge is selected among the edges adjacent to v.

Problem:
still sensitive to noise

- There are as many clusters as there are local maxima of the density function, which varies with neighborhood graph, so also quite sensitive to noise!





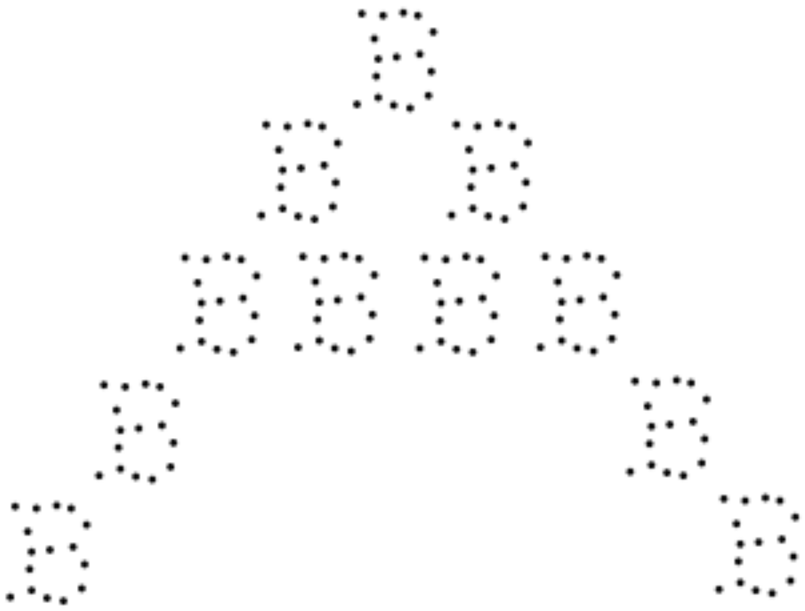


First: what is homology?

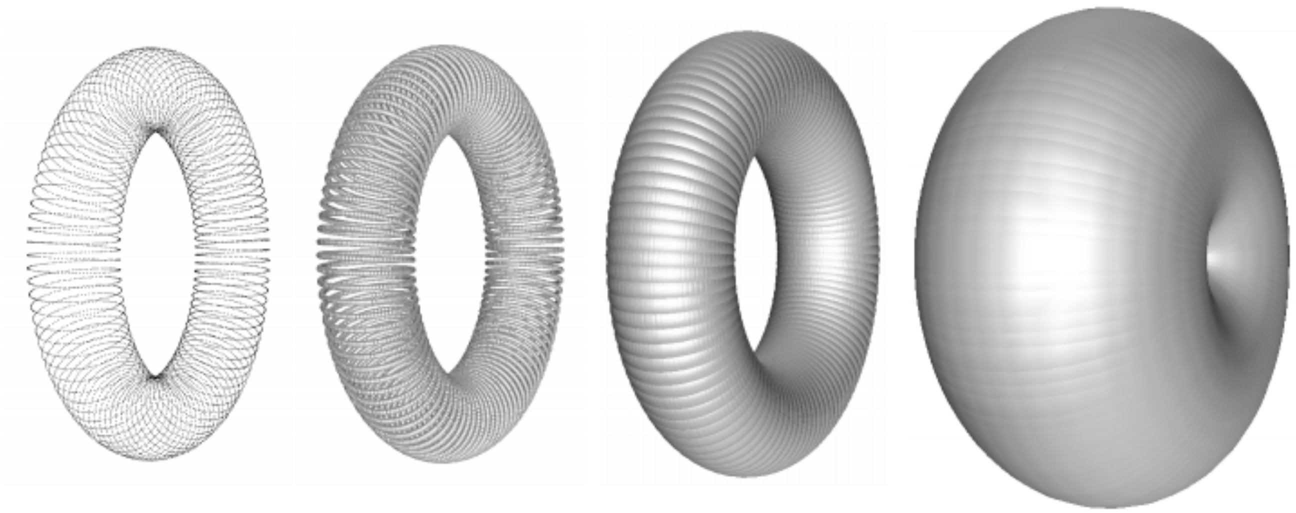
- More precisely, more often than not, people are interested in computing the ranks of the homology groups
    - Formally defined as the maximum number of linearly independent generators of a particular group
- I won’t get into math, but at a high level, we have 1 to k-dimensional homology groups for any k-dimensional structure
    - Rank of H_0 is the number of connected components
    - Rank of H_1 is the number of “handles” (or genus) in an orientable 2-manifold
    - Rank of H_3 captures the number of “voids” in a 3d-complex

Persistent homology motivation

Just to get us started:



Oudot 2015



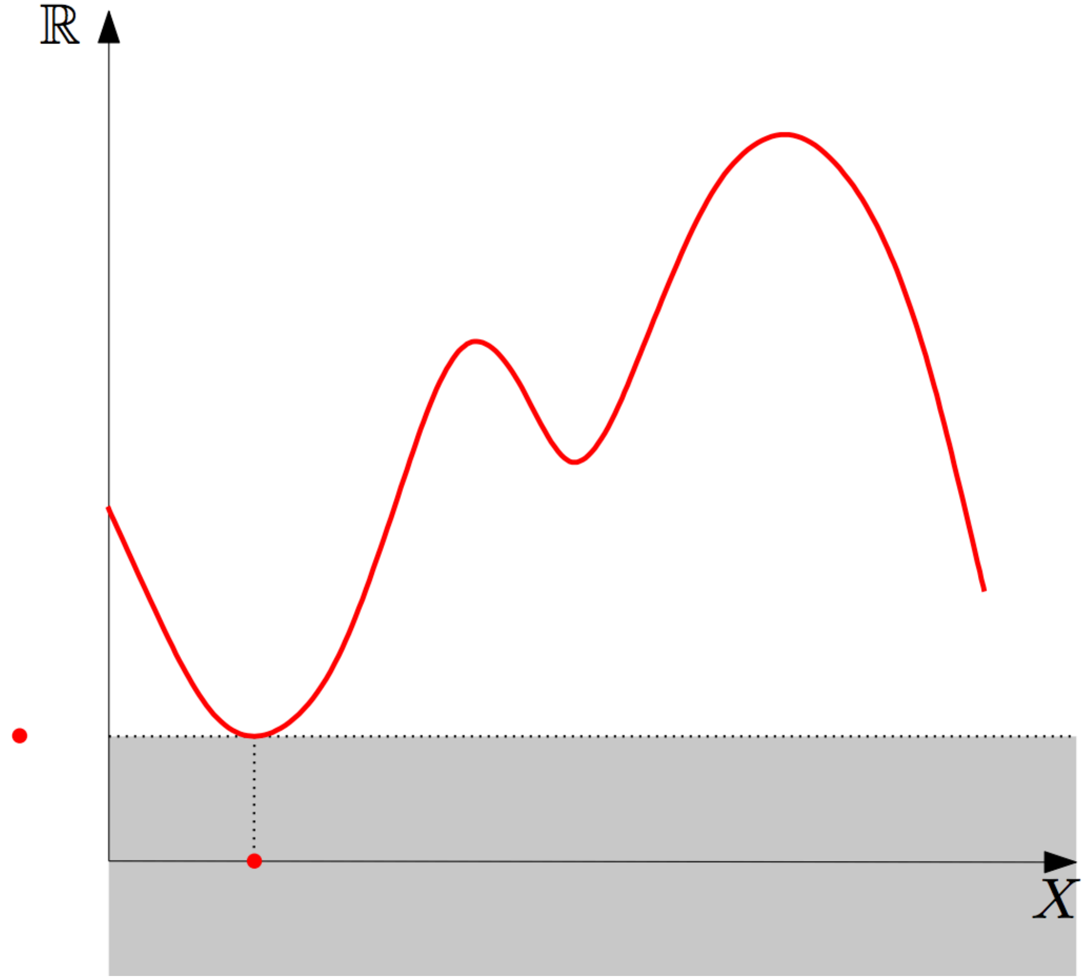
Oudot 2015

Key idea: depending on what scale we view the data, the question “what is the shape” may be quite different.

0-dimensional persistent homology on a function

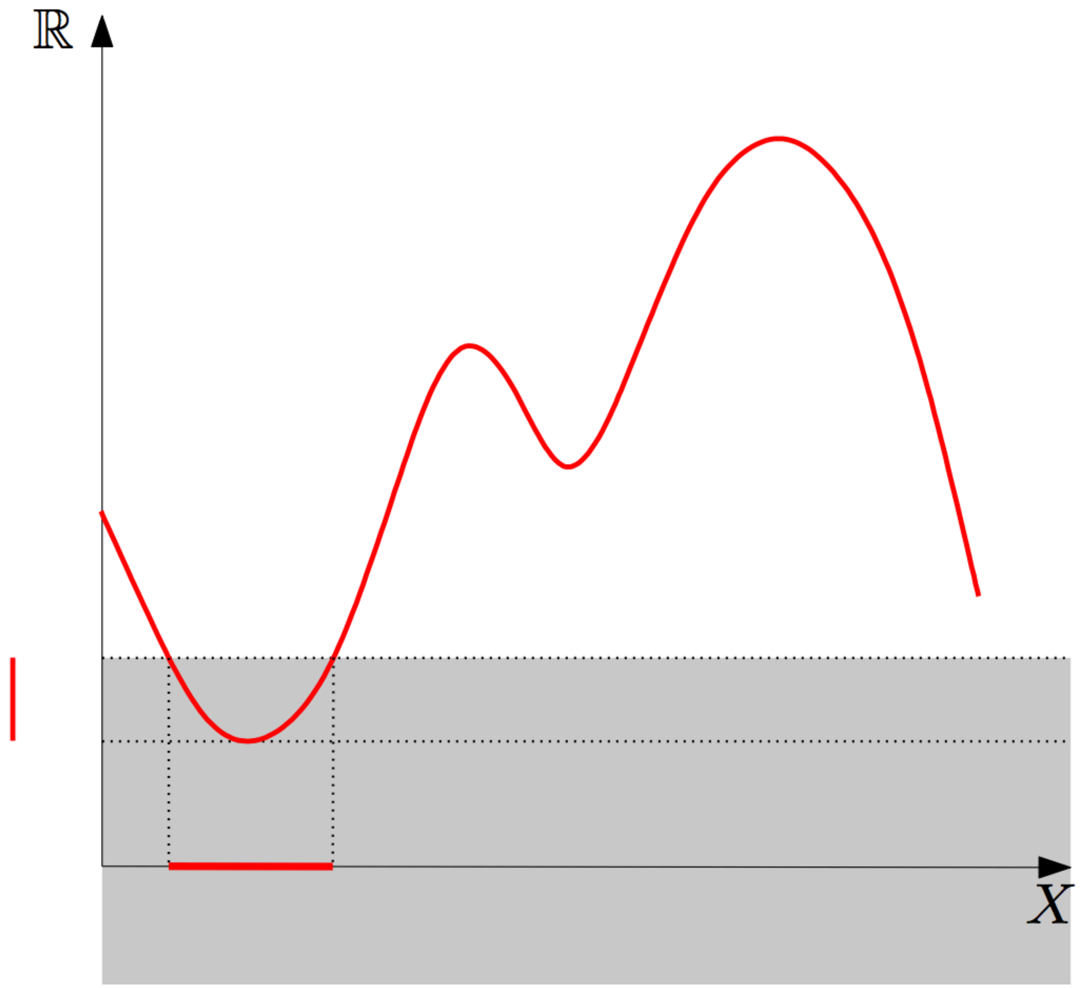
- Starting with a simple example:

  - Consider a function f, and track level sets and connectedness of $f^{-1}(-\infty,a)$, for a = -∞ to ∞:



0-dimensional persistent homology on a function

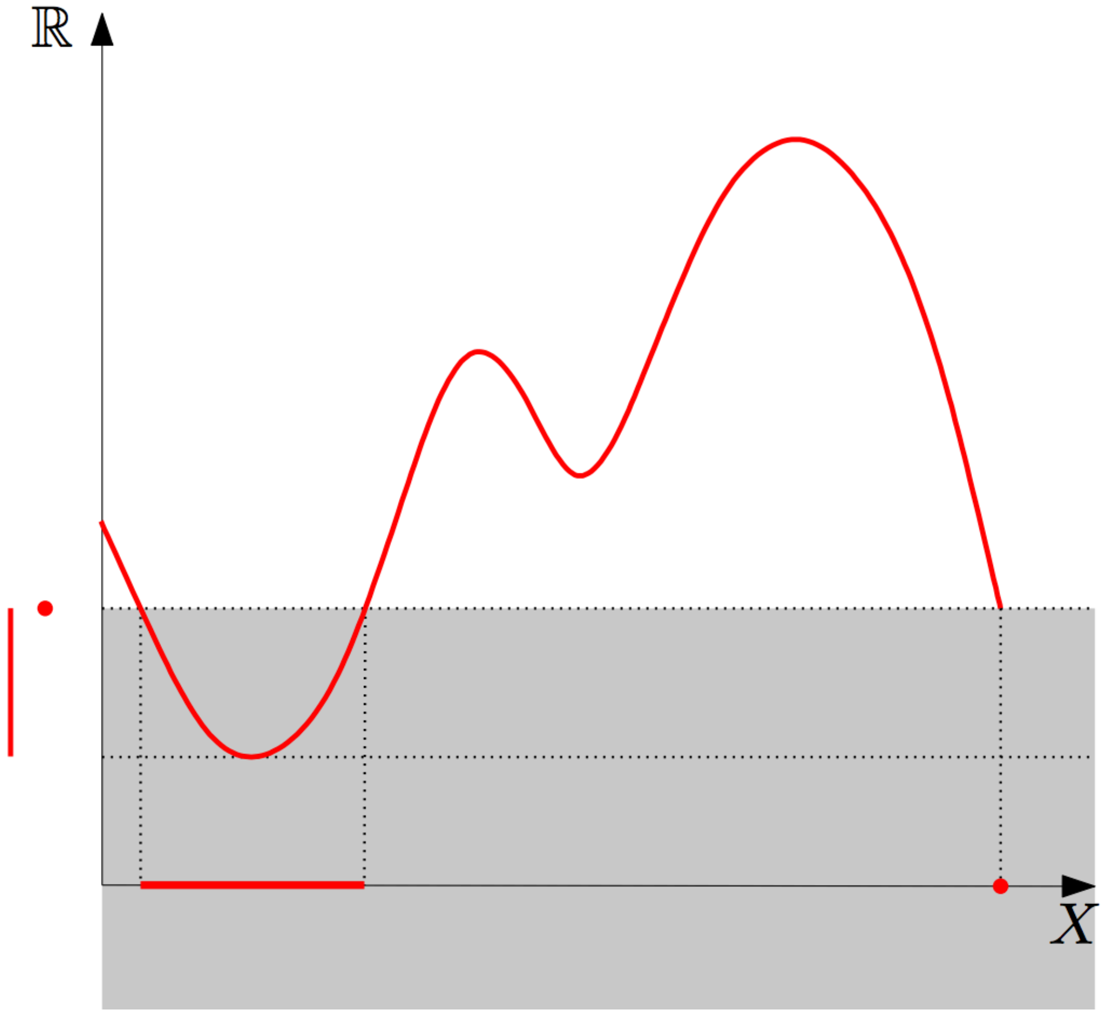
- Starting with a simple example:
  - Consider a function f, and track level sets and connectedness of $f^{-1}(-\infty,a)$, for a = $-\infty$ to ∞:



0-dimensional persistent homology on a function

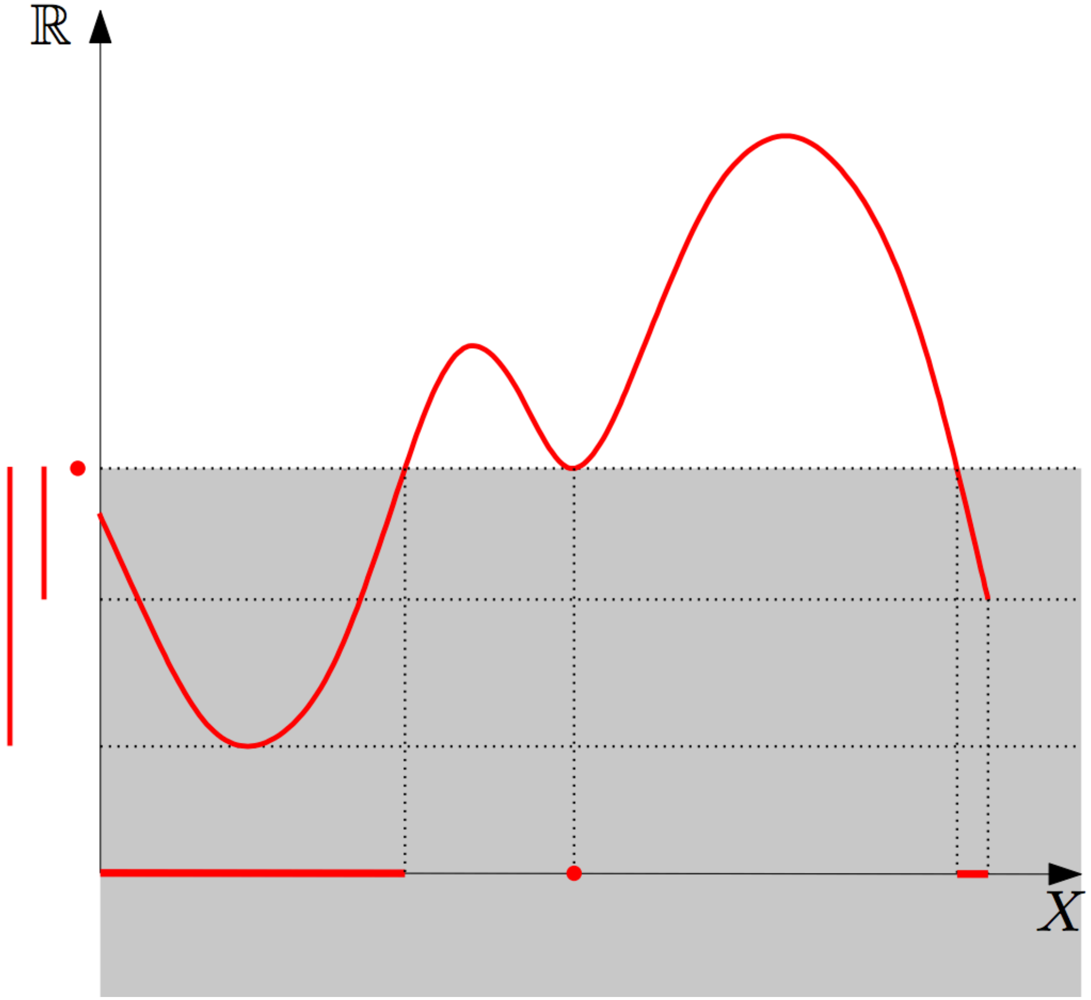
- Starting with a simple example:

  - Consider a function f, and track level sets and connectedness of $f^{-1}(-\infty,a)$, for a = -∞ to ∞:



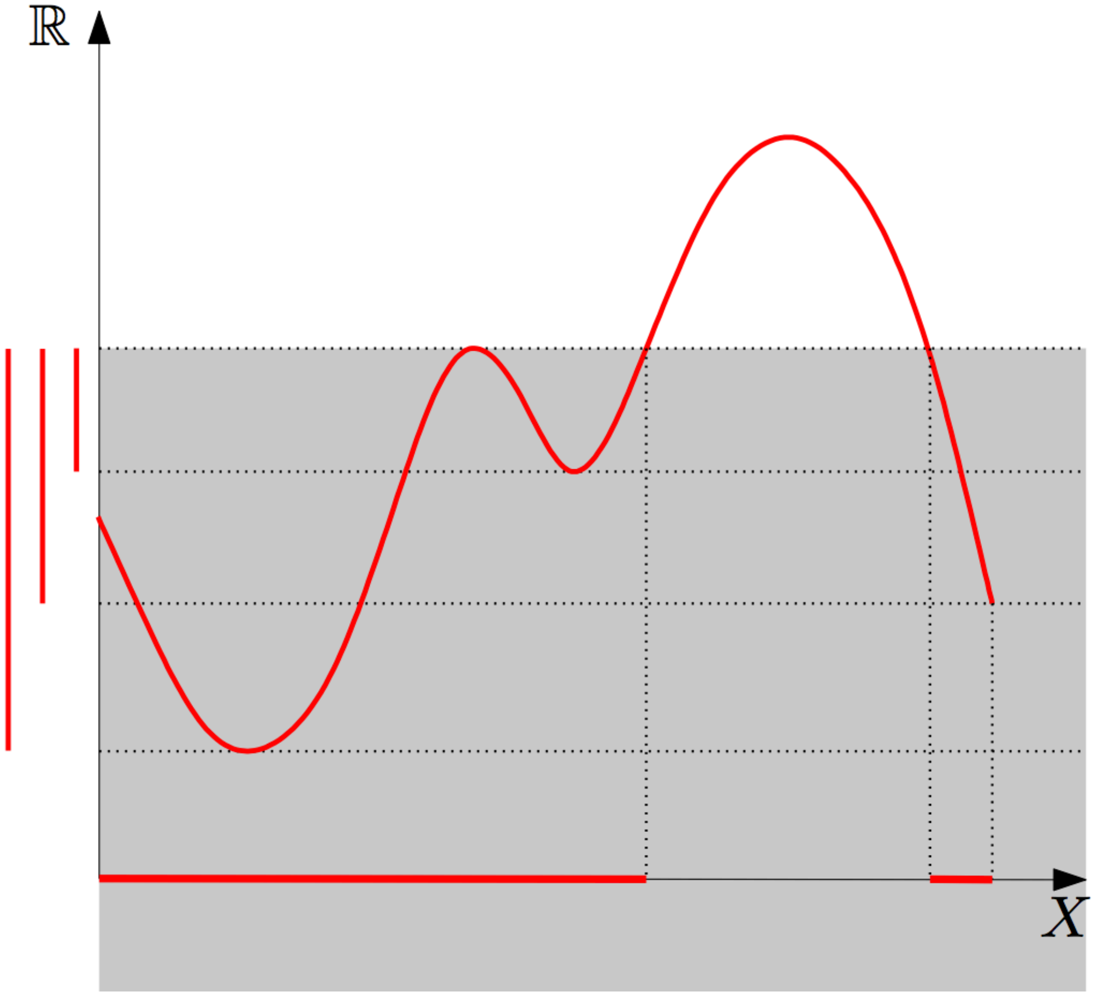
0-dimensional persistent homology on a function

- Starting with a simple example:
  - Consider a function f, and track level sets and connectedness of $f^{-1}(-\infty,a)$, for a = -∞ to ∞:



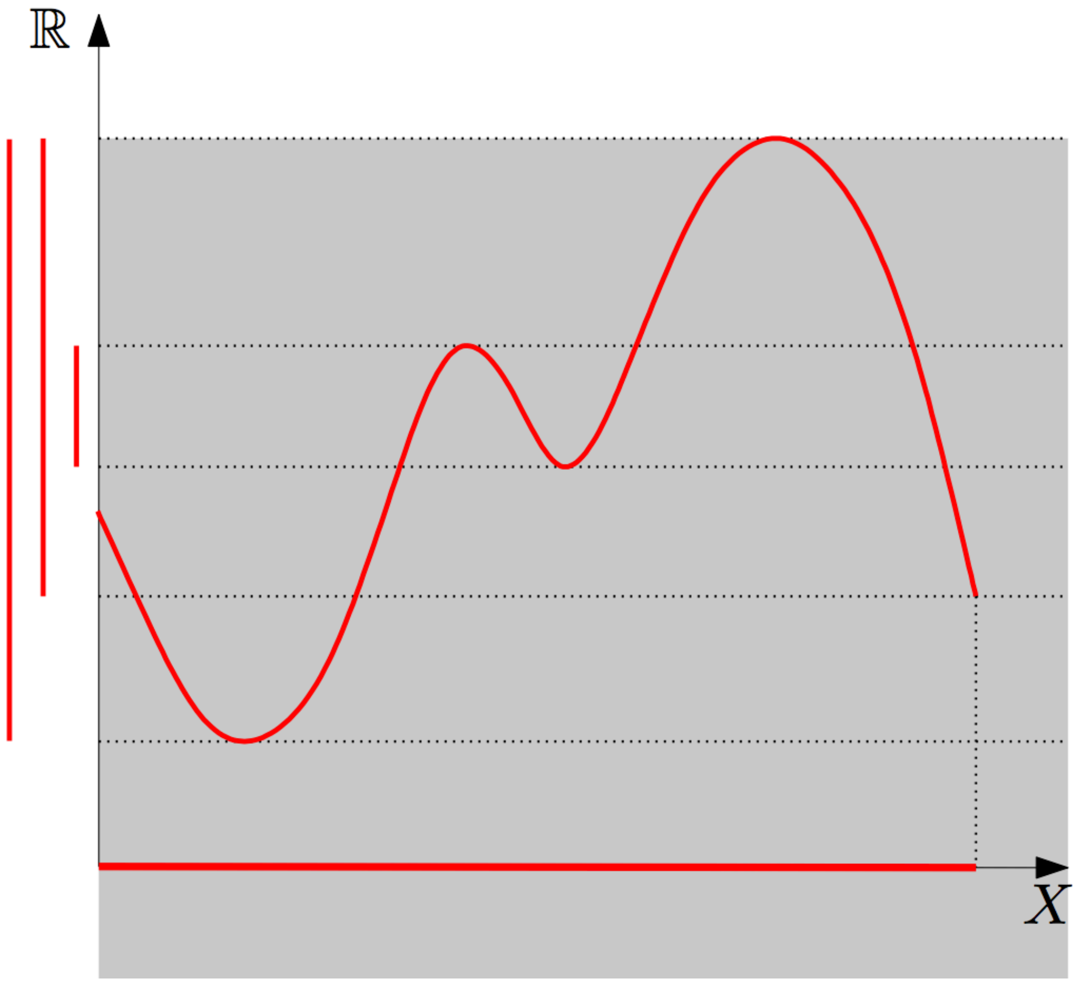
0-dimensional persistent homology on a function

- Starting with a simple example:
  - Consider a function f, and track level sets and connectedness of $f^{-1}(-\infty, a)$, for a = $-\infty$ to ∞:



0-dimensional persistent homology on a function

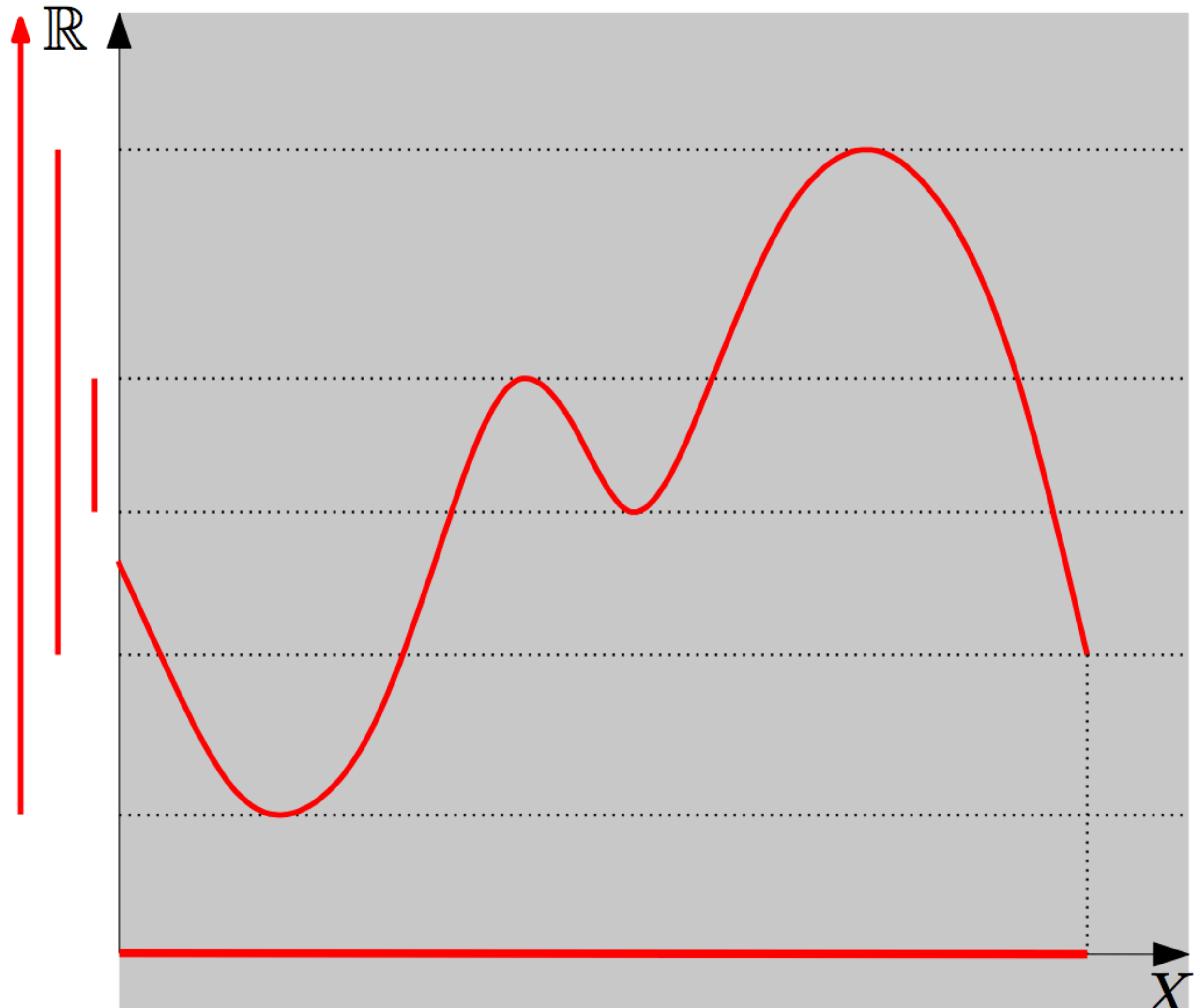
- Starting with a simple example:
  - Consider a function f, and track level sets and connectedness of $f^{-1}(-\infty,a)$, for a = $-\infty$ to ∞:



0-dimensional persistent homology on a function

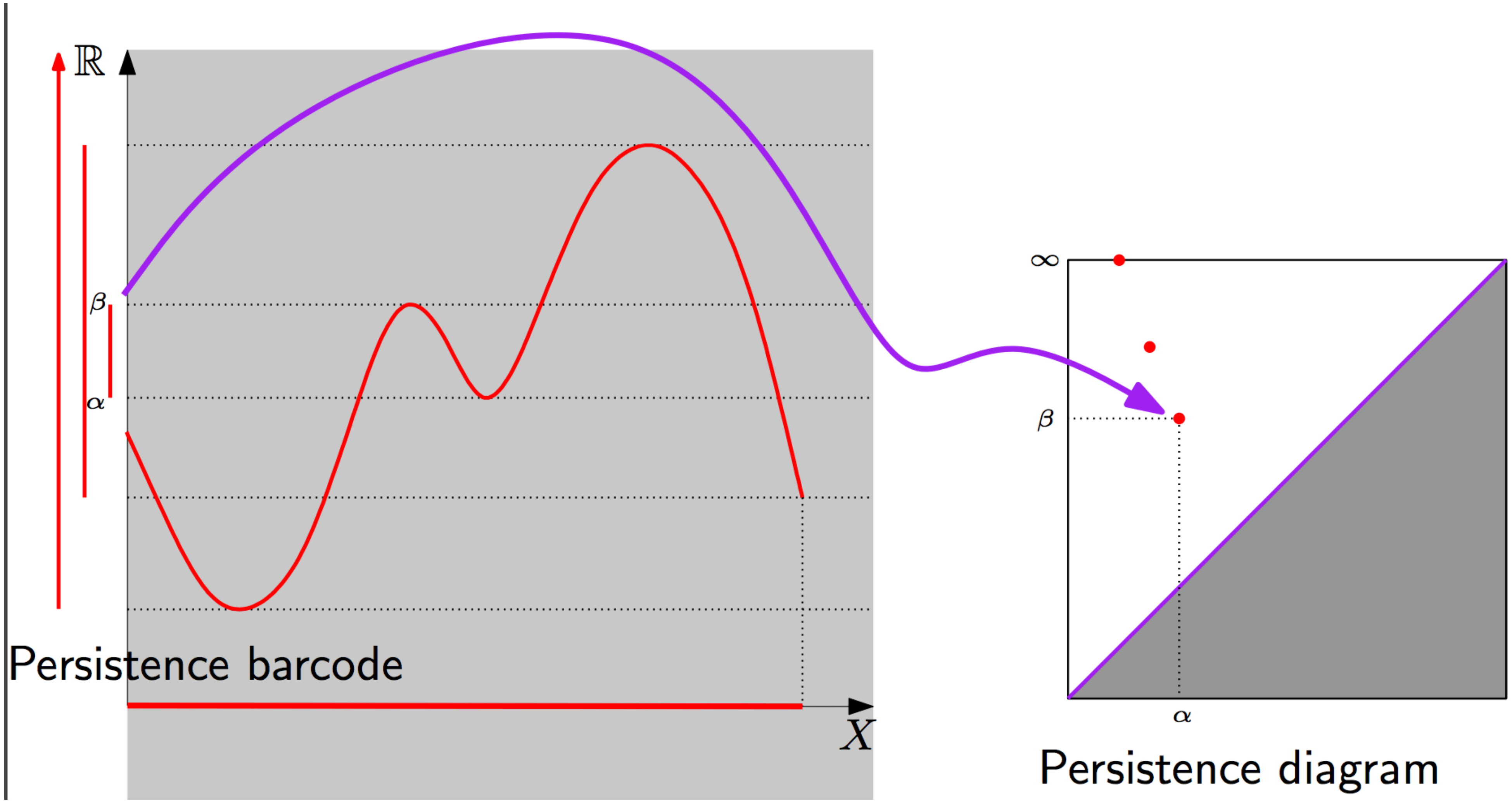
- Starting with a simple example:

  - Consider a function f, and track level sets and connectedness of $f^{-1}(-\infty,a)$, for a = -∞ to ∞:



0-dim PH of a function:

- The result: a representation of which features are "persistent", in the sense that they last a longer or shorter time as the input evolves.



How to go beyond functions?

- Core notion:

  - A k-simplex is a k-dimensional polytope which is the convex hull of its k+1 (affinely independent) vertices

  - A simplicial complex K is a collection of simplices where (1) any face of a simplex is in K and (2) the intersection of any two simplicies in K is also a simplex

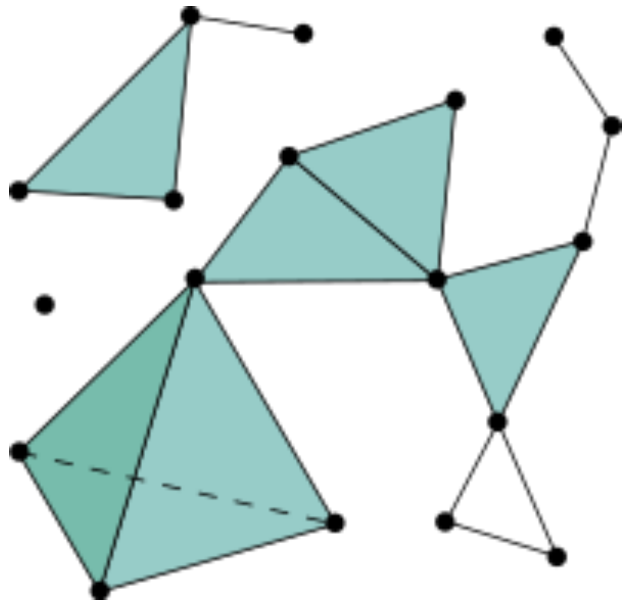


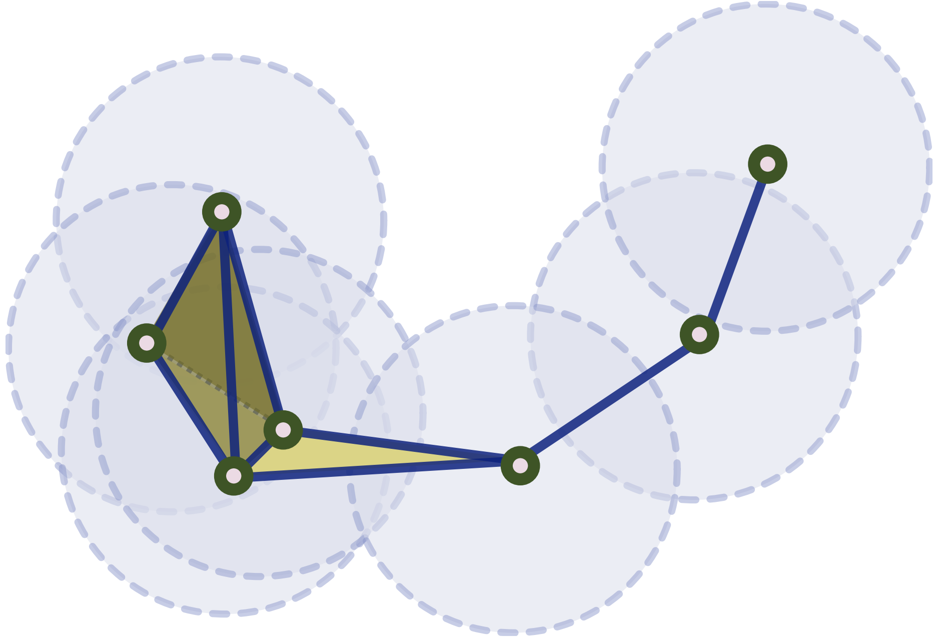
Image courtesy of wikipedia

The Ćech complex

- Given a point set X, consider a cover built from all radius ε balls centered at points x∈X.

- The Ćech complex is the nerve of this covering, where we fill in a k-simplex $\{i_0, i_1,\ldots,i_k\}$ is included whenever $Bi_0 \cap\ldots\cap Bi_k \neq \varnothing$.

    - Essentially, a k-simplex appears whenever k balls intersect.



Vietoris-Rips complex

- Finally, the Rips complex includes all simplicies where pairwise distances are less than 2r (even if the common intersection is empty)

  - $VR(x,r) = \{\sigma \subseteq X \mid B_r(x) \cap B_r(y) \neq \varnothing \text{ for all } x,y \in \sigma\}$

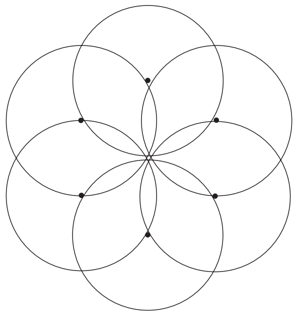
- This is strictly larger than the Ćech: can have pairwise distances ≤ 2r even when no common point.

- Both are commonly used to represent those neighborhood “graphs” of input points.

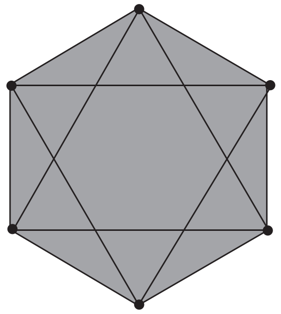
Rips versus Ćech

- The topology of these can be VERY different.

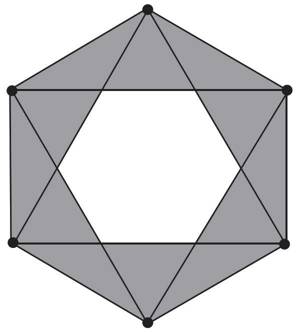
- Even for planar point sets:



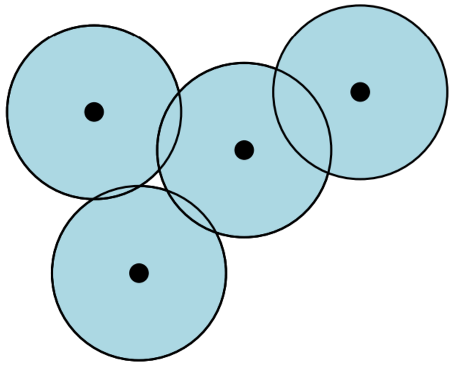
points

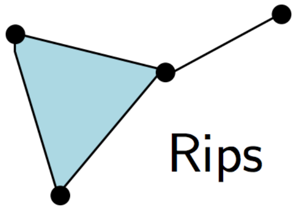


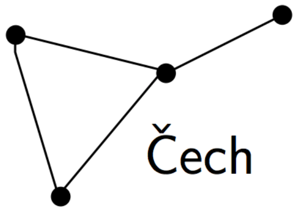
Rips



Ćech

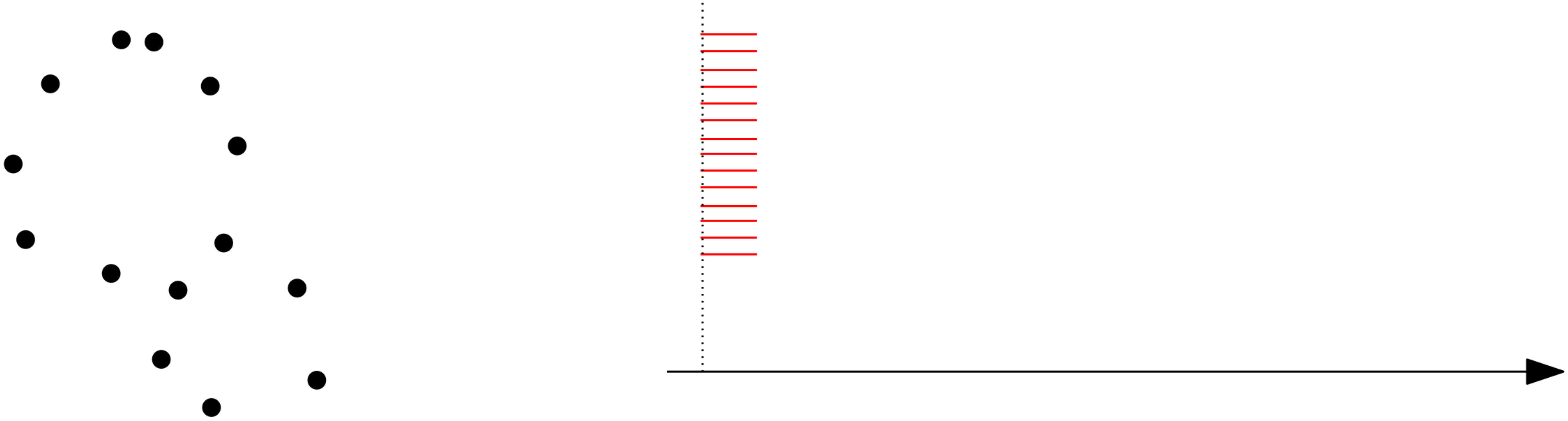






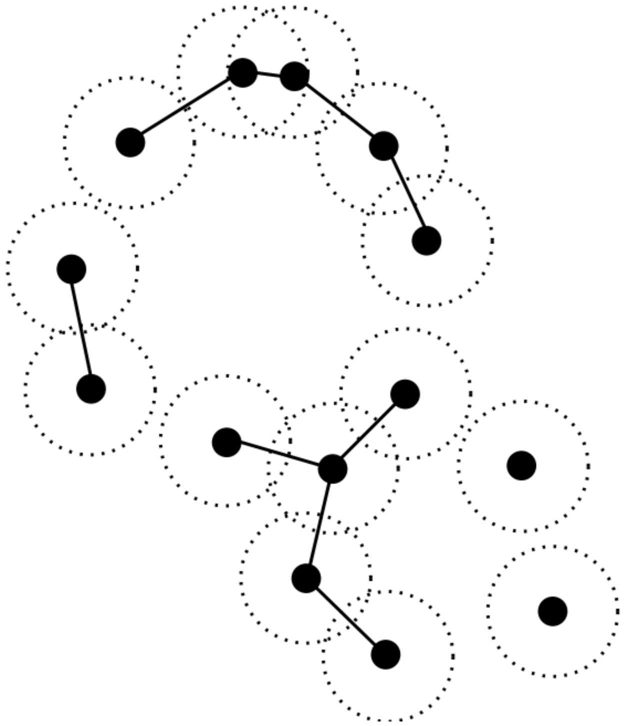
Persistence for point cloud data

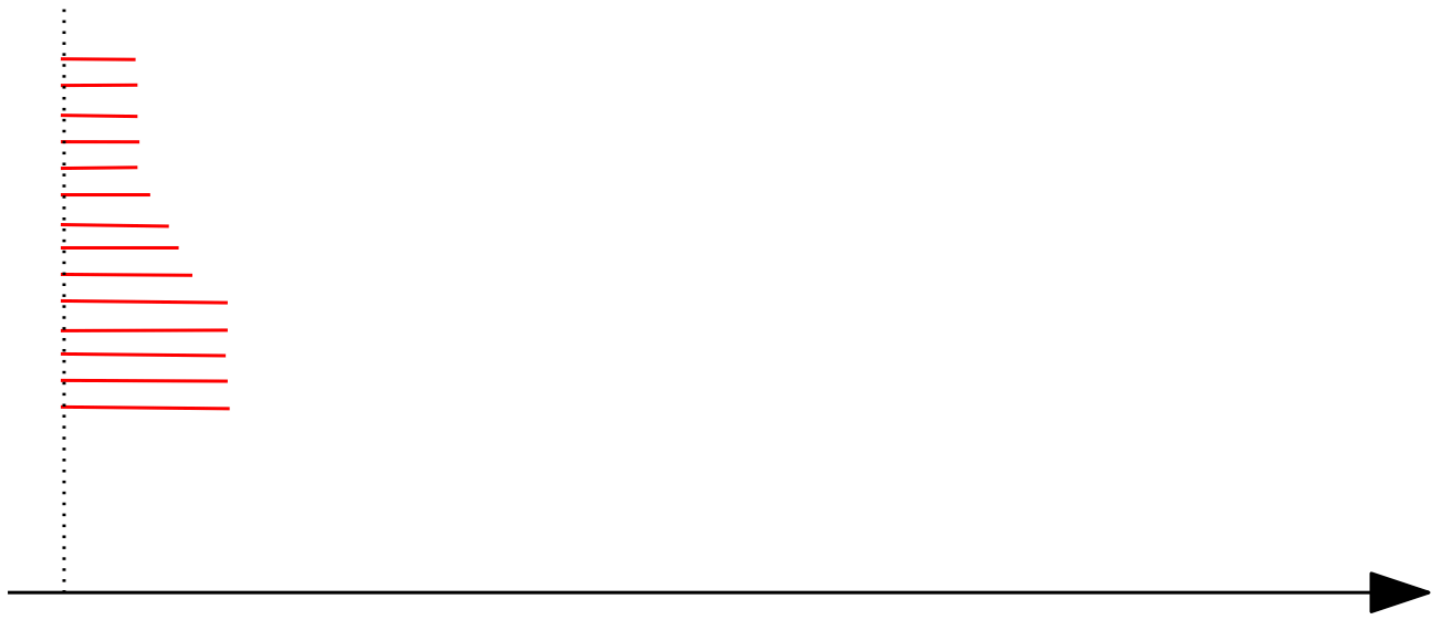
- Build the same connectivity in 2d:



Persistence for point cloud data

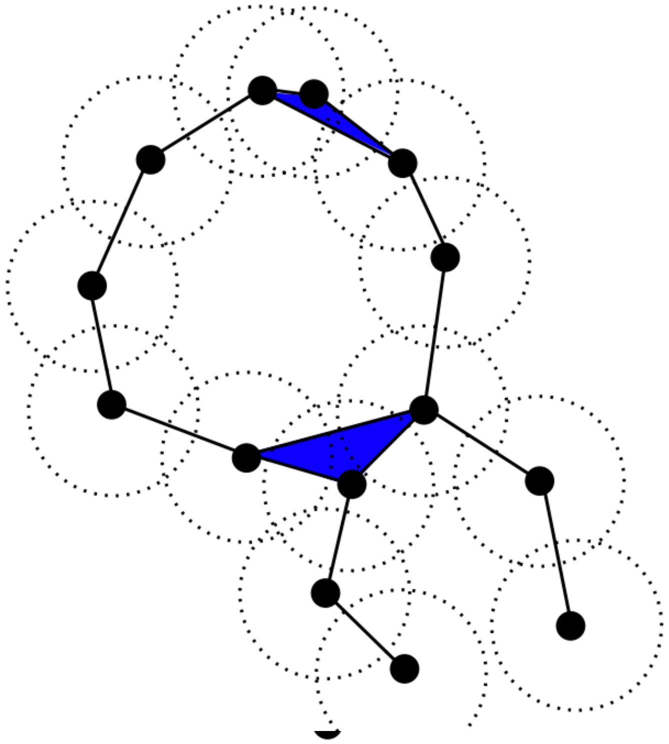
- Build the same connectivity in 2d:

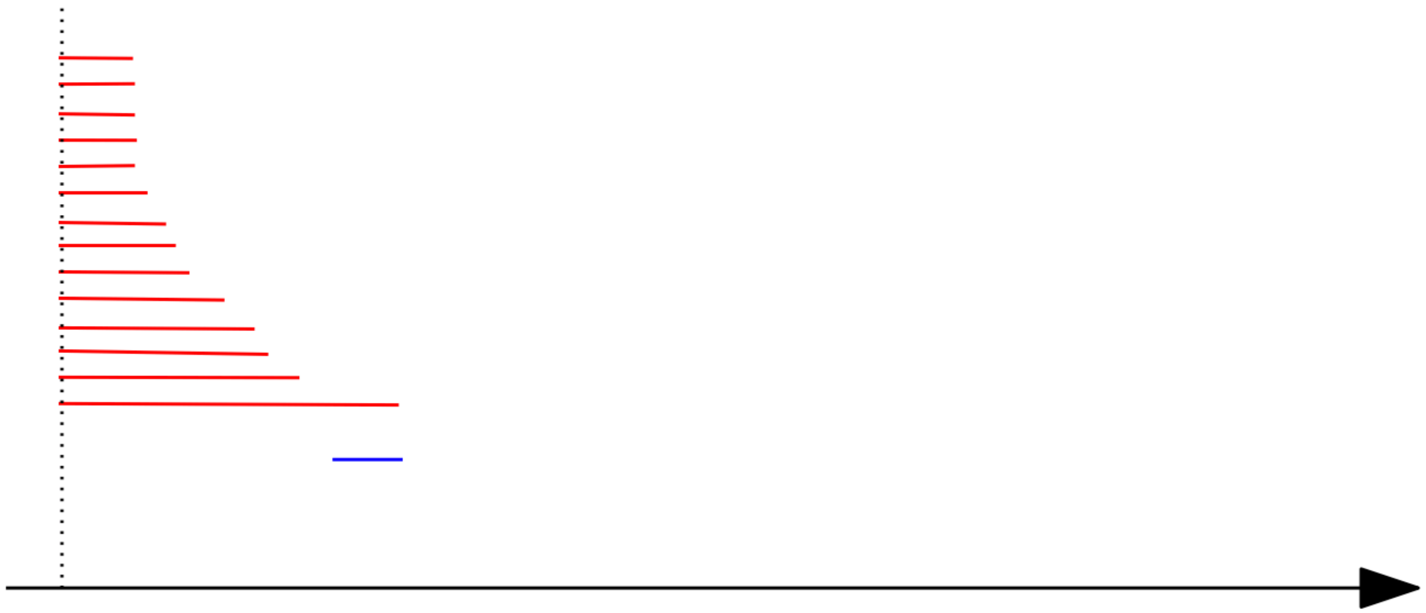




Persistence for point cloud data

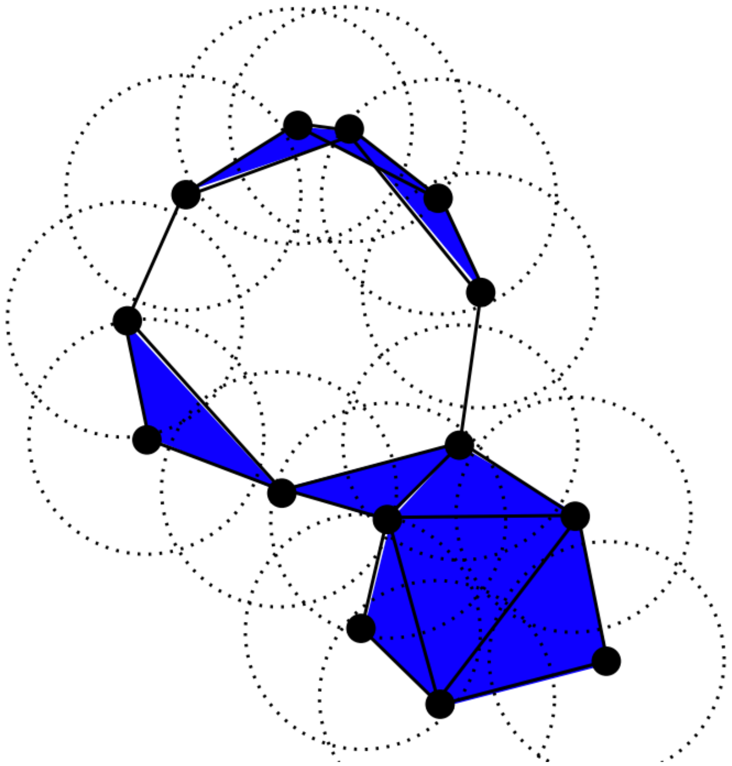
- Build the same connectivity in 2d:

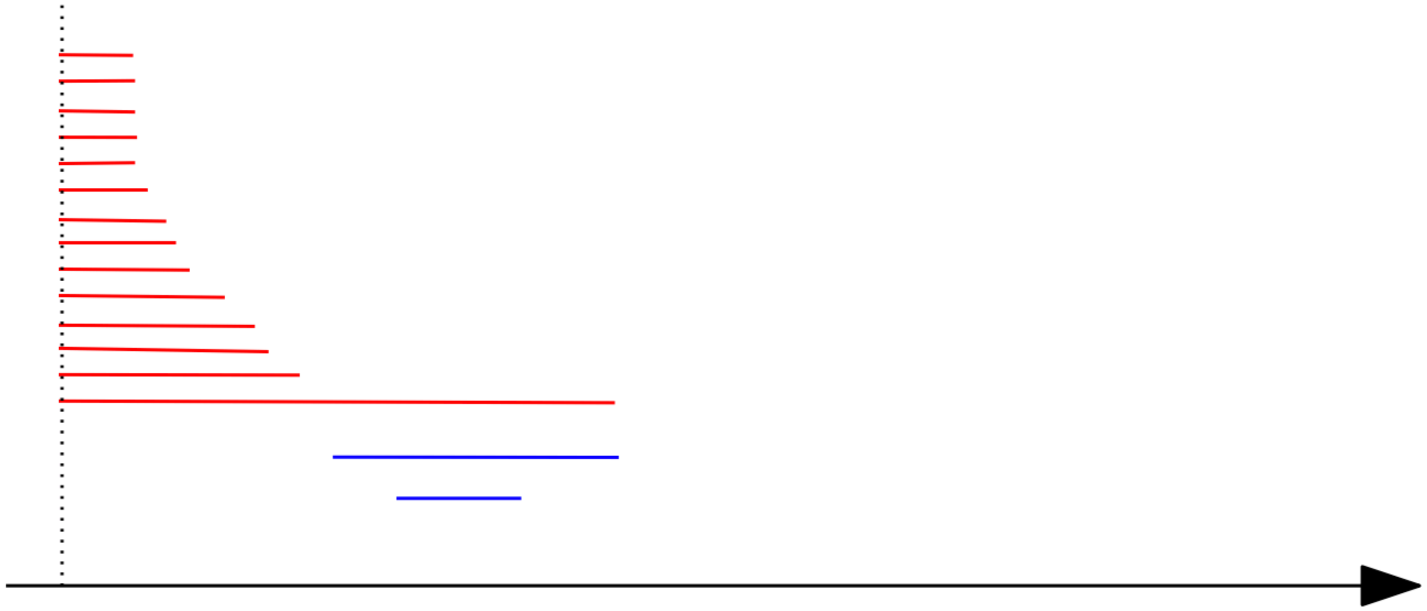




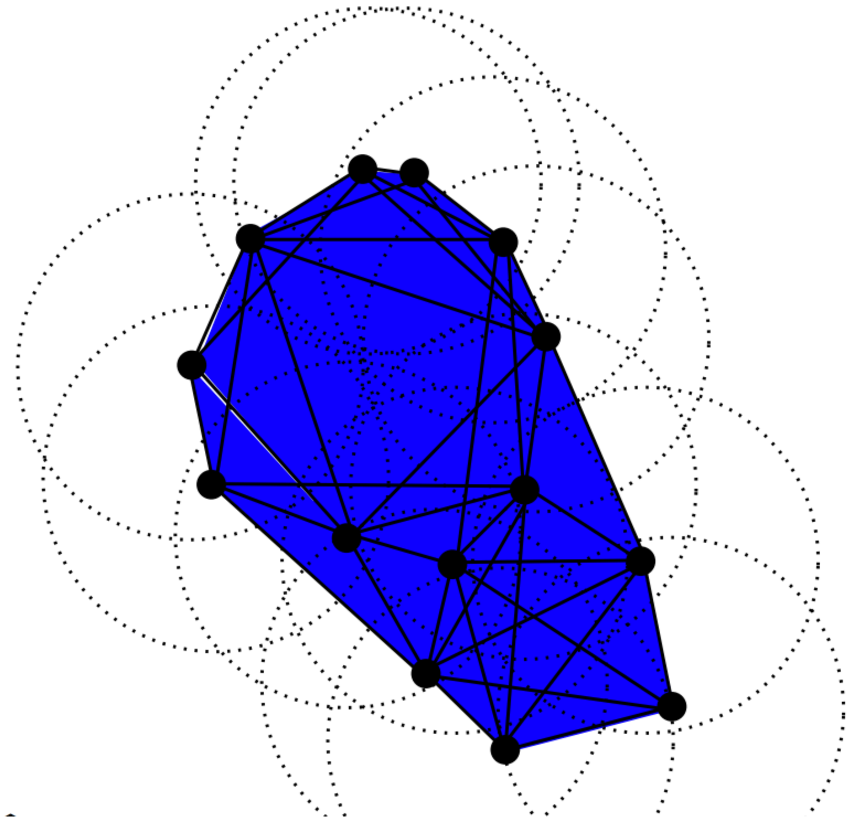
Persistence for point cloud data

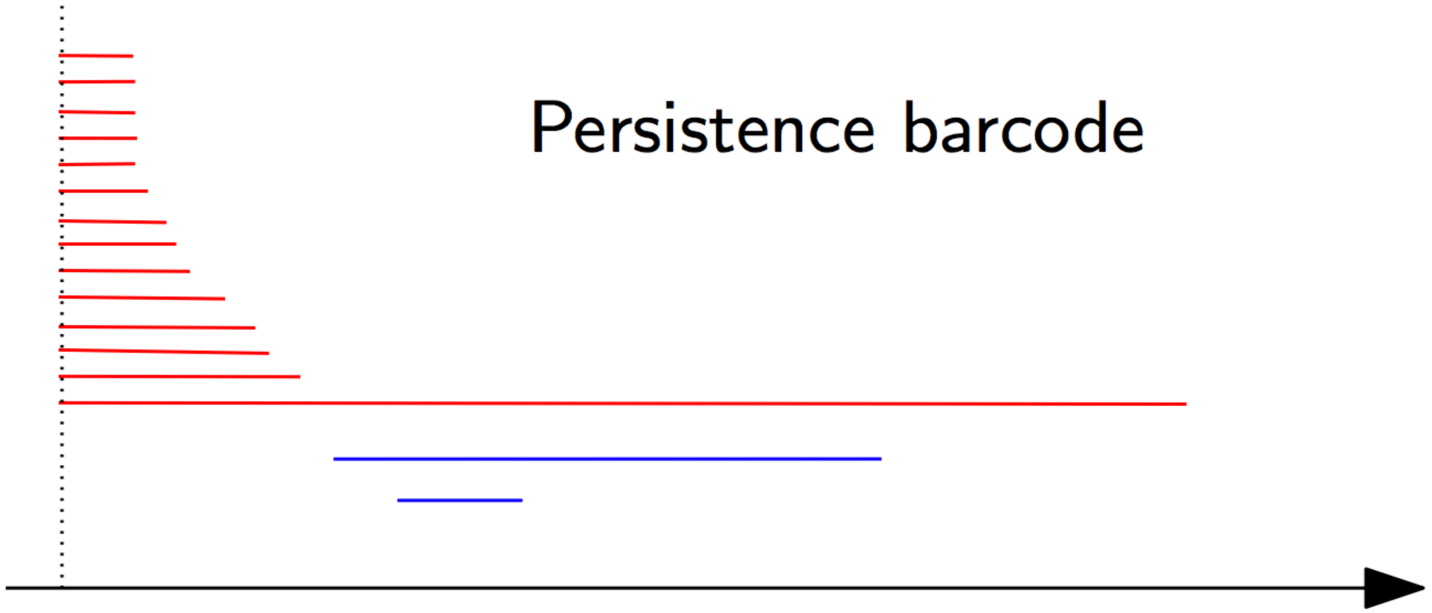
- Build the same connectivity in 2d:

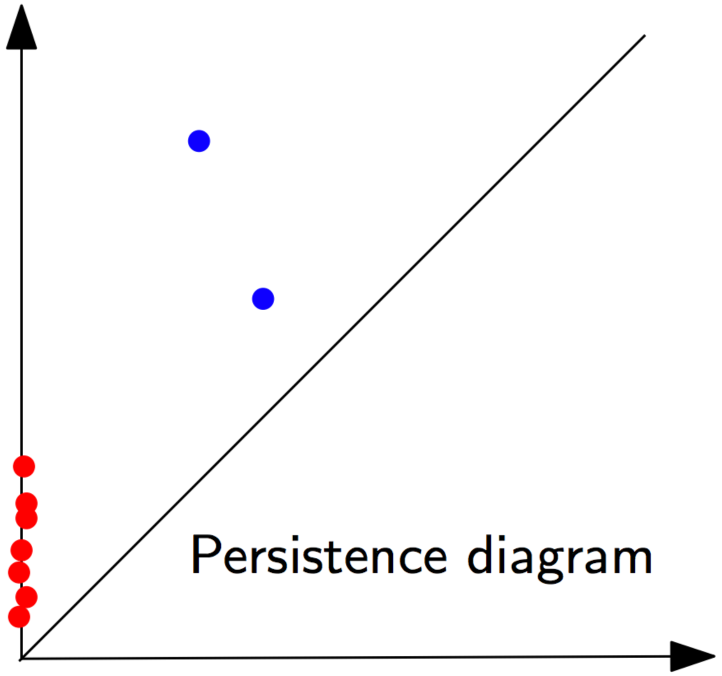




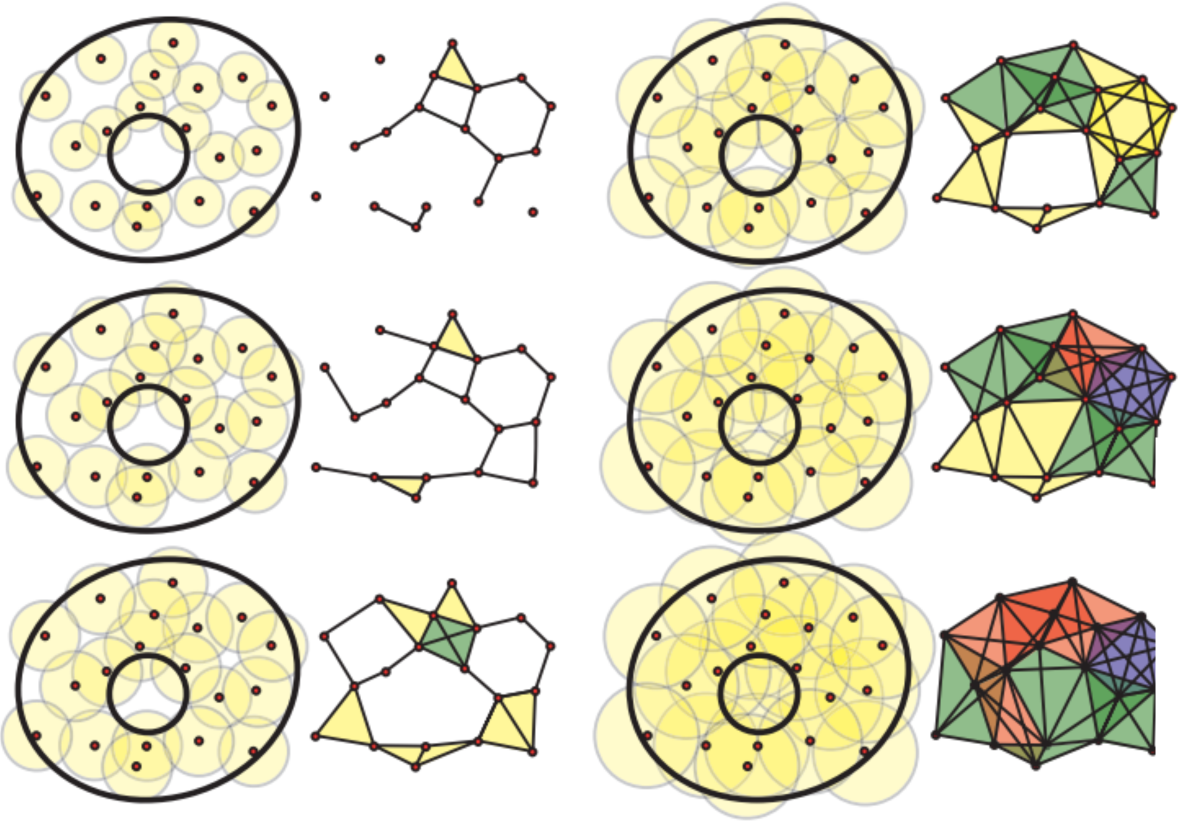
Persistence for point cloud data



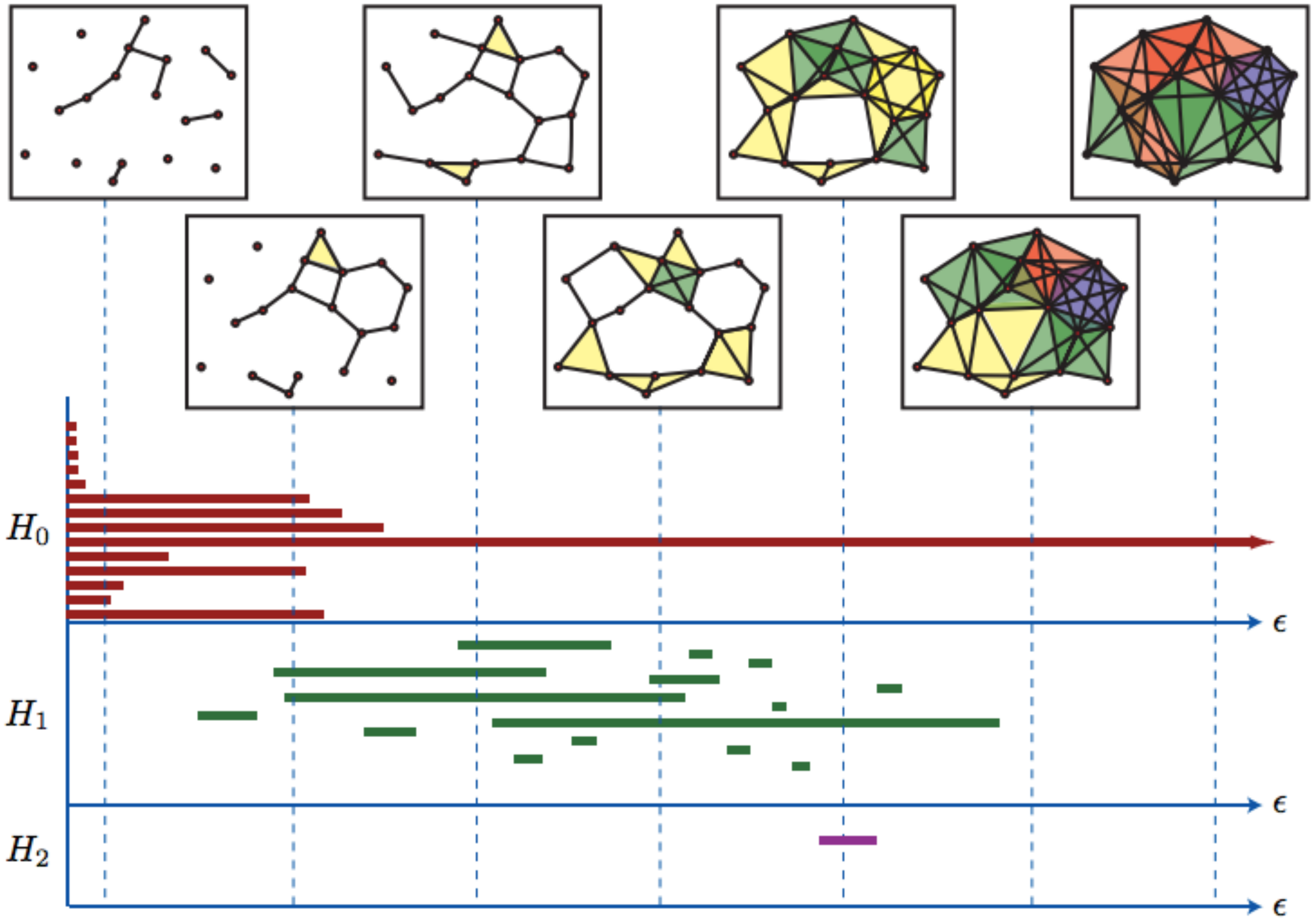




A more complex example: Rips complexes

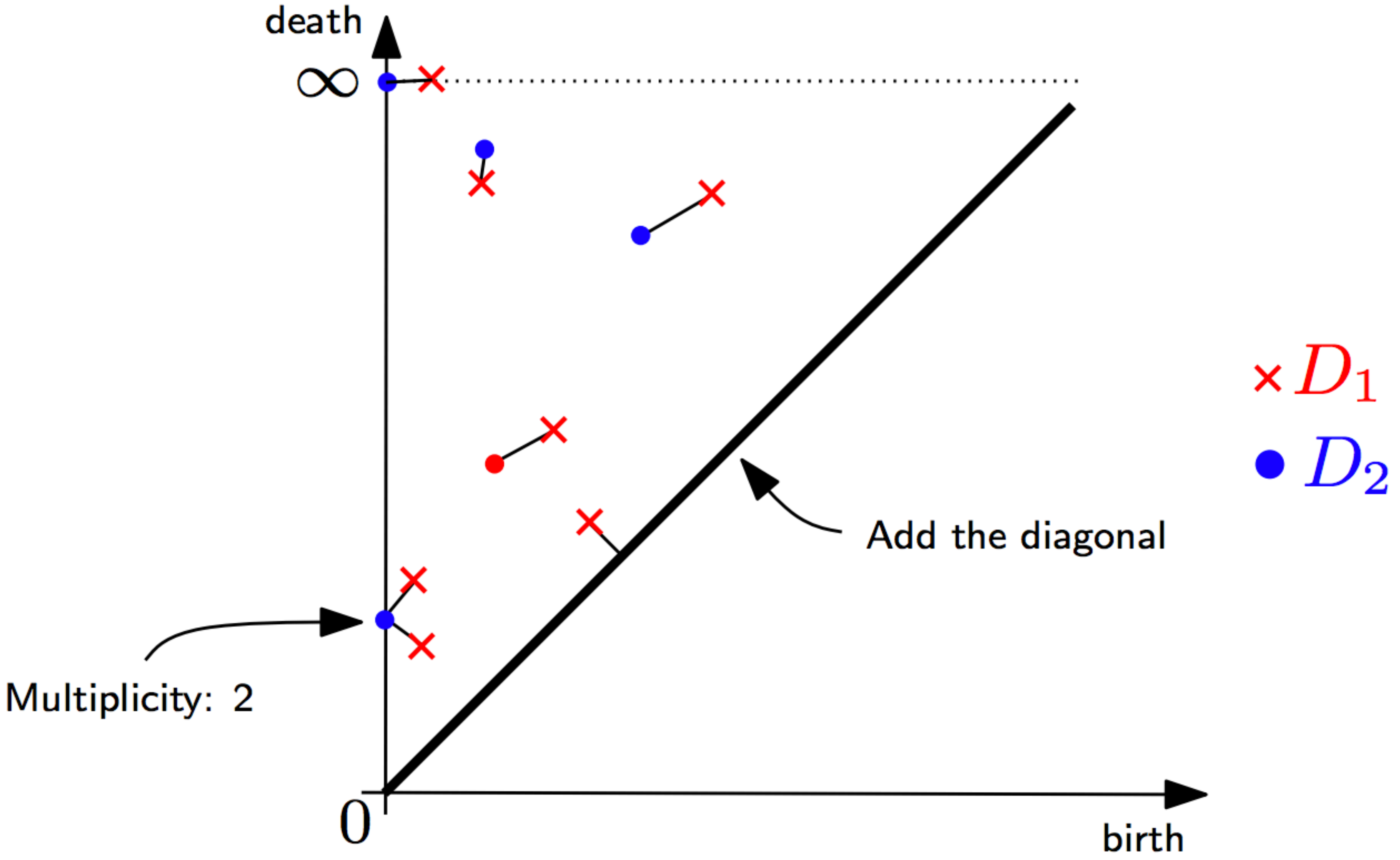


Ghrist 2008



Why do we care?

- Stability! We can define a distance, called the bottleneck distance, between these persistence diagrams:



Another example to illustrate stability

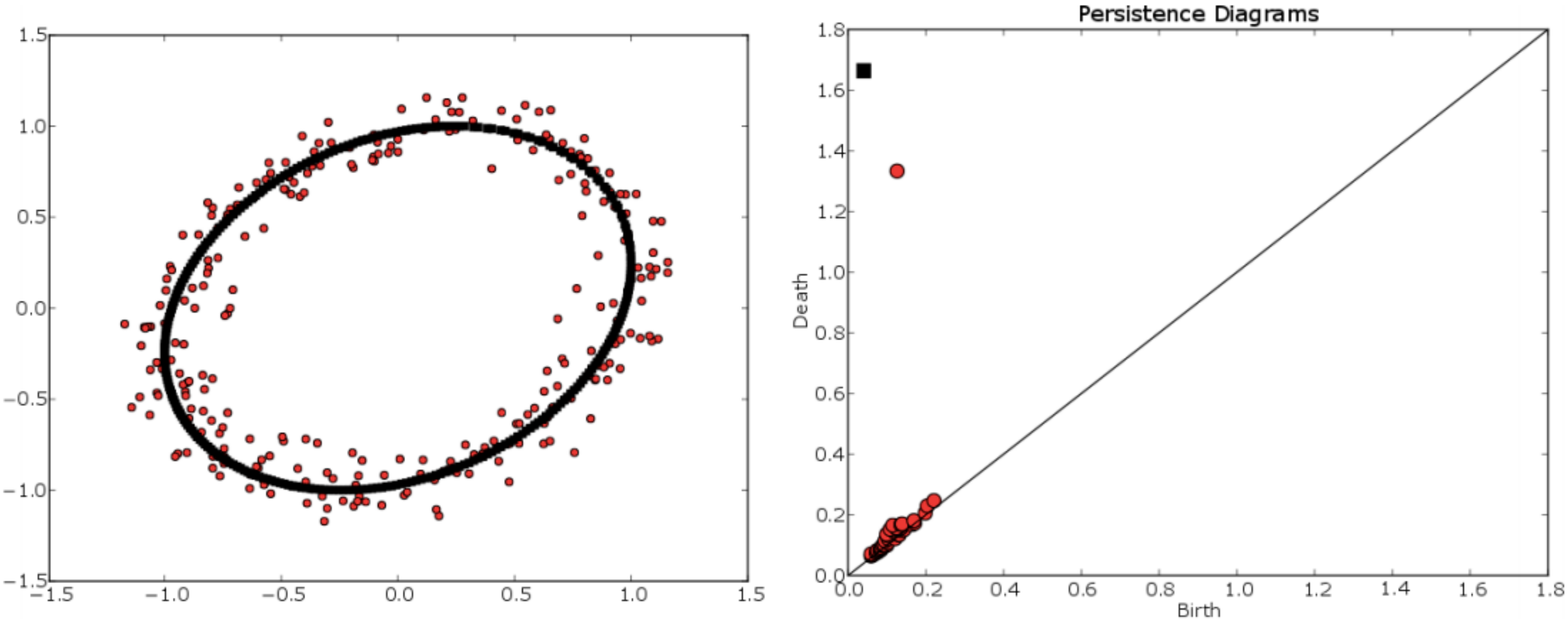
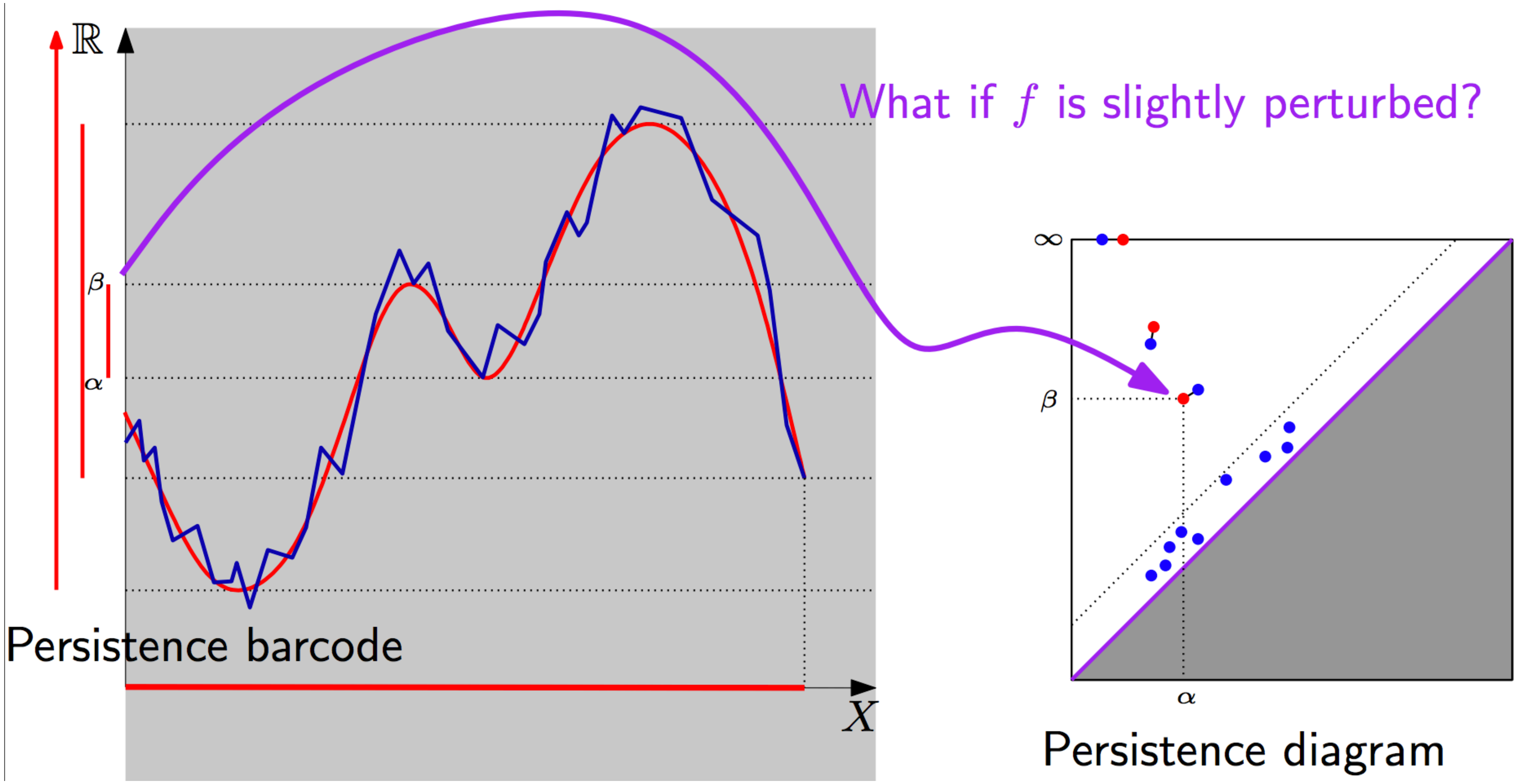


Figure 4: Two example point clouds are overlaid at left, and their persistence diagrams are overlaid at right. Notice that the point clouds are close in some sense. The fact that the persistence diagrams are also close is a result of the stability theorem for persistence.

Munch 2017

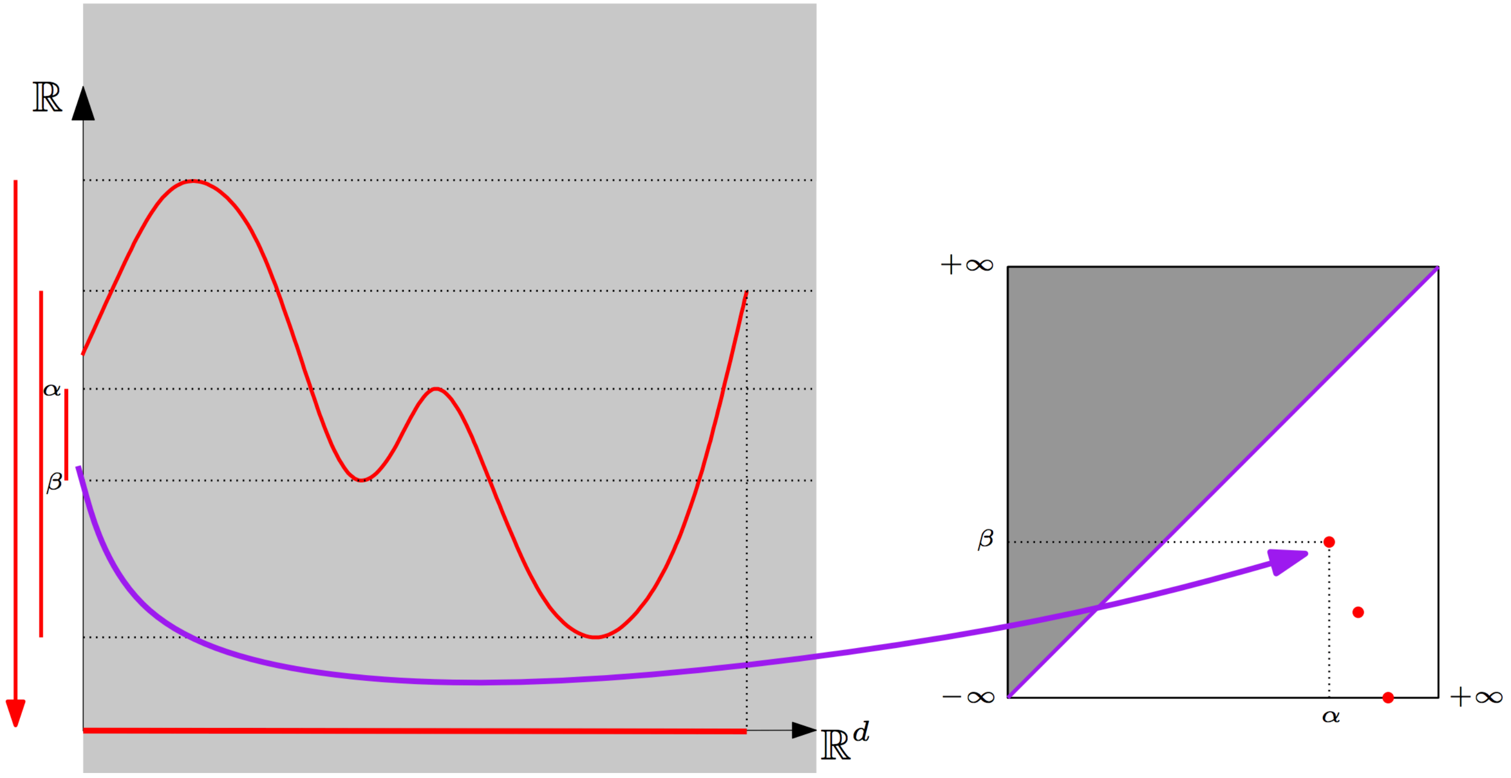
Stability result

- Theorem: [Cohen-Steiner et al 2005, Chazal et all 2009, de Silva et al 2012]: For any 2 “tame” functions f and g:
$$d(D_f, D_g) \leq ||f - g||_\infty$$


What if f is slightly perturbed?

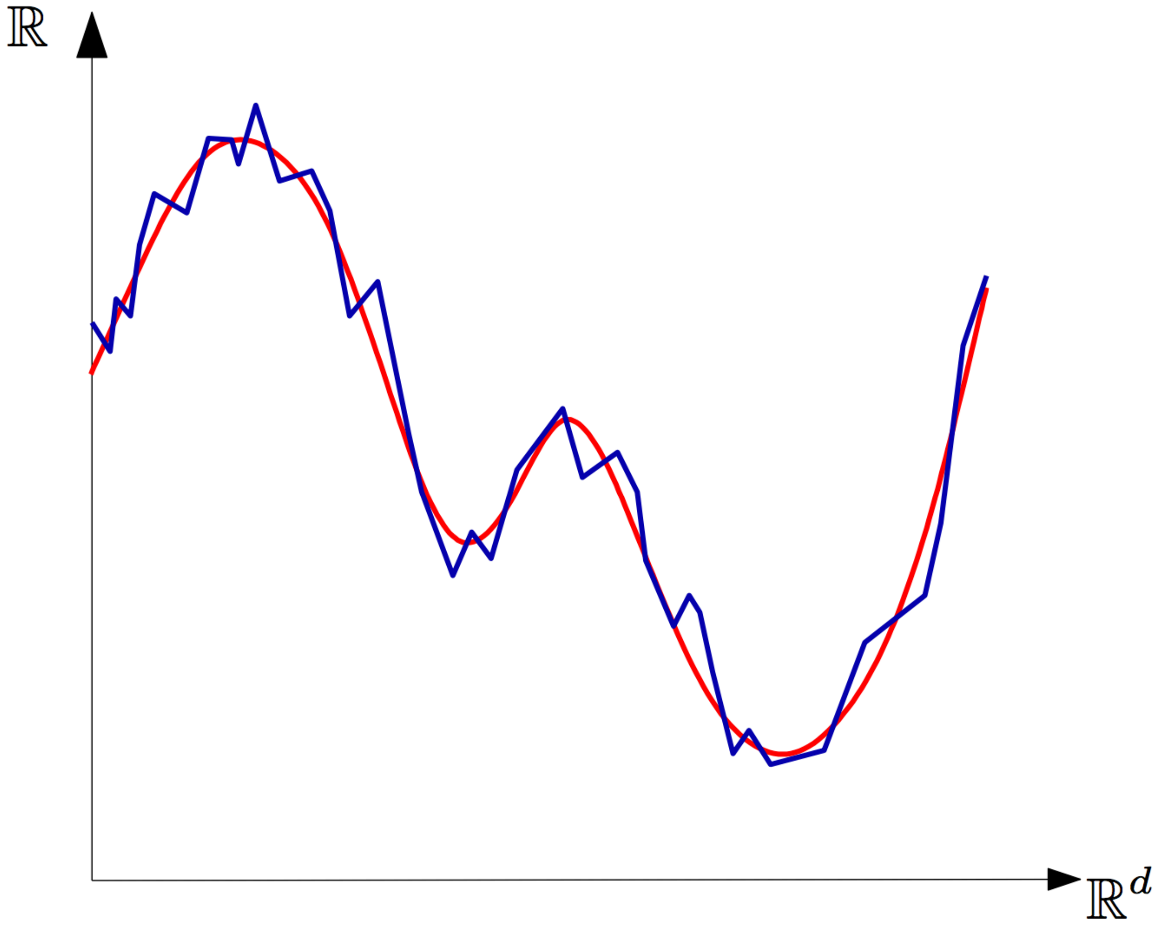
Now, back to clustering

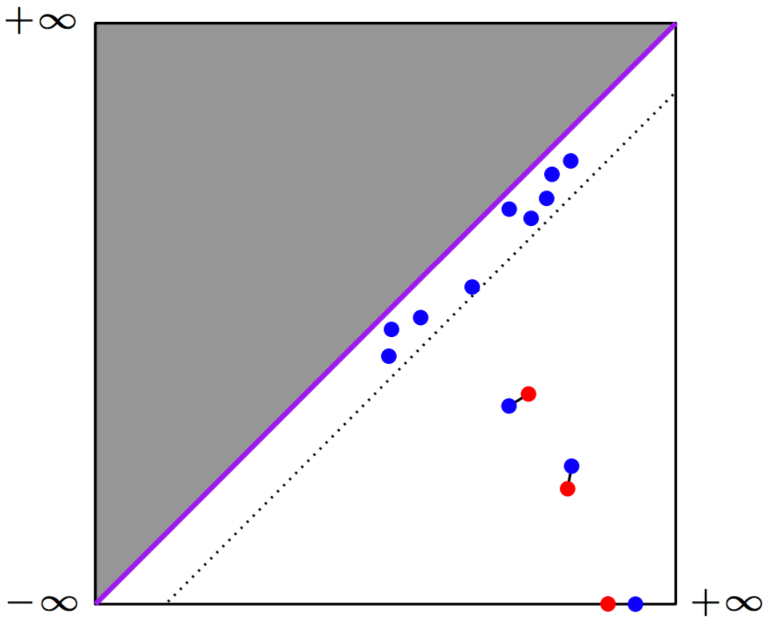
- We’ll redo the hierarchical clustering algorithm, but use it to track 0-dimensional homology.
  - Once two things are grouped in the same cluster, their “level sets” combine, just like in the function example



Now use stability

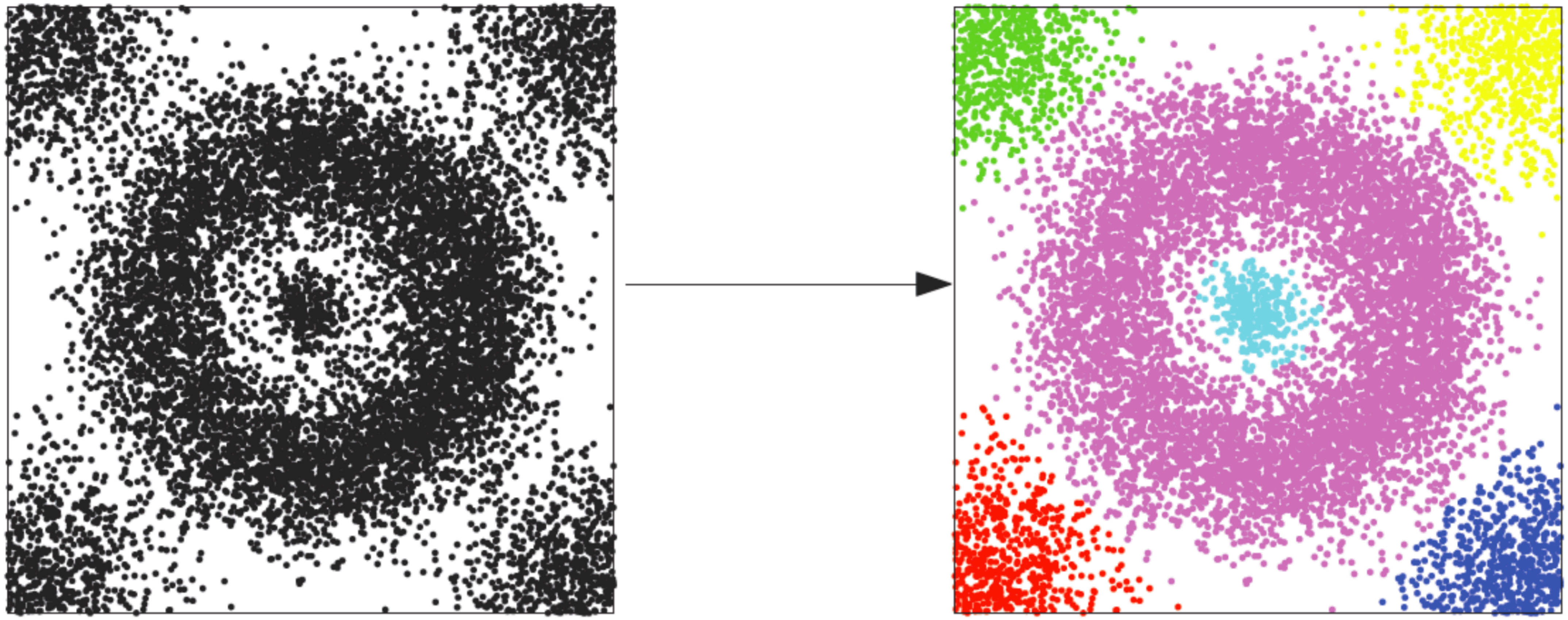
- Now, what if we don't have perfect information, but just a "guess" (or density approximation)?





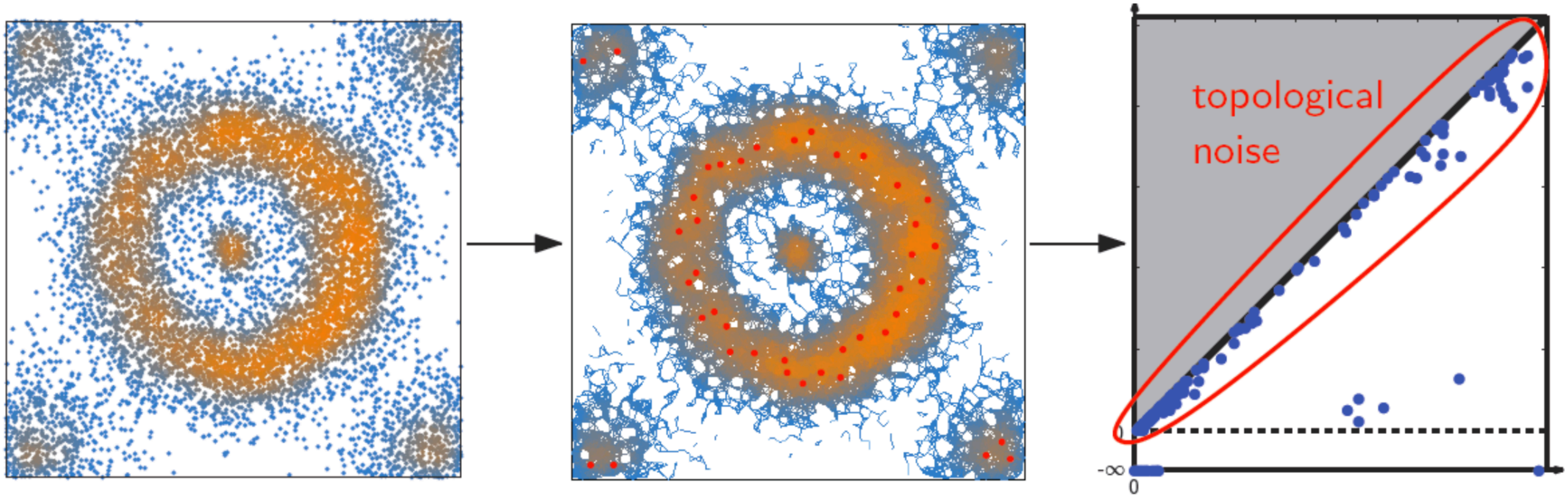
Final result

- Persistence based clustering: take in a finite point set and density estimate, and partition the function using persistence. [Chazal et al, J ACM 2013]



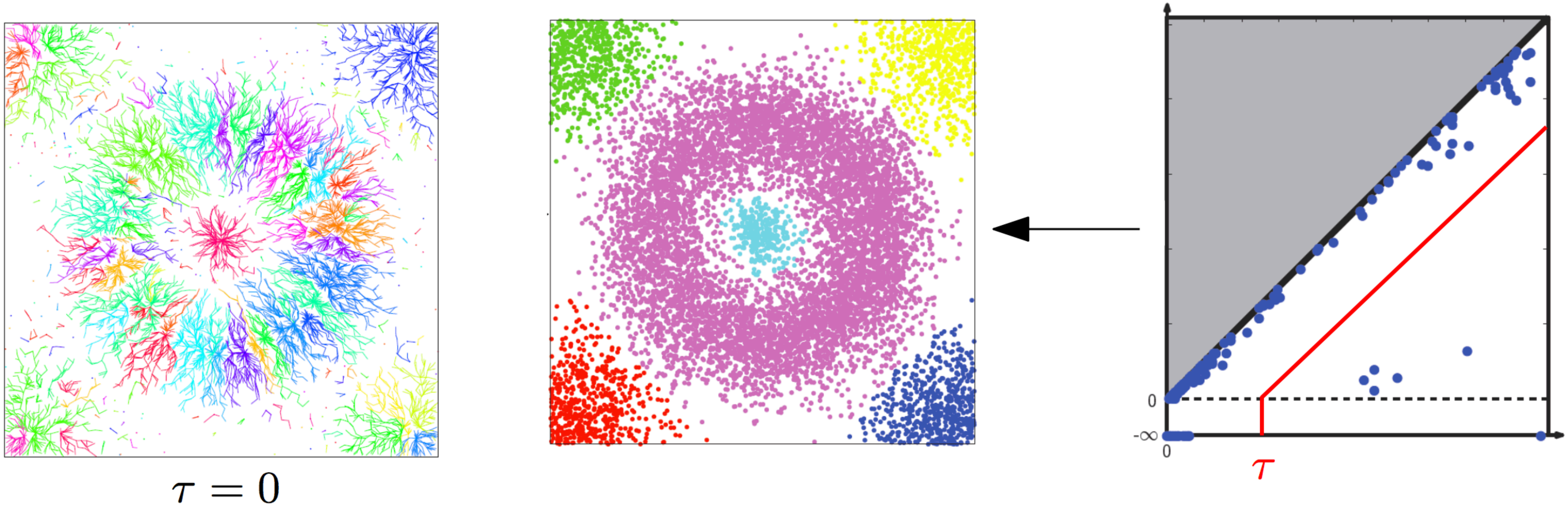
Persistence-based clustering

- Underneath, they just remove the “noise” using persistence, and grab the most prominent peaks of the function to cluster with:



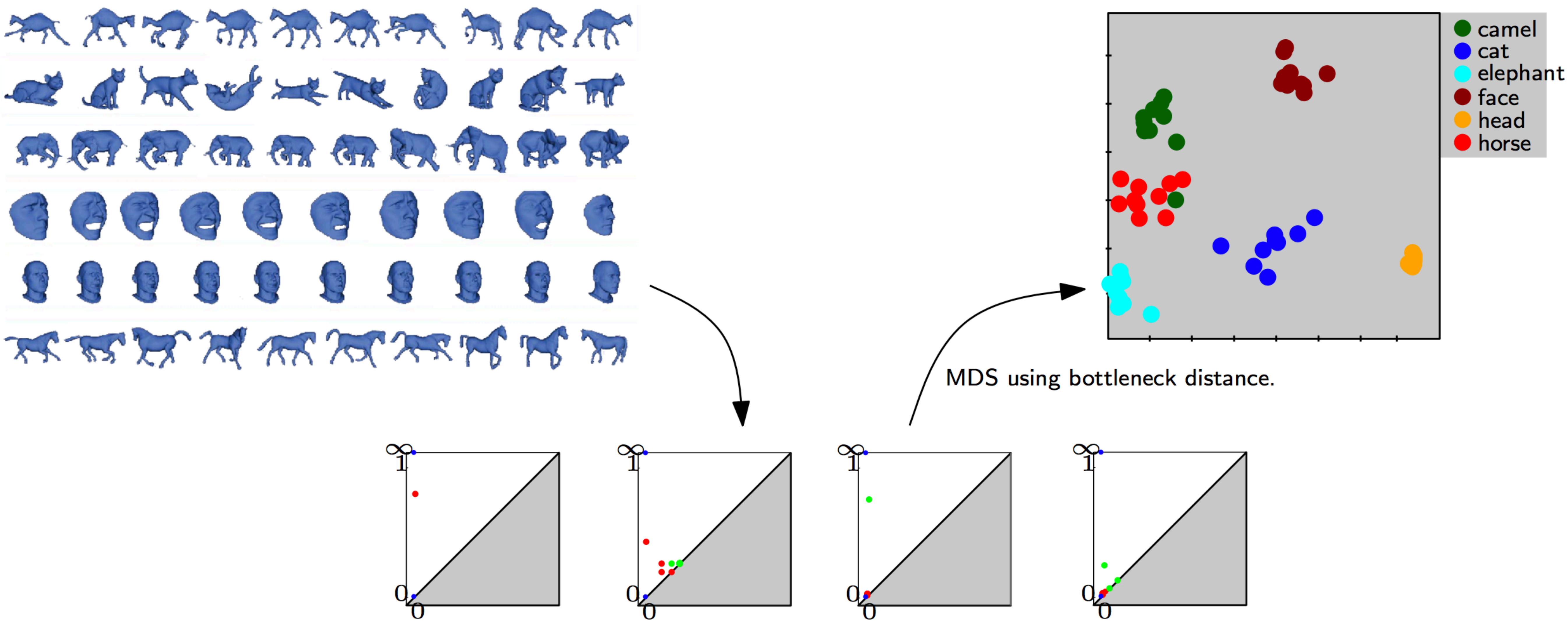
Persistence-based clustering

- You do still need a threshold function, just like in graph based approach:



A different application: Shape classification

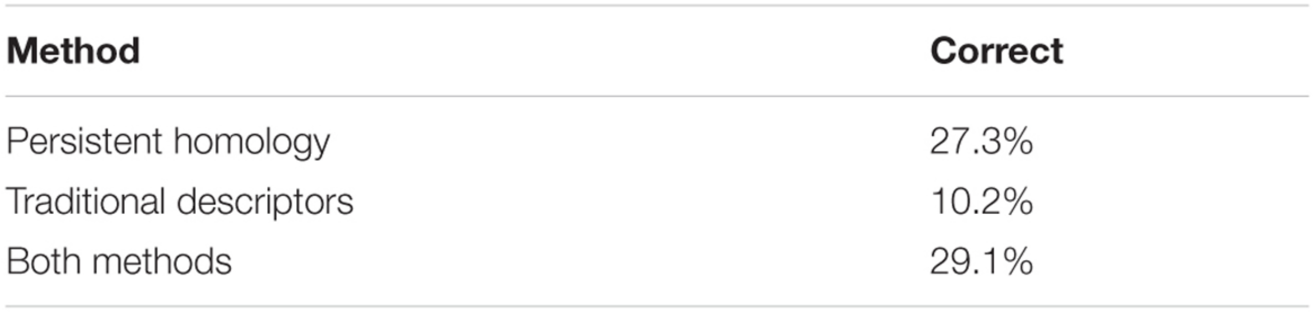
- Many other applications in terms of shape recognition [Chazal et al, SGP 2009]



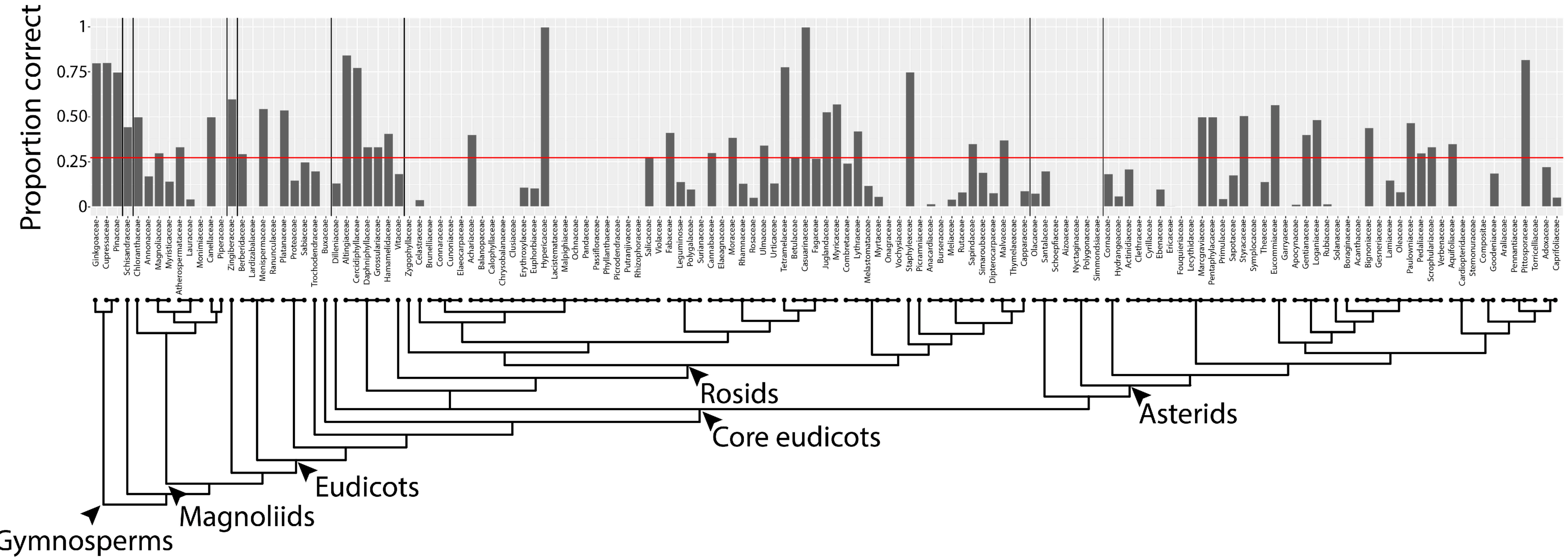
MDS using bottleneck distance.

Biological applications:

- [Mao et al, 2018]: Using persistence to identify leaf “morphospace”



Method	Correct
Persistent homology	27.3%
Traditional descriptors	10.2%
Both methods	29.1%



Biological applications:

- [Kovacev-Nikolic et al, 2016]: Using persistence to study proteins and detect structures

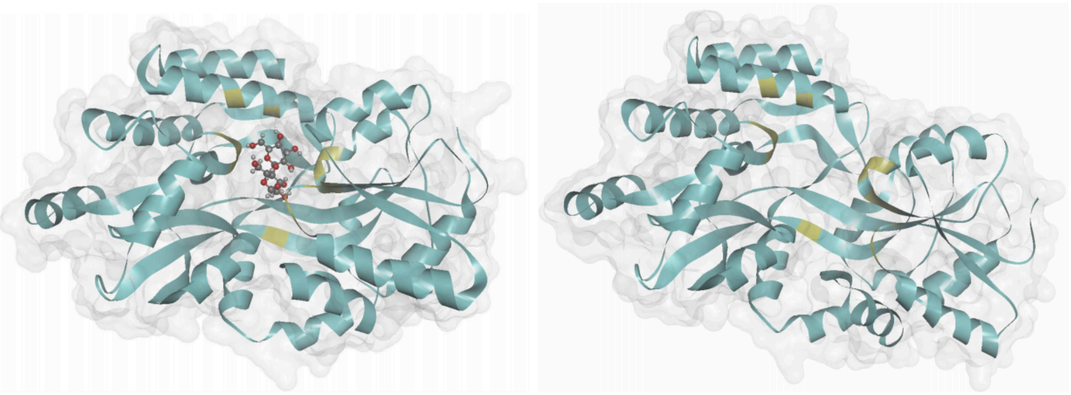


Figure 1: The biological assembly for the closed-holo 1MPD conformal structure (left, Shilton et al. (1996)) and the open-apo 1OMP conformal structure (right, Sharff et al. (1992)). Secondary structures and solvent accessible surfaces of both proteins are shown as blue flat ribbons and gray transparent surfaces, respectively. Active sites in ribbon representations have yellow color and interact with ligand maltose shown here as ball and stick model embedded in 1MPD structure.

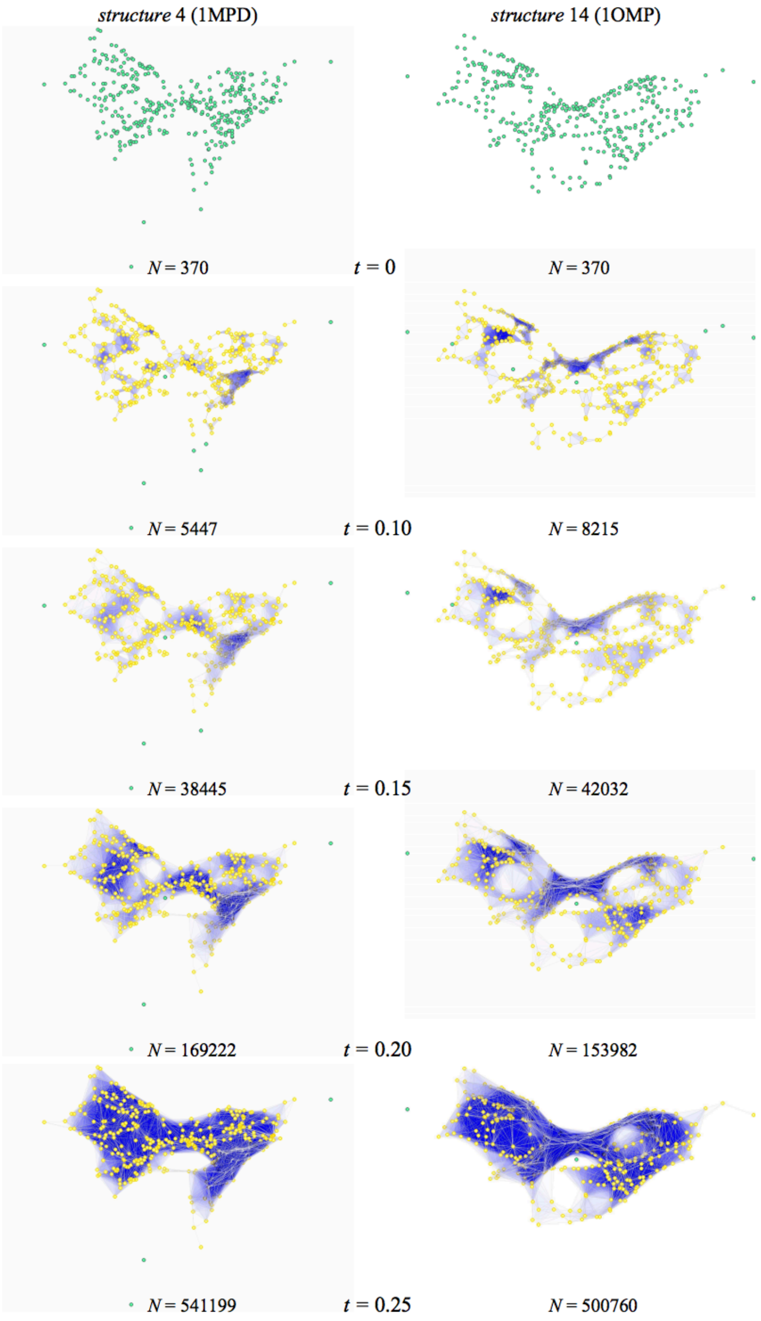
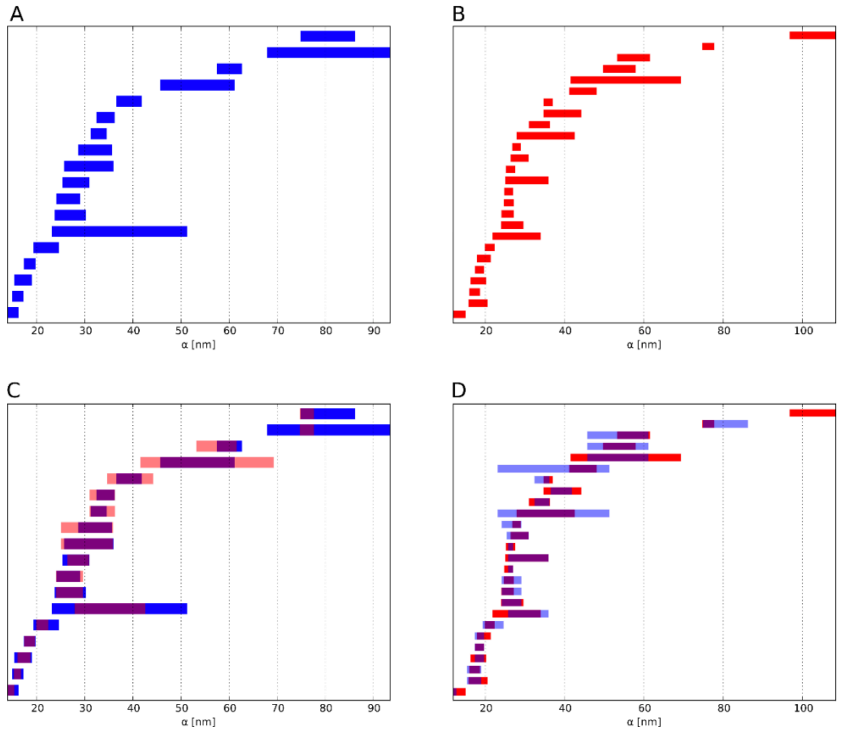


Figure 6: Five snapshots capture the evolution of the filtered Vietoris-Rips complex on the closed-holo 1MPD (left) and the open-apo 1OMP (right) structure of the maltose-binding protein. The complex is constructed on 370 vertices (green circles). The number of vertices that enter the complex (yellow circles) rapidly increases with filtration values. N counts the total number of simplices.

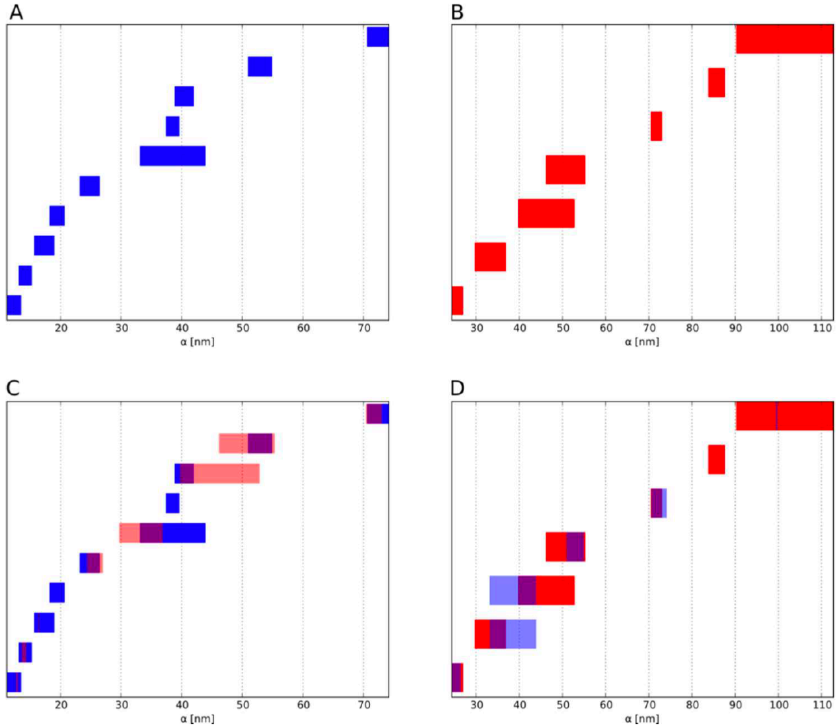
Biological applications:

- [Hofmann et al 2018]: Finding DNA double strand breaks induced by radiation

- “The aim of this article is to demonstrate a new approach to analyze repair foci by their topology in order to obtain a cell independent method of categorization”



Similar Barcodes



Dissimilar Barcodes