DSC 214
Topological Data Analysis
Topic 9: Mapper

Instructor: Zhengchao Wan

- Persistent homology
  - One of the most important developments in computational topology in the last two decades
- Other topological structures for analyzing functions
  - Real valued functions, or more complex maps

Mapper

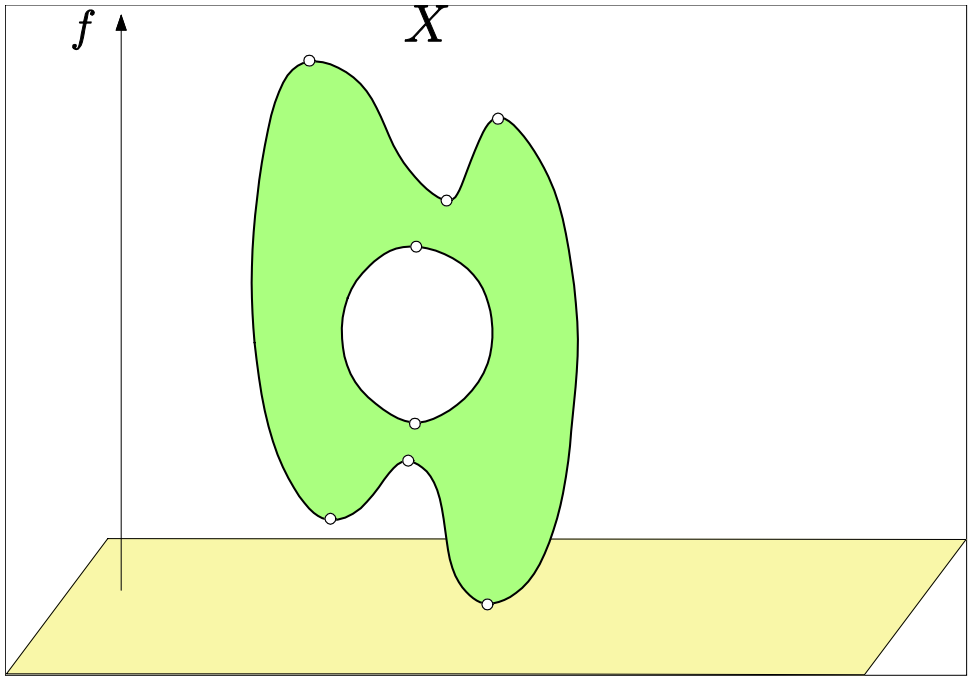
- [Singh, Mémoli, Carlsson, 2007]
- Dimension reduction through topological methods
- Data visualization

- Summarizing topological structure of a map $f : X \to Z$ into a graph

Section 0:
Reeb graph

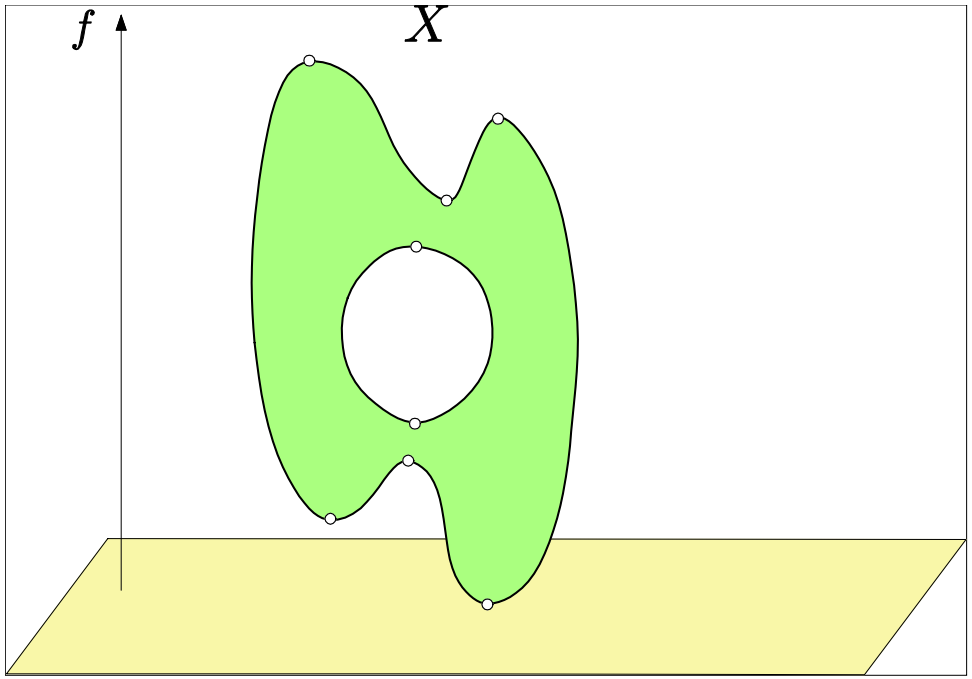
Introduction

- Given a topological space X and function $f: X \rightarrow R$



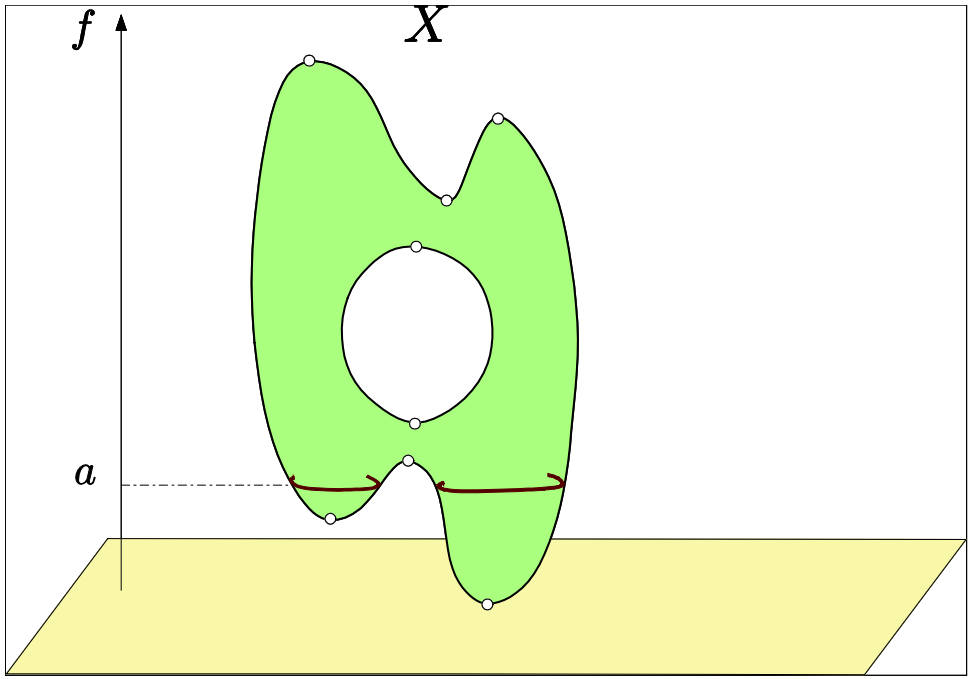
Introduction

- Given a topological space X and function $f: X \rightarrow R$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a \}$



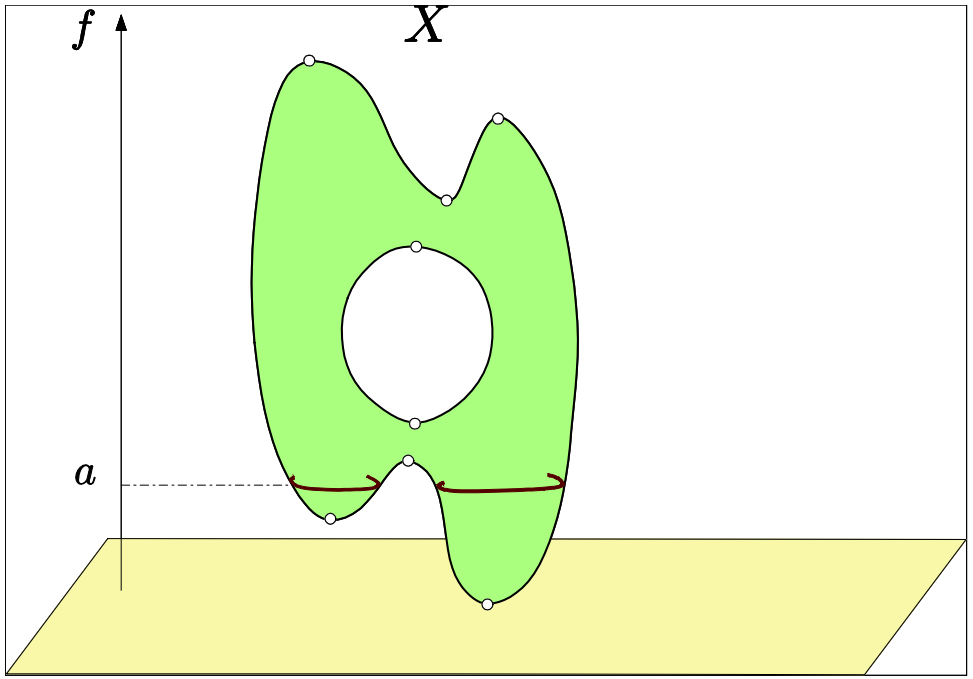
Introduction

- Given a topological space X and function $f: X \rightarrow R$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a \}$



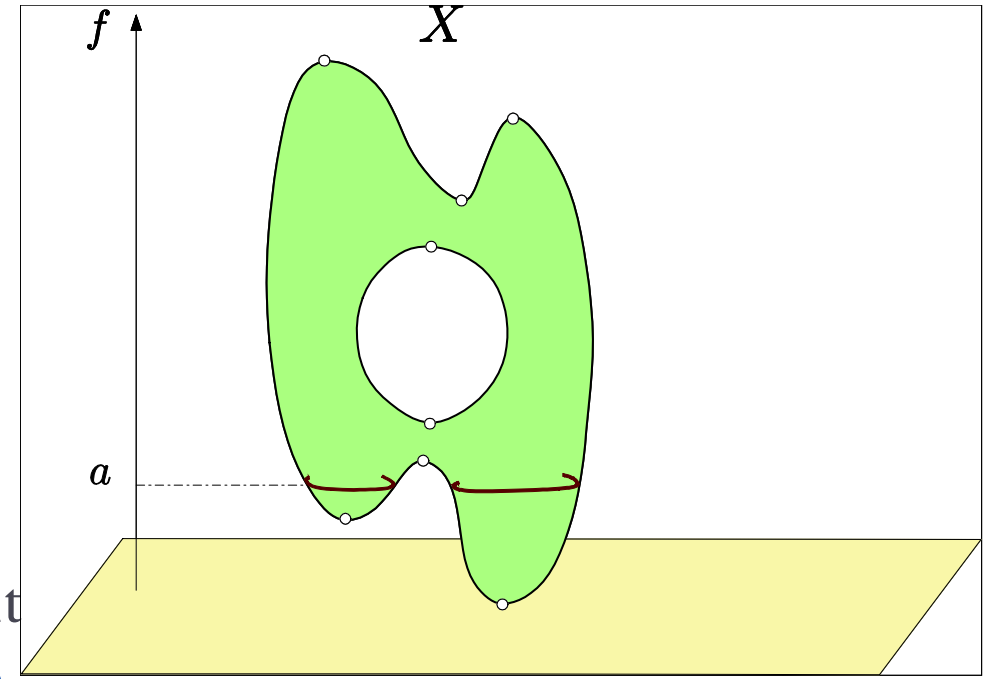
Introduction

- Given a topological space X and function $f: X \rightarrow R$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a \}$
- A *contour* at value a :
  - a connected component of X_a



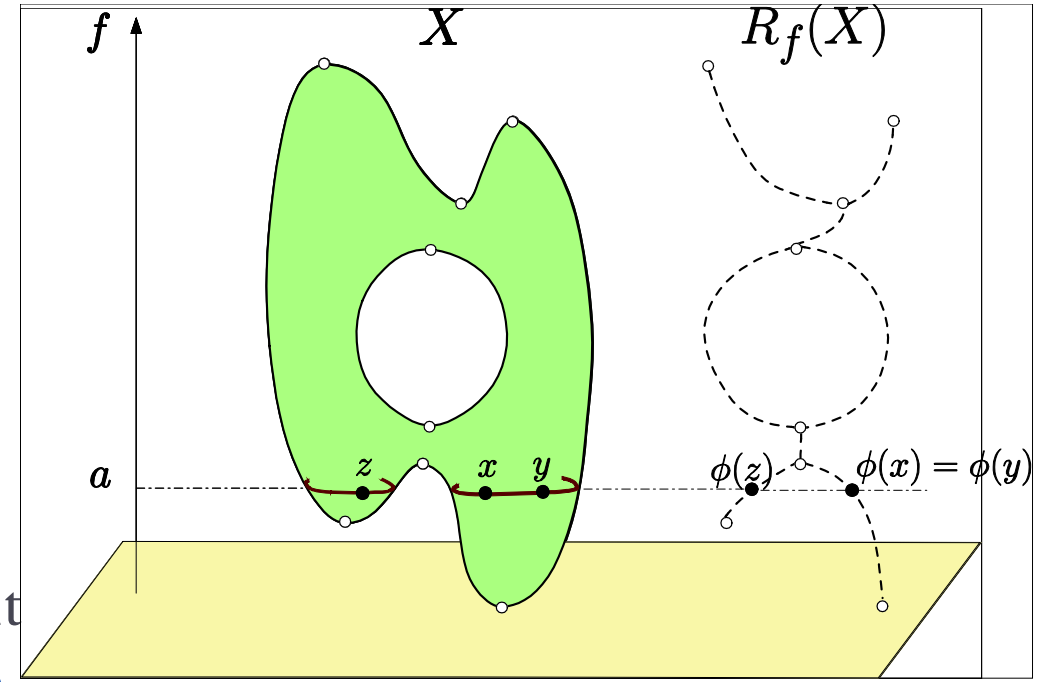
Introduction

- Given a topological space X and function $f: X \rightarrow R$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a \}$
- A *contour* at value a :
  - a connected component of X_a
- *Reeb graph* $R_f(X)$ of X w.r.t. f:
  - continuous collapsing of each contour of f to a point
  - A continuous surjection $\phi : X \rightarrow R_f(X)$ s.t, $\phi(x) = \phi(y)$ if and only if x and y is in the same contour



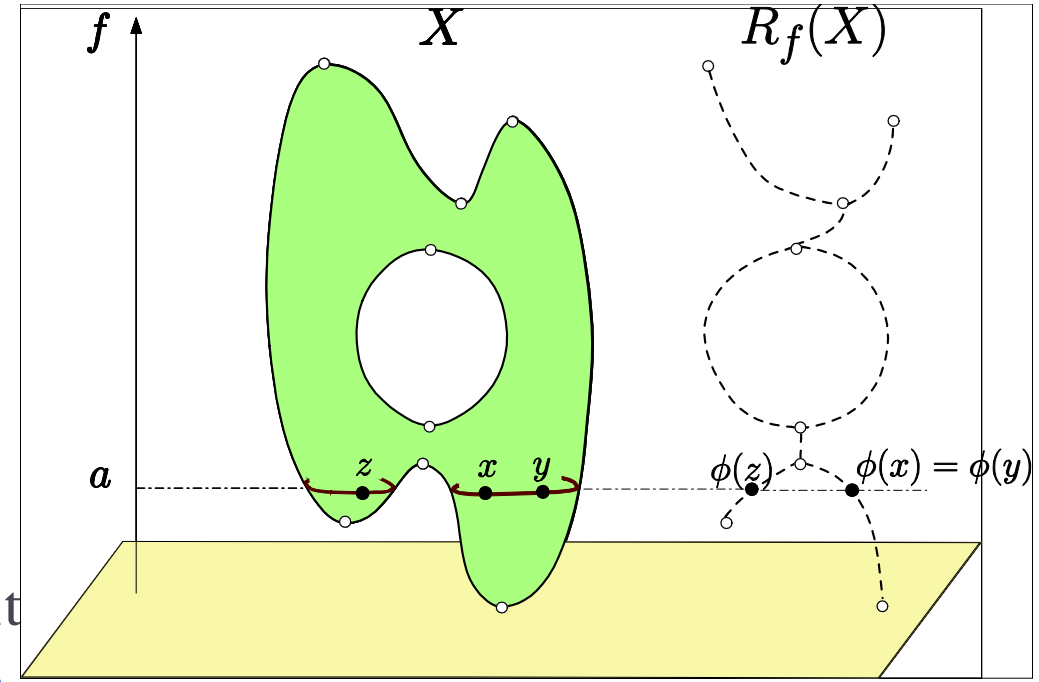
Introduction

- Given a topological space X and function $f: X \to R$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a \}$
- A *contour* at value a :
  - a connected component of X_a
- *Reeb graph* $R_f(X)$ of X w.r.t. f:
  - continuous collapsing of each contour of f to a point
  - A continuous surjection $\phi : X \to R_f(X)$ s.t, $\phi(x) = \phi(y)$ if and only if x and y is in the same contour



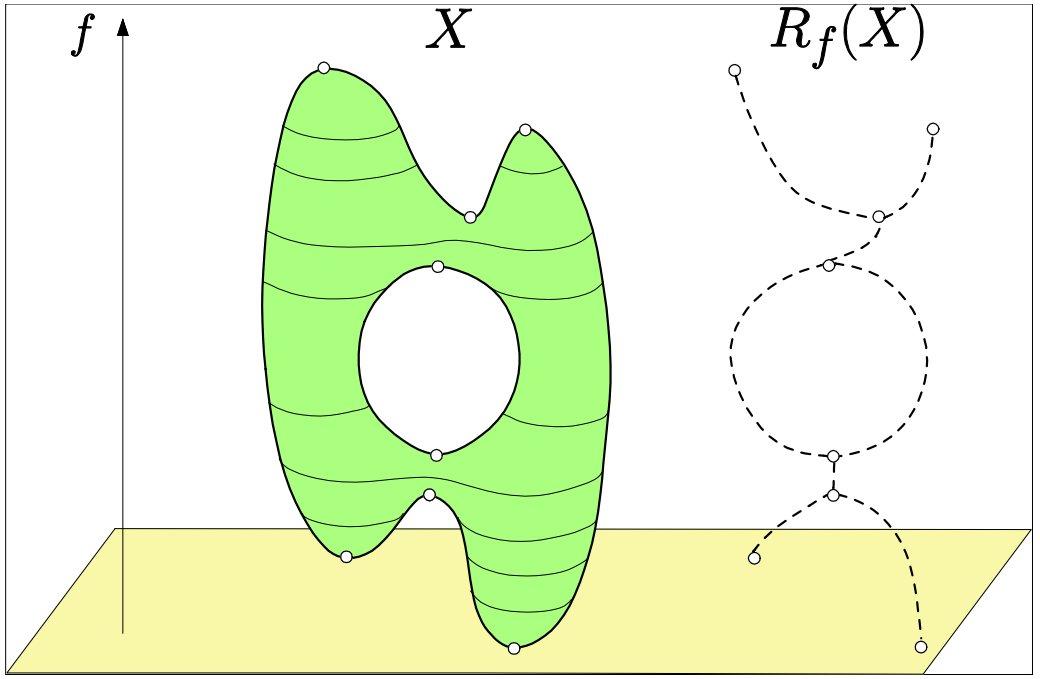
Introduction

- Given a topological space X and function $f: X \to R$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a \}$
- A *contour* at value a :
  - a connected component of X_a
- *Reeb graph* $R_f(X)$ of X w.r.t. f:
  - continuous collapsing of each contour of f to a point
  - A continuous surjection $\phi : X \to R_f(X)$ s.t, $\phi(x) = \phi(y)$ if and only if x and y is in the same contour



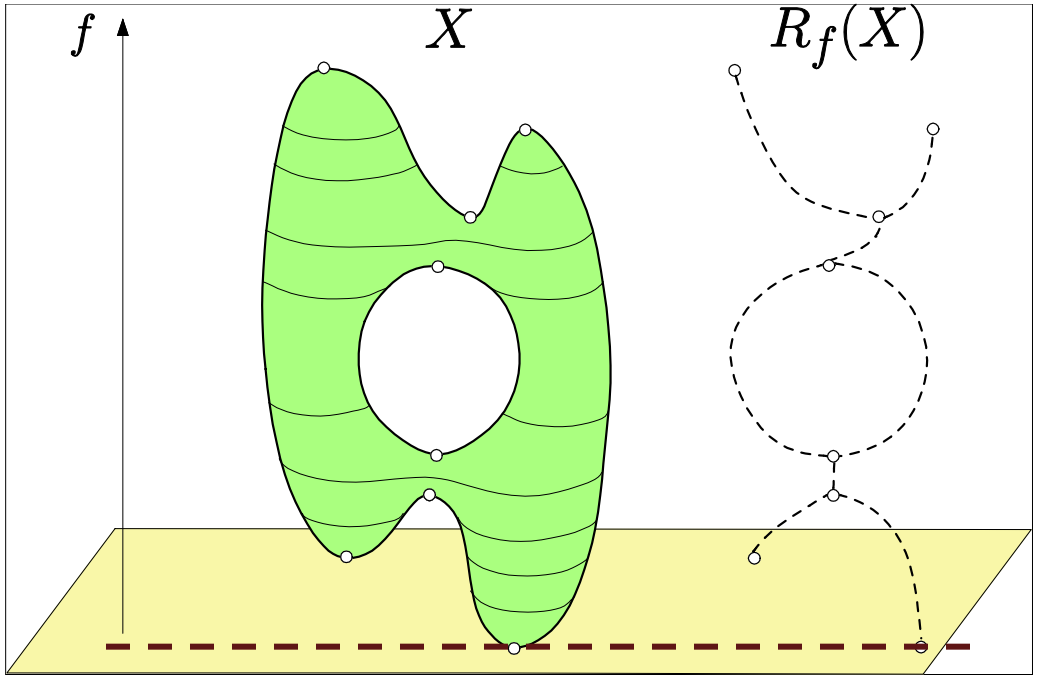
More on Reeb Graph

- Imagine sweeping X in increasing order of f
  - Track the changes in 0-th homology of level sets
    - i.e, changes in contours
  - Node:
    - where changes happen
  - Arc:
    - evolution of a single contour



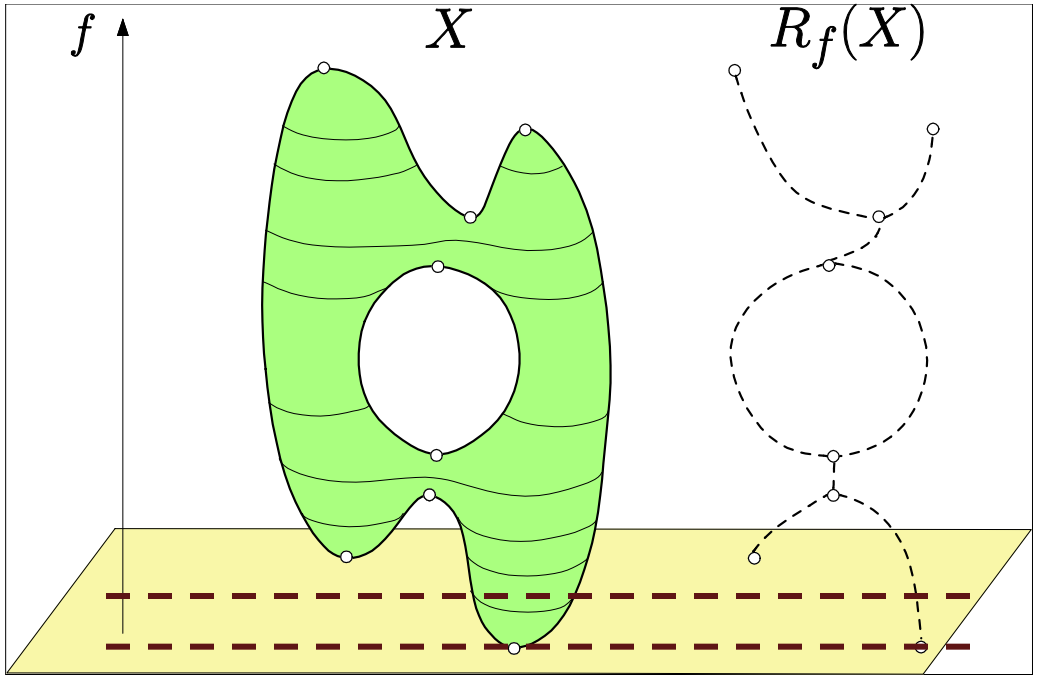
More on Reeb Graph

- Imagine sweeping X in increasing order of f
  - Track the changes in 0-th homology of level sets
    - i.e, changes in contours
  - Node:
    - where changes happen
  - Arc:
    - evolution of a single contour



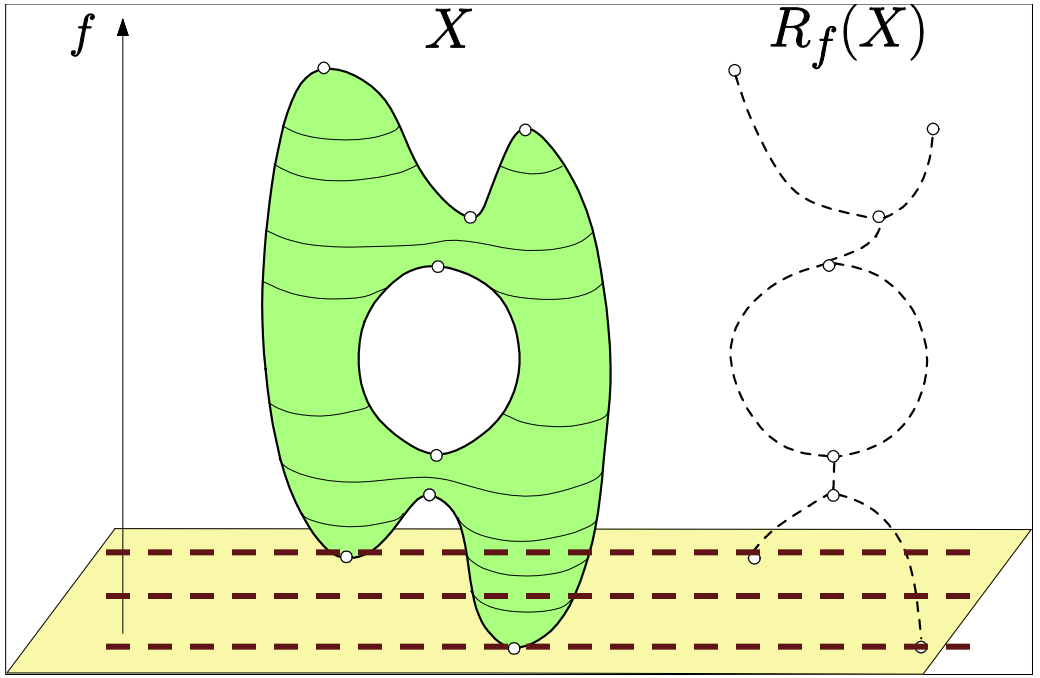
More on Reeb Graph

- Imagine sweeping X in increasing order of f
  - Track the changes in 0-th homology of level sets
    - i.e, changes in contours
  - Node:
    - where changes happen
  - Arc:
    - evolution of a single contour



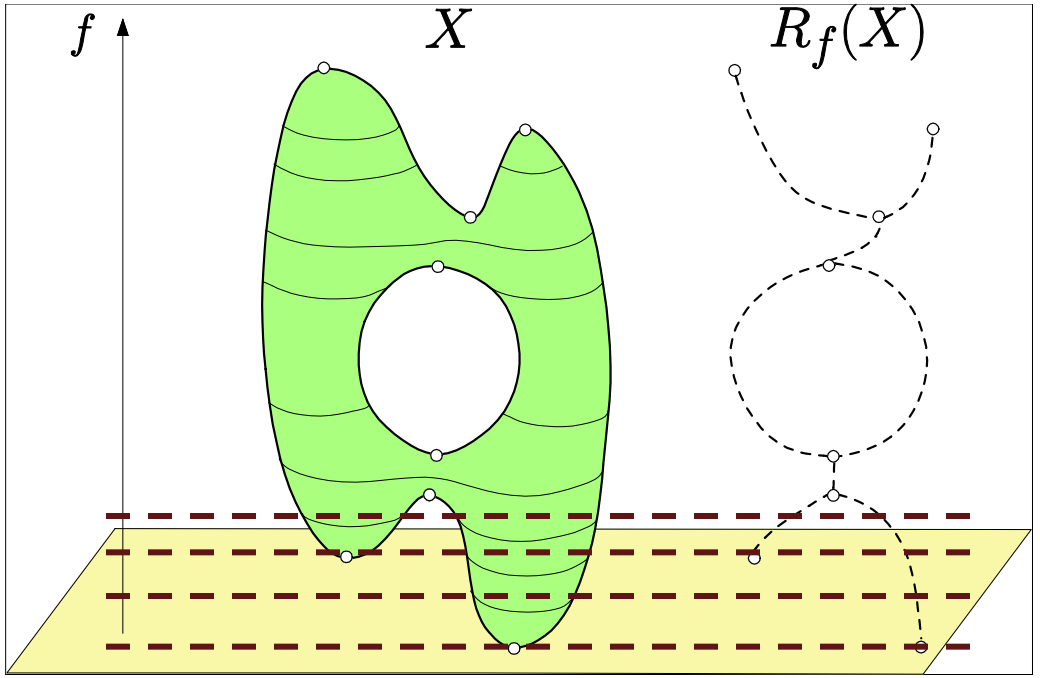
More on Reeb Graph

- Imagine sweeping X in increasing order of f
  - Track the changes in 0-th homology of level sets
    - i.e, changes in contours
  - Node:
    - where changes happen
  - Arc:
    - evolution of a single contour



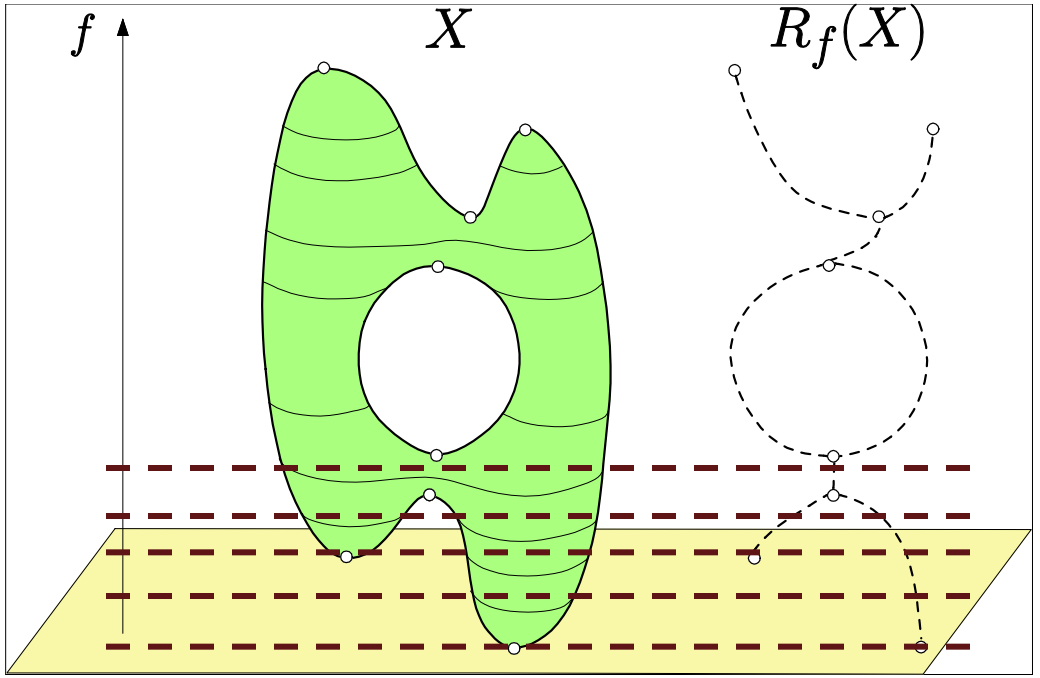
More on Reeb Graph

- Imagine sweeping X in increasing order of f
  - Track the changes in 0-th homology of level sets
    - i.e, changes in contours
  - Node:
    - where changes happen
  - Arc:
    - evolution of a single contour



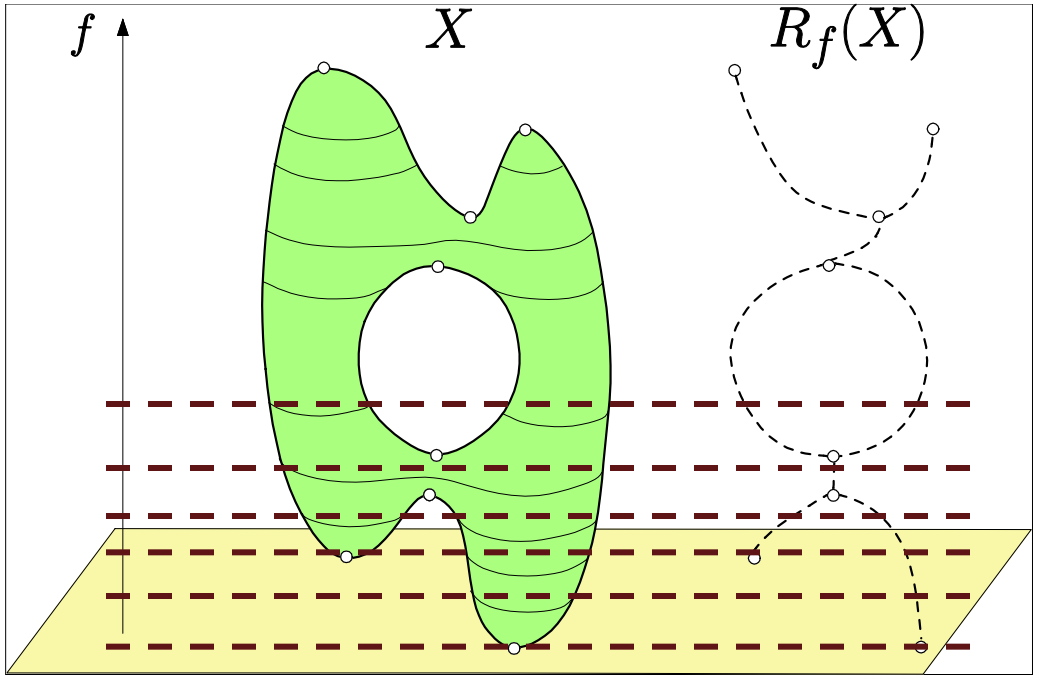
More on Reeb Graph

- Imagine sweeping X in increasing order of f
  - Track the changes in 0-th homology of level sets
    - i.e, changes in contours
  - Node:
    - where changes happen
  - Arc:
    - evolution of a single contour



More on Reeb Graph

- Imagine sweeping X in increasing order of f
  - Track the changes in 0-th homology of level sets
    - i.e, changes in contours
  - Node:
    - where changes happen
  - Arc:
    - evolution of a single contour



Reeb graph of Morse function

- Given an m-manifold M and $f: M \to R$,
  - A point $p \in M$ is *critical* if gradient of f vanishes at p
- A critical point is non-degenerate
  - if it has non-degenerate Hessian
- For every non-degenerate critical point

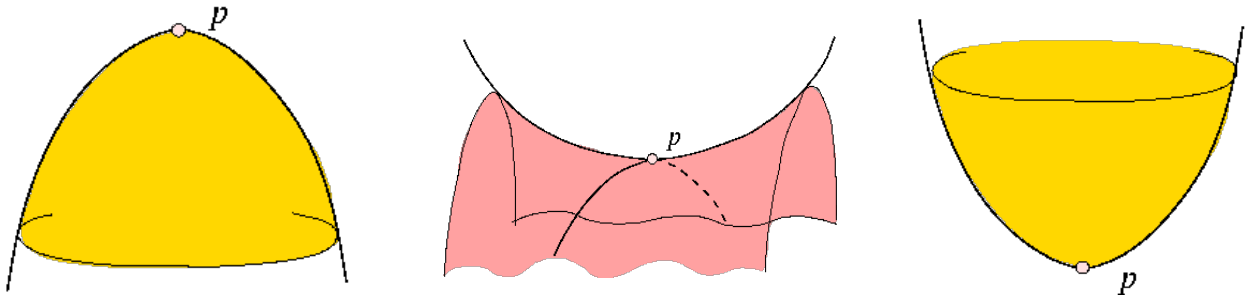
> MORSE LEMMA. Let u be a non-degenerate critical point of $f : \mathbb{M} \to \mathbb{R}$. There are local coordinates with $u = (0, 0, \ldots, 0)$ such that
>
> $$f(x) = f(u) - x_1^2 - \ldots - x_p^2 + x_{p+1}^2 + \ldots + x_d^2$$
>
> for every point $x = (x_1, x_2, \ldots, x_d)$ in a small neighborhood of u.

Critical Points cont.

- For non-degenerate critical points
- Suppose M is 2-manifold

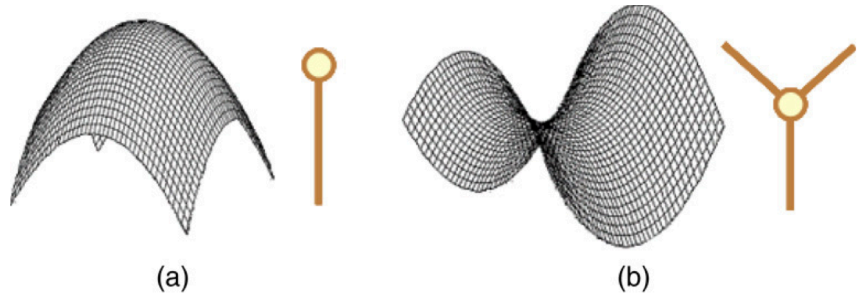
Critical Points cont.

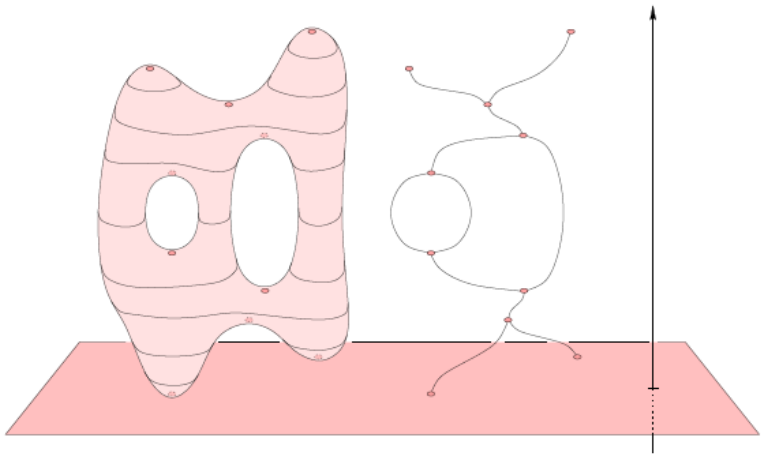
- For non-degenerate critical points
- Suppose M is 2-manifold



max saddle min

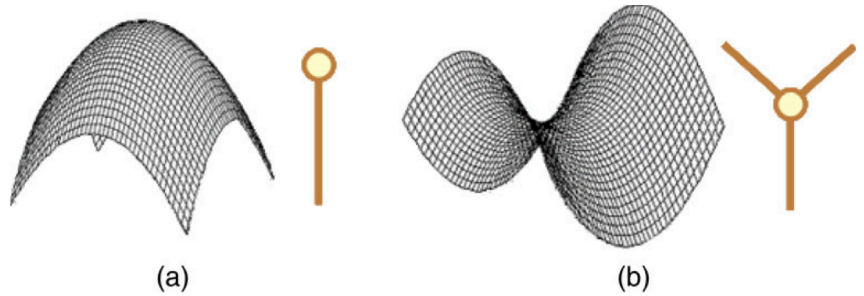
Bijection between critical points and tree nodes

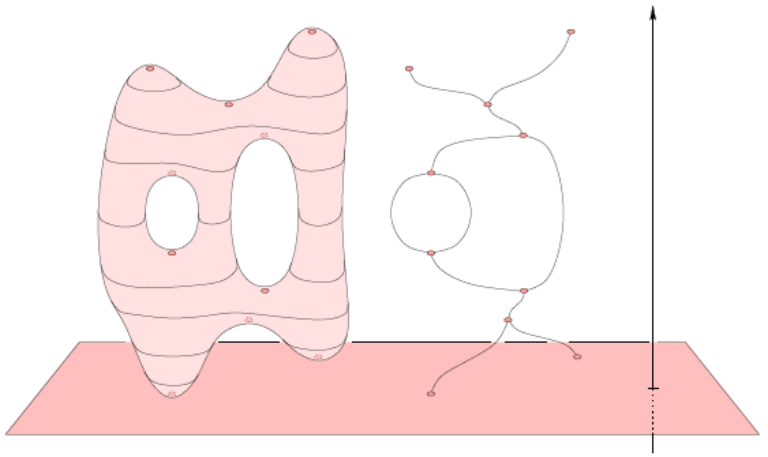




Bijection between critical points and tree nodes

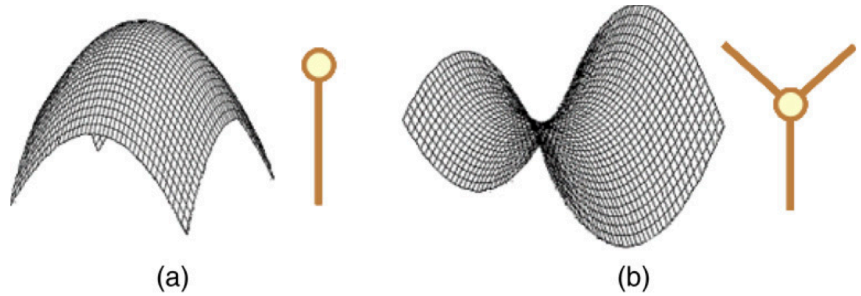
- If M is an d-manifold and $f: M \to R$ a Morse function

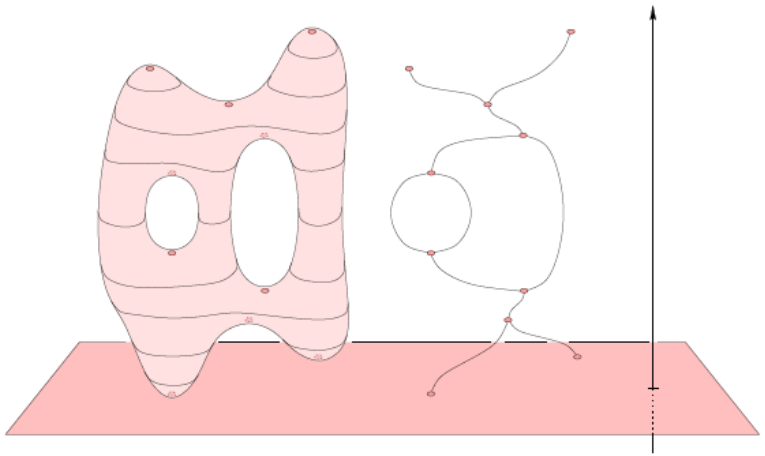




Bijection between critical points and tree nodes

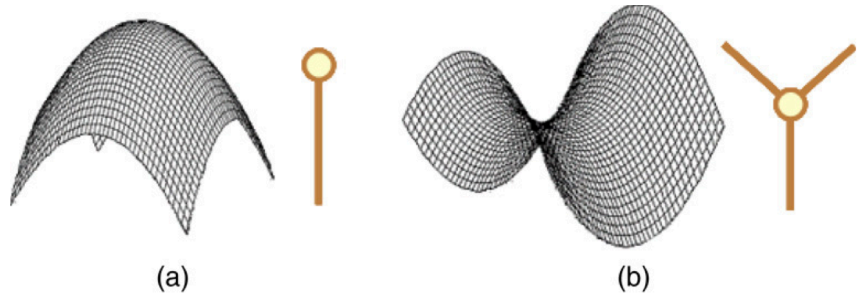
- If M is an d-manifold and $f: M \to R$ a Morse function
  - Degree 1 nodes:
    - Minimum or maximum

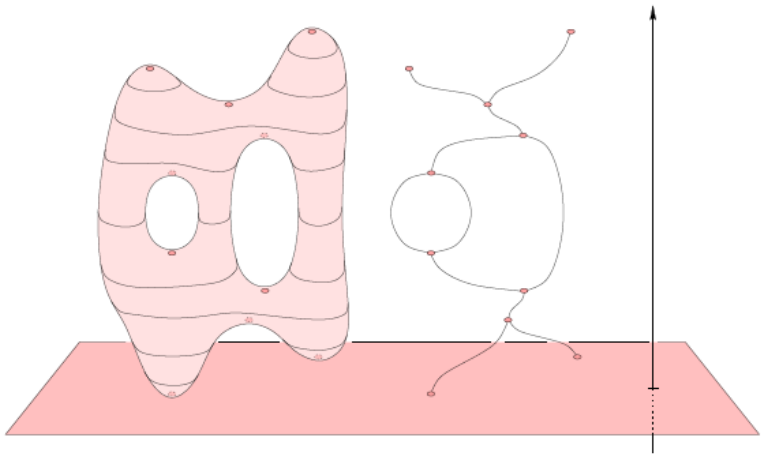




Bijection between critical points and tree nodes

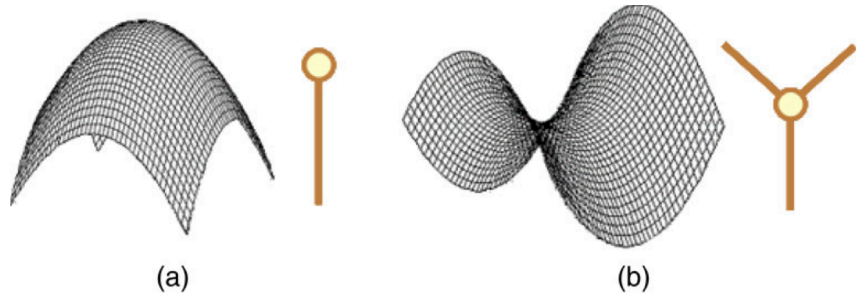
- If M is an d-manifold and $f: M \to R$ a Morse function
  - Degree 1 nodes:
    - Minimum or maximum
  - Degree *3* nodes:
    - 1-saddles that merge two contours (merging forks)

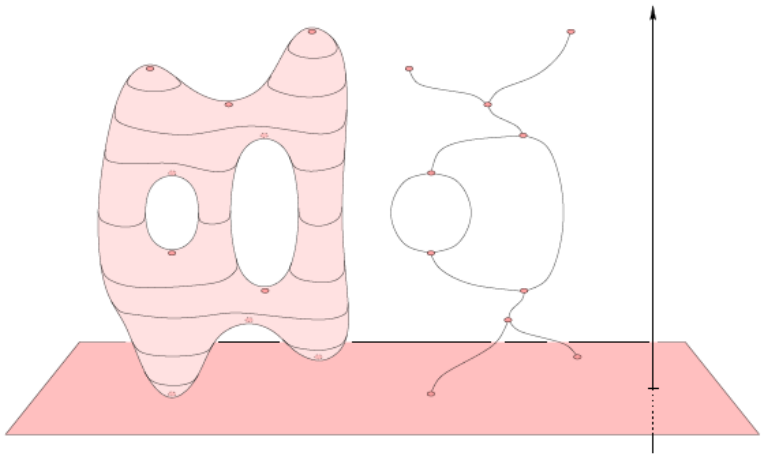




Bijection between critical points and tree nodes

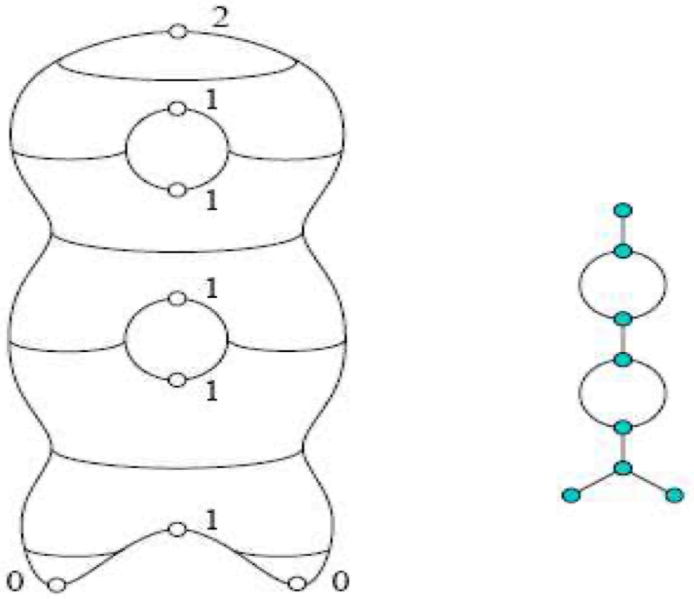
- If M is an d-manifold and $f: M \to R$ a Morse function
  - Degree 1 nodes:
    - Minimum or maximum
  - Degree 3 nodes:
    - 1-saddles that merge two contours (merging forks)
    - or (d-1)-saddles that split a contour into two (splitting forks)
  - Degree 2 nodes:
    - All other nodes





M is a 2-Manifold

The Reeb graph of a Morse function on a connected, orientable *2*-manifold of genus *g* has *g* loops.



Homology Relations

- Reeb graph contains less topological information than its original space

Lemma

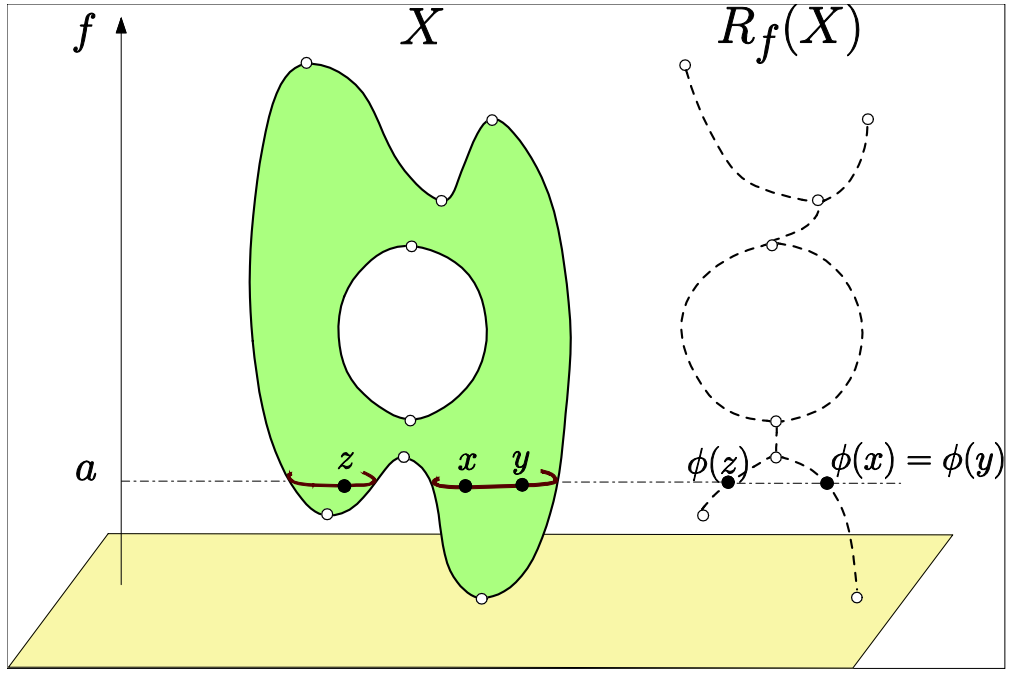
$$\beta_0(R_f(X)) = \beta_0(X)$$

$$\beta_1(R_f(X)) \leq \beta_1(X)$$

In general, the Reeb graph of a function $f: X \rightarrow R$ captures the so-called 1st *vertical homology* of X w.r.t. f.

[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f: X \rightarrow R$ captures the so-called 1st *vertical homology* of X w.r.t. f.

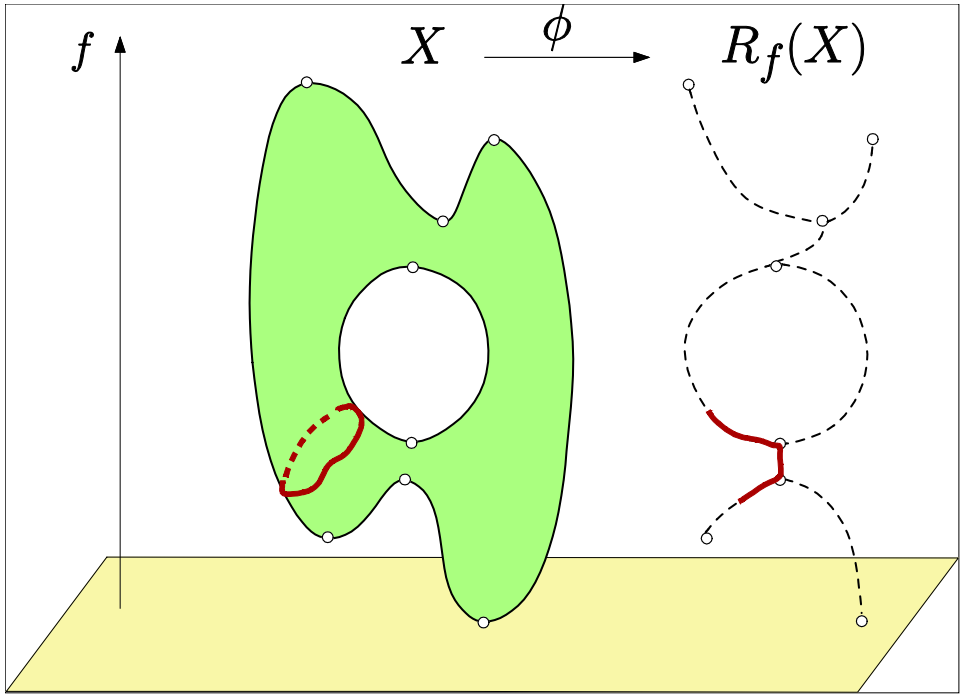


[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f: X \rightarrow R$ captures the so-called 1st *vertical homology* of X w.r.t. f.

[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f: X \rightarrow R$ captures the so-called 1st *vertical homology* of X w.r.t. f.

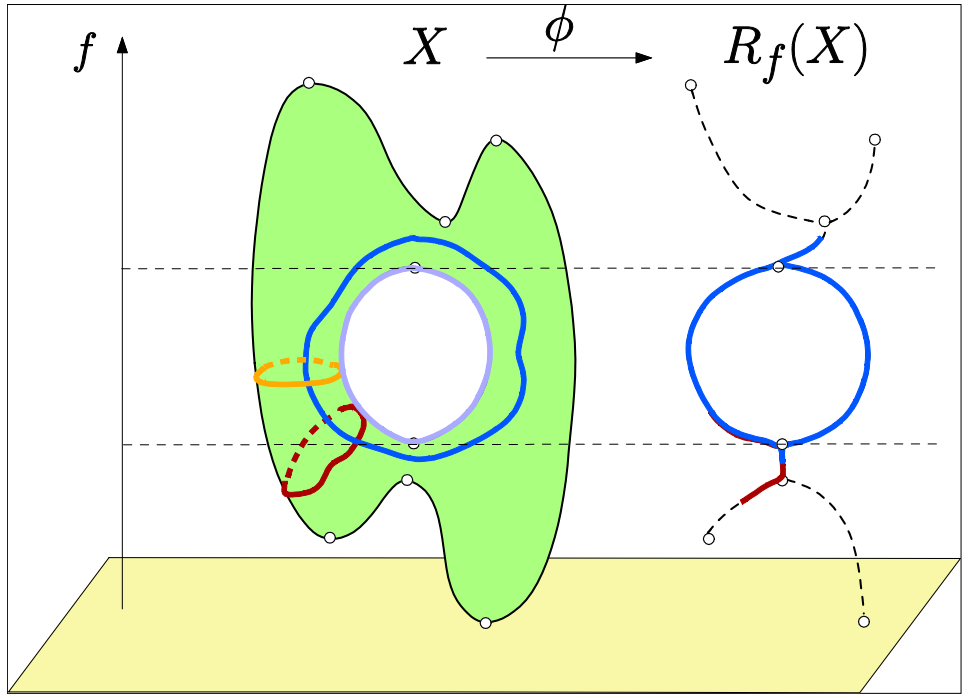


[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f: X \rightarrow R$ captures the so-called 1st *vertical homology* of X w.r.t. f.

[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f: X \rightarrow R$ captures the so-called 1st *vertical homology* of X w.r.t. f.

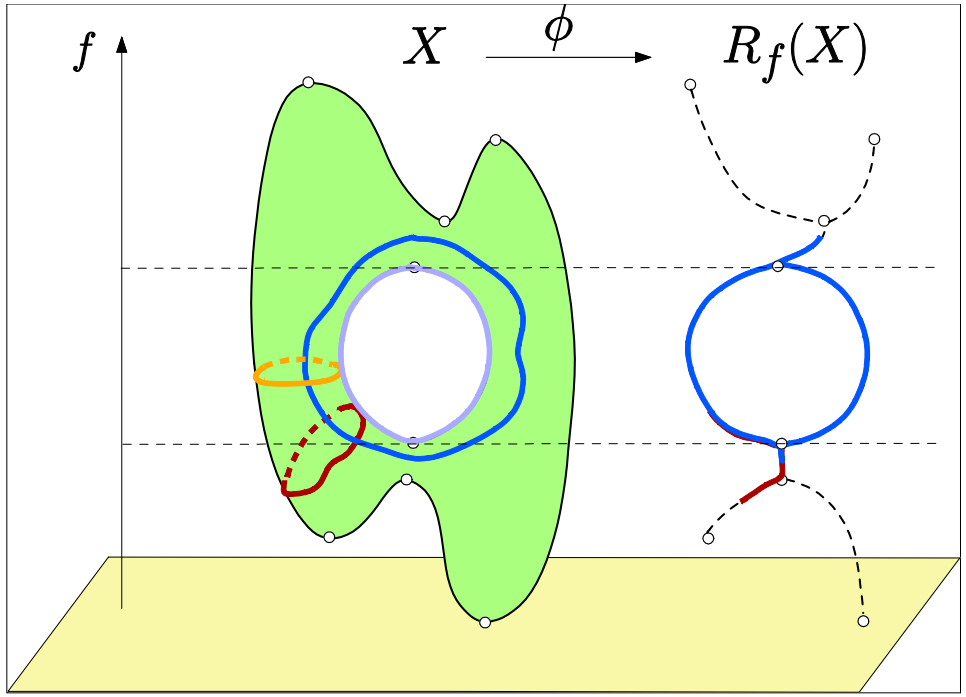


[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f: X \rightarrow R$ captures the so-called 1st *vertical homology* of X w.r.t. f.

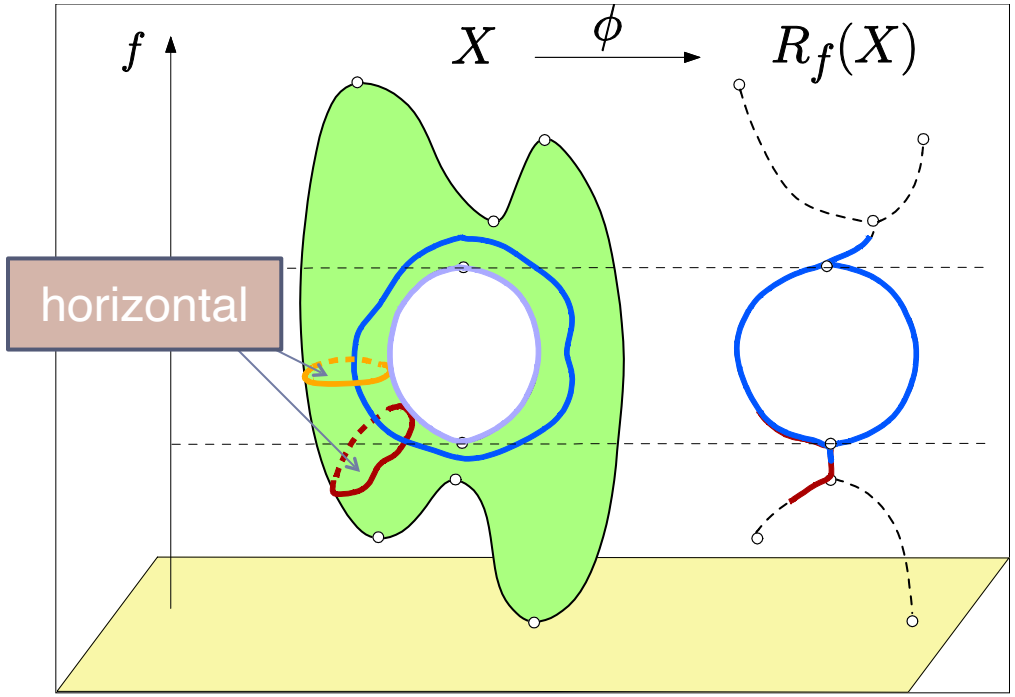
[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f: X \rightarrow R$ captures the so-called 1st *vertical homology* of X w.r.t. f.



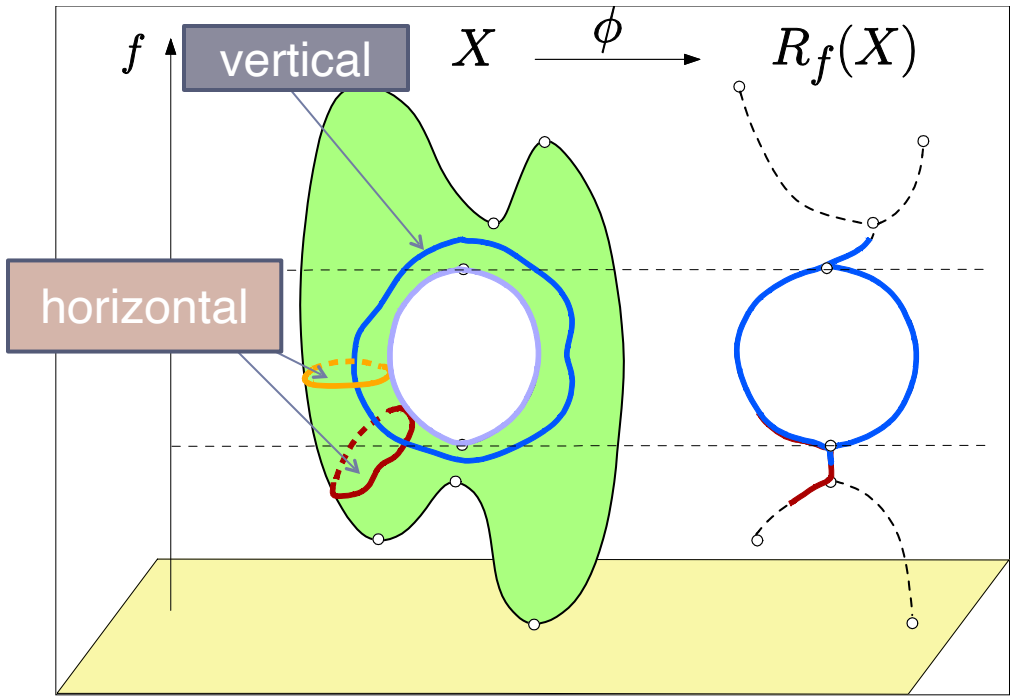
[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f: X \rightarrow R$ captures the so-called 1st *vertical homology* of X w.r.t. f.



[Dey and Wang, DCG2012]

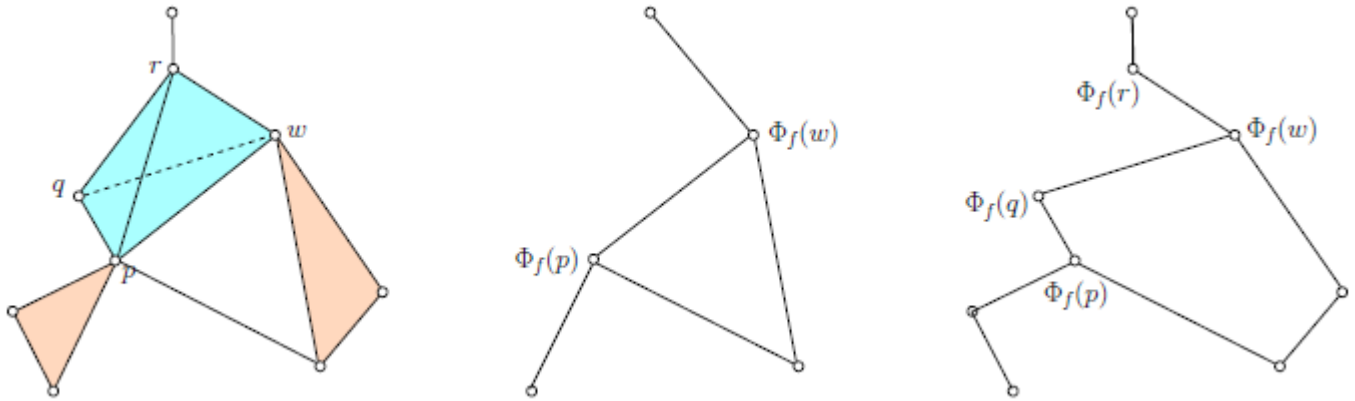
In general, the Reeb graph of a function $f: X \rightarrow R$ captures the so-called 1st *vertical homology* of X w.r.t. f.



[Dey and Wang, DCG2012]

PL Setting

- PL function *f* defined on simplicial complex *K*
  - *f* is decided by function values on the vertices *V* of *K*
  - only 2-skelenton *(V,E,T)* of *K* matters
  - Reeb graph $R_f(X)$ can be computed in $O(m \log n)$ time
    - *m*: number of vertices, edges, and triangles of *X*,
    - *n*: number of vertices



Applications

- Handle removal
- Skelentonize a shape
- Shape matching

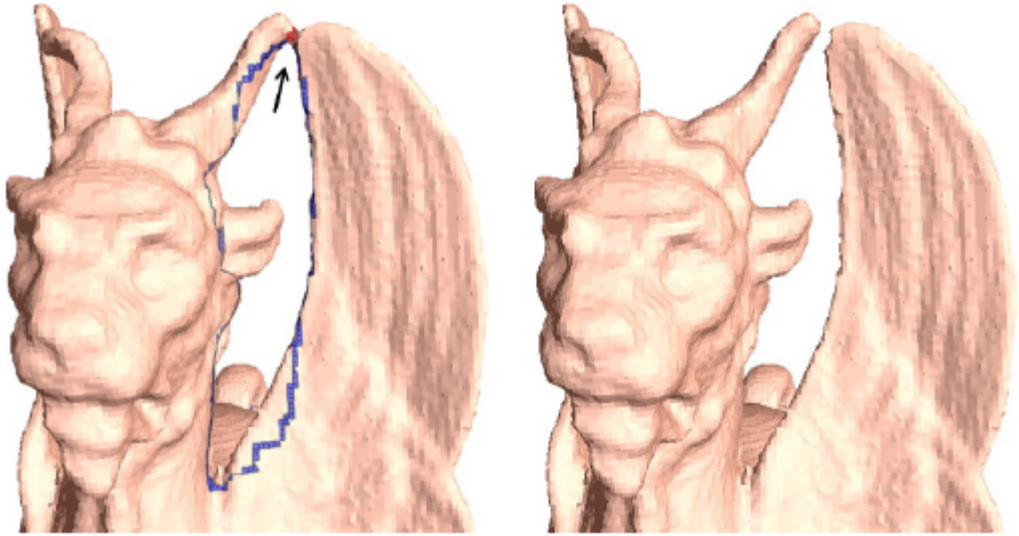
Applications

- Handle removal
- Skelentonize a shape
- Shape matching

Courtesy of Wood et al. 2002

Applications

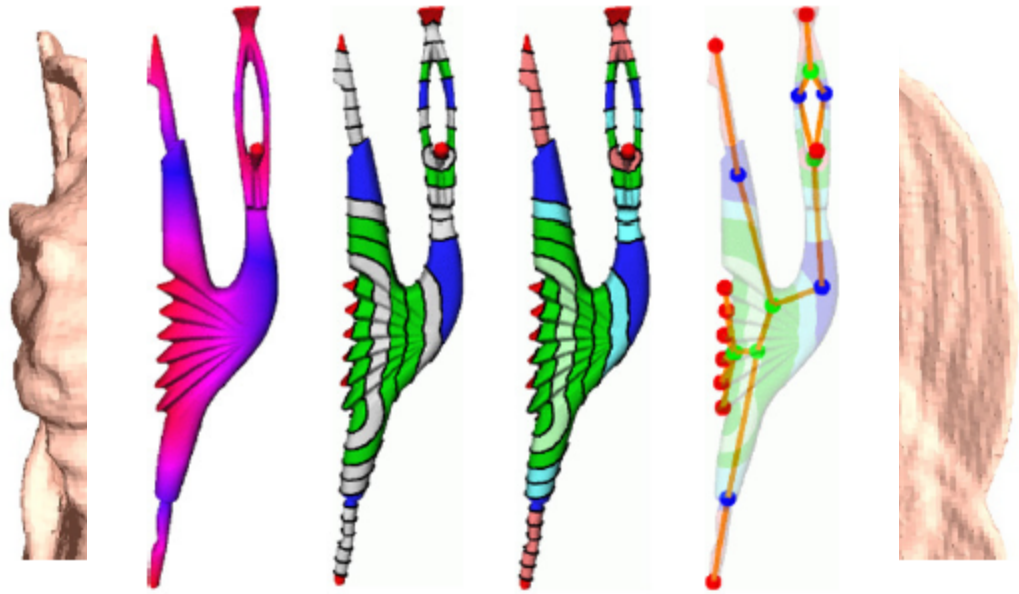
- Handle removal
- Skelentonize a shape
- Shape matching



Courtesy of Wood et al. 2002

Applications

- Handle removal
- Skelentonize a shape
- Shape matching



Courtesy of Biosotti et al. 2008

Applications

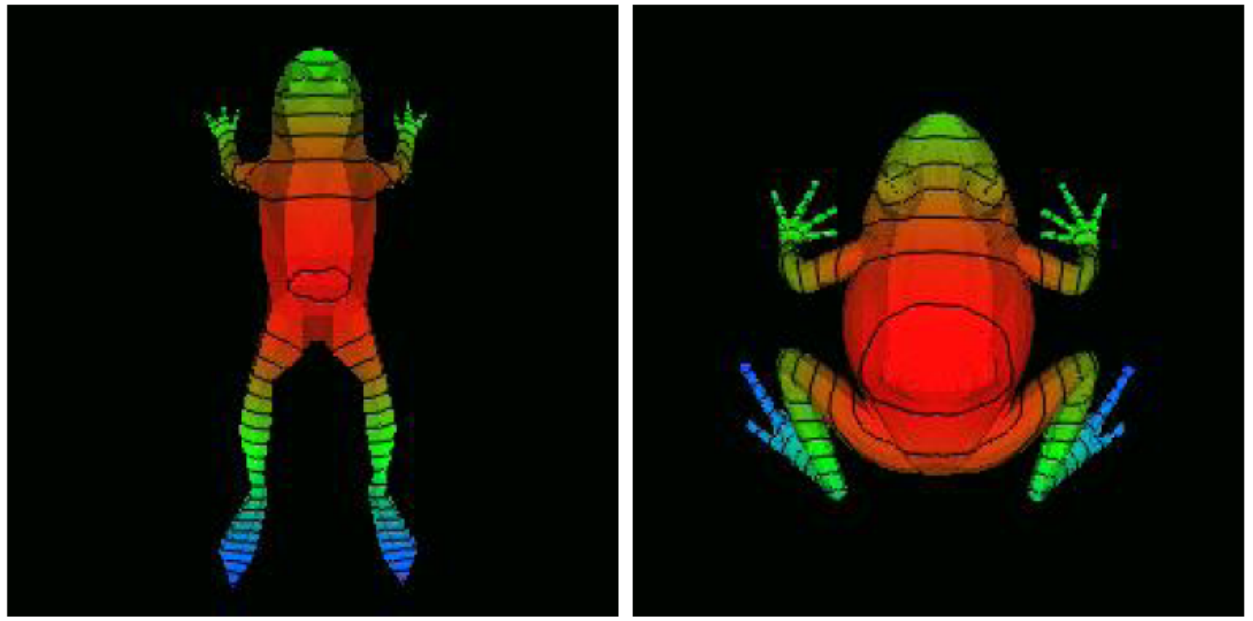
- Handle removal
- Skelentonize a shape
- Shape matching

(a) Our output (b) PGA [19] (c) LDPC [23]

Courtesy of Ge et al. 2011

Applications

- Handle removal
- Skelentonize a shape
- Shape matching



Courtesy of Hilaga et al. 2001

Reeb Space

- Given a topological space X and function $f : X \to Z$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a \}$
- A *contour* at value a :
  - a connected component of X_a
- *Reeb space* $RS_f(X)$ of X w.r.t. f:
  - continuous collapsing of each contour of f to a point
  - A continuous surjection $\phi : X \to R_f(X)$ s.t, $\phi(x) = \phi(y)$ if and only if x and y is in the same contour

Reeb Space

- Given a topological space X and function $f : X \to Z$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a \}$
- A *contour* at value a :
  - a connected component of X_a
- *Reeb space* $RS_f(X)$ of X w.r.t. f:
  - continuous collapsing of each contour of f to a point

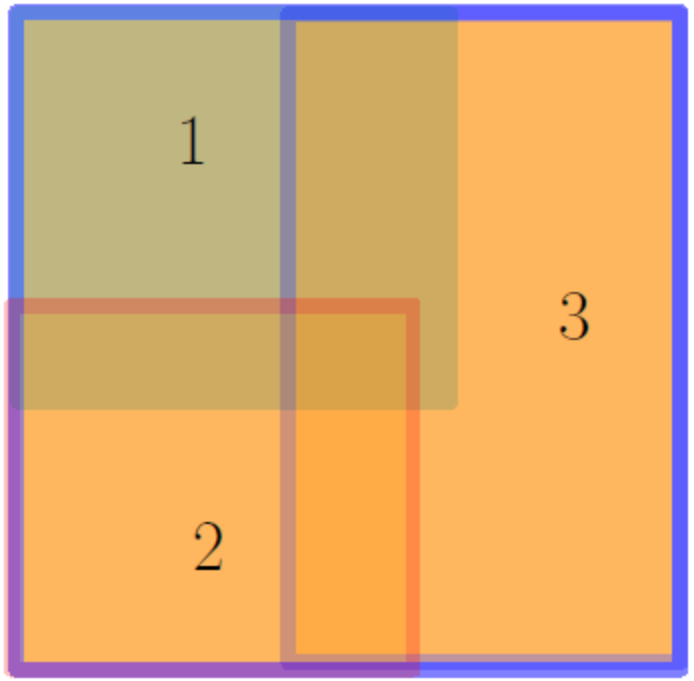
That is, Reeb space tracks connected components in the pre-image of any point in the co-domain Z.

- Study of Reeb space structure is intricate though. In general, it is not easy to compute the Reeb space.
  - [Edelsbrunner, Harer and Patel, 2008]
- The idea of viewing the structure of X from the lens of the map $f: X \to Z$ is interesting
  - In particular, often in practice, we may not know X but we can have several observations at points in X
- The Mapper construction!
  - [Singh, Mémoli and Carlsson, 2007]
  - Instead of considering pullbacks of all points in Z, and track their components, now considering pullbacks of elements in a cover of co-domain Z .
  - Tracking of components in such pullbacks is achieved via taking the nerve of these components.

Section 1:
Mapper: A topological summary of high dimensional data

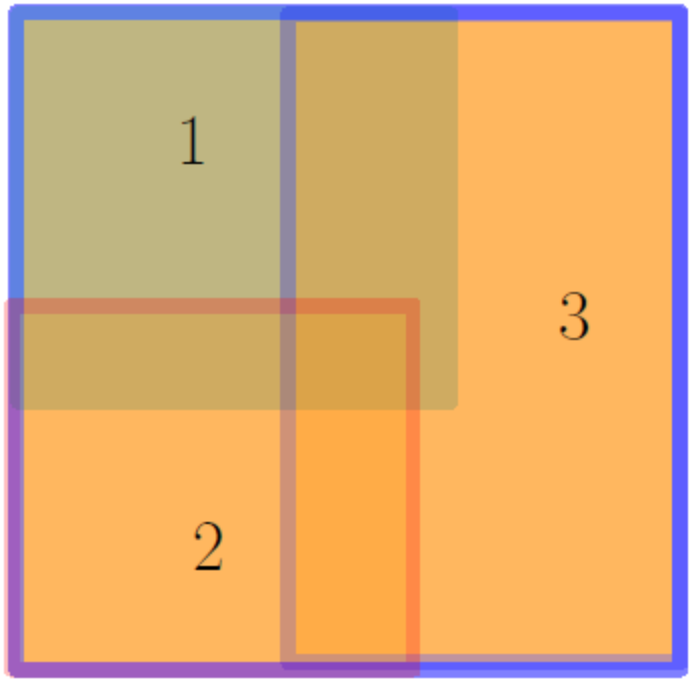
Covers and nerves

- A finite cover $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ of a space Y



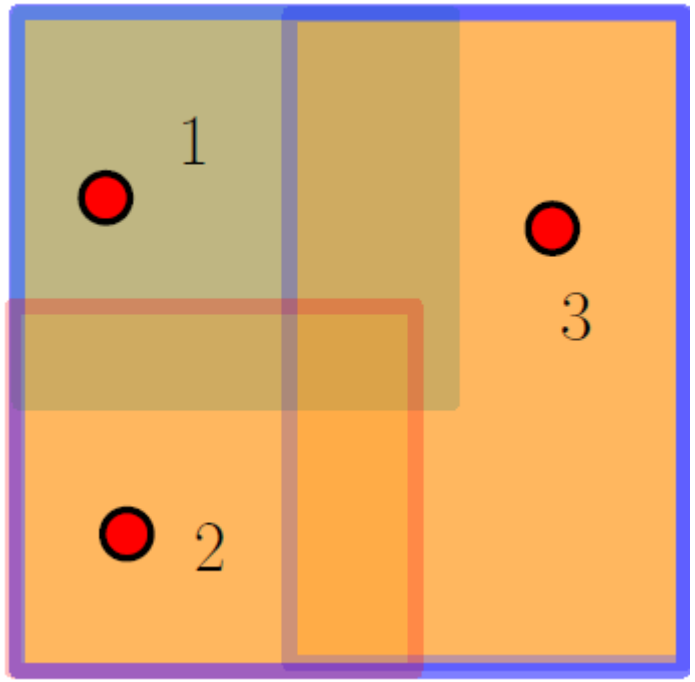
Covers and nerves

- A finite cover $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ of a space Y
- Nerve of $\mathcal{U}$: $\mathrm{N}(\mathcal{U})$
  - with vertex set A, and



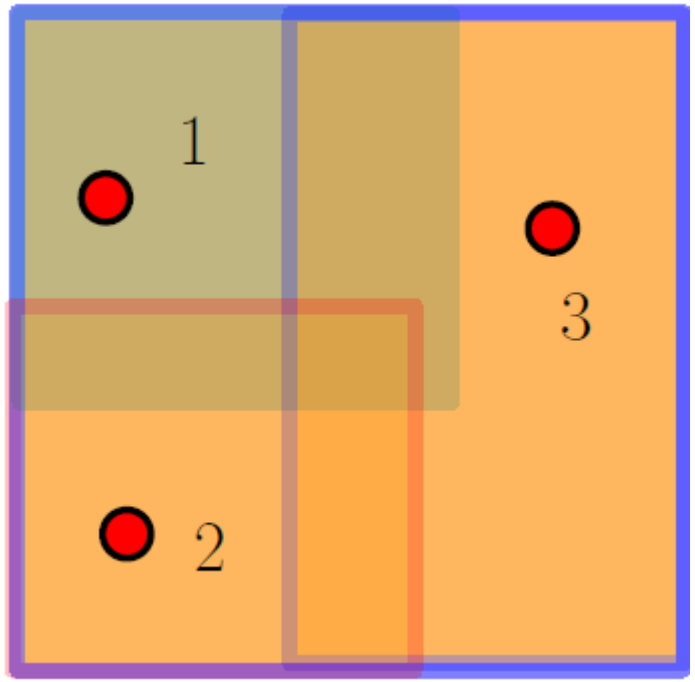
Covers and nerves

- A finite cover $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ of a space Y
- Nerve of $\mathcal{U}$: $\mathrm{N}(\mathcal{U})$
  - with vertex set A, and



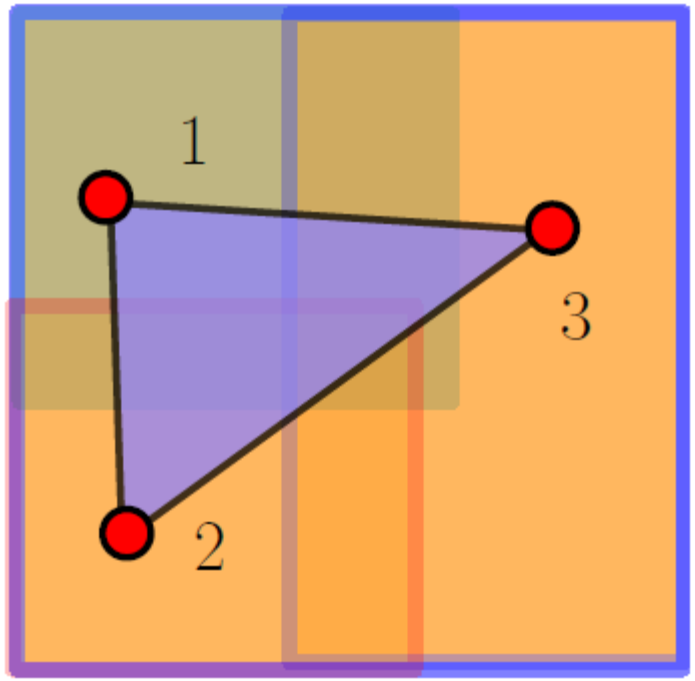
Covers and nerves

- A finite cover $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ of a space Y
- Nerve of $\mathcal{U}$: $\mathrm{N}(\mathcal{U})$
  - with vertex set A, and
  - simplex $(\alpha_0, \ldots, \alpha_k)$ iff $U_{\alpha_0} \cap U_{\alpha_1} \cap \cdots \cap U_{\alpha_k} \neq 0$



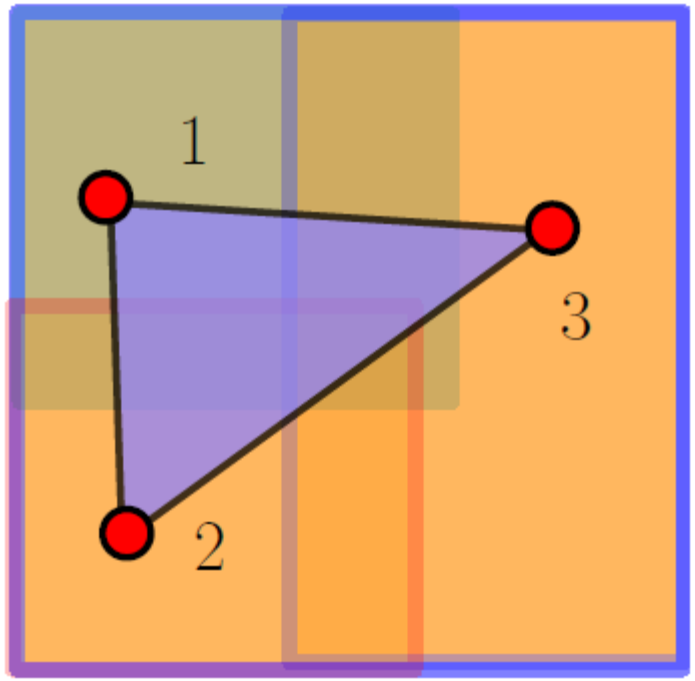
Covers and nerves

- A finite cover $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ of a space Y
- Nerve of $\mathcal{U}$: $\mathrm{N}(\mathcal{U})$
  - with vertex set A, and
  - simplex $(\alpha_0, \ldots, \alpha_k)$ iff $U_{\alpha_0} \cap U_{\alpha_1} \cap \cdots \cap U_{\alpha_k} \neq 0$



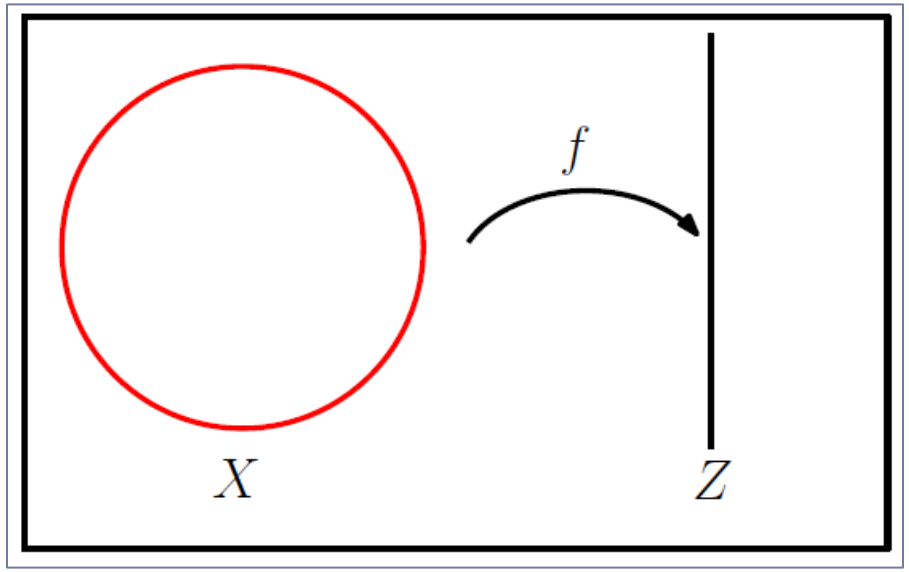
Covers and nerves

- A finite cover $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ of a space Y
- Nerve of $\mathcal{U}$: $\mathrm{N}(\mathcal{U})$
  - with vertex set A, and
  - simplex $(\alpha_0, \ldots, \alpha_k)$ iff $U_{\alpha_0} \cap U_{\alpha_1} \cap \cdots \cap U_{\alpha_k} \neq 0$
- One can view the nerve of a space as a discrete representation of the space via the cover
  - the cover provides the discretization



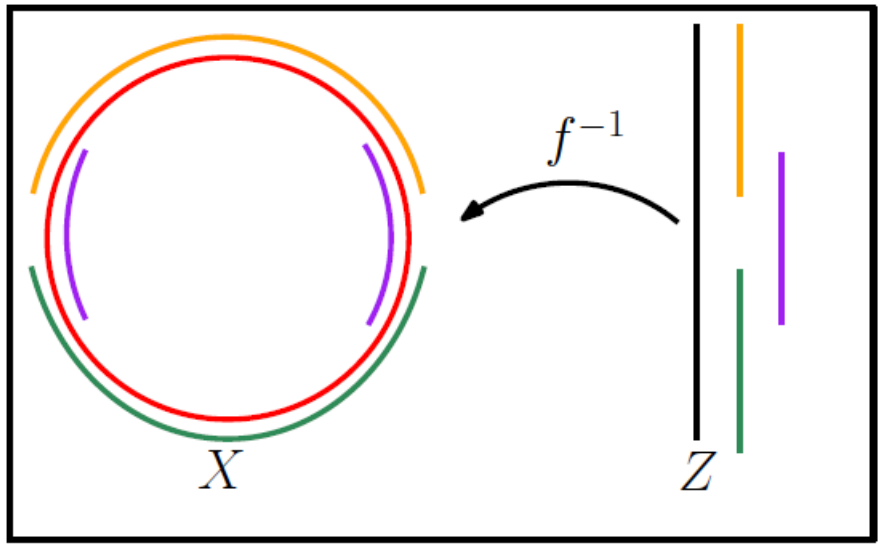
Pullback Cover

- Let $f: X \to Z$ be continuous and well-behaved, $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ a finite cover of Z



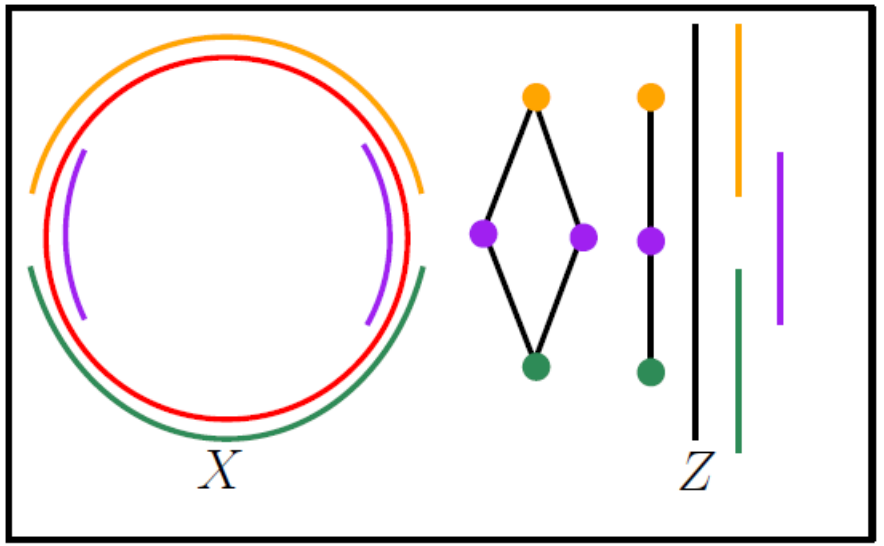
Pullback Cover

- Let $f: X \to Z$ be continuous and well-behaved, $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ a finite cover of Z
- Pullback cover via f and $\mathcal{U}$:
  - Connected components of $f^{-1}(U_\alpha) = \bigcup_{i=1}^{j_\alpha} V_{\alpha,i}$, for all $\alpha \in A$, form a cover $f^*(\mathcal{U})$ of X

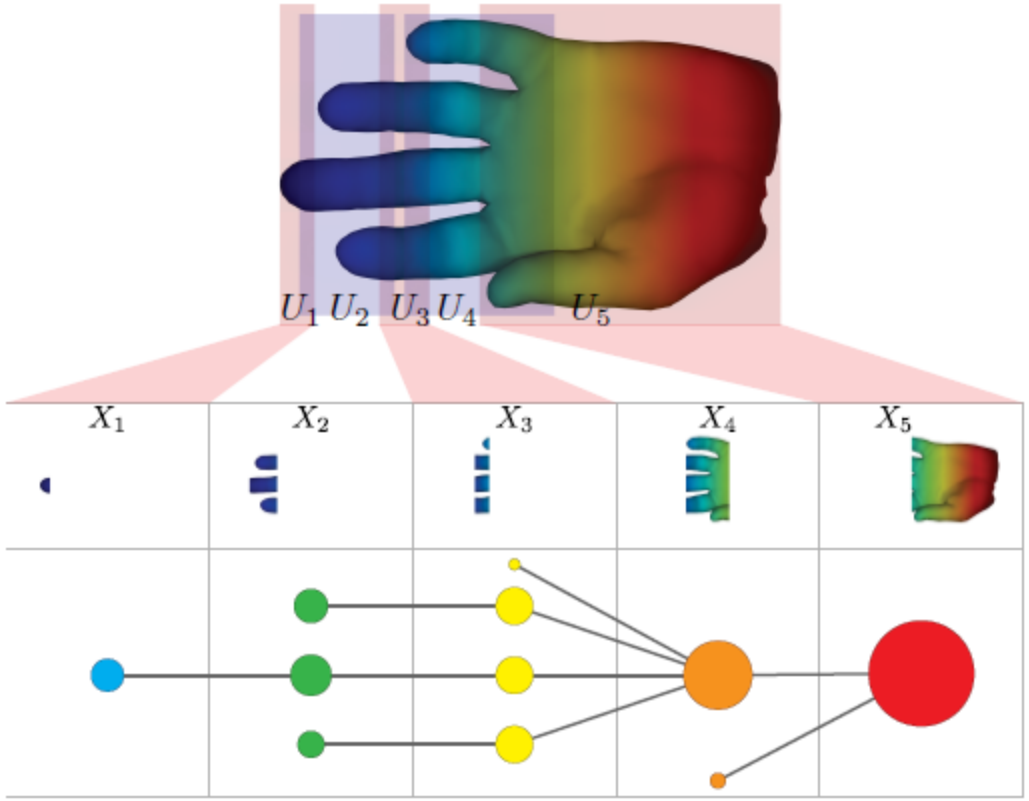


Pullback Cover

- Let $f: X \to Z$ be continuous and well-behaved, $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ a finite cover of Z
- Pullback cover via f and $\mathcal{U}$:
  - Connected components of $f^{-1}(U_\alpha) = \bigcup_{i=1}^{j_\alpha} V_{\alpha,i}$, for all $\alpha \in A$, form a cover $f^*(\mathcal{U})$ of X
- Mapper: $M(\mathcal{U}, f) := N(f^*(\mathcal{U}))$ the nerve of the pullback cover $f^*(\mathcal{U})$!

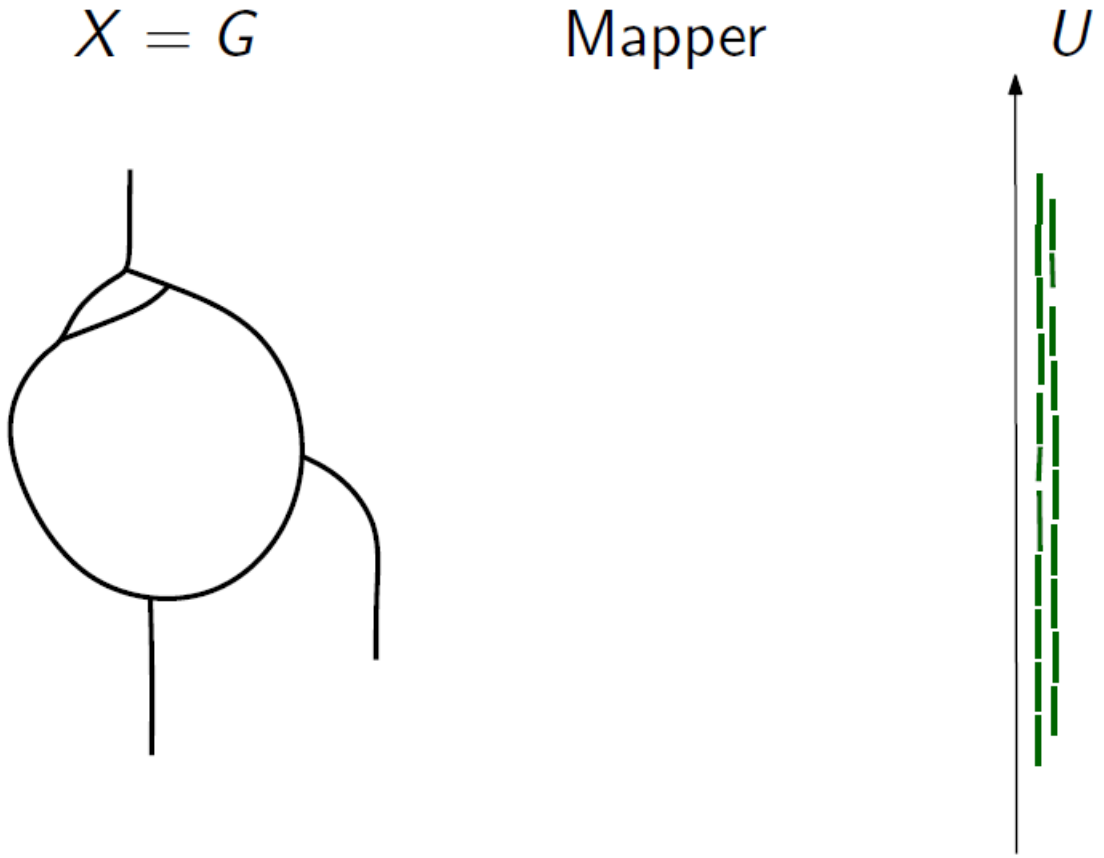


Another example



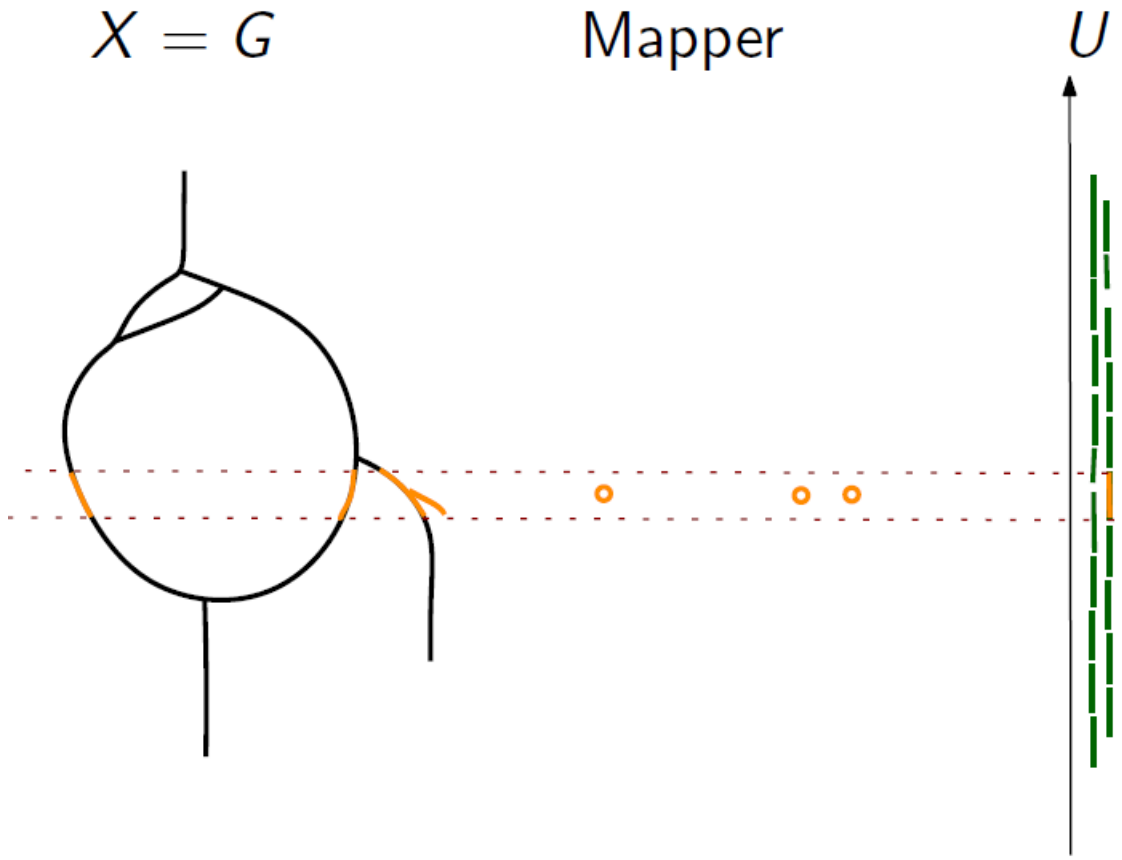
Mapper vs Reeb Space

- Consider a real-valued function $f: X \to R$ (i.e, $Z = R$)



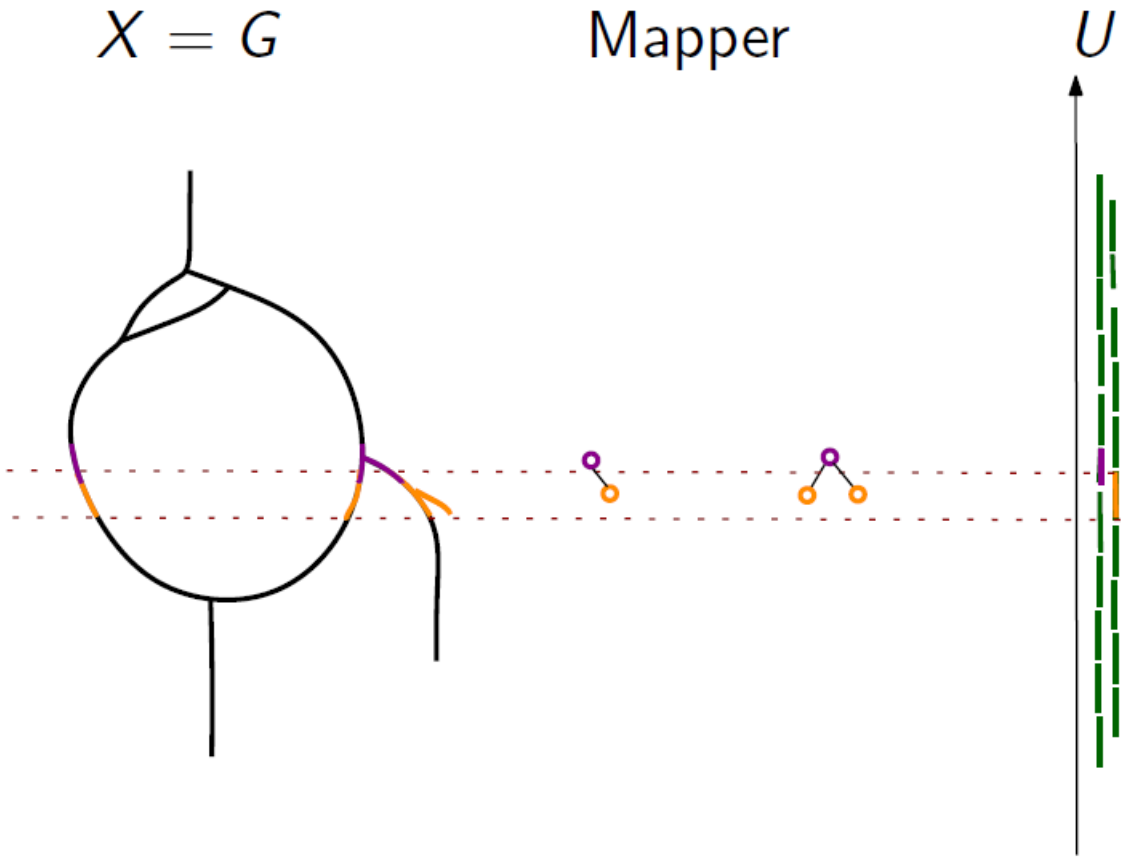
Mapper vs Reeb Space

- Consider a real-valued function $f: X \to R$ (i.e, $Z = R$)



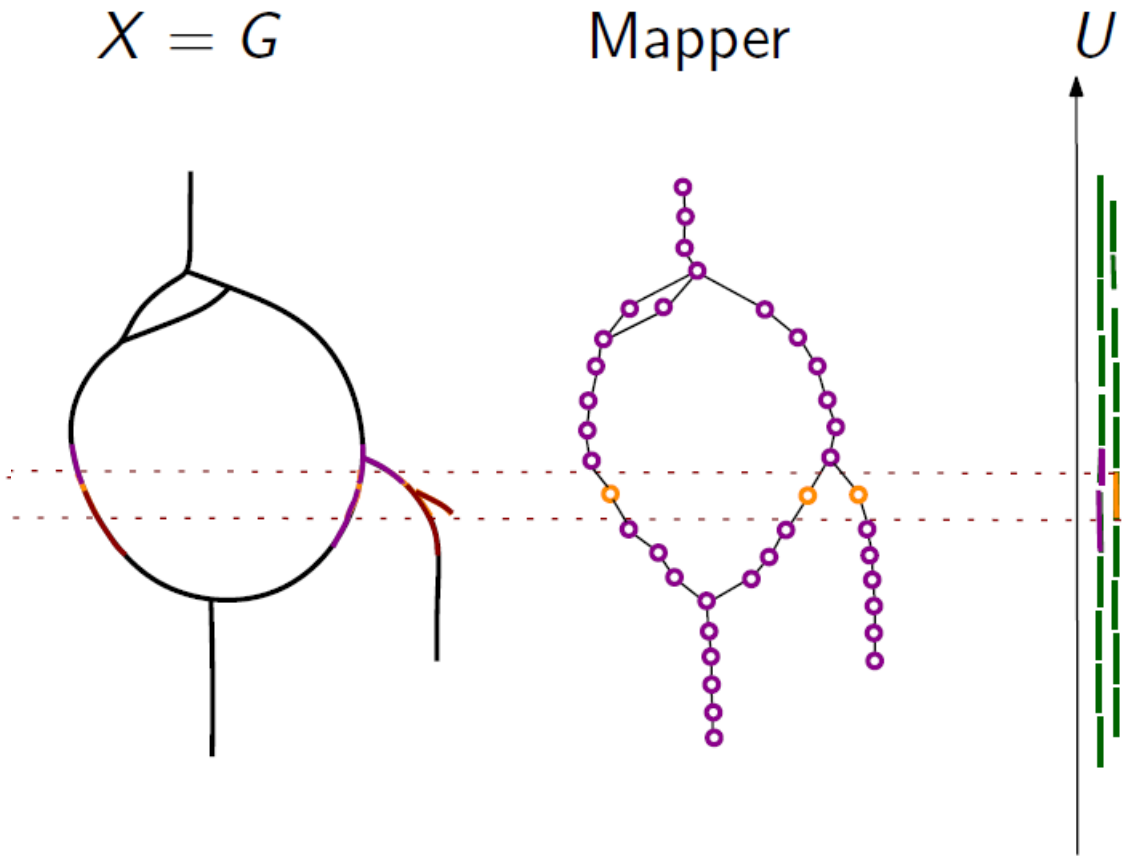
Mapper vs Reeb Space

- Consider a real-valued function $f: X \to R$ (i.e, $Z = R$)



Mapper vs Reeb Space

- Consider a real-valued function $f: X \to R$ (i.e, $Z = R$)



Mapper vs. Reeb Space

- In some sense, mapper structure can be considered as a coarsening of Reeb space via the coarsening of the co-domain (via a cover of it)
  - Certain convergences results known [Munch, B. Wang, 2016], [Dey, Mémoli, Wang, 2017]

Theorem 32. *Under the conditions above,*

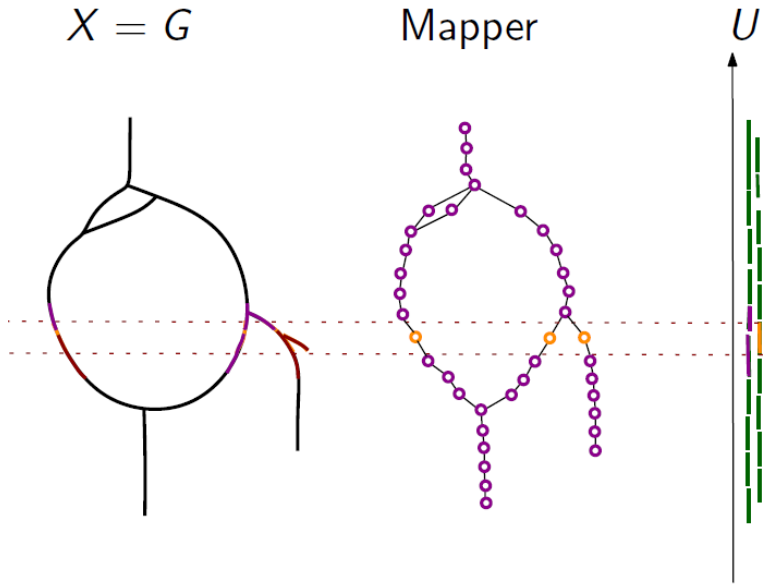
$$d_{GH}\big((R_f, \tilde{d}_f), (P_\delta, d_\delta)\big) \leq 5\delta.$$

Mapper vs. Reeb Space

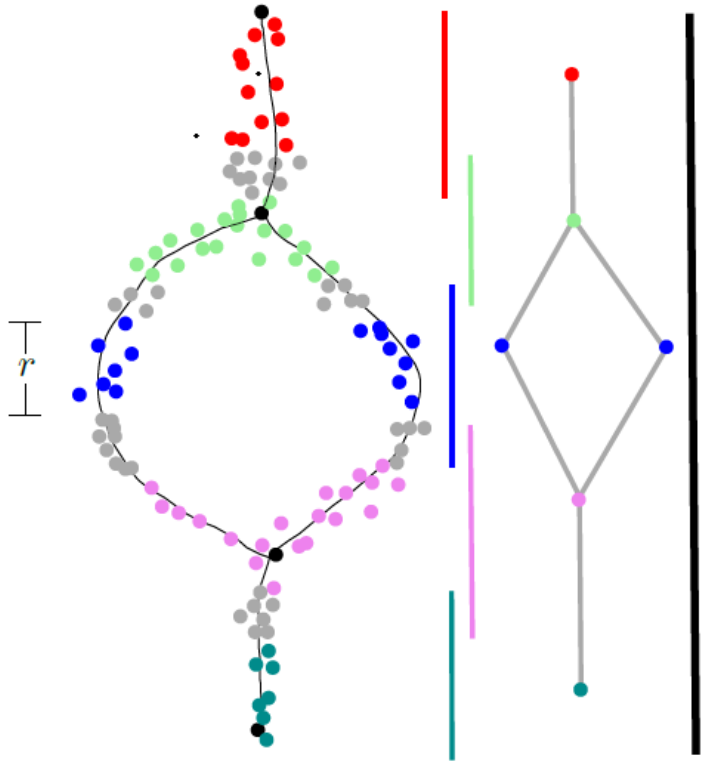
- In some sense, mapper structure can be considered as a coarsening of Reeb space via the coarsening of the co-domain (via a cover of it)
  - Certain convergences results known [Munch, B. Wang, 2016], [Dey, Mémoli, Wang, 2017]

Theorem 32. *Under the conditions above,*

$$d_{GH}\big((R_f, \tilde{d}_f), (P_\delta, d_\delta)\big) \leq 5\delta.$$

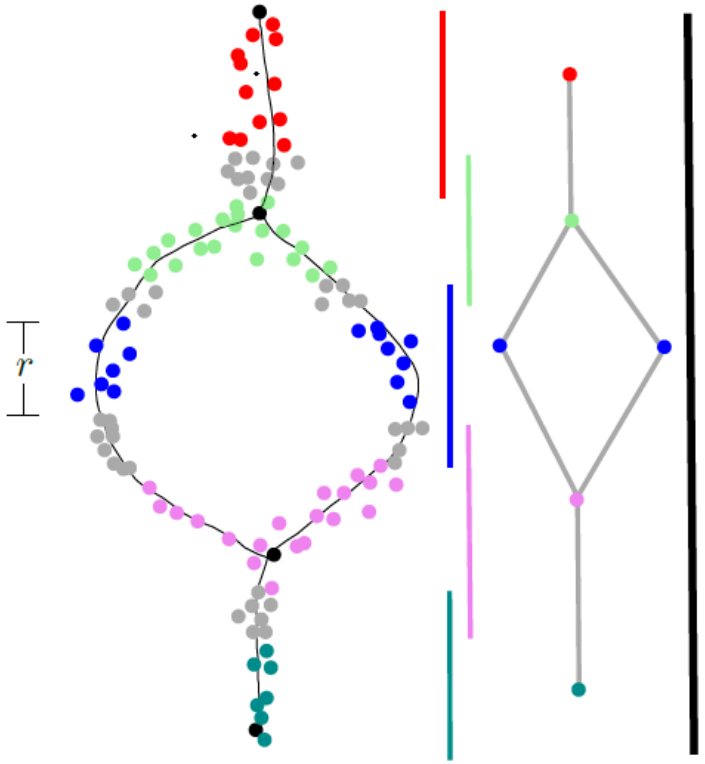


Mapper in Practice



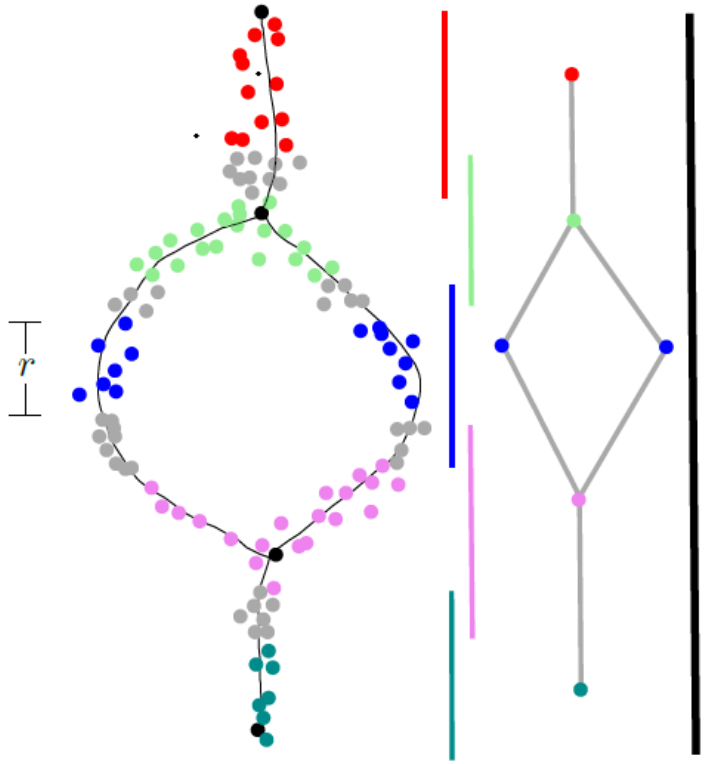
Mapper in Practice

- If X is discrete, then $f^{-1}U_\alpha$ is also discrete and is just a union of points. The nerve doesn't make sense now
- Use clustering algorithm to construct clusters from the set of points $f^{-1}U_\alpha$
  - DBSCAN
  - Single linkage clustering
  - ...



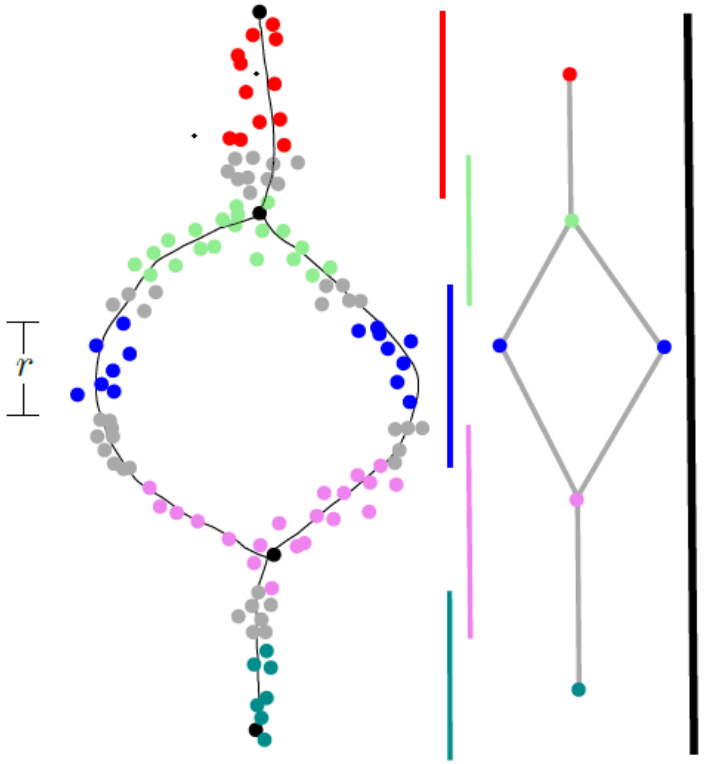
Mapper in Practice

- In practice, for point clouds data, a clustering algorithm can be applied to the pullback of each cover elements to identify the components



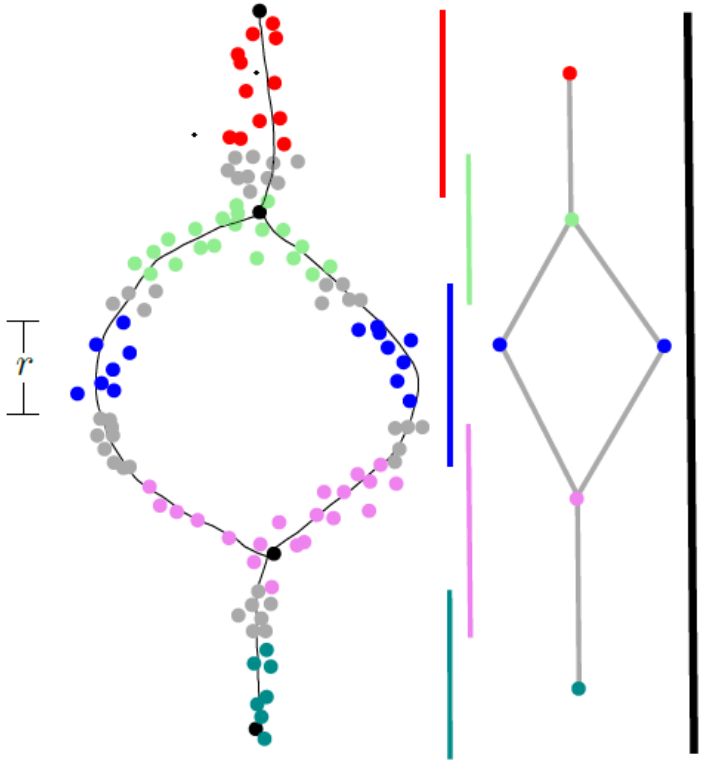
Mapper in Practice

- In practice, for point clouds data, a clustering algorithm can be applied to the pullback of each cover elements to identify the components
- 1-skeleton of the mapper structure is often used, as a platform for data exploration



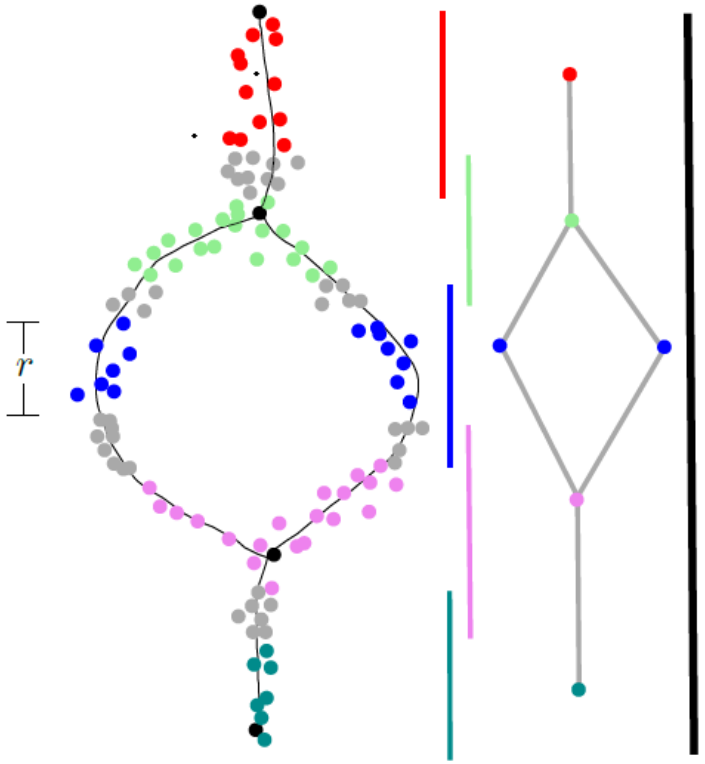
Mapper in Practice

- In practice, for point clouds data, a clustering algorithm can be applied to the pullback of each cover elements to identify the components
- 1-skeleton of the mapper structure is often used, as a platform for data exploration
- Mapper can be used as a replacement for dimensionality reduction
  - serving as a low-dimensional metaphor for the continuous space of high dimensional data

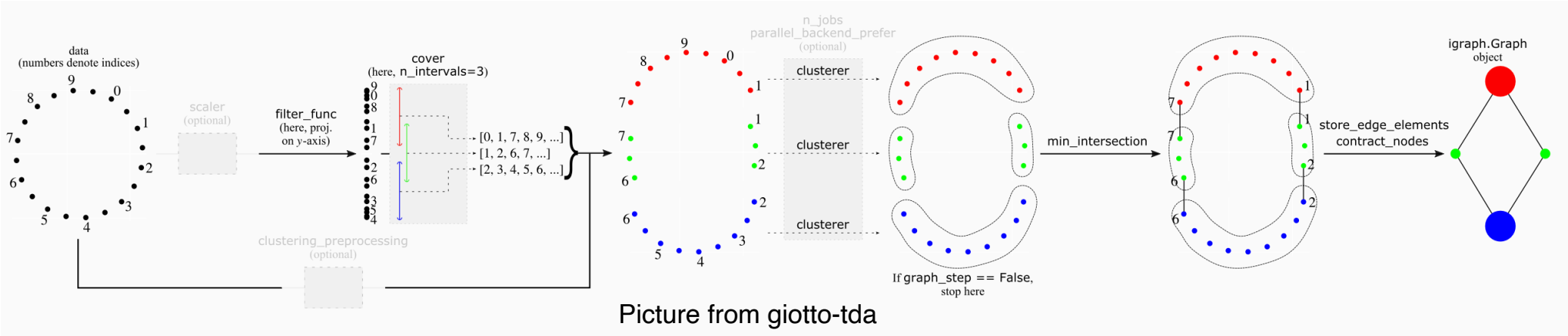


Mapper in Practice

- In practice, for point clouds data, a clustering algorithm can be applied to the pullback of each cover elements to identify the components
- 1-skeleton of the mapper structure is often used, as a platform for data exploration
- Mapper can be used as a replacement for dimensionality reduction
  - serving as a low-dimensional metaphor for the continuous space of high dimensional data
- Input data can be just point cloud data
  - helper functions (called filter functions) will be used to serve as $f: X \to R^d$



A standard Mapper pipeline in practice



Picture from giotto-tda

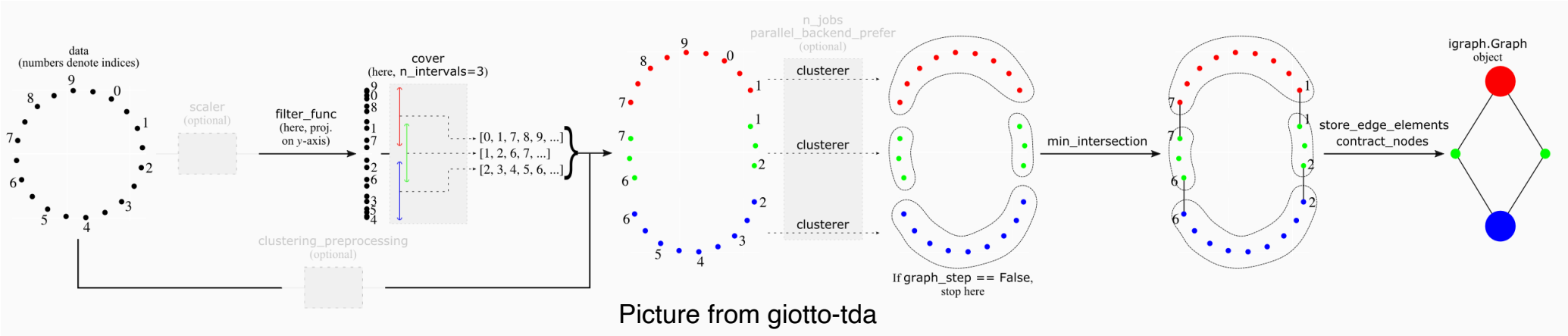
- Input: high dimensional PCD P

  - Step 1: Choose a few (d) filter functions $F: P \to R^d$, which could incorporate domain knowledge (or can be as simple as eigenfunctions from PCA)

  - Step 2: Create the Mapper structure (upto 2-skeleton, i.e, vertices, edges and triangles; often just 1-skeleton) w.r.t. F and some cover of R^d (i.e, just some ``rectangular"-tiling), where connected components in pullbacks are computed by some clustering algorithm

  - Step 3: Visualize the 1-skeleton (graph skeleton) of Mapper structure using a graph layout algorithm

A standard Mapper pipeline in practice



Picture from giotto-tda

- Parameters
  - Filter function $f : X \to \mathbb{R}$
  - Cover of $im(f)$ by open intervals
  - Clustering method and its parameters

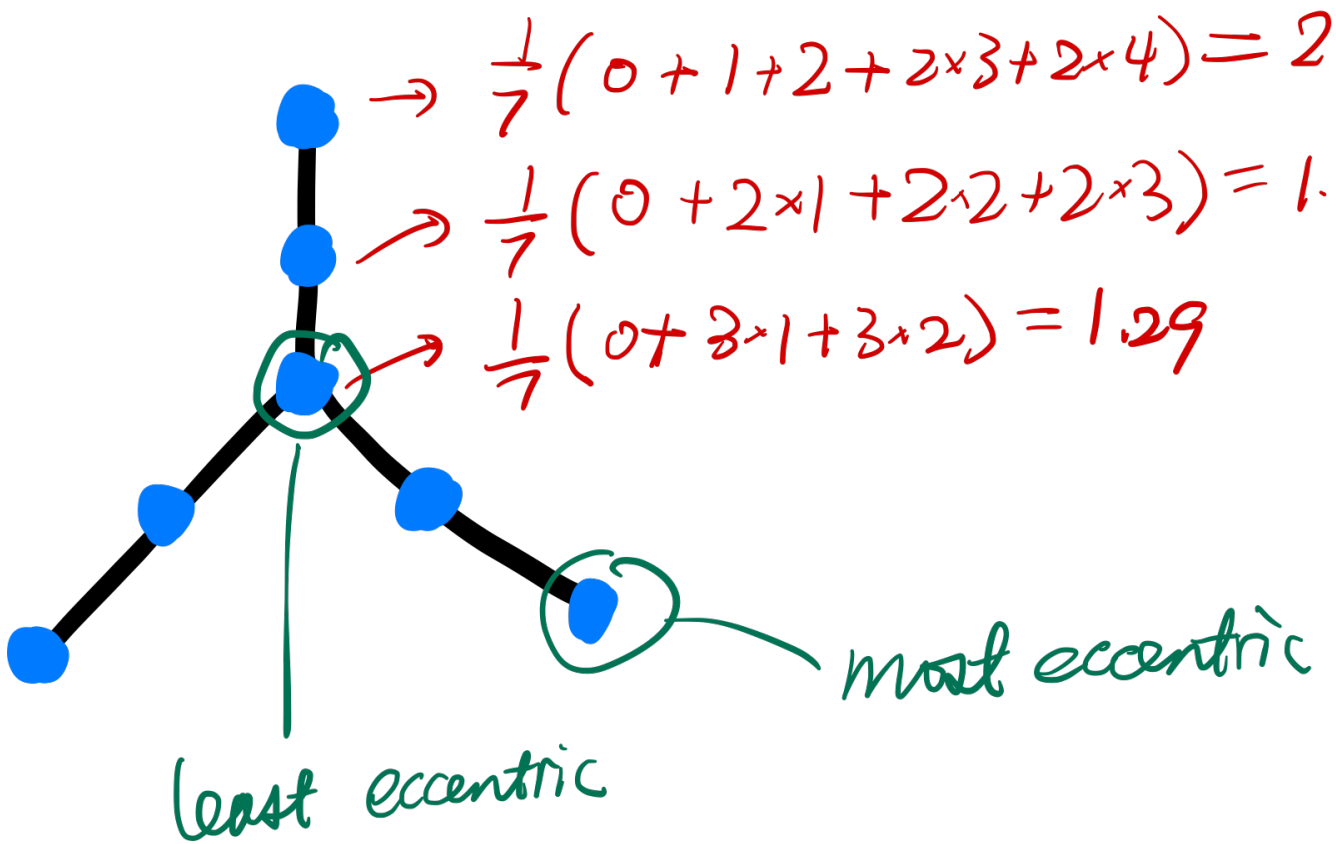
Examples of filter functions

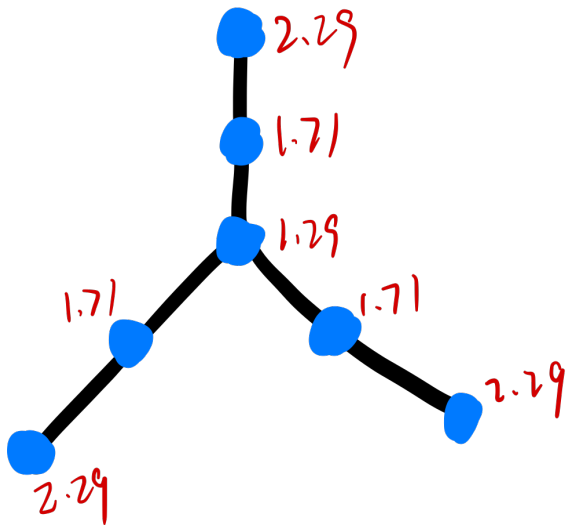
- Data driven function: The dataset X could come with a function. For example, points in X are locations on earth, $f : X \to \mathbb{R}$ could be temperature

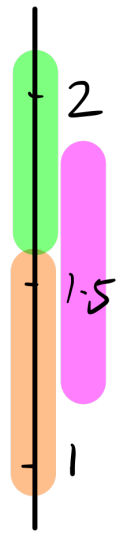
- Density: suppose $X \subset \mathbb{R}^N$. A density estimator is a function which takes X as input and returns a probability density $s_X : \mathbb{R}^N \to \mathbb{R}$

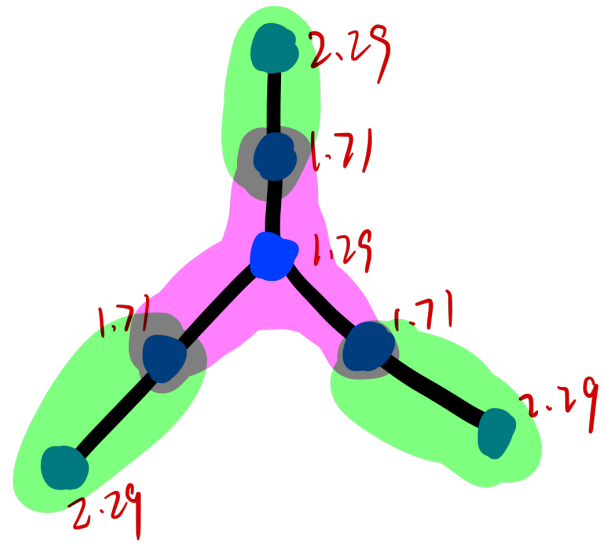
$$\widehat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\Big(\frac{x - x_i}{h}\Big),$$

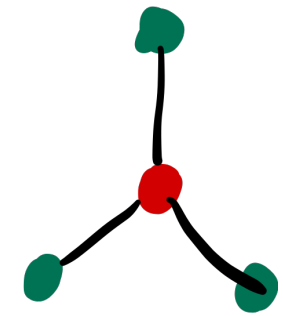
- Eccentricity: Let (X, d_X) be a metric space. For $p \in \mathbb{Z}_{\geq 0}$, the p-eccentricity function $e_p : X \to \mathbb{R}$ of X is defined by sending $x \in X$ to $\sum_{x' \in X} d_X(x, x')^p / |X|$




2.29
1.71
1.29
1.71
1.71
2.29
2.29

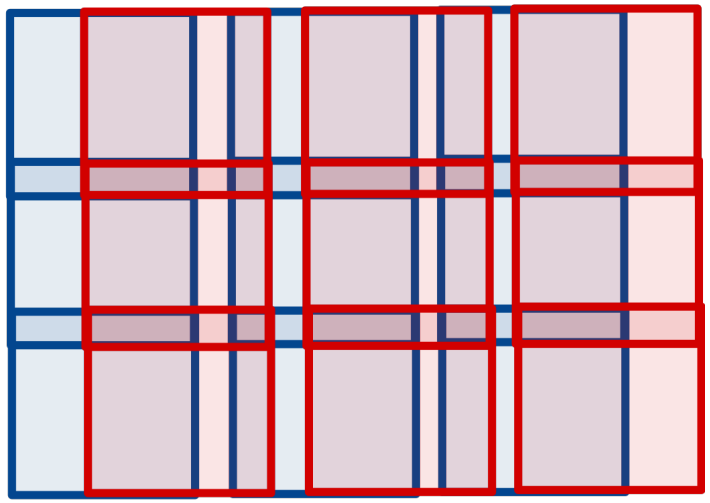

2
1.5
1


2.29
1.71
1.29
1.71
1.71
2.29
2.29



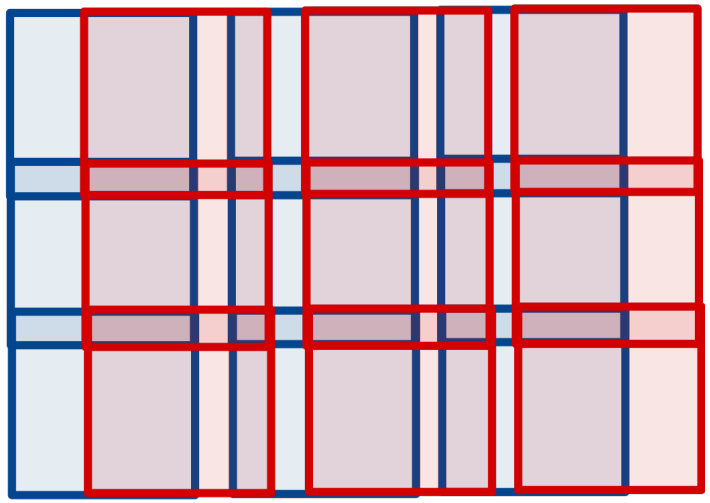
Choice of cover

- Z is usually chosen to be $\mathbb{R}$ or $\mathbb{R}^2$
- Intervals or hypercube covers



Choice of cover

- Z is usually chosen to be $\mathbb{R}$ or $\mathbb{R}^2$
- Intervals or hypercube covers



Demo

Mapper for high-D data exploration

- Visualizing and exploring high dimensional data
  - Clustering is limited, ignoring the continuous space behind data
  - Dimensionality reduction can be misleading due to that the target dimension is much smaller than intrinsic dimension
- Mapper provides a low-D metaphor for the continuous space behind high dimensional data, via the lens of filter functions

Mapper in Applications

- Extracting insights from the shape of complex data using topology, Lum et al., Nature, 2013
- Topological Data Analysis for Discovery in Preclinical Spinal Cord Injury and Traumatic Brain Injury, Nielson et al., Nature, 2015
- Using Topological Data Analysis for Diagnosis Pulmonary Embolism, Rucco et al., arXiv preprint, 2014
- Topological Methods for Exploring Low-density States in Biomolecular Folding Pathways, Yao et al., J . Chemical Physics, 2009
- CD8 T-cell reactivity to islet antigens is unique to type 1 while CD4 T-cell reactivity exists in both type 1 and type 2 diabetes, Sarikonda et al., J . Autoimmunity , 2013
- Innate and adaptive T cells in asthmatic patients: Relationship to severity and disease mechanisms, Hinks et al., J . Allergy Clinical Immunology , 2015

Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival

Monica Nicolau[a], Arnold J. Levine[b,1], and Gunnar Carlsson[a,c]

[a]Department of Mathematics, Stanford University, Stanford, CA 94305; [b]School of Natural Sciences, Institute for Advanced Study, Princeton, NJ 08540; and [c]Ayasdi, Inc., Palo Alto, CA 94301

Contributed by Arnold J. Levine, February 25, 2011 (sent for review July 23, 2010)

High-throughput biological data, whether generated as sequencing, transcriptional microarrays, proteomic, or other means, continues to require analytic methods that address its high dimensional aspects. Because the computational part of data analysis ultimately identifies shape characteristics in the organization of data sets, the mathematics of shape recognition in high dimensions continues to be a crucial part of data analysis. This article introduces a method that extracts information from high-throughput microarray data and, by using topology, provides greater depth of information than current analytic techniques. The method, termed *Progression Analysis of Disease* (*PAD*), first identifies robust aspects of cluster analysis, then goes deeper to find a multitude of biologically meaningful shape characteristics in these data. Additionally, because *PAD* incorporates a visualization tool, it provides a simple picture or graph that can be used to further explore these data. Although *PAD* can be applied to a wide range of high-throughput data types, it is used here as an example to analyze breast cancer transcriptional data. This identified a unique subgroup of *Estrogen Receptor*-positive (*ER*$^+$) breast cancers that express high levels of *c-MYB* and low levels of innate inflammatory genes. These patients exhibit 100% survival and no metastasis. No supervised step beyond distinction between tumor and healthy patients was used to identify this subtype. The group has a clear and distinct, statistically significant molecular signature, it highlights coherent biology but is invisible to cluster methods, and does not fit into the accepted classification of *Luminal A/B*, *Normal-like* subtypes of *ER*$^+$ breast cancers. We denote the group as *c-MYB*$^+$ breast cancer.

applied topology | p53 | systems biology

Increasingly it has become clear that, for most cancers, understanding the disease demands exploring biological processes as complex functioning systems and the pathology observed as a disruption in the coordinated performance of such systems. This viewpoint necessitates incorporating high-throughput data in the study of these diseases and consequently demands the continued development of mathematical analytic methods geared specifically to such data. The fundamental mathematical challenges in extracting meaningful information from high-throughput biological data stem, ultimately, from the difficulty in understanding the intrinsic shape of data in high dimensions (1). Shape characteristics such as kurtosis, modality, or the presence of outliers have always played a crucial role in the analysis of data, but the high dimensionality of genomic data poses mathematical difficulties in identifying its geometry. Additionally, biological phenomena are intrinsically highly variable and stochastic in nature, and notions of biological similarity are less rigid. Consequently, analysis methods for biomedical data need to identify shape characteristics that are fairly robust to changes by rescaling of distances and therefore become more qualitative in nature. This has led us to use methods adapted from the mathematics area of topology, which studies precisely the characteristics of shapes that are not rigid. The particular method we introduce in the present article is intermediate between clustering and more distance-sensitive methods like *Principal Component Analysis* (*PCA*) and multidimensional scaling. This hybrid approach is able to extract unique biology from data sets. As an example, we applied our method of analysis to breast cancer transcriptional genomic data and identified a molecularly distinct unique breast cancer subgroup of *Estrogen Receptor*-positive (*ER*$^+$) tumors that have 100% overall survival and whose molecular signature is distinct from normal tissue and other breast cancers.

This article introduces *Progression Analysis of Disease* (*PAD*), an approach to data analysis of disease that unravels the geometry of data sets and provides an easily accessible picture of the outcome. This method is an application of *Mapper* (2), a mathematical tool that builds a simple geometric representation of data along preassigned guiding functions called filters. *Mapper* provides both a method for mathematical data analysis and a visualization tool; the filter functions introduced through *Mapper* define a framework for supervised analysis. The output of the analysis approximates a collapse of the data into a simple, low dimensional shape, and the filter functions act as guides along which the collapse is done. *Mapper* has already been used successfully to uncover unique subtle aspects of the folding patterns of RNA (3). Here we define an application of *Mapper* to the analysis of transcriptionally genomic data from disease, with guiding filter functions provided by *Disease-Specific Genomic Analysis* (*DSGA*) (4). *DSGA* is a method of mathematical analysis of genomic data that highlights the component of data relevant to disease, by defining a transformation that measures the extent to which diseased tissue deviates from healthy tissue. *DSGA* has been shown to both (*i*) outperform traditional methods of analysis, and (*ii*) highlight unique biology. In combination with *Mapper*, *DSGA* transformations provide a means to define the guiding filter function, essentially by unraveling the data according to the extent of overall deviation from a healthy state.

We make *PAD* available as a Web tool, with options for *DSGA* only, *Mapper* only, or a combination of the two (5).

Our method, *PAD*, is able to identify geometric characteristics of these data that are obscured when using cluster analysis. Long gradual drifts in the graphs of these data are visible, as for example are expected when the results consist of patients with progressively advanced stages of disease. More importantly, by preserving the geometry of these data, *PAD* has identified a unique subset of breast cancers that exhibit clear and coherent clinical characteristics. Specifically, we applied *PAD* to breast cancer transcriptional microarray data (6) and identified two

Author contributions: M.N., A.J.L., and G.C. designed research; M.N. performed research; M.N., A.J.L., and G.C. analyzed data; and M.N., A.J.L., and G.C. wrote the paper.

The authors declare no conflict of interest.

Freely available online through the PNAS open access option.

[1]To whom correspondence should be addressed. E-mail: alevine@ias.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1102826108/-/DCSupplemental.

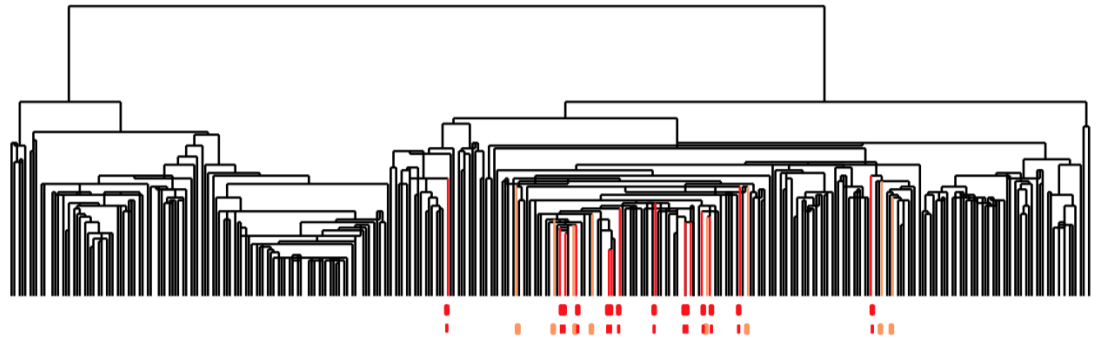
- Disease-Specific Genomic Analysis

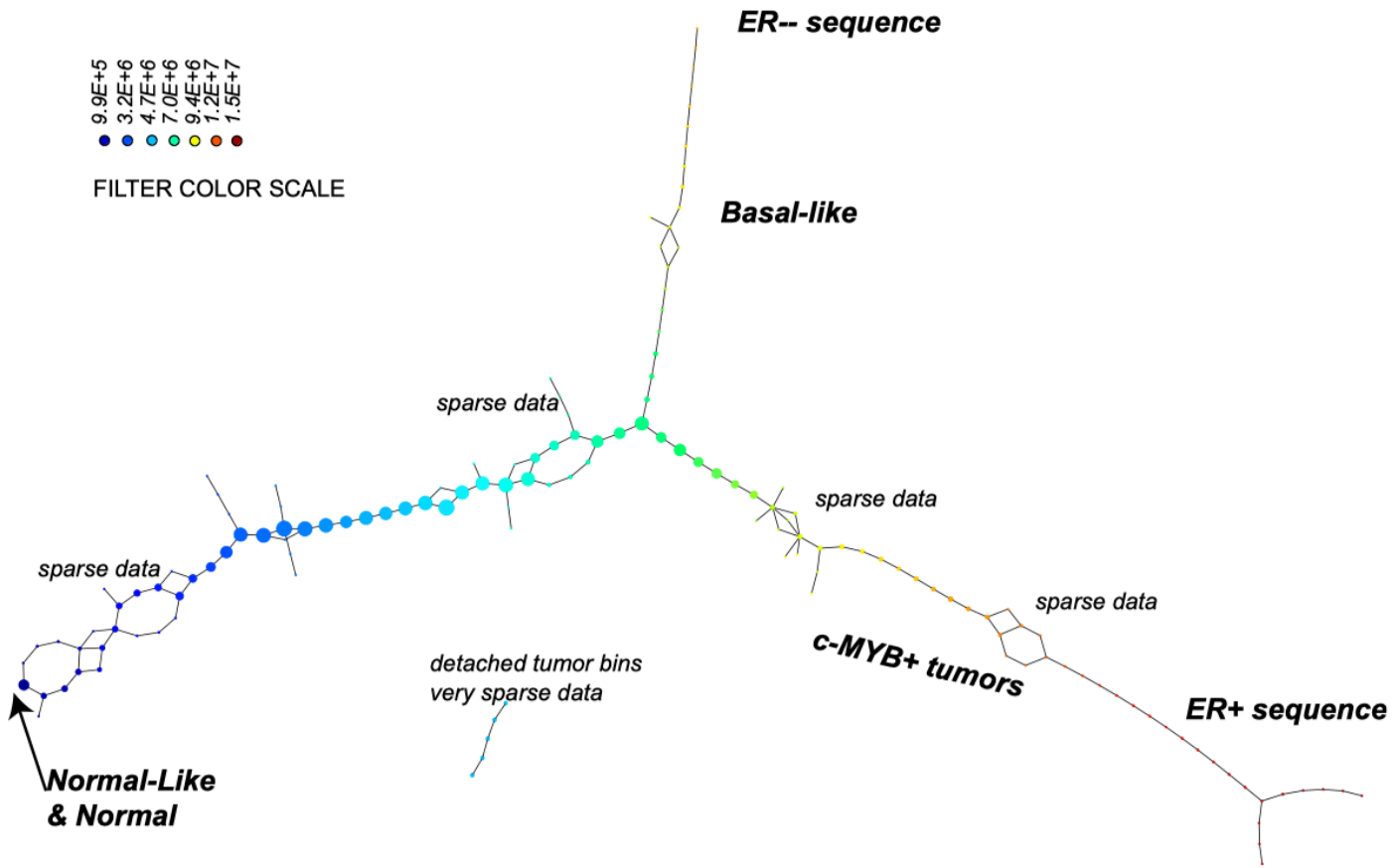
$$\vec{T} = Nc.\vec{T} + Dc.\vec{T}.$$

- Filter function measures gene expression deviation from normal tissue

$$\vec{V} = <g_1, g_2, \ldots, g_s>.$$

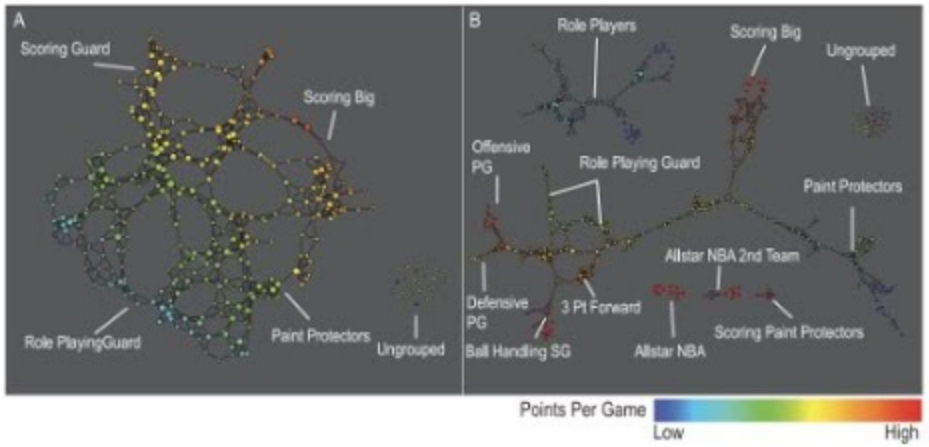
$$f_{p,k}(\vec{V}) = [\Sigma |\, g_r|^p]^{k/p}.$$

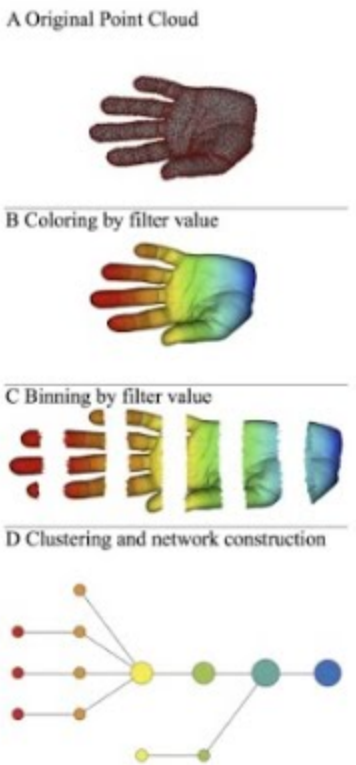



9.9E+5
3.2E+6
4.7E+6
7.0E+6
9.4E+6
1.2E+7
1.5E+7
FILTER COLOR SCALE
ER-- sequence
Basal-like
sparse data
sparse data
sparse data
sparse data
detached tumor bins
very sparse data
c-MYB+ tumors
ER+ sequence
Normal-Like
& Normal

Using TDA to Define College Basketball Positions with Mapper

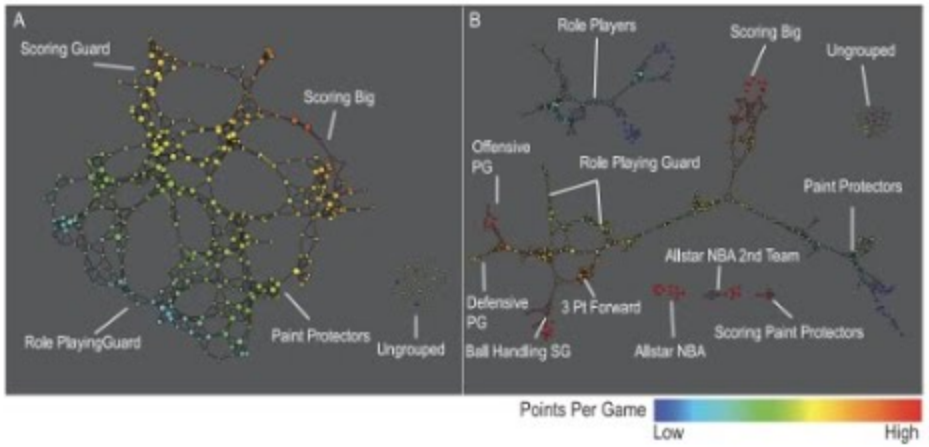
Mark Yukelis and Alan Suh

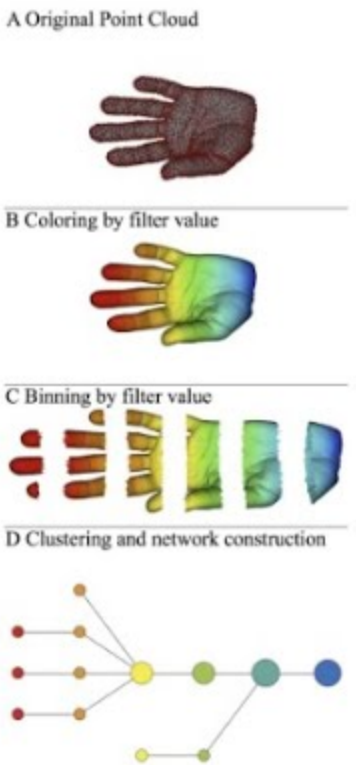




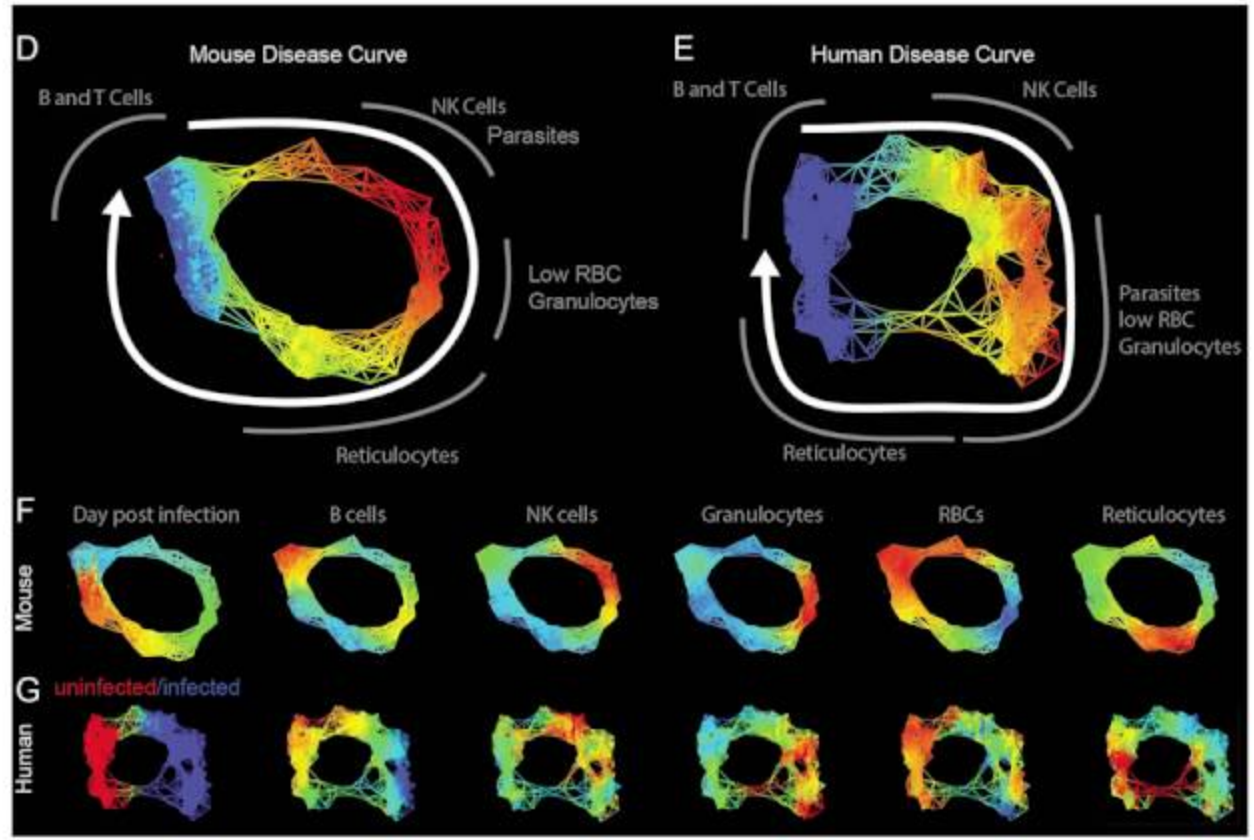
Using TDA to Define College Basketball Positions with Mapper

Mark Yukelis and Alan Suh





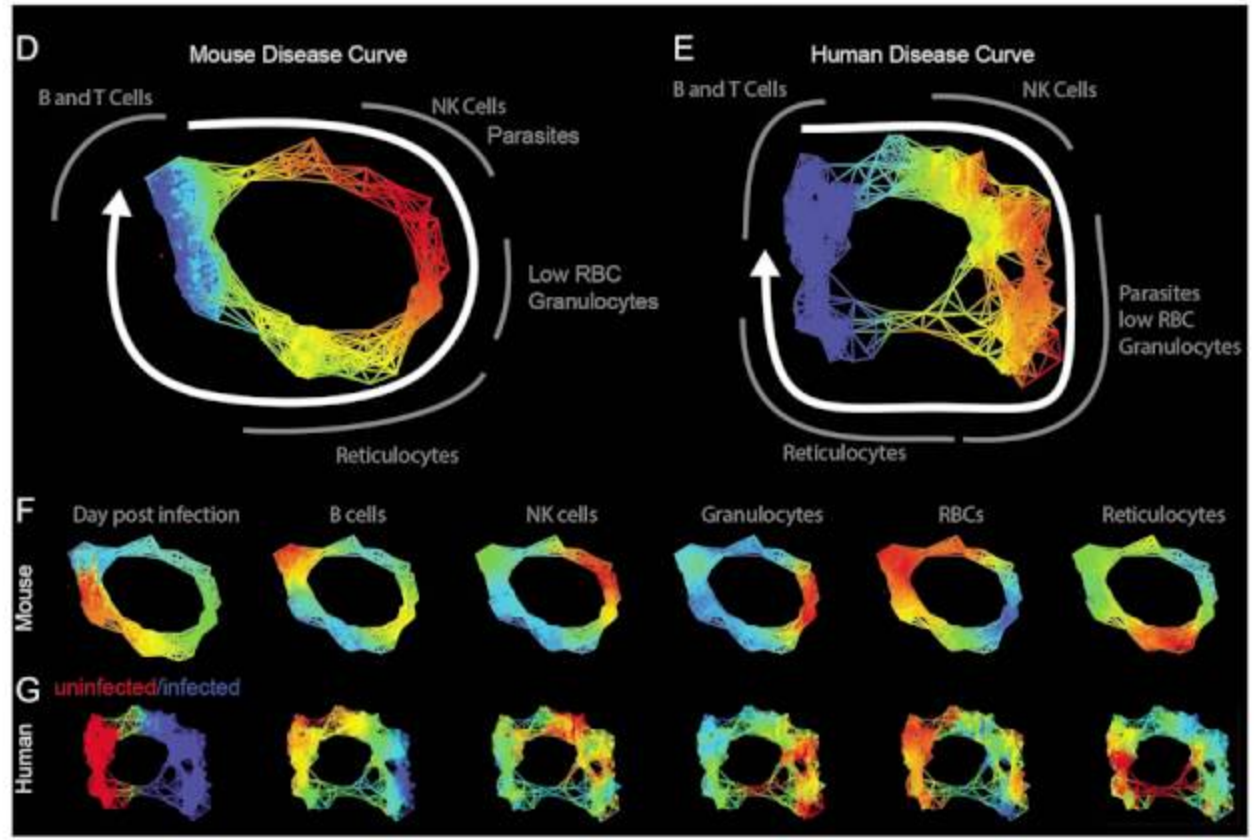
Mapping



Torres et al, PLOS Biology, 2016

ICME Institute for Computational & Mathematical Engineering

Mapping



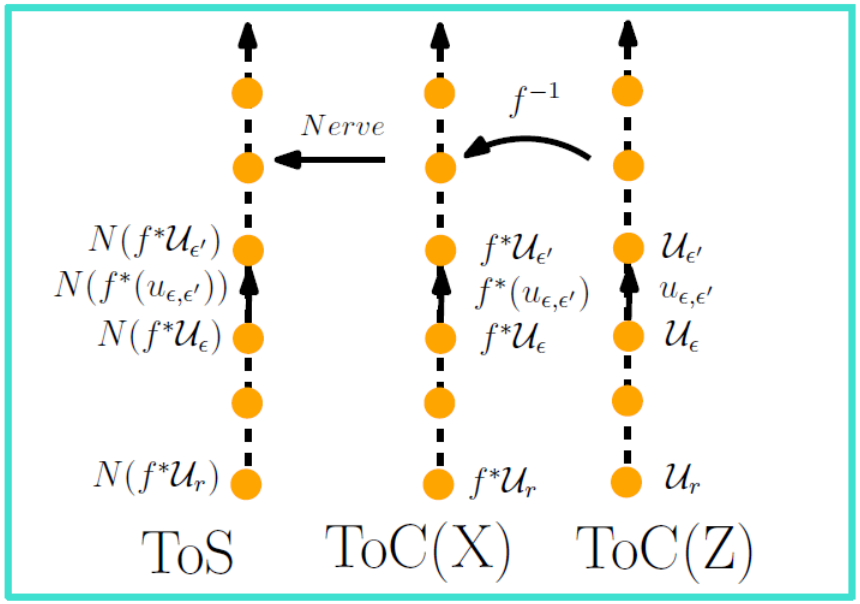
Torres et al, PLOS Biology, 2016

ICME Institute for Computational & Mathematical Engineering

- Mapper is the key behind the company AyasdiAI
- The company was acquired by SymphonyAI in 2019

Section 2:
Multi-mapper: A multiscale representation of general maps

Main idea



- Consider a sequence of coarser and coarser covers (aka, look at the discretization at coarser and coarser resolution)
- Pull back, look at the sequence of coarser and coarser Mapper structures
- This gives rise to a sequence of simplicial complexes, called multiscale mapper, and we can compute its persistent homology.

Motivation

- Mapper is at a fixed scale
  - How to choose the scale? Why not look at all scales?

Motivation

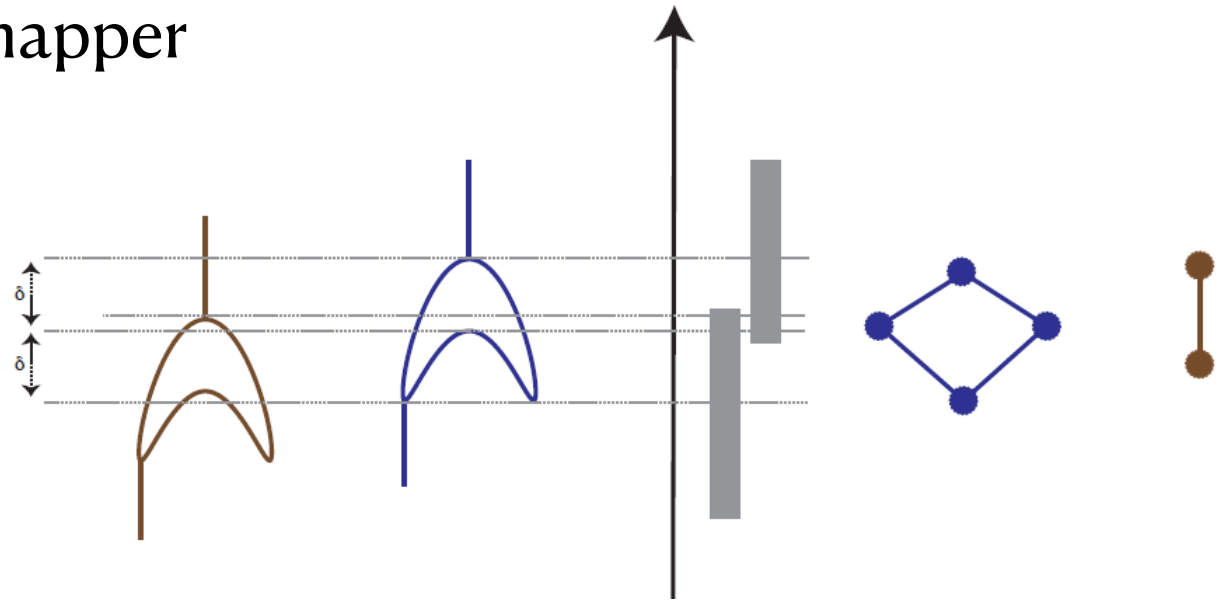
- Mapper is at a fixed scale
  - How to choose the scale? Why not look at all scales?
- Mapper is a structure, not an easy feature representation to compare
  - Can we obtain a persistence-like summary?

Motivation

- Mapper is at a fixed scale
  - How to choose the scale? Why not look at all scales?
- Mapper is a structure, not an easy feature representation to compare
  - Can we obtain a persistence-like summary?
- Un-stability of mapper

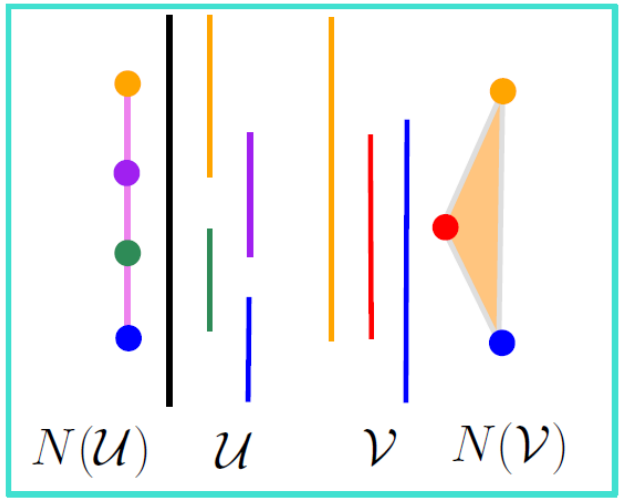
Motivation

- Mapper is at a fixed scale
  - How to choose the scale? Why not look at all scales?
- Mapper is a structure, not an easy feature representation to compare
  - Can we obtain a persistence-like summary?
- Un-stability of mapper

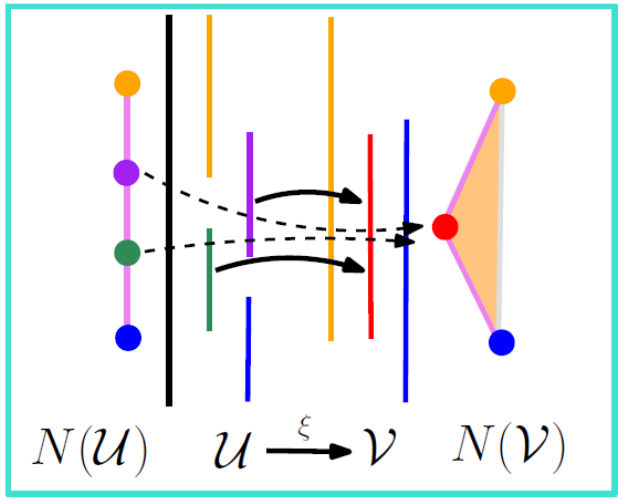


Maps between covers

Maps between covers



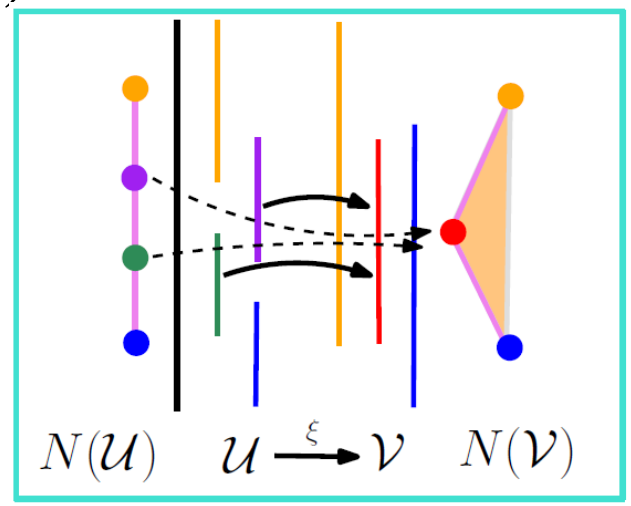
Maps between covers



Maps between covers

- Given two covers $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ and $\mathcal{V} = \{V_\beta\}_{\beta \in B}$ of the same space Y,
  - a cover map $\xi: \mathcal{U} \to \mathcal{V}$ is any set map $\xi: A \to B$ such that $U_\alpha \subseteq V_{\xi(\alpha)}$ for all $\alpha \in A$
  - Intuitively, a cover map can connect covers at different resolutions (B is coarser than A)
- A cover map $\xi: A \to B$ induces a simplicial map in nerves:

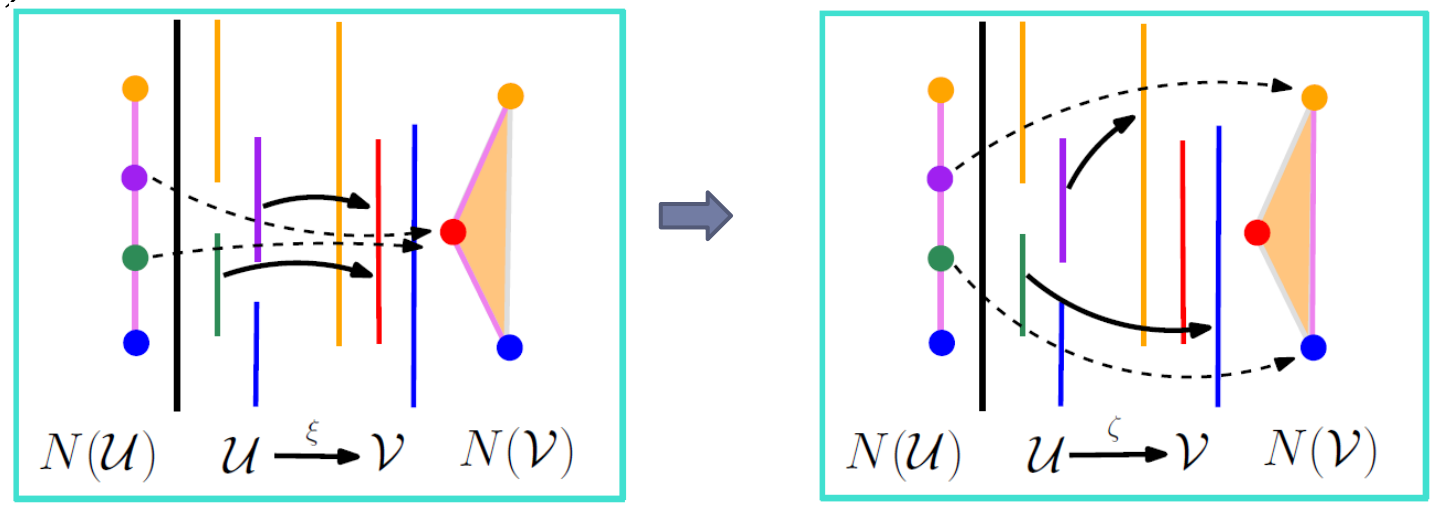
$$N(\xi): N(\mathcal{U}) \to N(\mathcal{V})$$



Maps between covers

- Given two covers $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ and $\mathcal{V} = \{V_\beta\}_{\beta \in B}$ of the same space Y,
  - a cover map $\xi: \mathcal{U} \to \mathcal{V}$ is any set map $\xi: A \to B$ such that $U_\alpha \subseteq V_{\xi(\alpha)}$ for all $\alpha \in A$
  - Intuitively, a cover map can connect covers at different resolutions (B is coarser than A)
- A cover map $\xi: A \to B$ induces a simplicial map in nerves:

$$N(\xi): N(\mathcal{U}) \to N(\mathcal{V})$$



Maps between covers

- Given two covers $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ and $\mathcal{V} = \{V_\beta\}_{\beta \in B}$ of the same space Y,
  - a cover map is any set map $\xi: A \to B$ such that $U_\alpha \subseteq V_{\xi(\alpha)}$ for all $\alpha \in A$
  - Intuitively, a cover map can connect covers at different resolutions (B is coarser than A)
- A cover map $\xi: A \to B$ induces a simplicial map in nerves:

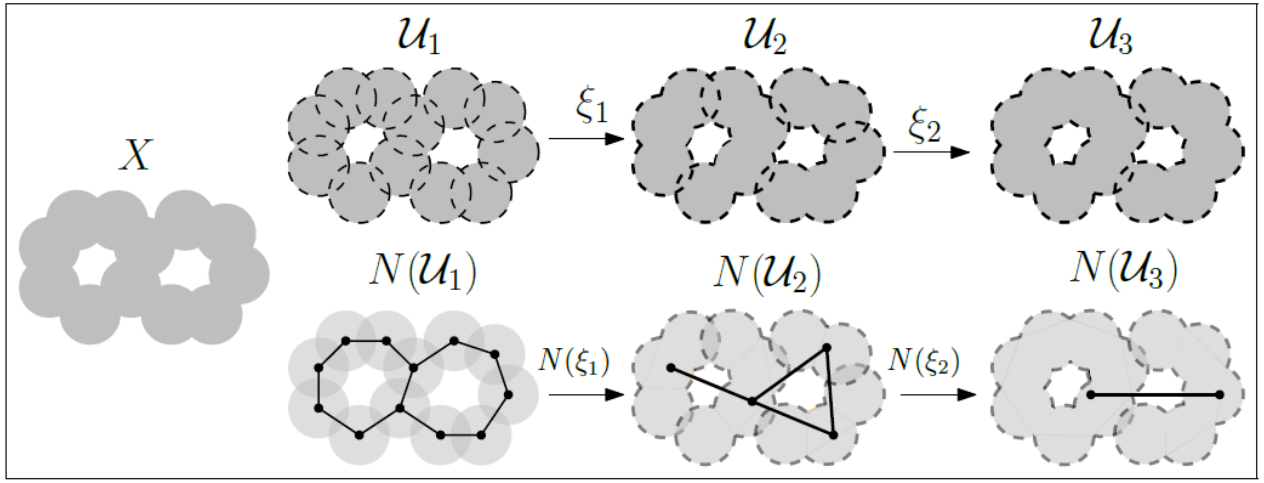
  $N(\xi): N(\mathcal{U}) \to N(\mathcal{V})$ if $\mathcal{U} \xrightarrow{\xi_1} \mathcal{V} \xrightarrow{\xi_2} \mathcal{W}$, then $N(\xi_2 \circ \xi_1) = N(\xi_2) \circ N(\xi_1)$

Maps between covers

- Given two covers $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ and $\mathcal{V} = \{V_\beta\}_{\beta \in B}$ of the same space Y,
  - a cover map is any set map $\xi: A \to B$ such that $U_\alpha \subseteq V_{\xi(\alpha)}$ for all $\alpha \in A$
  - Intuitively, a cover map can connect covers at different resolutions (B is coarser than A)
- A cover map $\xi: A \to B$ induces a simplicial map in nerves:

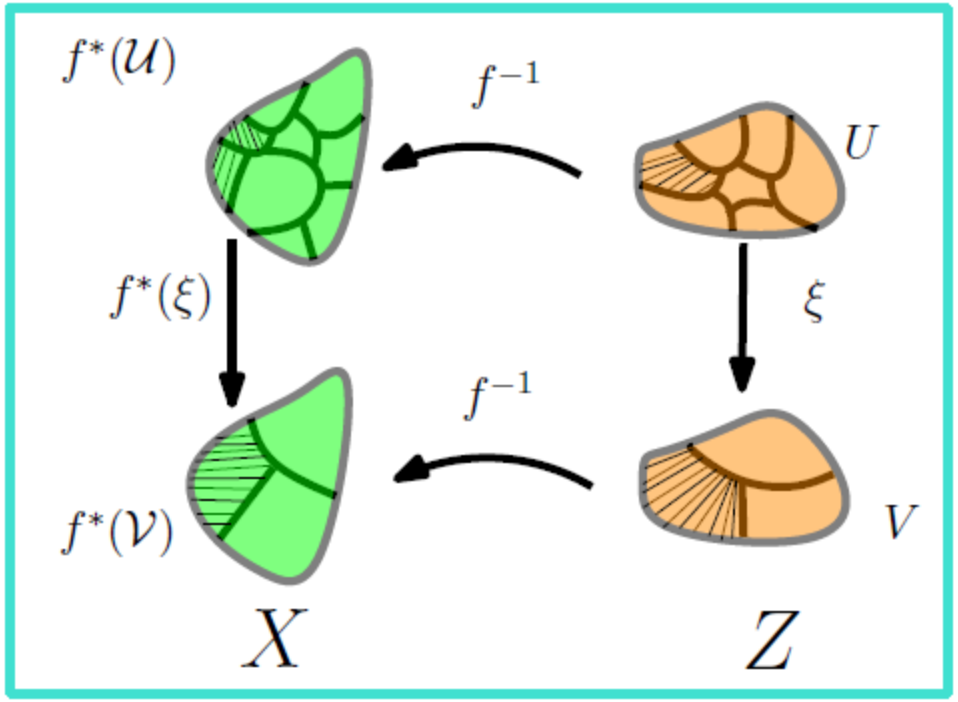
$N(\xi): N(\mathcal{U}) \to N(\mathcal{V})$

if $\mathcal{U} \xrightarrow{\xi_1} \mathcal{V} \xrightarrow{\xi_2} \mathcal{W}$, then $N(\xi_2 \circ \xi_1) = N(\xi_2) \circ N(\xi_1)$

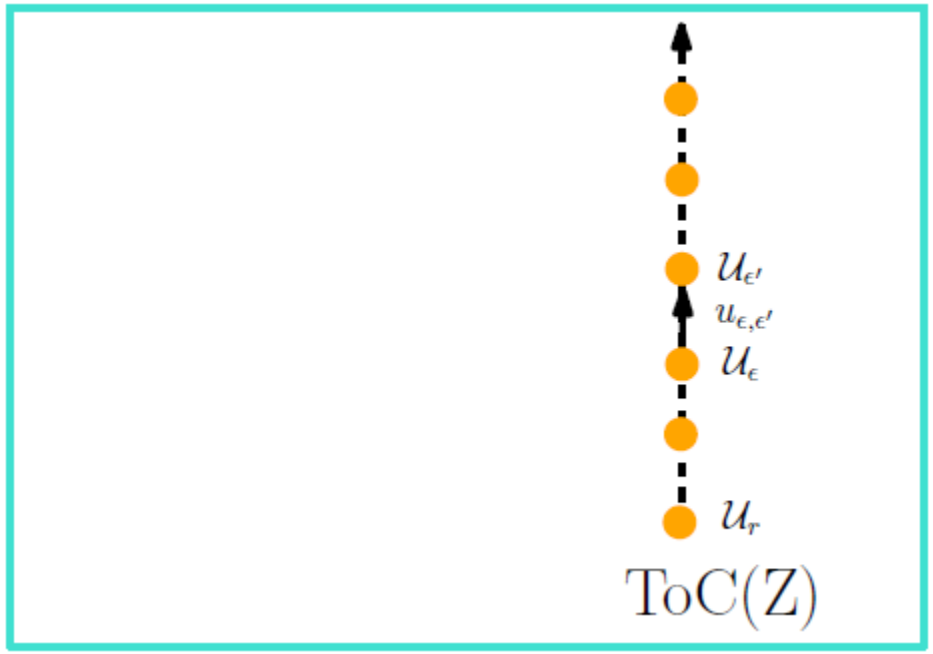


Pullback covers

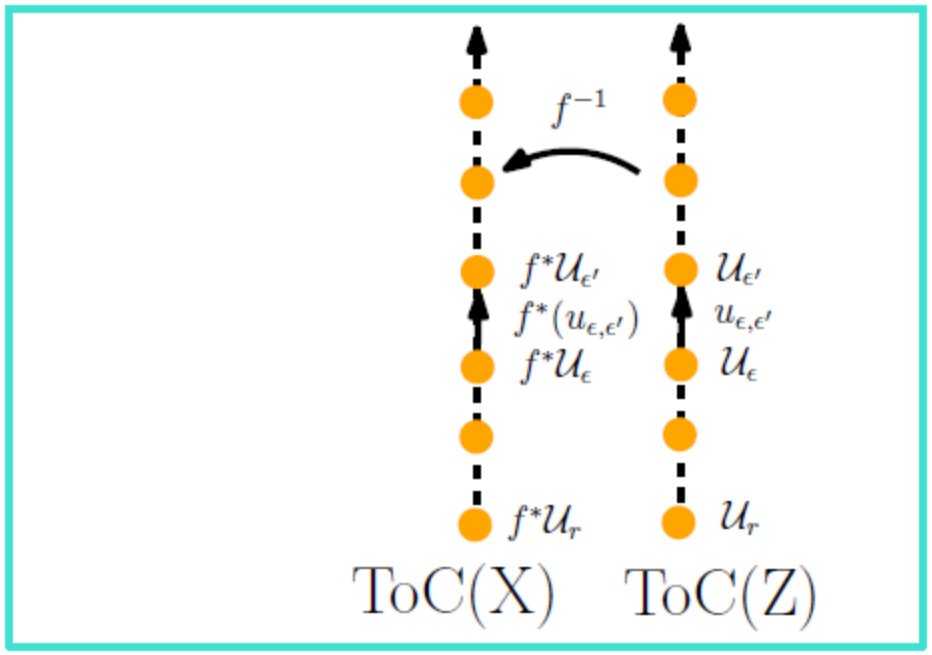
- $f: X \to Z$ continuous, and well-behaved
- A map $\xi: \mathcal{U} \to \mathcal{V}$ between covers of Z
- $\Rightarrow$ a cover map for pullback covers of X
  - $f^*(\xi): f^*(\mathcal{U}) \to f^*(\mathcal{V})$



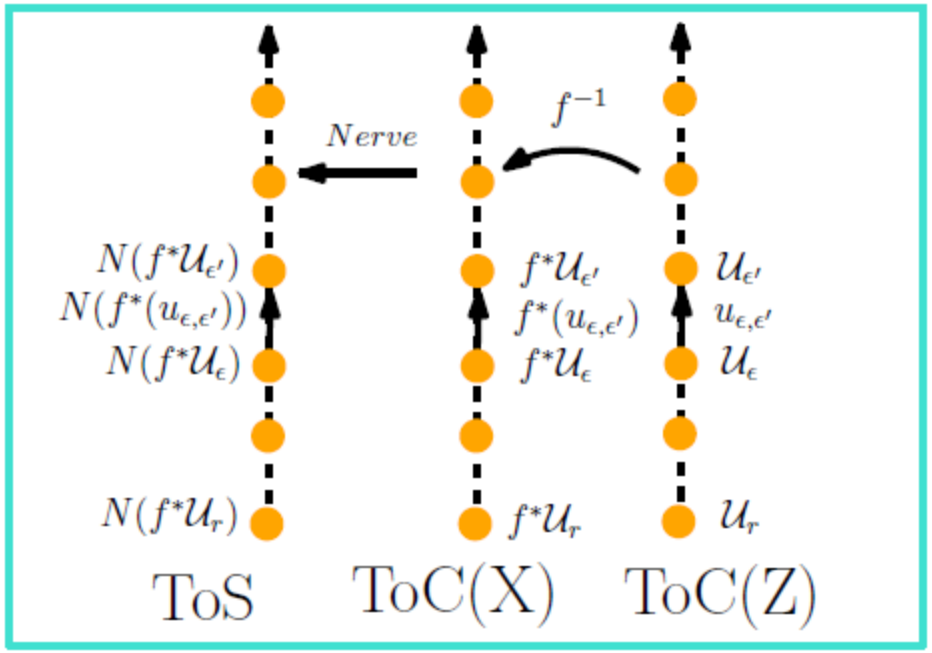
Multiscale Mapper



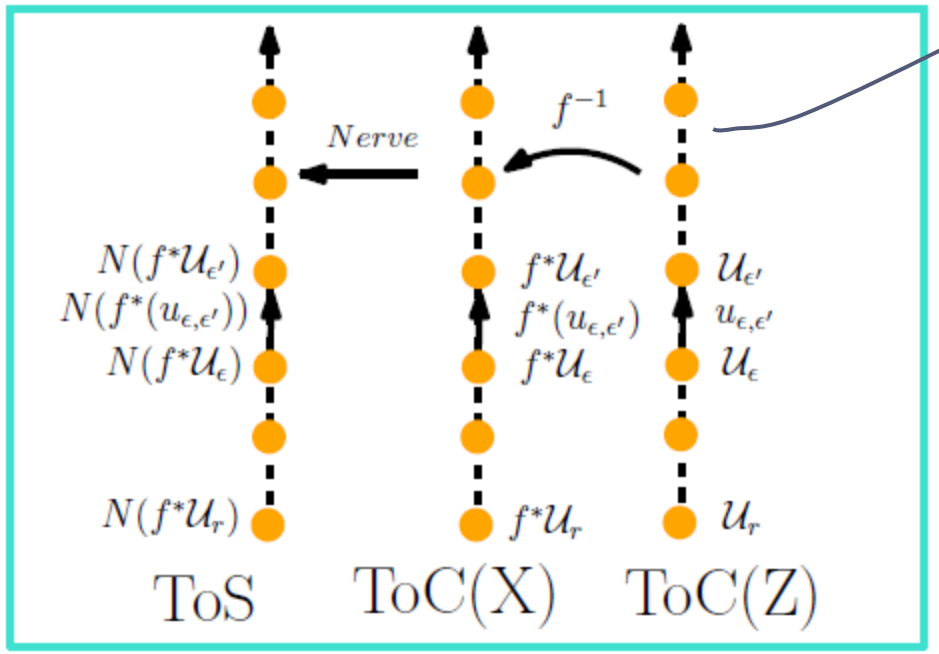
Multiscale Mapper



Multiscale Mapper



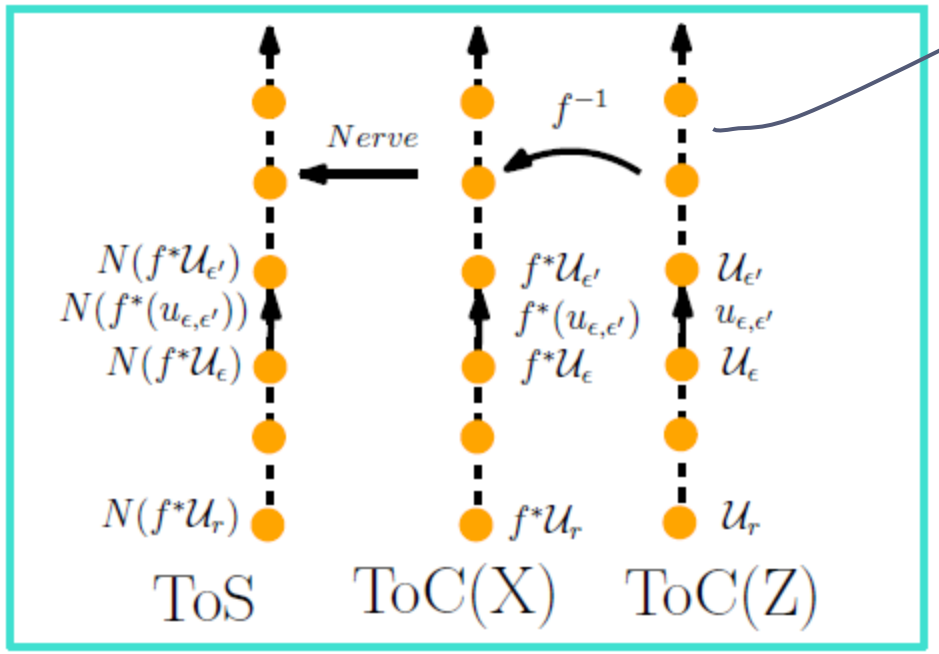
Multiscale Mapper



a tower of covers

$\mathfrak{A} = \{\mathcal{U}_\varepsilon\}_\varepsilon$

Multiscale Mapper



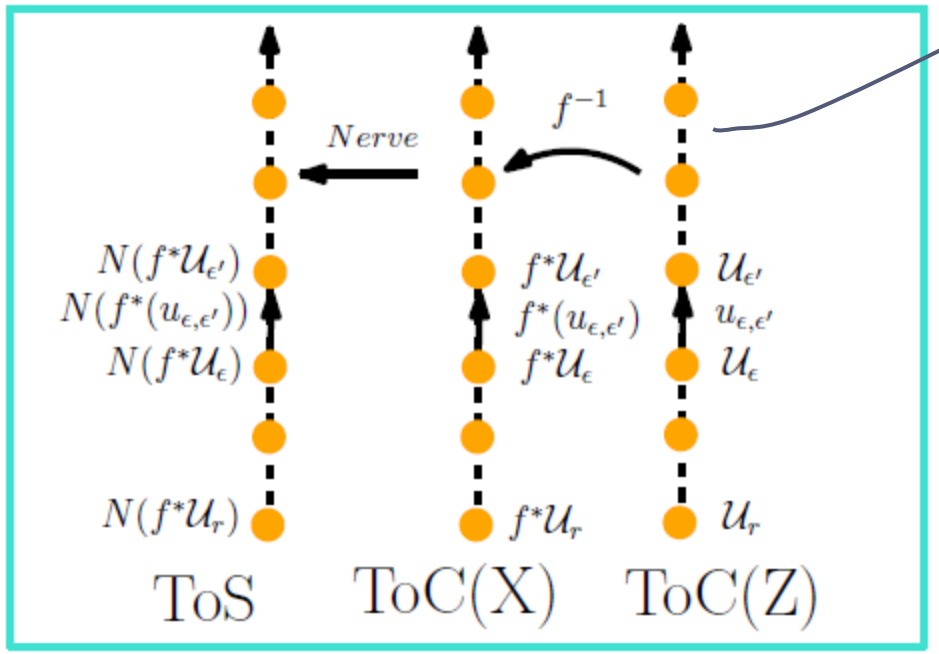
a tower of covers

$$\mathfrak{A} = \{\mathcal{U}_\varepsilon\}_\varepsilon$$

Multiscale Mapper:

$$MM(\mathfrak{A},\ f) := N(f^*(\mathfrak{A}))$$

Multiscale Mapper



a tower of covers
$\mathfrak{A} = \{\mathcal{U}_\varepsilon\}_\varepsilon$

Multiscale Mapper:

$$MM(\mathfrak{A},\ f) := N(f^*(\mathfrak{A}))$$

$\mathrm{D_k MM}(\mathfrak{U}, f)$= persistence diagram of:

$$\mathrm{H}_k\left(N(f^*(\mathcal{U}_{\varepsilon_1}))\right) \to \mathrm{H}_k\left(N(f^*(\mathcal{U}_{\varepsilon_2}))\right) \to \cdots \to \mathrm{H}_k\left(N(f^*(\mathcal{U}_{\varepsilon_n}))\right)$$

Remarks

- Intuitively, persistent homology of multiscale mapper captures important features of X through the lens of the map f and a tower of covers (across resolutions) of co-domain Z

Remarks

- Intuitively, persistent homology of multiscale mapper captures important features of X through the lens of the map f and a tower of covers (across resolutions) of co-domain Z

- For H_1, turns out that we cannot create new features, only kill existing features at the highest resolution gradually
  - time a feature is killed (ie., its persistence) corresponds to its importance

Remarks

- Intuitively, persistent homology of multiscale mapper captures important features of X through the lens of the map f and a tower of covers (across resolutions) of co-domain Z
- For H_1, turns out that we cannot create new features, only kill existing features at the highest resolution gradually
  - time a feature is killed (ie., its persistence) corresponds to its importance
- The multiscale mapper and its PH summaries have several stability properties w.r.t. perturbation of functions and tower of covers.

Remarks

- Intuitively, persistent homology of multiscale mapper captures important features of X through the lens of the map f and a tower of covers (across resolutions) of co-domain Z
- For H_1, turns out that we cannot create new features, only kill existing features at the highest resolution gradually
  - time a feature is killed (ie., its persistence) corresponds to its importance
- The multiscale mapper and its PH summaries have several stability properties w.r.t. perturbation of functions and tower of covers.
- Finally, there is an interleaving distance between multiscale mappers, much like the one for persistent homology.

FIN