

# Advanced Data Visualization

CS 6965

Fall 2019

Prof. Bei Wang Phillips

University of Utah



Lecture 04

# Beyond PCA & t-SNE: Visual Interactions with DR



HD

# Visual Interactions with DR

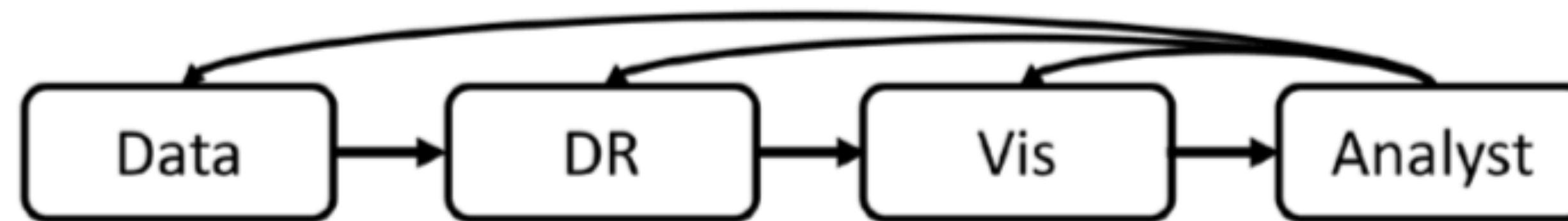


Fig. 1: A basic DR pipeline maps data to a DR algorithm. The results are visualized and presented to the analyst. Interaction feeds back to the pipeline components.

# Visual Interactions with DR

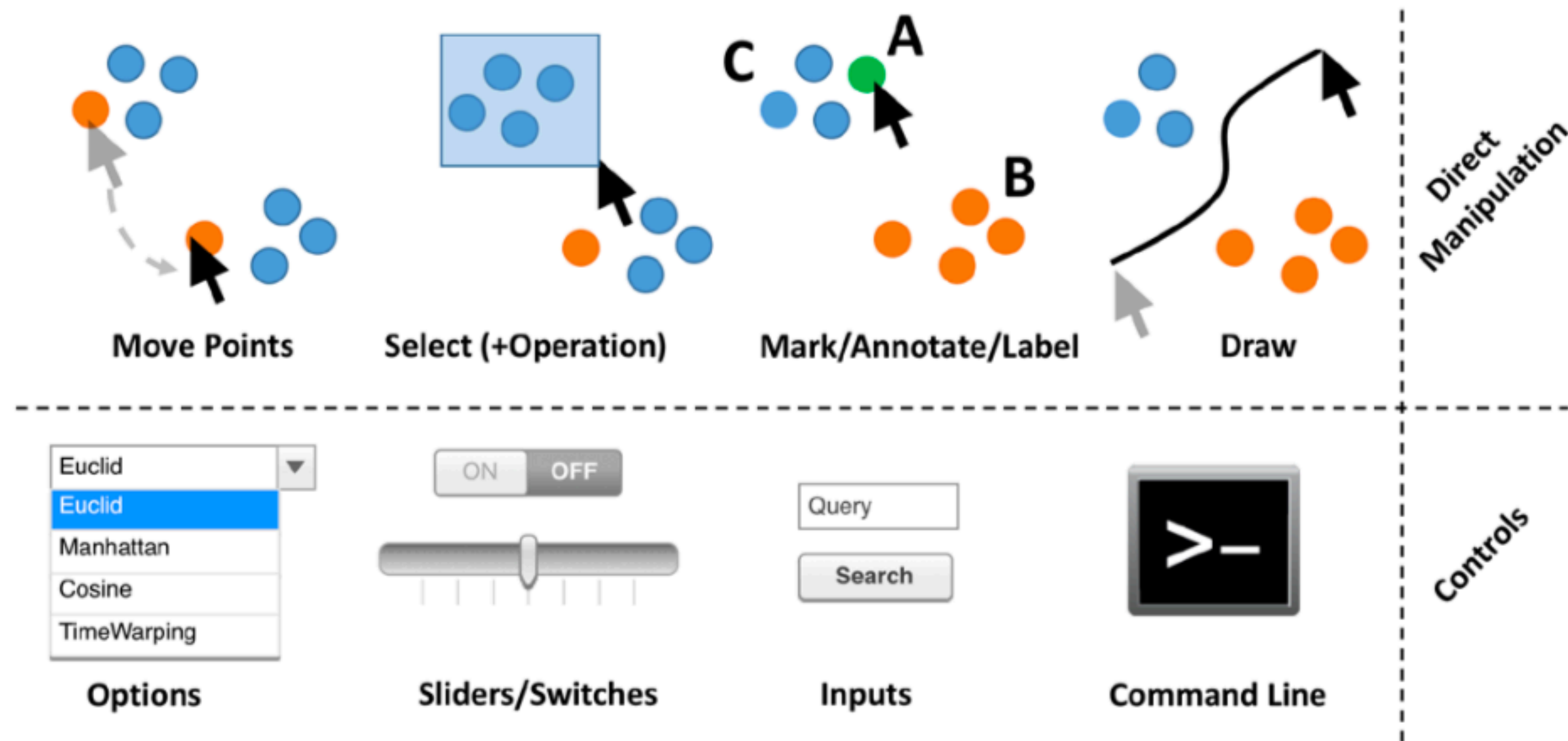


Fig. 5: Different interaction paradigms: Typical *Direct Manipulation* interactions are shown in the upper half. On the bottom, control elements are shown. DR-Interfaces are usually composed of both.

# Visual Interactions with DR

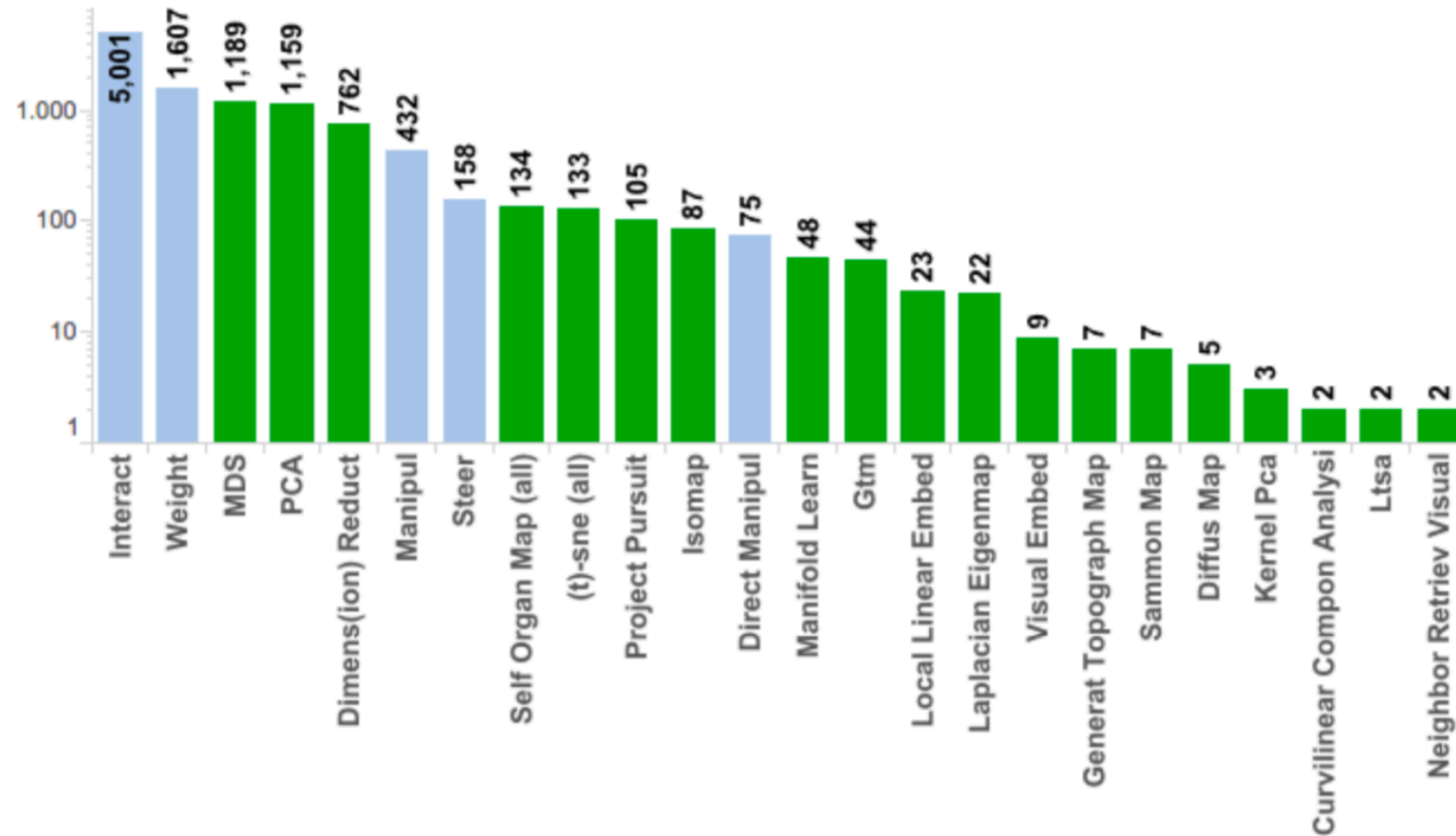


Fig. 3: The top keyword occurrences in the automatically identified papers shown in a log-scale histogram. DR keywords are colored in green and interaction keywords are colored in light blue.

# 7 Guiding Scenarios for DR Interaction

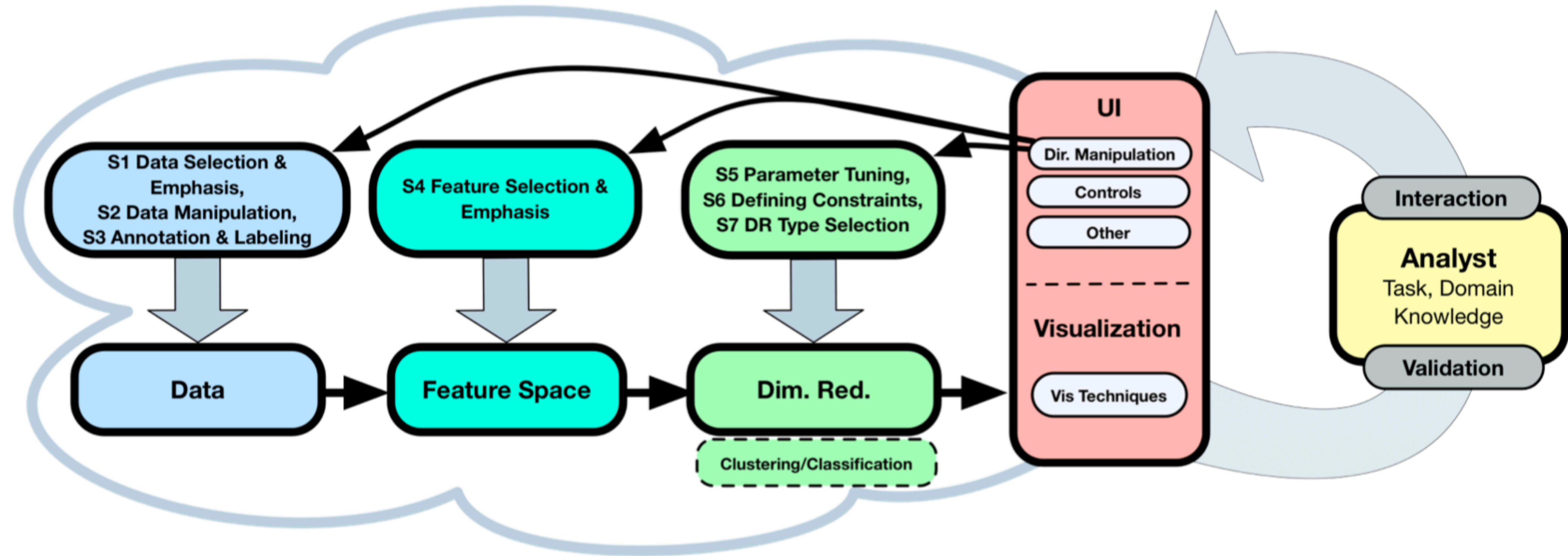
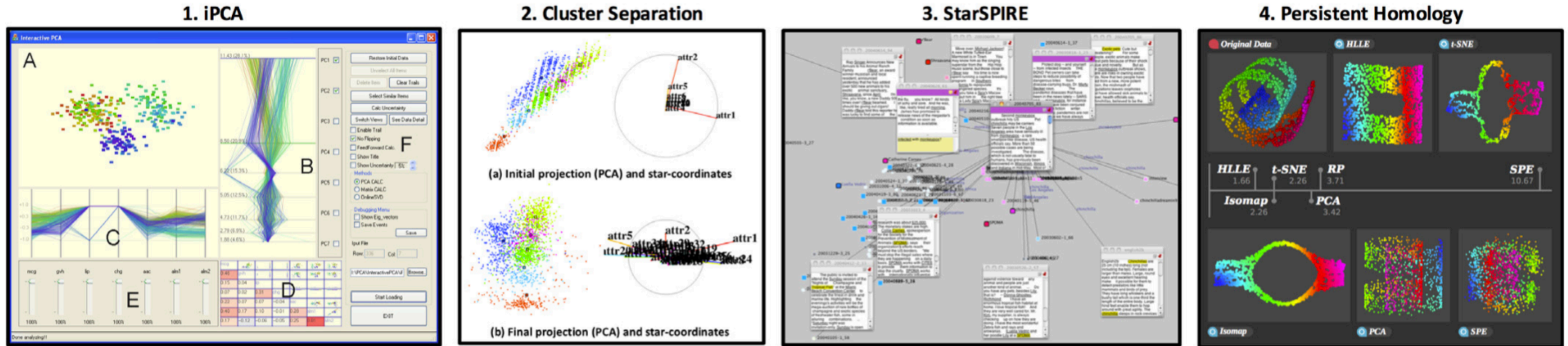
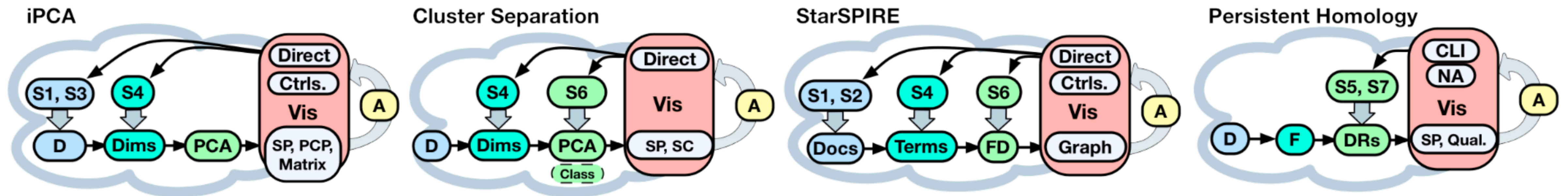


Fig. 6: Proposed “human in the loop” process model for interactive DR. The analyst can iteratively refine the analysis by interacting with the DR pipeline. The visualization interface serves as a “lens” that interactively mediates between the DR pipeline and the analyst, presenting DR results or updates and accepting feedback.

# Visual Interactions with DR



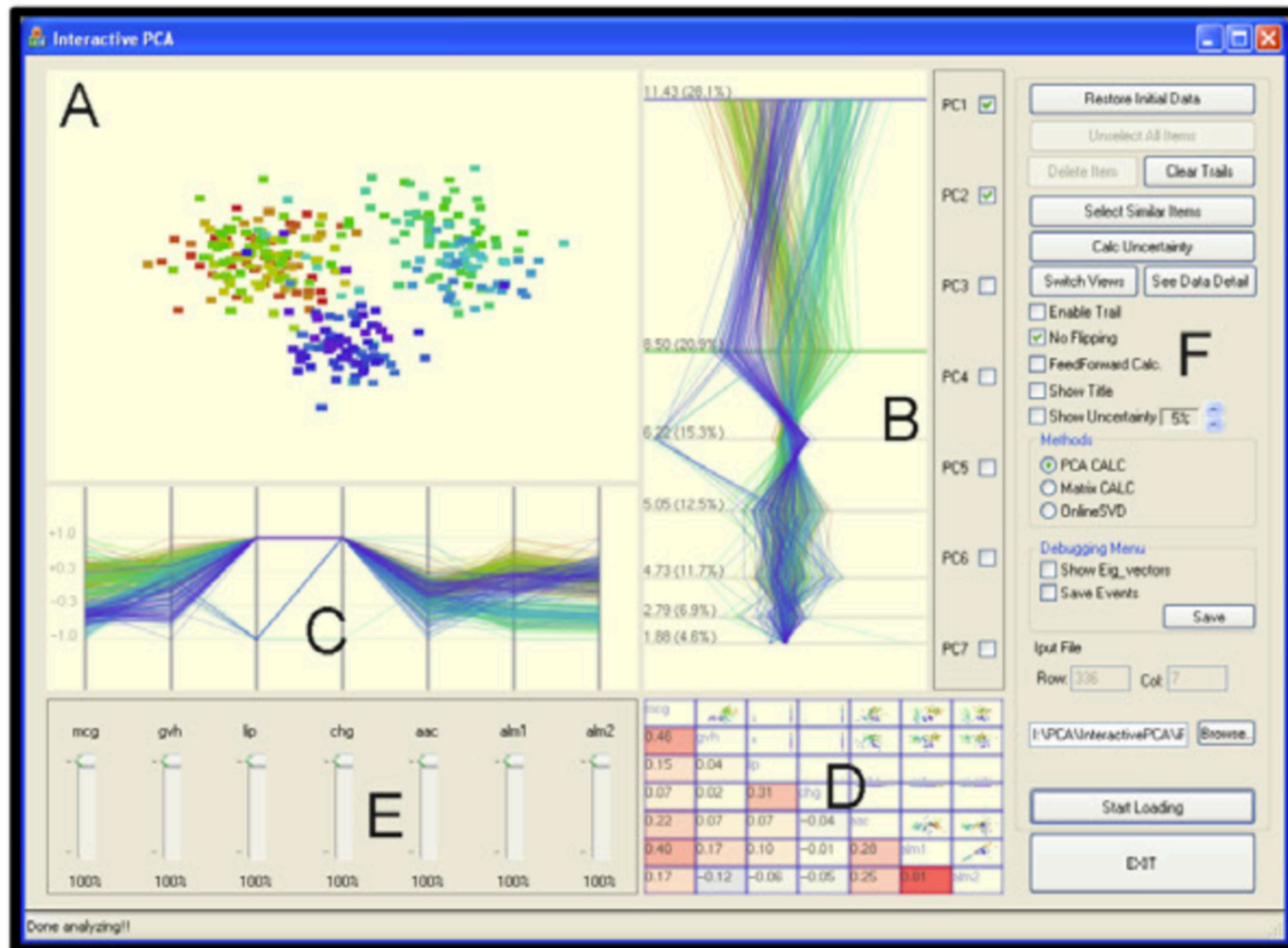
(a) Images for each example.



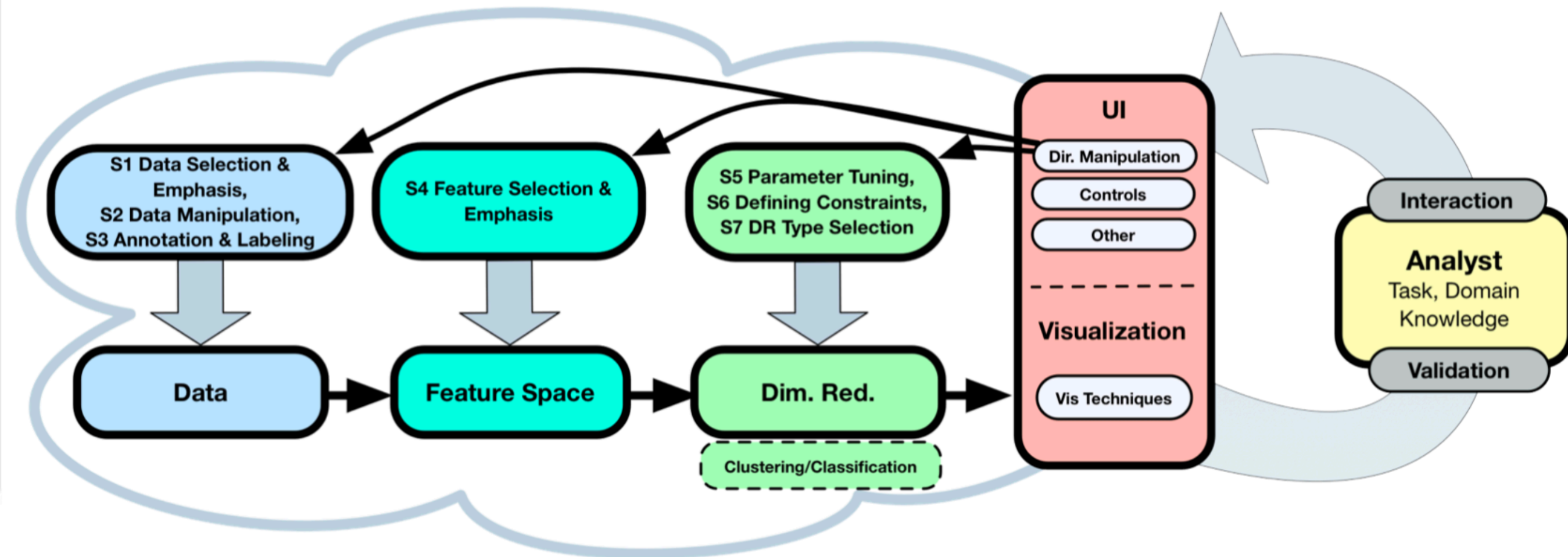
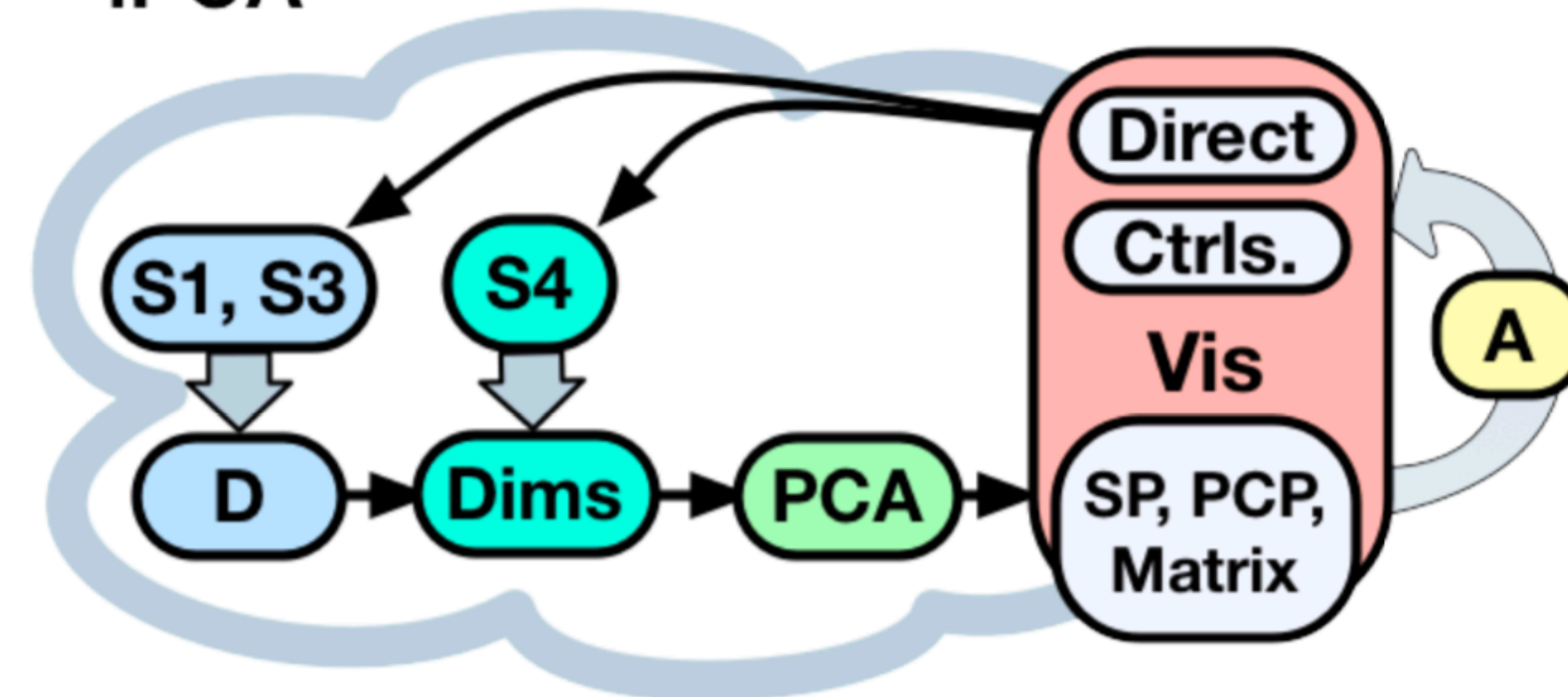
(b) Interactive DR process model instances for each example.

# iPCA and DR Interaction

## 1. iPCA



## iPCA





# Some more maths on PCA and t-SNE

- See white board for some more math

## Automatic Selection of t-SNE Perplexity

**Yanshuai Cao**    `YANSHUAI.CAO@RBC.COM` and **Luyu Wang**    `LUYU.WANG@RBC.COM`

# t-SNE Revisited

$$\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \quad d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|.$$

$$p_{j|i} = \frac{\exp(-d(\mathbf{x}_i, \mathbf{x}_j)^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-d(\mathbf{x}_i, \mathbf{x}_k)^2 / 2\sigma_i^2)}, \quad p_{i|i} = 0 \quad (1)$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}. \quad (2)$$

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}, \quad q_{ii} = 0. \quad (3)$$

$$C(\mathcal{E}) = KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}. \quad (4)$$

# SNE/t-SNE Revisited

$\sigma_i$  induces a probability distribution,  $P_i$ , over all of the other datapoints. This distribution has an entropy which increases as  $\sigma_i$  increases. SNE performs a binary search for the value of  $\sigma_i$  that produces a  $P_i$  with a fixed perplexity that is specified by the user.<sup>3</sup> The perplexity is defined as

$$\text{Perp}(P_i) = 2^{H(P_i)},$$

where  $H(P_i)$  is the Shannon entropy of  $P_i$  measured in bits

$$H(P_i) = - \sum_j p_{j|i} \log_2 p_{j|i}.$$

The perplexity can be interpreted as a smooth measure of the effective number of neighbors. The performance of SNE is fairly robust to changes in the perplexity, and typical values are between 5 and 50.

# SNE/t-SNE Revisited

**perplexity : *float, optional (default: 30)***

The perplexity is related to the number of nearest neighbors that is used in other manifold learning algorithms. Larger datasets usually require a larger perplexity. Consider selecting a value between 5 and 50. Different values can result in significantly different results.

<https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html#sklearn.manifold.TSNE>

# Perplexity auto-tuning

This suggests that trading off between the final KL divergence and  $Perp$  could potentially lead to good embeddings. Based on this intuition, we design the following criteria:

$$S(Perp) = 2\text{KL}(P||Q) + \log(n)\frac{Perp}{n} \quad (4)$$

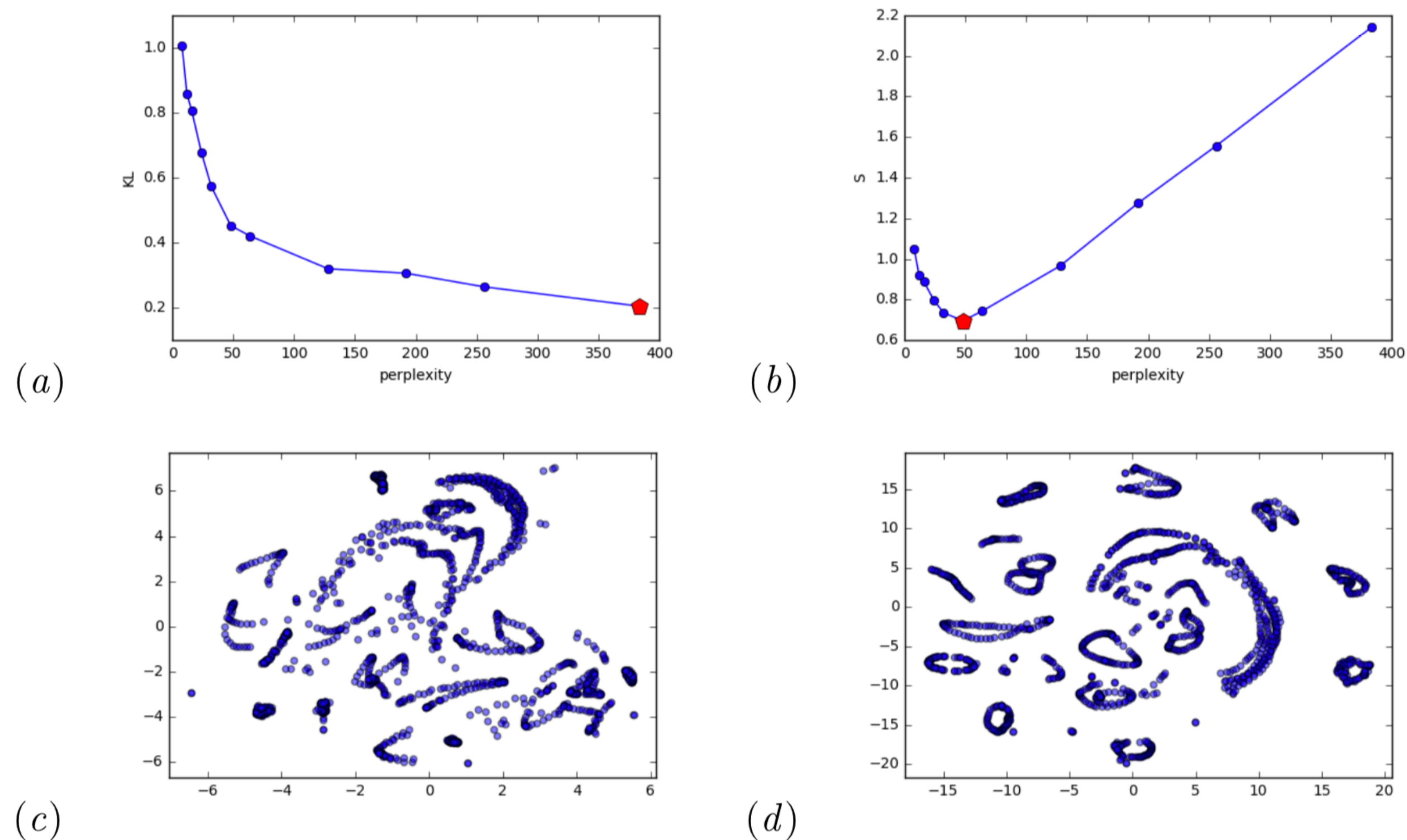


Figure 1: KL divergence (1(a)) and  $S$  (1(b)) as function of  $Perp$  on Coil20 dataset, along with t-SNE maps (1(c) and 1(d)) at their respective argmin locations marked by red markers.

# Further reading

## Ten quick tips for effective dimensionality reduction

Lan Huong Nguyen, Susan Holmes 

Published: June 20, 2019 • <https://doi.org/10.1371/journal.pcbi.1006907>

# Mapper, Clustering & beyond

HD

# The Mapper Algorithm: History and Overview

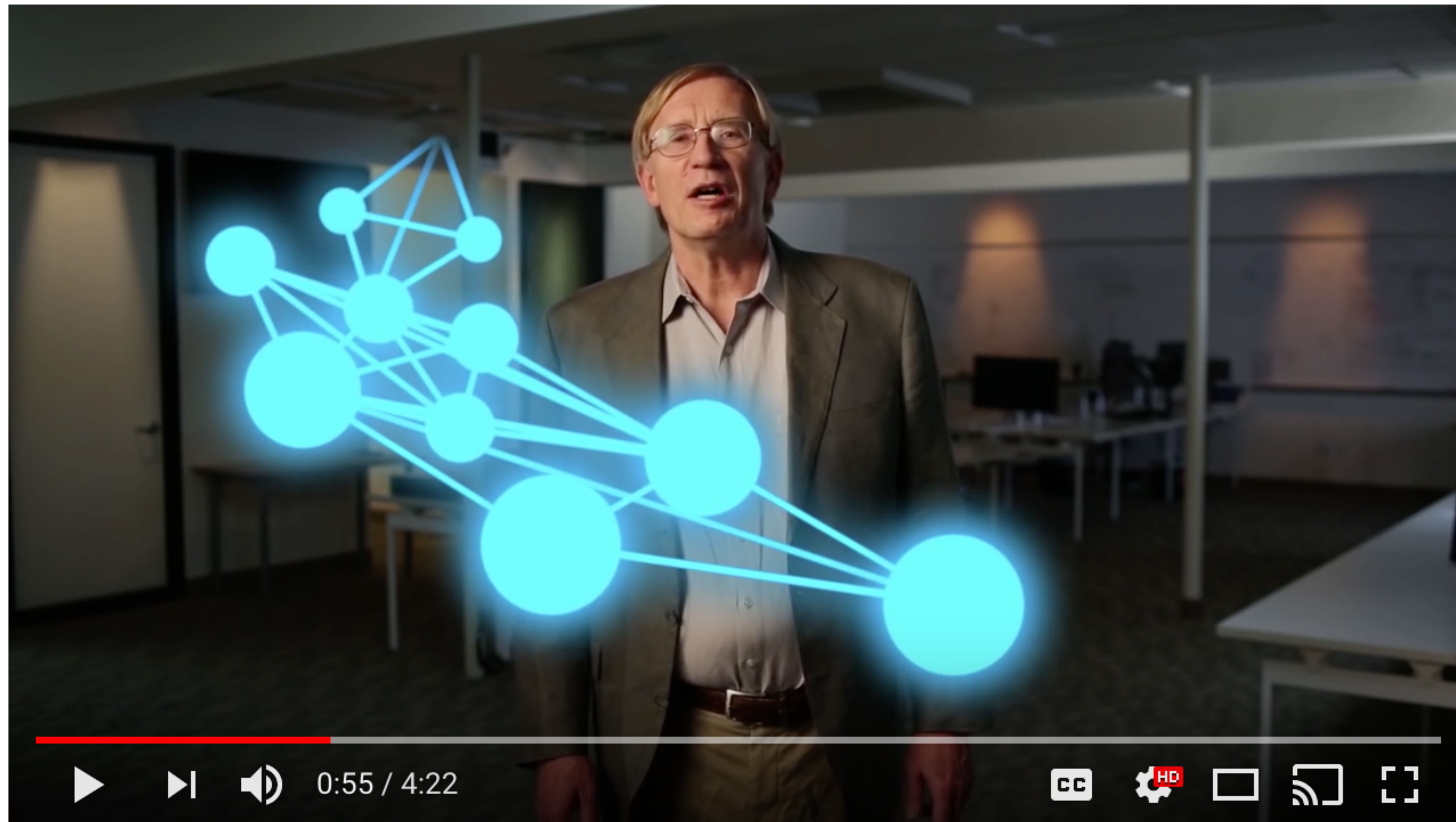
A tool for high-dimensional data analysis and visualization



# History of mapper algorithm

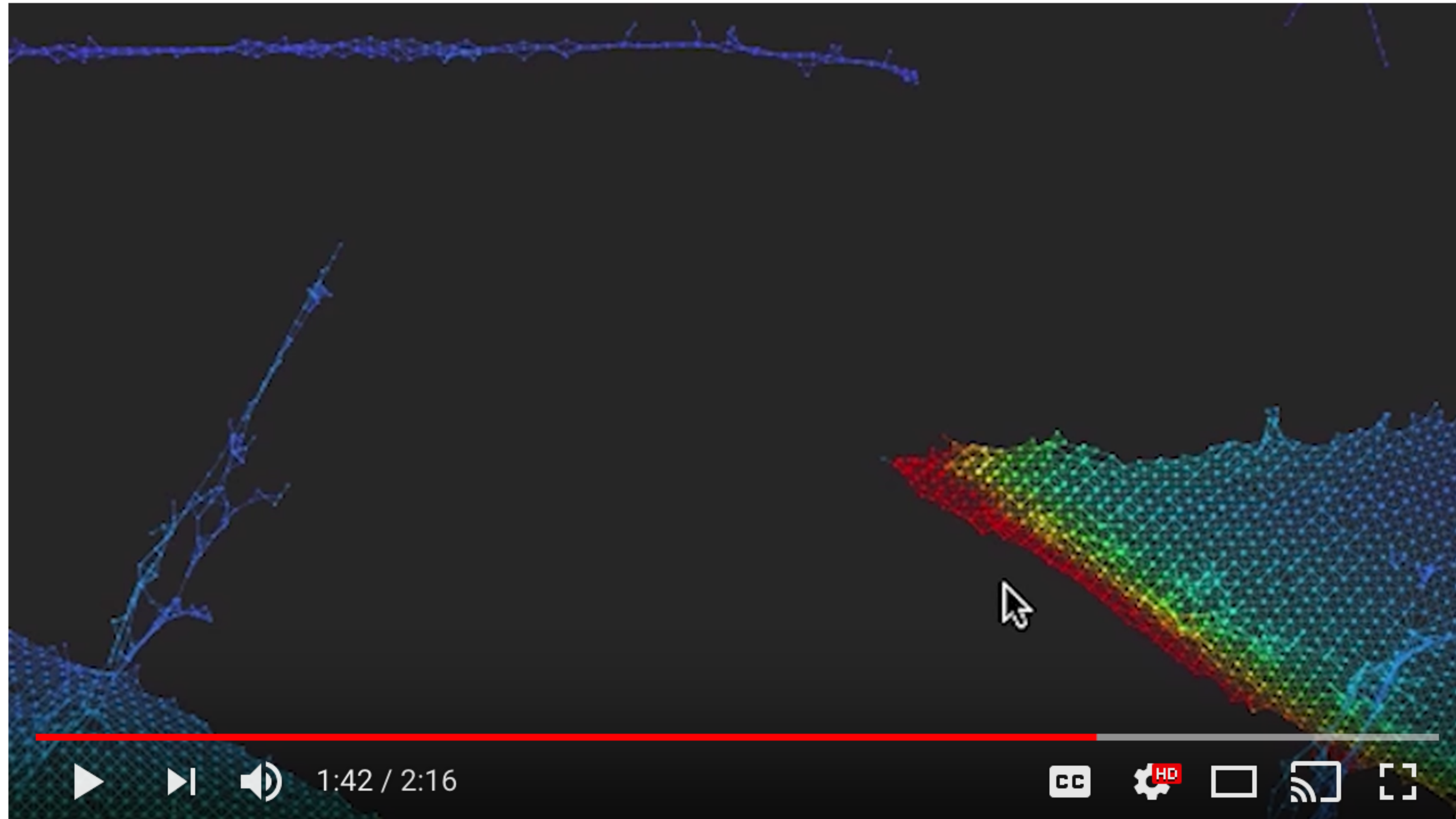
- At the core of at least two data analysis startups:
  - Ayasdi: topological data analysis, machine learning and visualization
    - <https://www.ayasdi.com/>
  - Alpine Data: (topological) data analysis at scale,
    - <http://alpinedata.com/>

# Ayasdi



<https://www.youtube.com/watch?v=XfWibrh6stw>

# Ayasdi: Fraud detection



<https://www.youtube.com/watch?v=L8o4an5nh4E>

# Ayasdi: Patient Stratification




<https://www.youtube.com/watch?v=FmfIJ3-Uual>

# Alpine Data

Enterprise Scale Topological Data Analysis Using Spark

## What we'll talk about

- What's TDA and why should you care
- Deep dive into Mapper and bottlenecks
- **Betti Mapper** - scaling Mapper to the enterprise



SPARK SUMMIT 2016

SPARK SUMMIT 2016  
MORE VIDEOS

4:14 / 32:29

YouTube

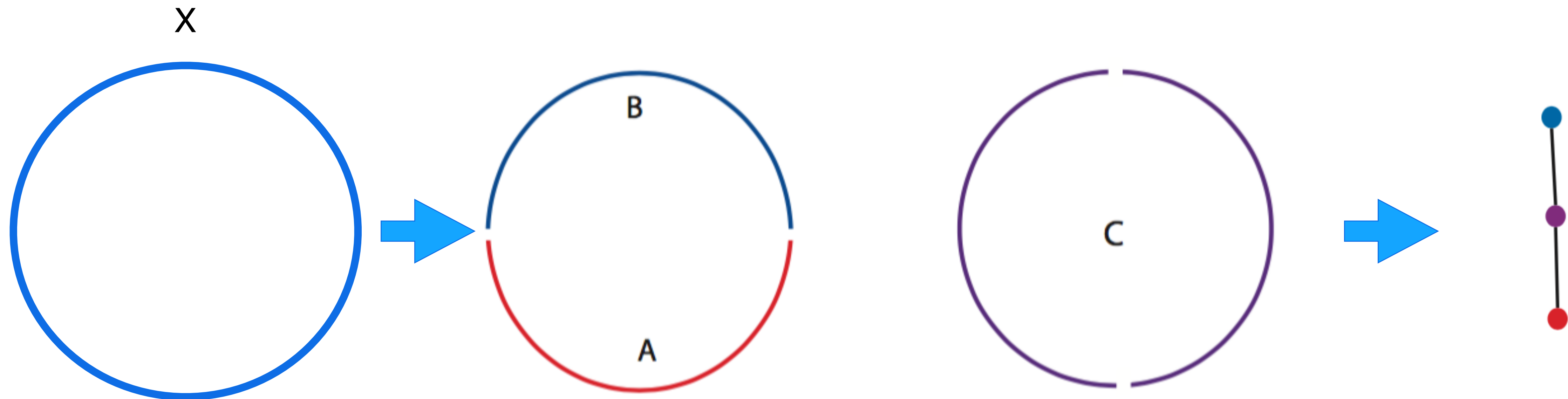
The image shows a video player interface. The main content area displays a slide with a title, a subtitle, and a bulleted list. To the right of the slide is a large, colorful circular graphic with the 'Spark Summit' logo in the center. Below the slide is a video player with a progress bar, a 'MORE VIDEOS' button, and a small video thumbnail showing a speaker. The video player controls include play, volume, and a time indicator showing 4:14 / 32:29. The YouTube logo and other interface icons are visible at the bottom right of the player.

<https://databricks.com/session/enterprise-scale-topological-data-analysis-using-spark>

# Mapper Algorithm and Visualization

- A qualitative understanding of high-dim point cloud data through direct visualization
- Combining DR with graph visualization
- Desirable properties of visualization for high-dimensional data
  - Insensitive to metric (approximation to similarity): robust to small changes to the metric
  - Understanding sensitivity to parameter changes: provide useful summary of behavior under all choices of parameters, exploratory
  - Multi-scale representation: at various levels of resolution, comparison
- “Features which are seen at multiple scales will be viewed as more likely to be actual features as opposed to more transient features which could be viewed as artifacts of the imaging method.”

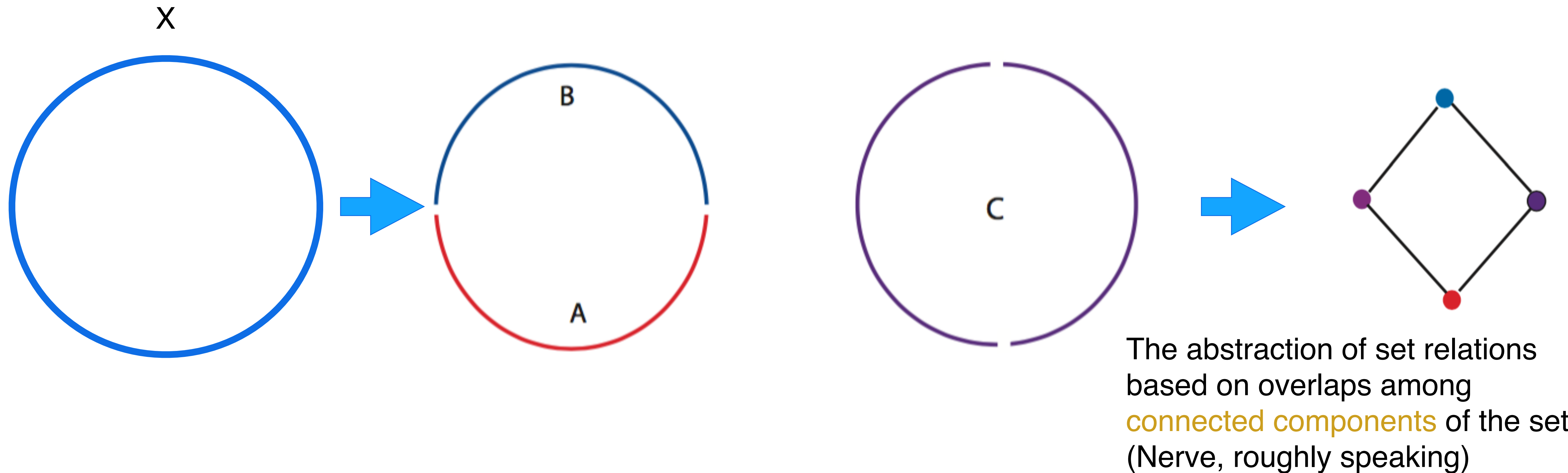
# Covering a circle by sets



The abstraction of set relations based on overlaps of **sets**

**Example.** Let  $X$  denote the unit circle, and let a covering  $\mathcal{U}$  of  $X$  be given by the three sets  $A = \{(x, y) \mid y < 0\}$ ,  $B = \{(x, y) \mid y > 0\}$ , and  $C = \{(x, y) \mid y \neq \pm 1\}$ .

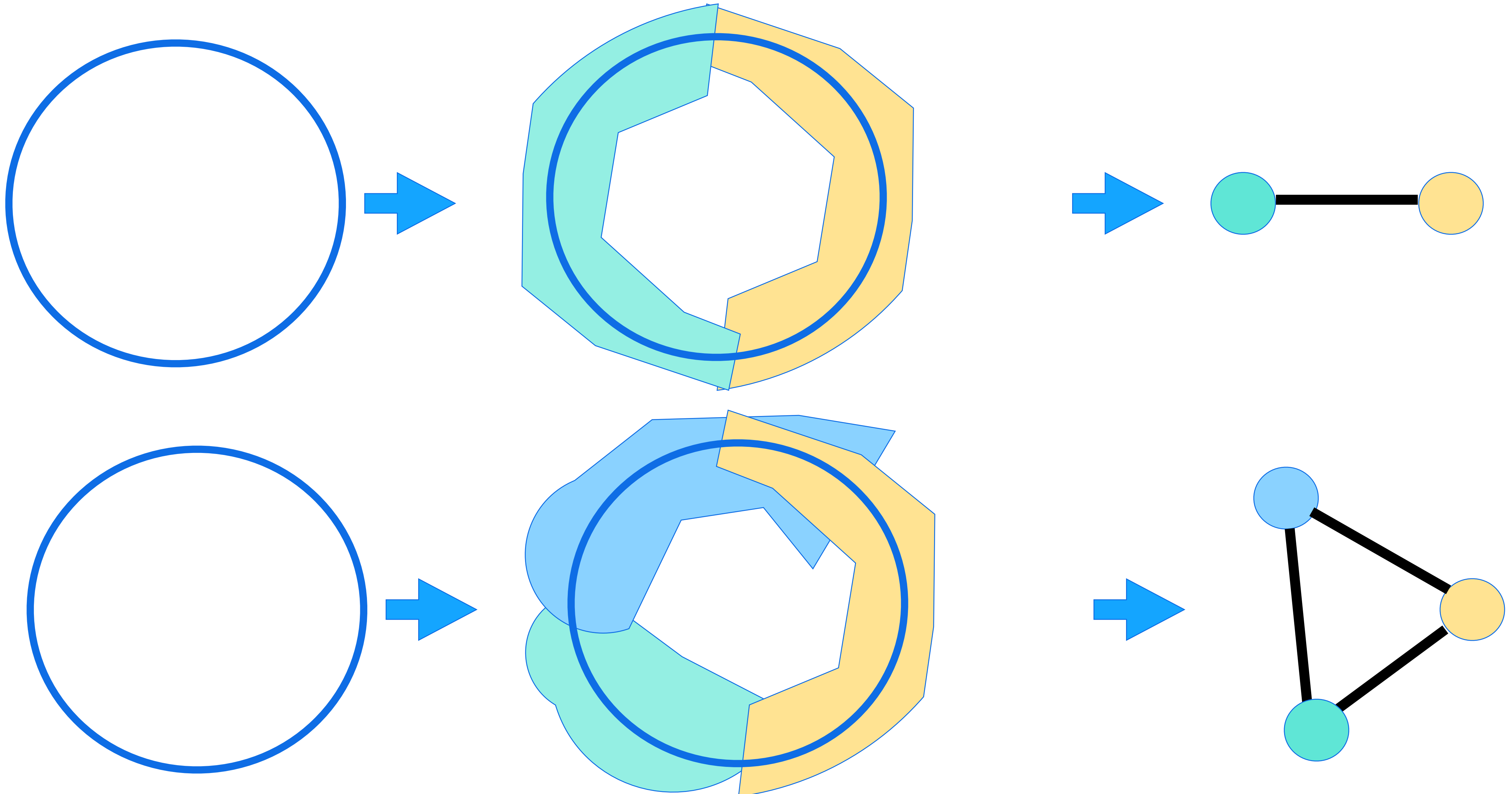
# Covering a circle by sets



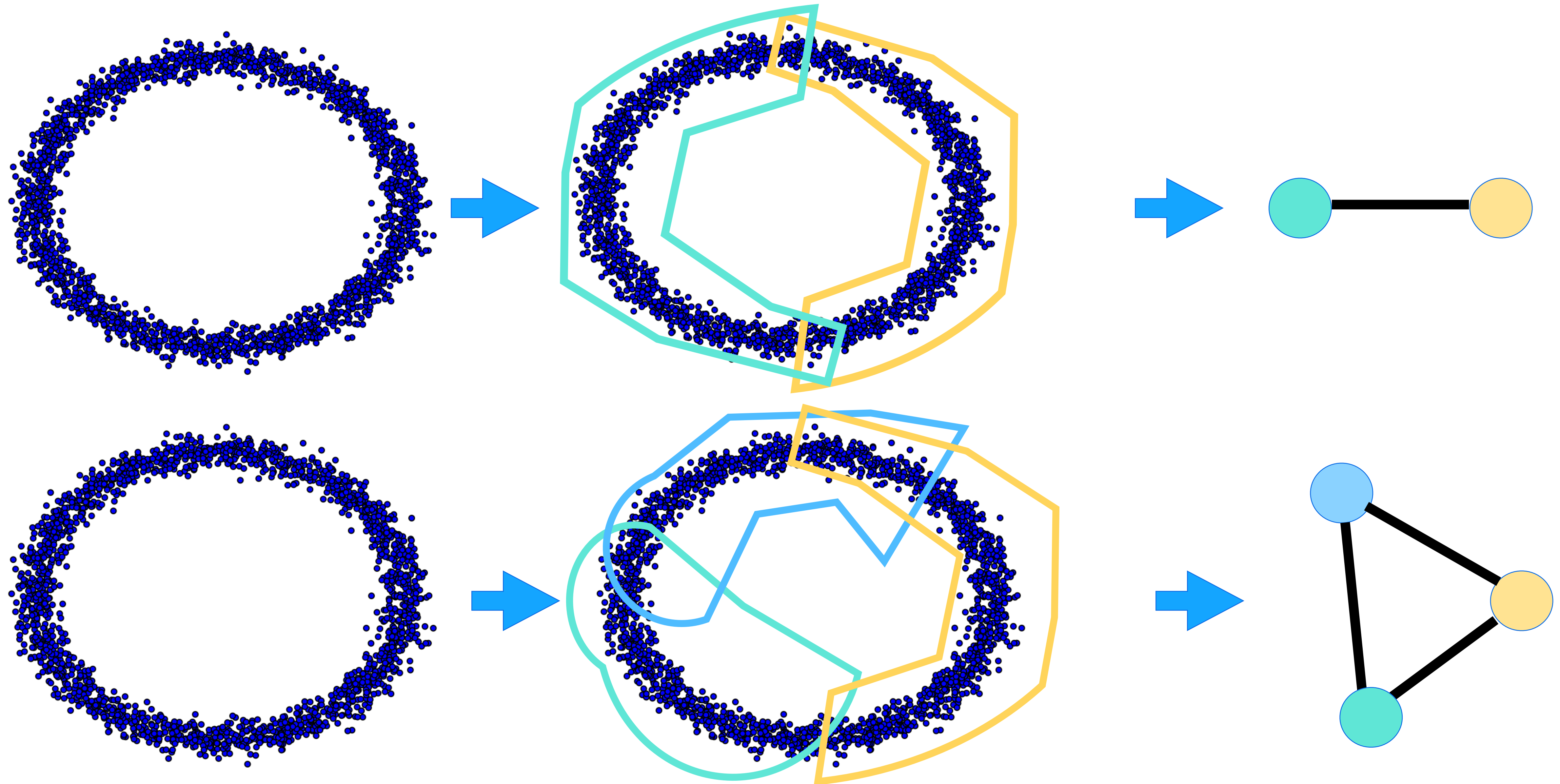
**Example.** Let  $X$  denote the unit circle, and let a covering  $\mathcal{U}$  of  $X$  be given by the three sets  $A = \{(x, y) \mid y < 0\}$ ,  $B = \{(x, y) \mid y > 0\}$ , and  $C = \{(x, y) \mid y \neq \pm 1\}$ .



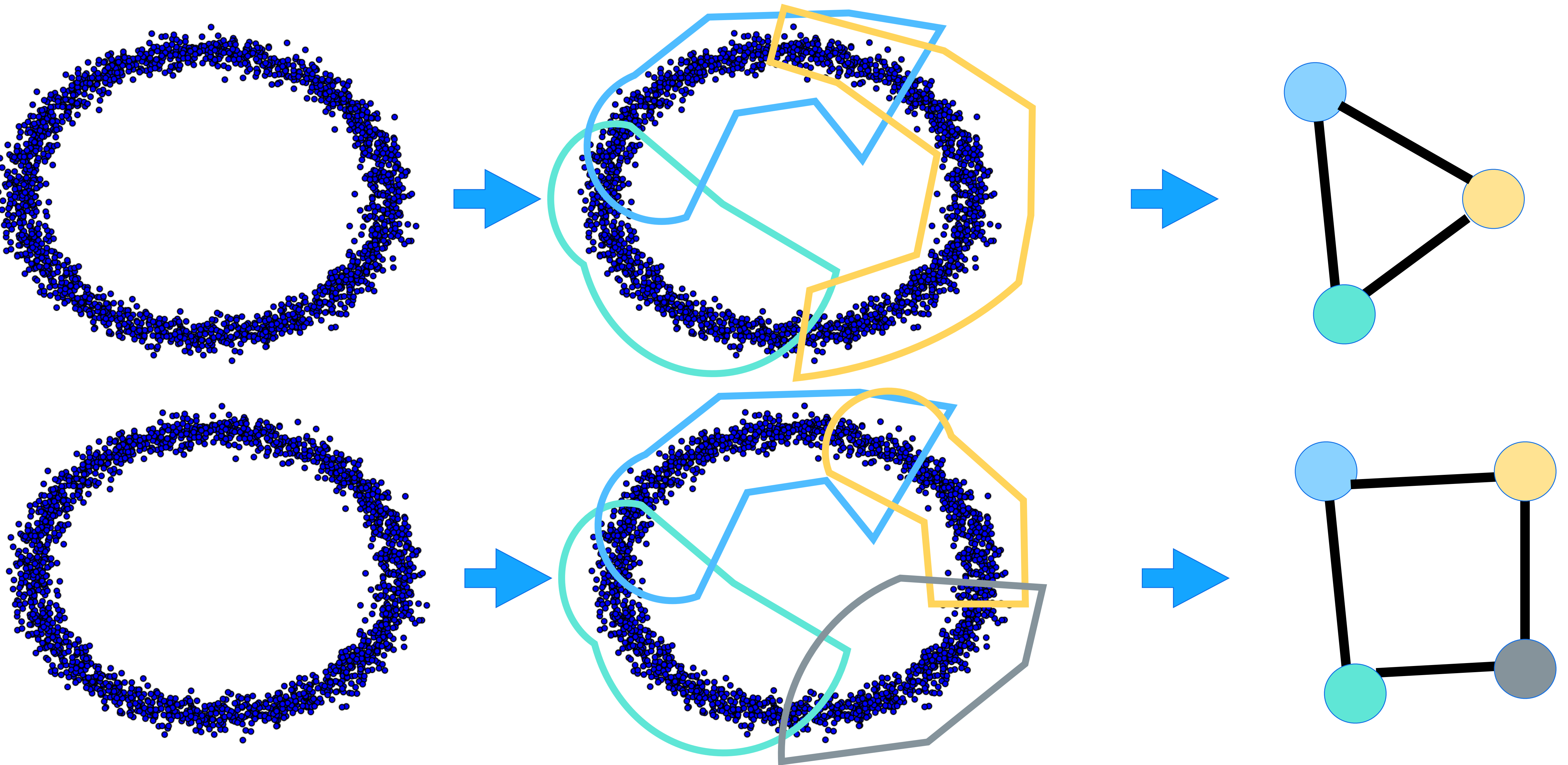
# Covering a circle by sets



# Point cloud data: soft clustering



# Change of scale



# Covering of a point cloud

- Given point cloud data and a covering...
- Taking the **nerve** of the covering can sometimes capture the shape of the data at the **right** scale(s)

# Mapper Algorithm at a Glance

Qualitative analysis, simplification and visualization of high-dimensional **data sets** and **functions on these data sets**:

- **Data summarization/skeletonization**: Extracting simple descriptions of high dimensional data sets in the form of simplicial complexes or graphs
- **Function-induced clustering**: partial clustering of the data guided by a set of functions defined on the data.
- **Flexibility**: any clustering algorithm may be used with Mapper.
- **Exploratory and multi-scale**: explore parameters at all scales if possible.

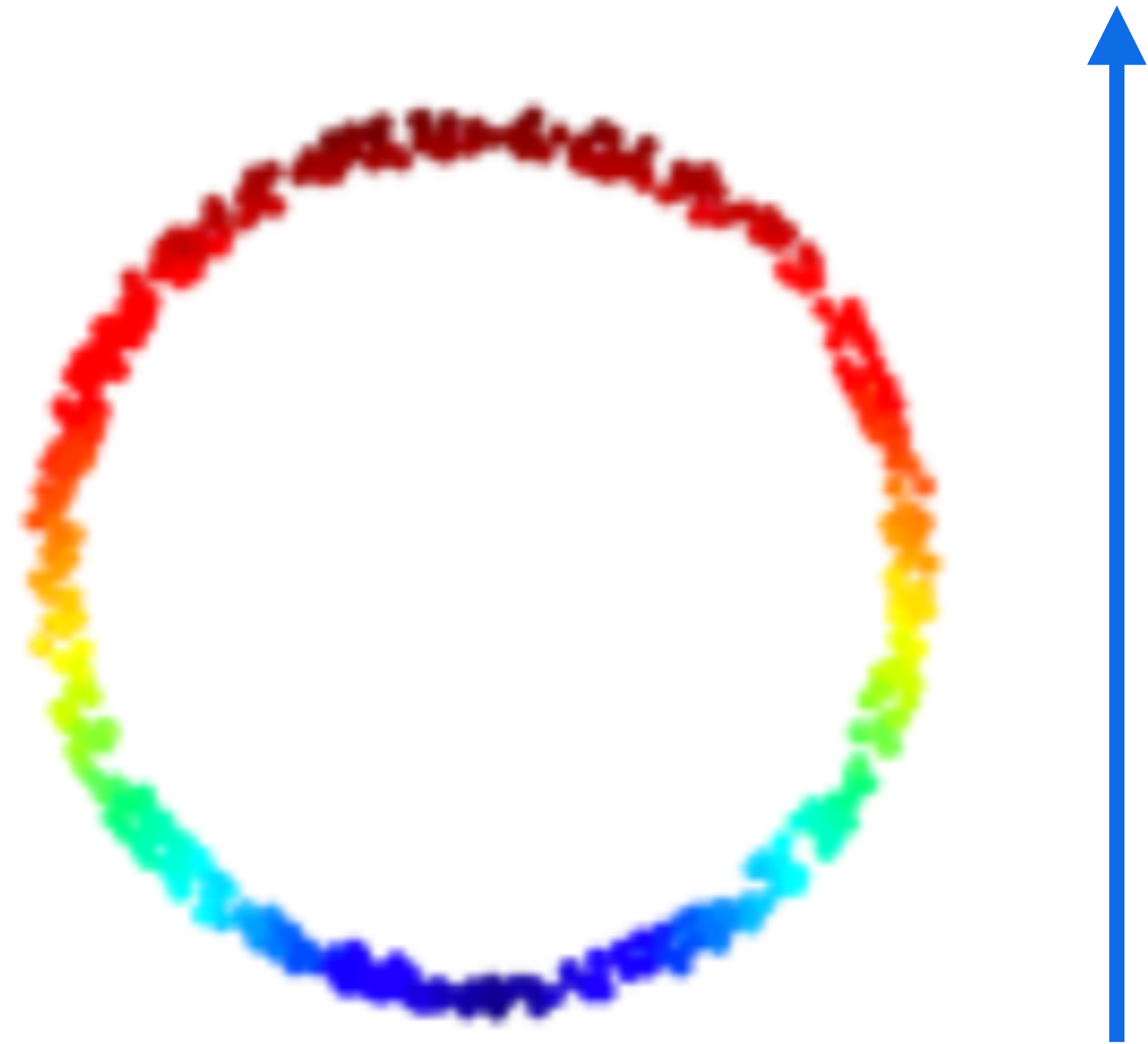
# Mapper Algorithm: Core

A main algorithm

# Mapper I/O, implementation

- Input:
  - Point cloud data, distance metric on the point cloud
  - Functions on the point cloud: filter function/lens
- Output:
  - (Interactive) visualization of a summary of the data as a graph or a simplicial complex based on function-induced clustering
  - Potentially interface with statistics and machine learning algorithms
- Parameters:
  - Parameters related to the chosen clustering algorithms
  - Filter functions
  - Number of intervals
  - Amount of interval overlap
  - Color functions, etc.

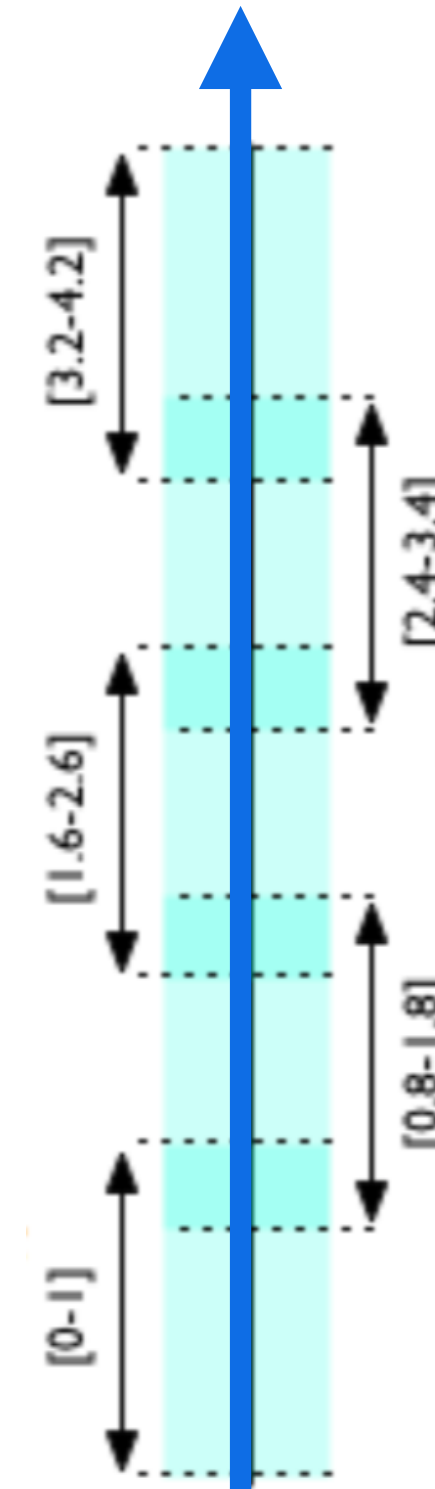
# Mapper Algorithm by example



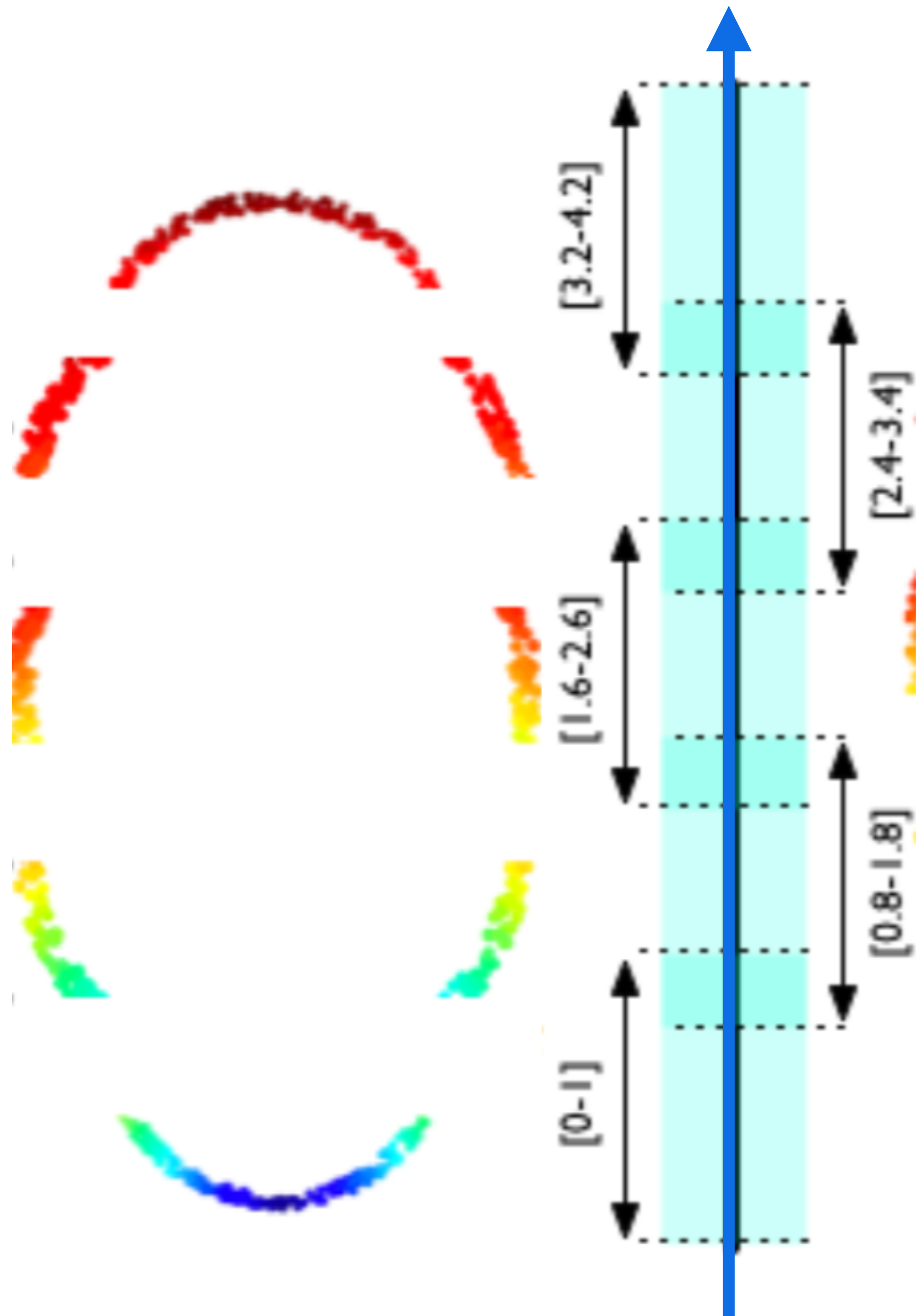
1. Input: a point cloud with a **filter** function e.g., a height function. Also assume that there is a distance (metric) defined between any two points in the point cloud.



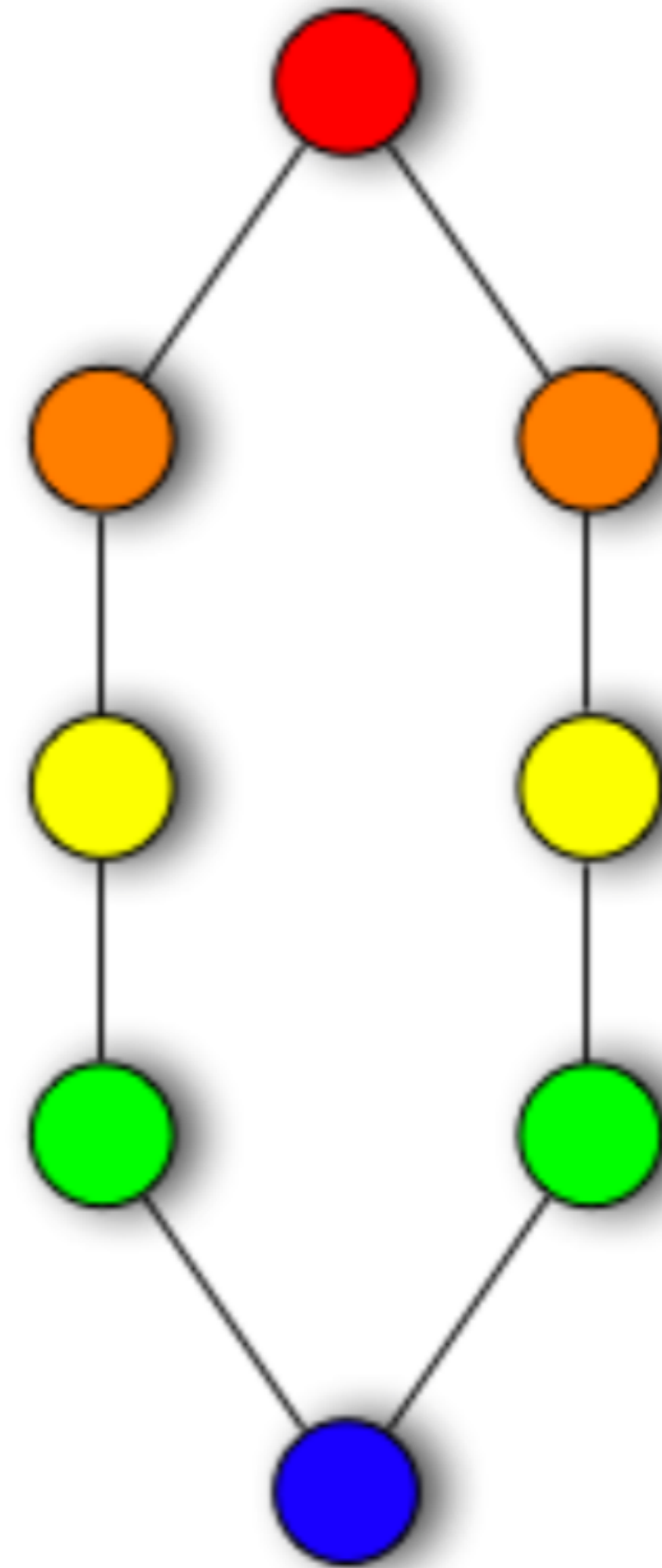
2. Cover the **range** of the function with intervals: using # of intervals, and amount % of overlap as parameters. E.g., # of intervals = 5, overlap = 25%.







3. Look at the points in the domain that falls into each interval, and apply clustering to these points. E.g., following the inverse map.



4. Obtaining the nerve of all clusters (a covering) in the domain. E.g., here it is a graph representation that summarizes the data.

Such a graph can interface with machine learning and interactive visualization...

# Clustering

- Almost any clustering algorithm can be used
- Assume there is a notion of distance (metric) between a pair of points in the data domain (distance can be computed or provided)
- Clustering is equivalent to a notion of connected component in the point cloud setting
- Commonly used clustering algorithms:
  - Density-based spatial clustering of applications with noise (DBSCAN)
  - Single-linkage clustering
  - K-means, etc.
- Desirable properties:
  - Not restricted to Euclidean distance; can take distance matrix input
  - Do not require specifying the number of clusters beforehand

# Parameters for the covering

- Number of intervals:  $k$ 
  - Increasing  $k$  will increase the # of clusters we observe
  - May create more empty clusters (small number of points per cluster)
  - Finer features of the data
  - If density varies, pick up clusters with high density
- Percentage of overlap:  $p$ 
  - Increasing  $p$  will increase the connectivities among the clusters
  - Sometimes robust in dealing with noise

# Filter functions

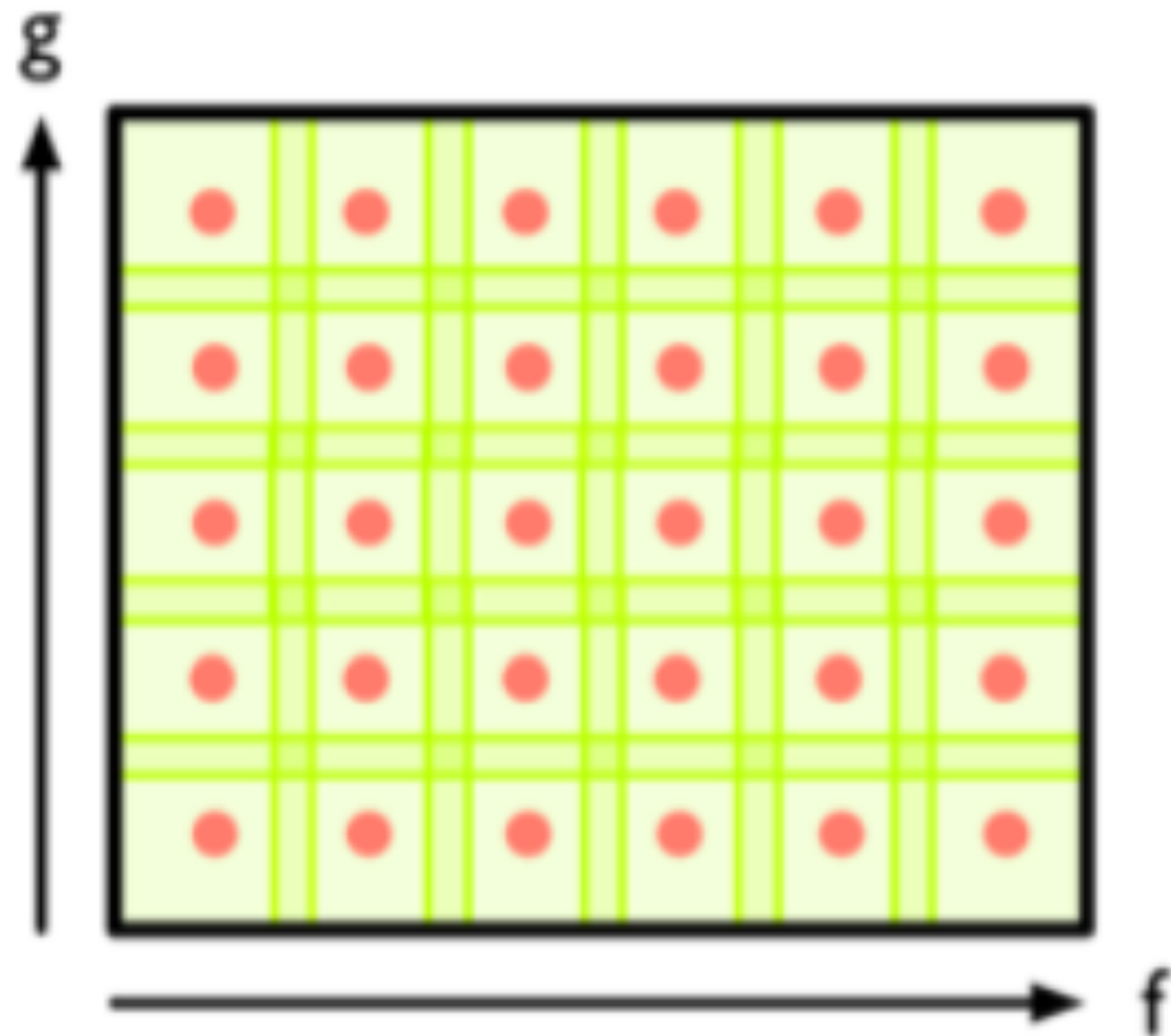
- A filter function can be given a prior, e.g. car purchasing price
- It can also be derived from the properties of the point cloud itself
  - Density estimation
  - Eccentricity
  - Distance to a point in the data
  - Graph laplacians

# Filter functions

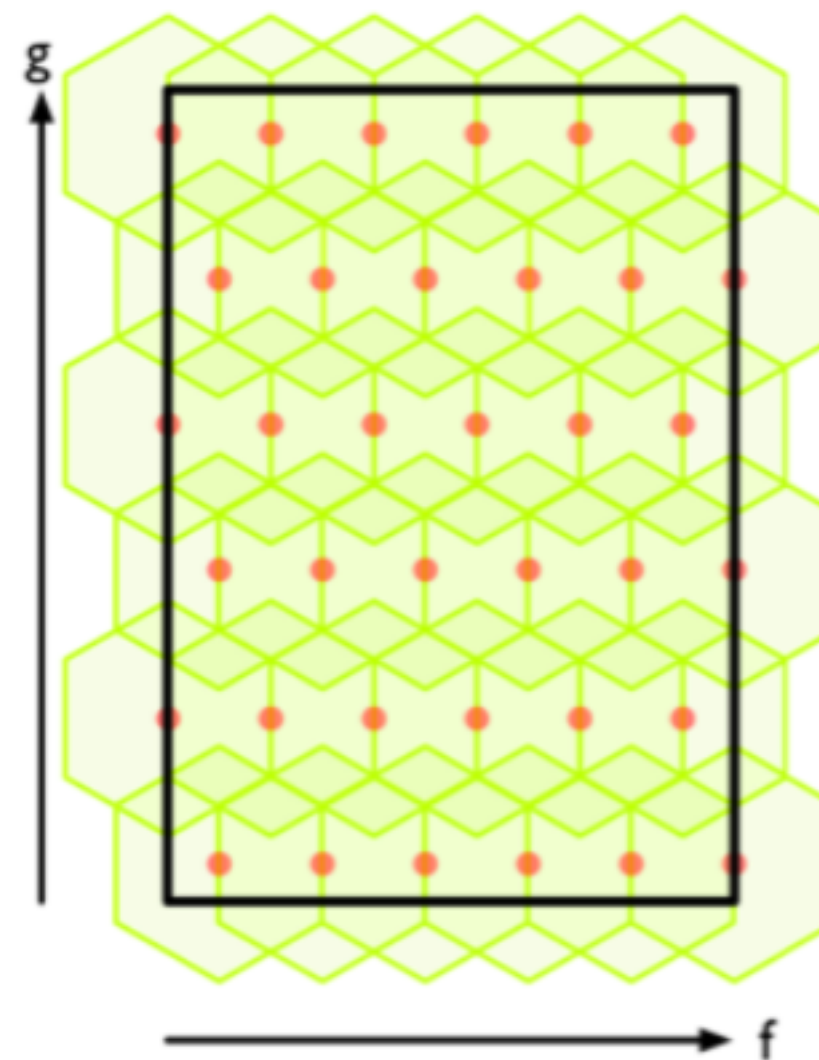
Density:  $f_{\varepsilon}(x) = C_{\varepsilon} \sum_y \exp\left(\frac{-d(x,y)^2}{\varepsilon}\right)$

Eccentricity  $E_p(x) = \left(\frac{\sum_{y \in X} d(x,y)^p}{N}\right)^{\frac{1}{p}}$

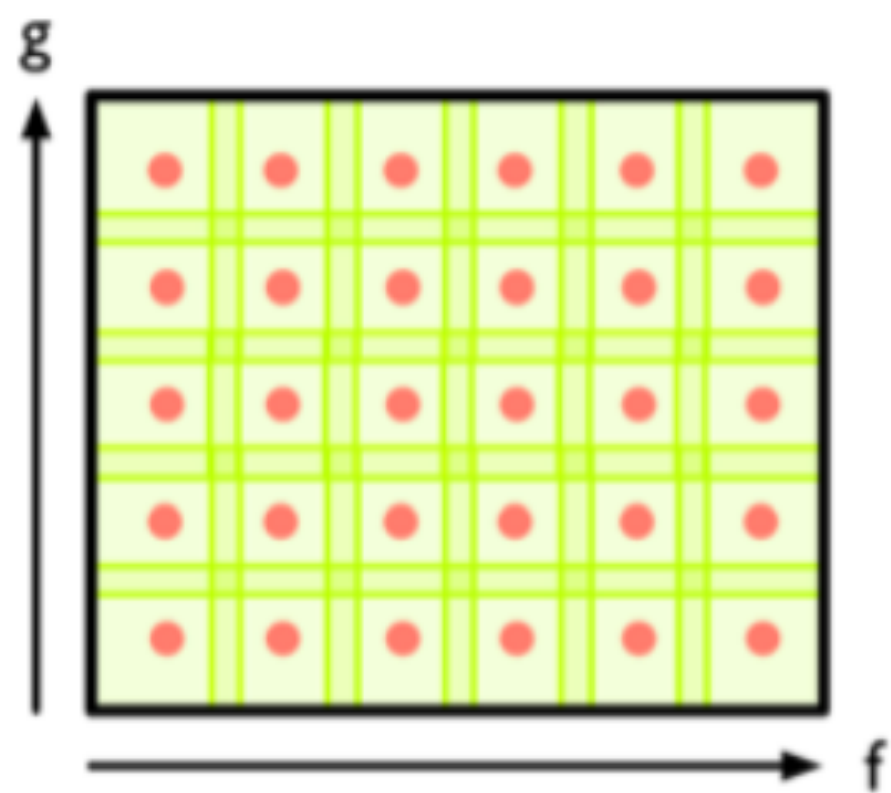
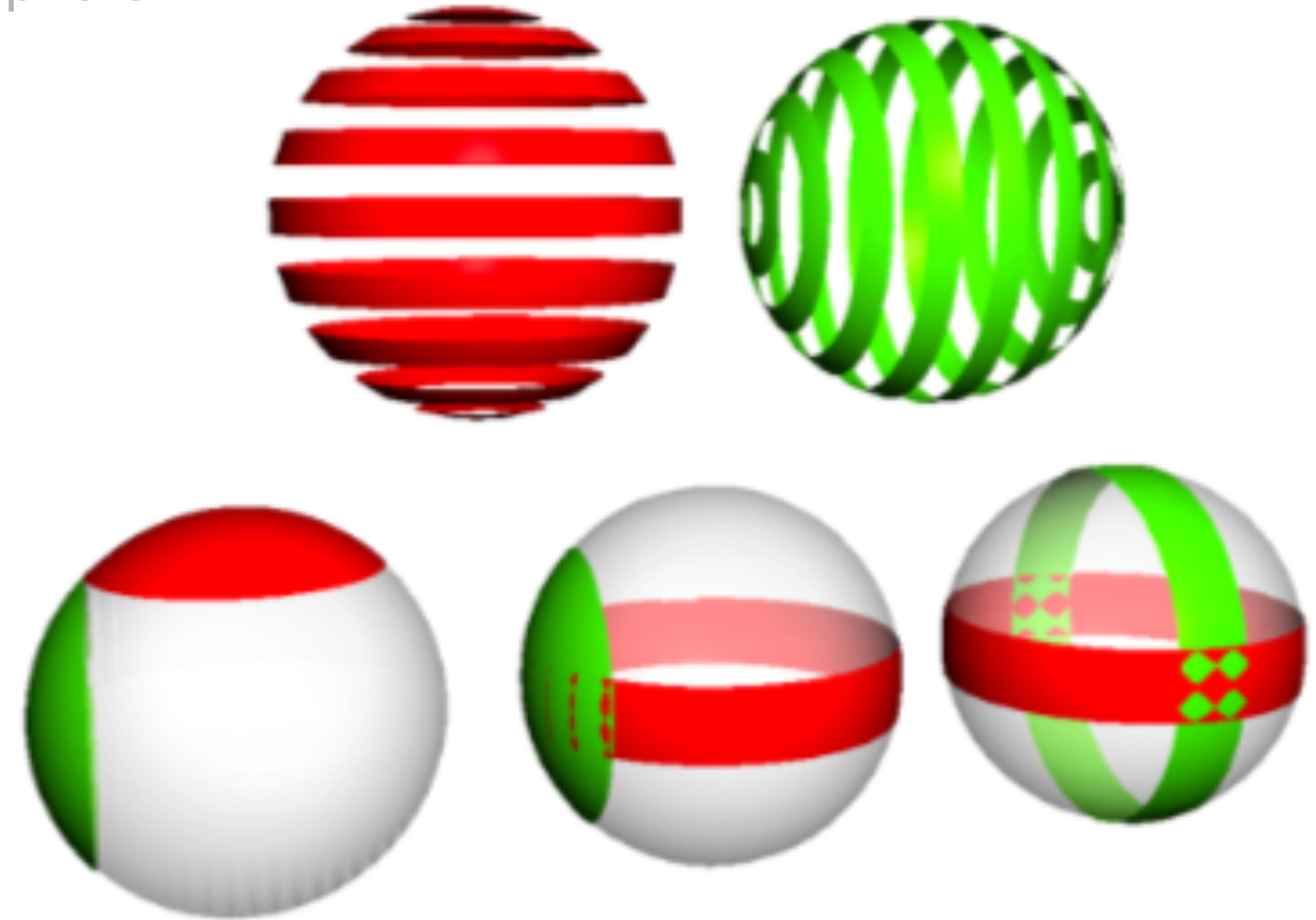
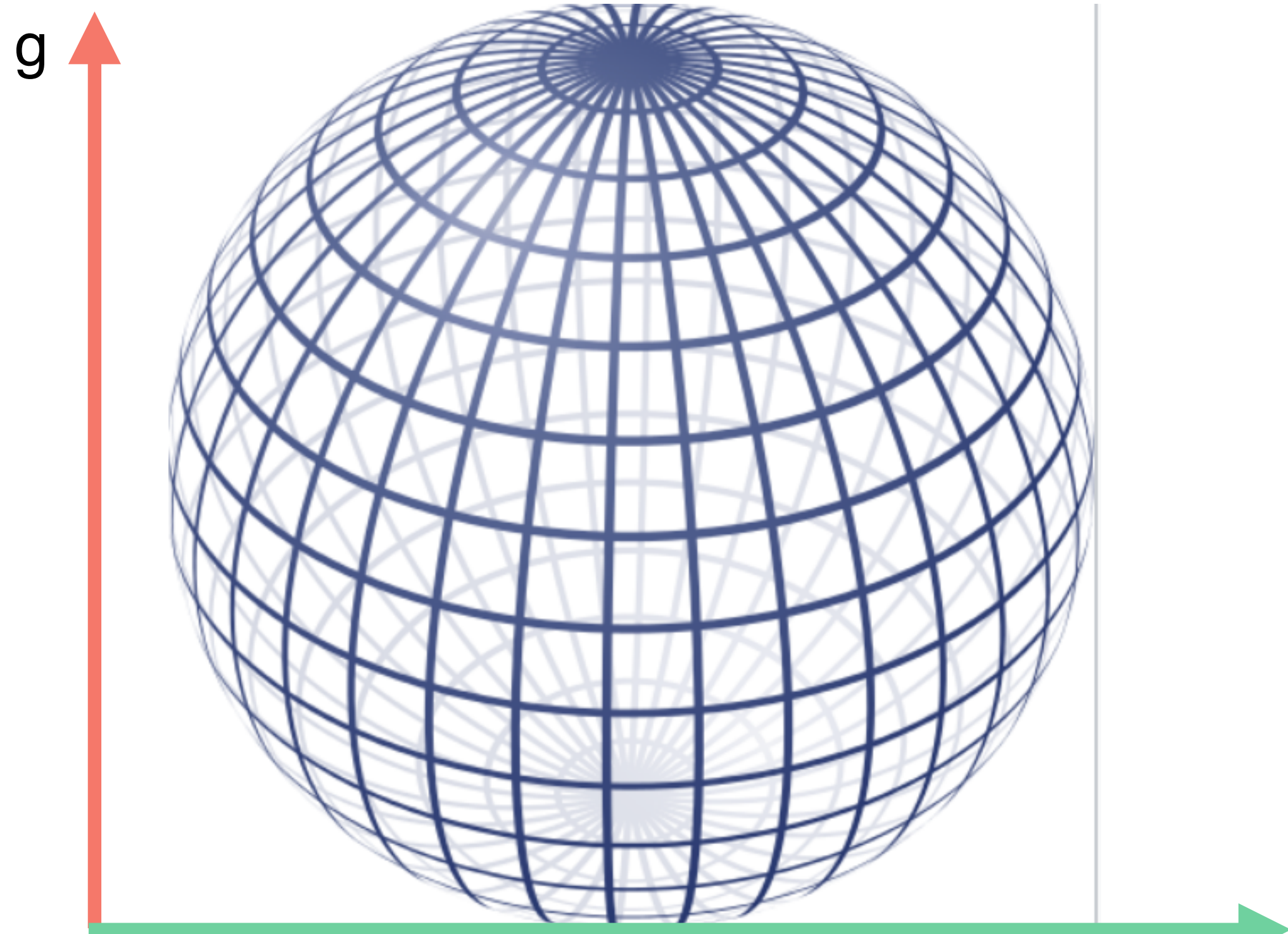
# 1D vs 2D Mapper



- 1D Mapper: a single filter function
- 2D Mapper: 2 filter functions
  - The covering of the domain of the function is no longer by intervals
  - Instead, by rectangles or other geometric shapes, etc.



# 2D Mapper



0	0	1	1	0	0
0	1	1	1	1	0
1	1	2	2	1	1
1	1	2	2	1	1
0	1	1	1	1	0
0	0	1	1	0	0

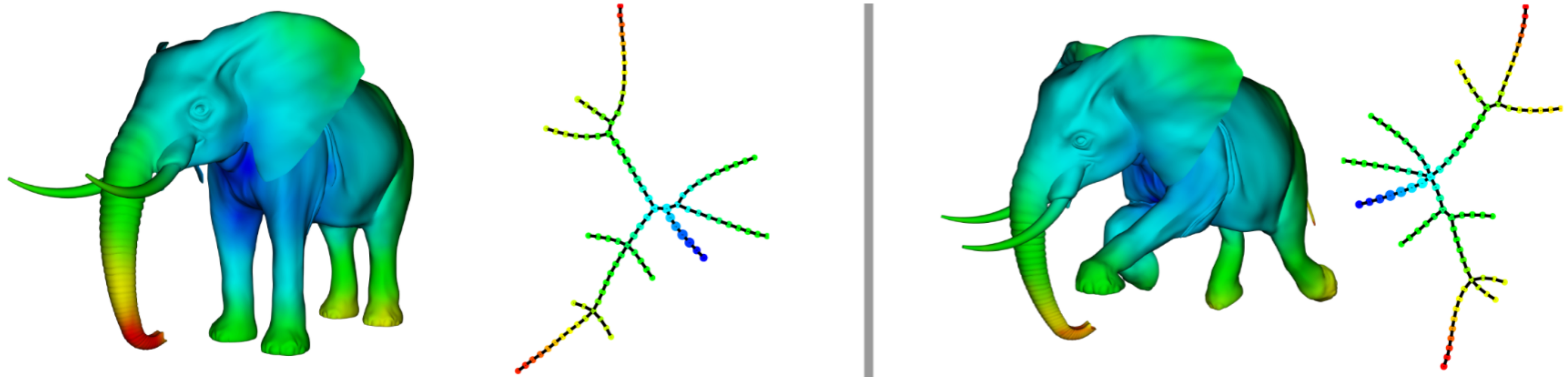
Count the number of connected components per "2D interval" (square in the range)

# Mapper Algorithm: Applications

A few examples

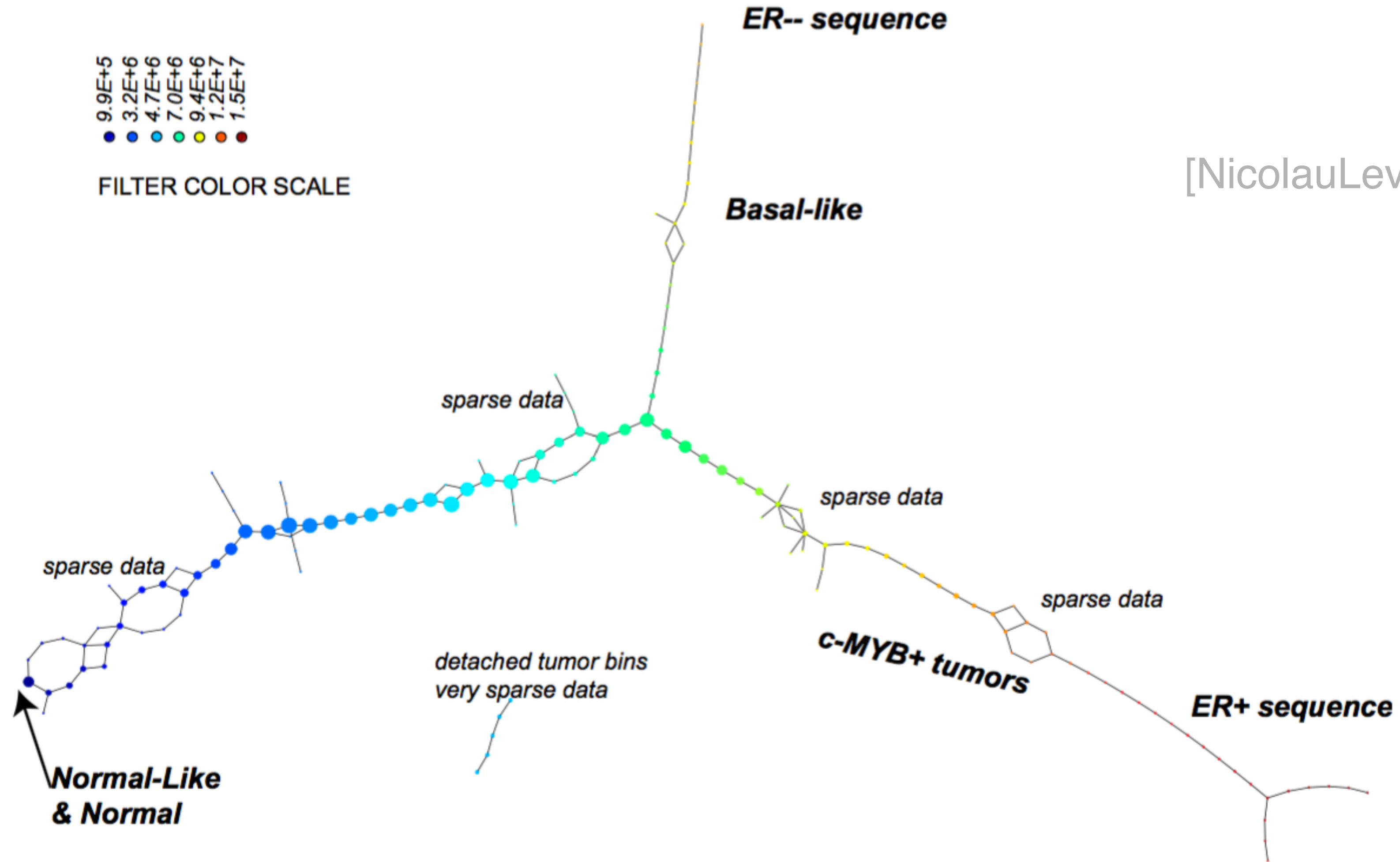


# Shape skeletonization & classification



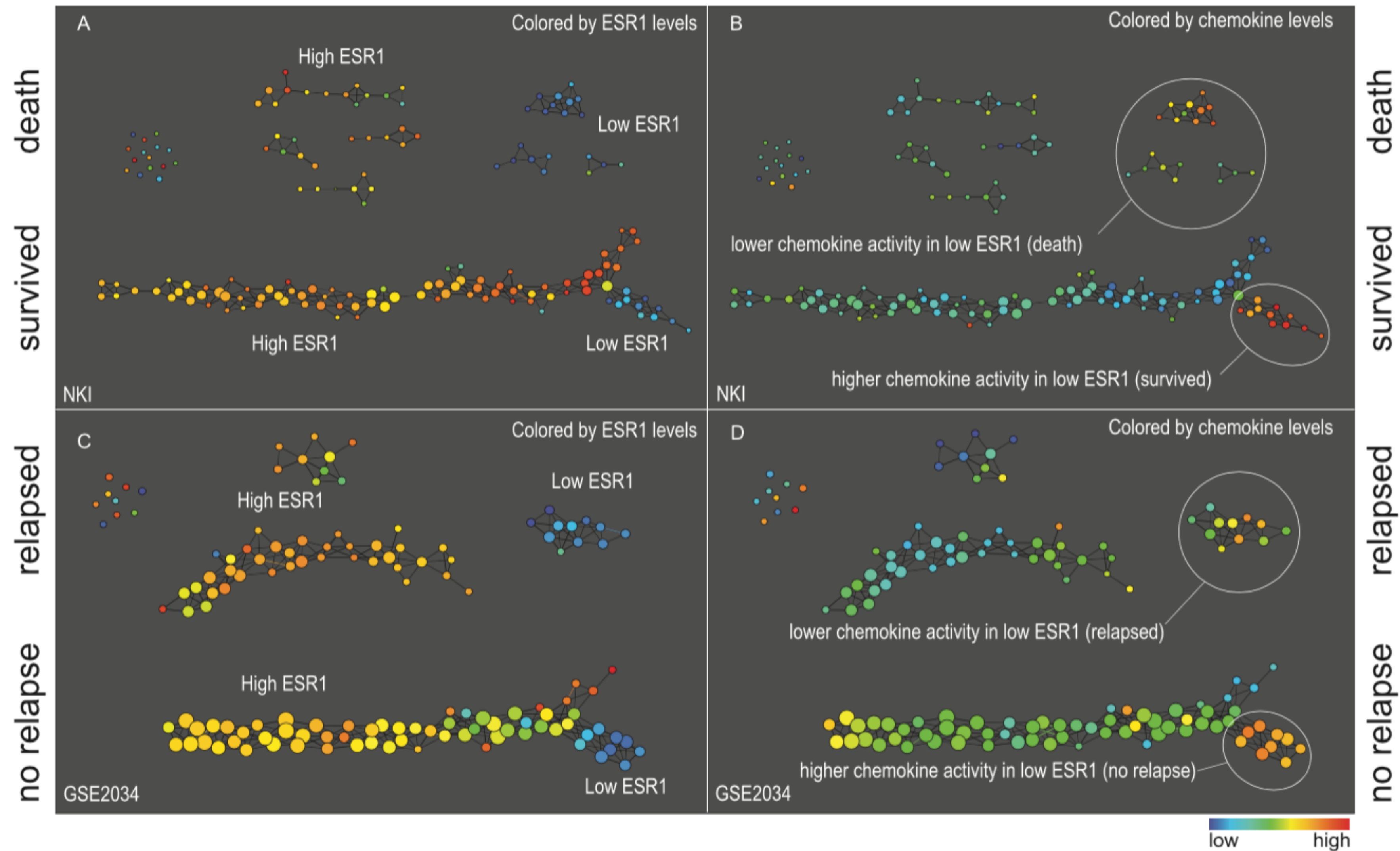
Also see Kepler Mapper demo examples: cat, lion, horse...

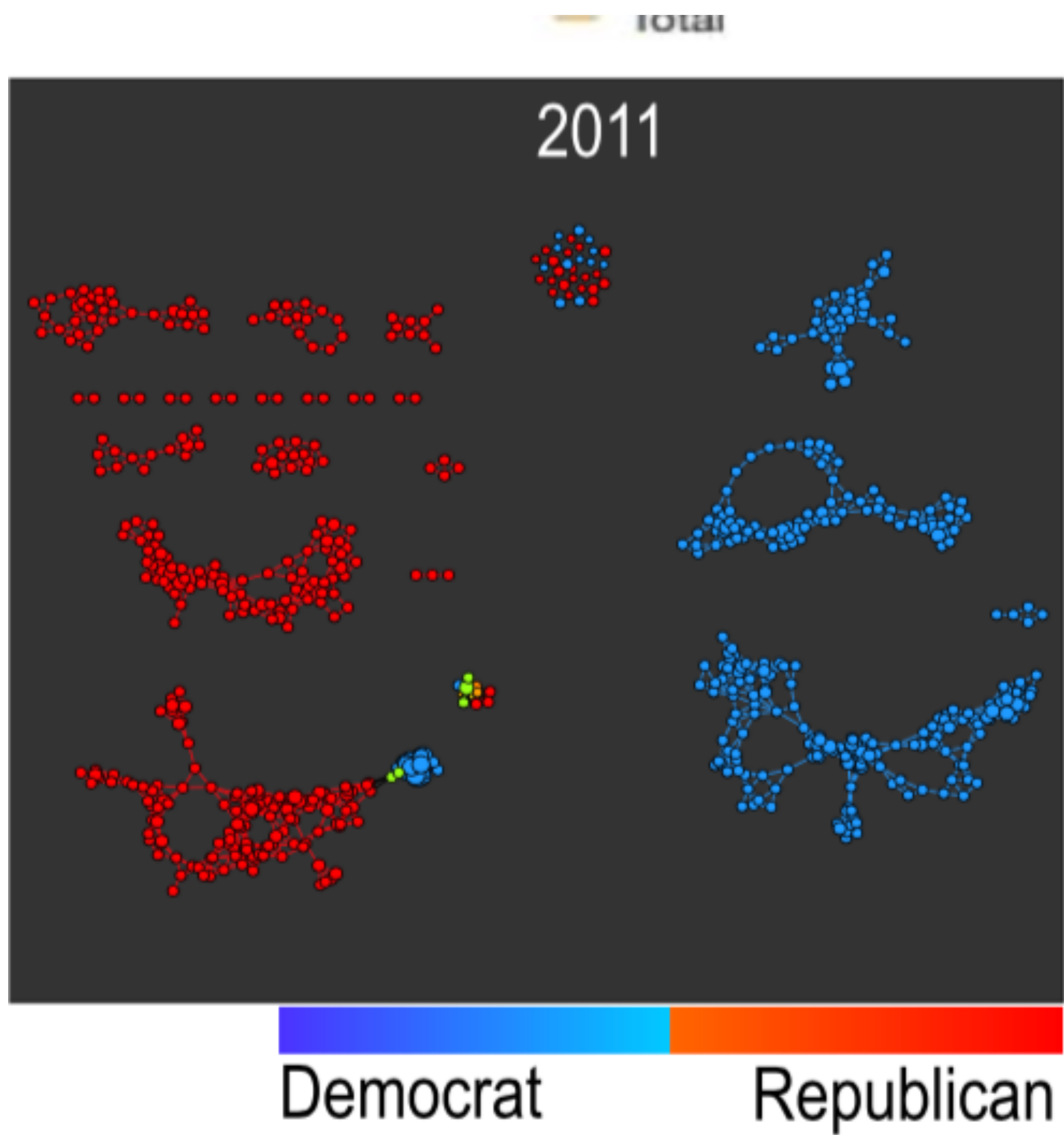
# Breast Cancer dataset



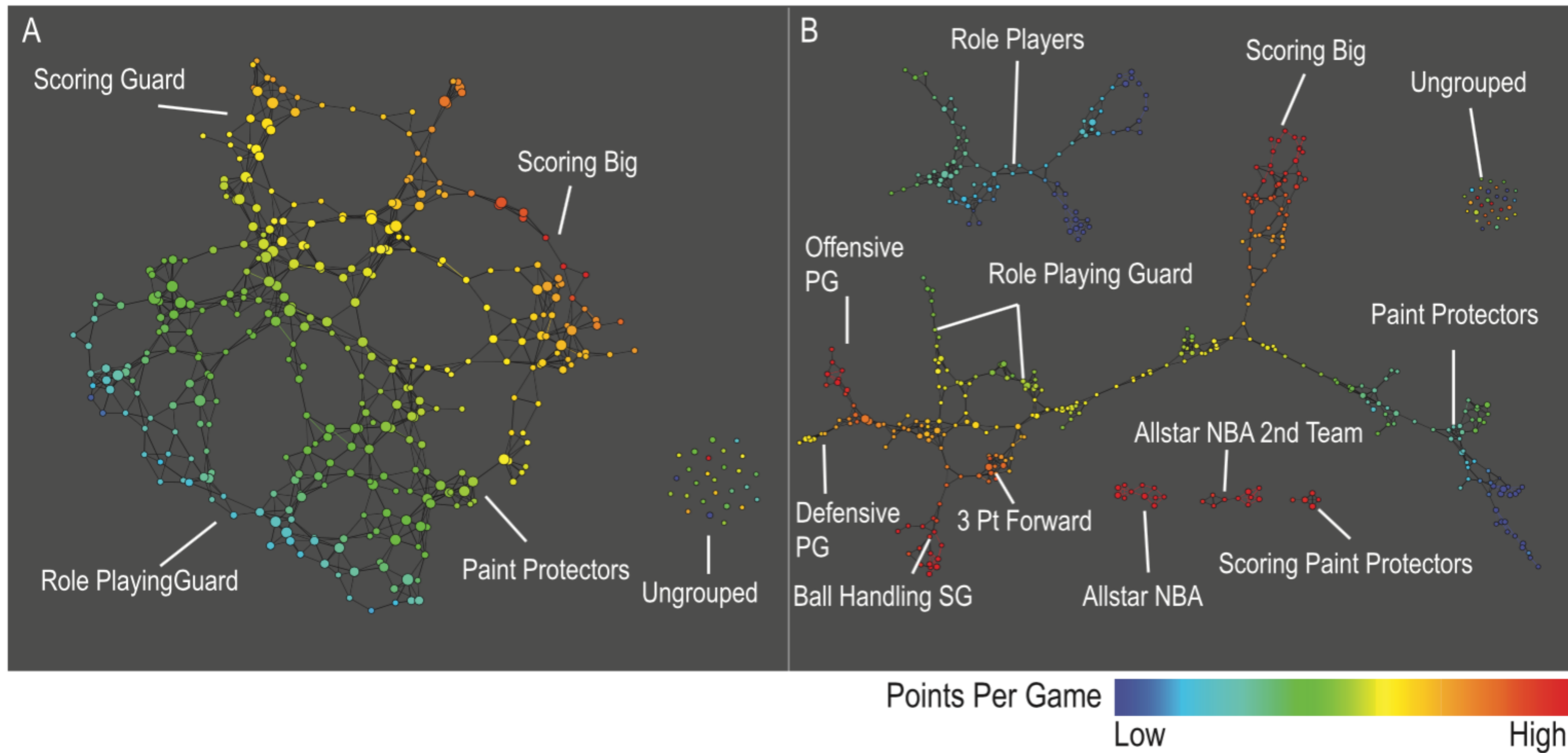
**Fig. 3.** PAD analysis of the NKI data. The output has three progression arms, because tumors (data points) are ordered by the magnitude of deviation from normal (the *HSM*). Each bin is colored by the mean of the filter map on the points. Blue bins contain tumors whose total deviation from *HSM* is small (normal and *Normal-like* tumors). Red bins contain tumors whose deviation from *HSM* is large. The image of  $f$  was subdivided into 15 intervals with 80% overlap. All bins are seen (outliers included). Regions of sparse data show branching. Several bins are disconnected from the main graph. The  $ER^-$  arm consists mostly of *Basal* tumors. The  $c-MYB^+$  group was chosen within the  $ER$  arm as the tightest subset, between the two sparse regions.

# Breast Cancer dataset 2





# Politics



# Sports

# Discussions

Future directions...

# Limitations of Mapper

- How to choose the stable range of parameters (k, p)
- How to choose the clustering algorithms
- How to choose the filter functions
- Obtain insights with the right color function...

# Future directions

- Better automatic parameter tuning
- Multi-scale mapper
- 2D Mapper: theoretical understanding
- What are the other possible variations? (Discussion)



# KepperMapper

A Demo

# Open sourced implementation

- Python Mapper
  - <http://danifold.net/mapper/index.html>
- R implementation: TDAmapper
  - <https://cran.r-project.org/web/packages/TDAmapper/index.html>
- Spark Mapper:
  - <https://github.com/log0ymxm/spark-mapper>

# Kepler-Mapper Demo

- The example with one circle
- The example with two circles
- Digits example
- Breast Cancer Example

# Project 1 Explained

A simple, first application of HD analysis

# Limitations of KeplerMapper

- KeplerMapper is still under active development:
  - The visualization capabilities are limited (one color function)
  - There are not much of interactive visualization
  - No integration with other machine learning algorithms

# Project 1 tips

- The questions on KeplerMapper is rather fundamental, the goal is for you to understand the inner-working of the code; try to read and understand `kmapper.py` as much as possible
- Start your project early; start it today
- Read the paper

# Final Project Idea

- A non-trivial extension to open sourced Mapper implementation
  - E.g. enhance the visualization capabilities of KeplerMapper
  - Scalable solution using Spark Mapper
- A non-trivial application of mapper algorithms to real-world data set
  - Need to solve or give insight to a real-world problem



# Thanks!

Any questions?

You can find me at: [beiwang@sci.utah.edu](mailto:beiwang@sci.utah.edu)



# CREDITS

Special thanks to all people who made and share these awesome resources for free:

- ☐ Presentation template designed by [Slidesmash](#)
- ☐ Photographs by [unsplash.com](#) and [pexels.com](#)
- ☐ Vector Icons by [Matthew Skiles](#)

# Presentation Design

This presentation uses the following typographies and colors:

## Free Fonts used:

<http://www.1001fonts.com/oswald-font.html>

<https://www.fontsquirrel.com/fonts/open-sans>

## Colors used

