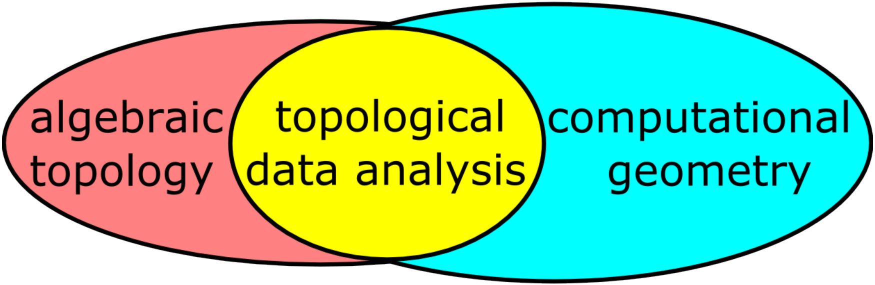
Topological Data Analysis

theory, applications and the future

Vitaliy Kurlin, http://kurlin.org
Materials Innovation Factory (MIF) and
Computer Science, University of Liverpool

TDA = topological data analysis

quantifies *persistent topological structures*
analysing unorganised data *across all scales*.



Goal: also use *machine learning* and *statistics*.
Carlsson, Topology and Data, Bulletin AMS 2009.

What are data in TDA?

Input: a cloud of points with pairwise distances

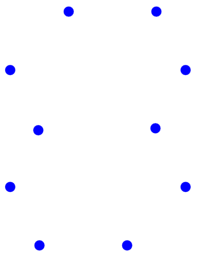
without any scale, $\#$ neighbours, noise bound.

2D cloud: edge pixels in an image, a noisy scan.

High-dim cloud: a vector of features, histogram.

Life story of a cloud: scale $\alpha = 0$

Blue cloud: unstructured set of points

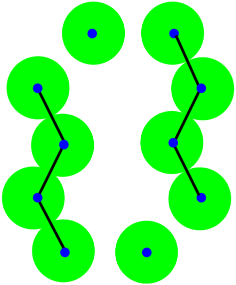


Question: how many holes?

Answer: not clear at scale 0

Idea: study it at all scales

Life story of a cloud: scale $\alpha \approx 1.1$



scale := radius of disks

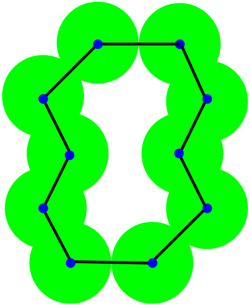
offset := union of disks

no holes are born yet

offsets are evolving if the scale is increasing

0 now $\approx$ 1.1

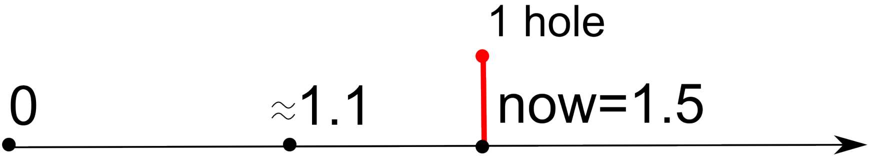
Life story of a cloud: scale $\alpha = 1.5$



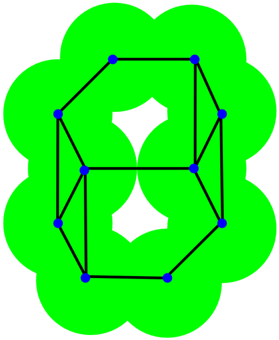
First hole is *born*

at scale = 1.5

continue ...

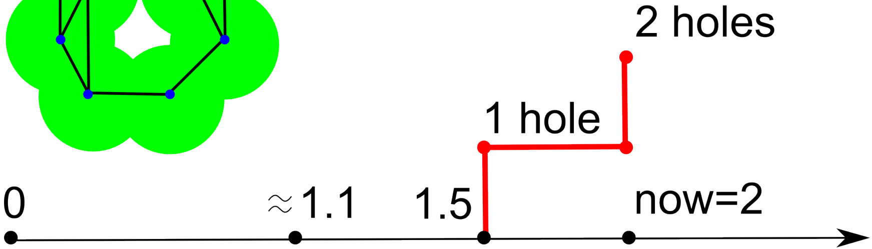


Life story of a cloud: scale $\alpha = 2$

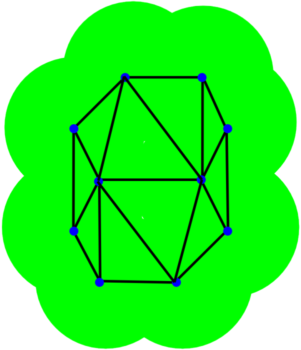


Second hole is *born* when

scale = 2 (radius of disks)



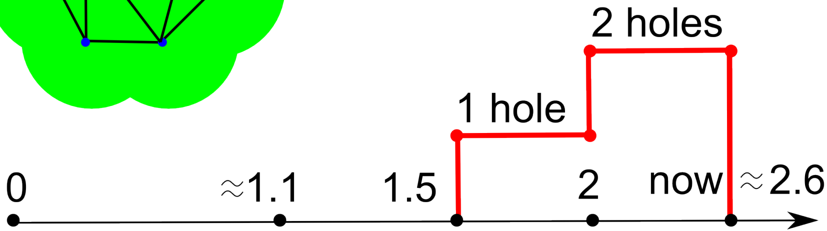
Life story of a cloud: scale $\alpha \approx 2.6$



Both holes *die* at scale $\approx$2.6

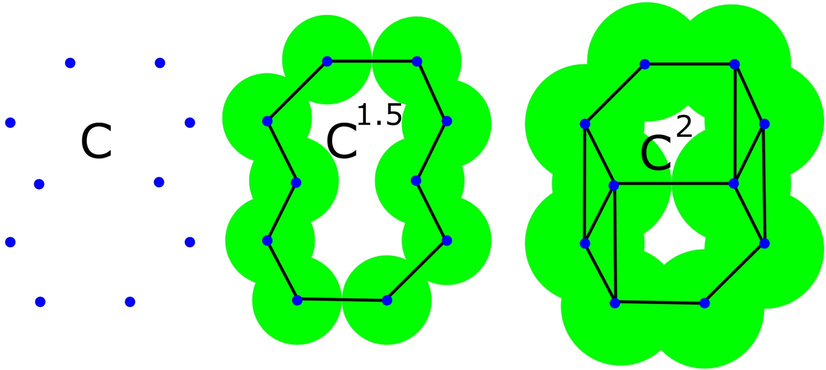
P(1 hole)$\approx$46.5%

P(2 holes)$\approx$53.5%



From a cloud C to a filtration

Def : the α-offset of a cloud C in a space M is the union of balls $C^\alpha = \bigcup_{p \in C} B(p; \alpha)$ of a radius α.



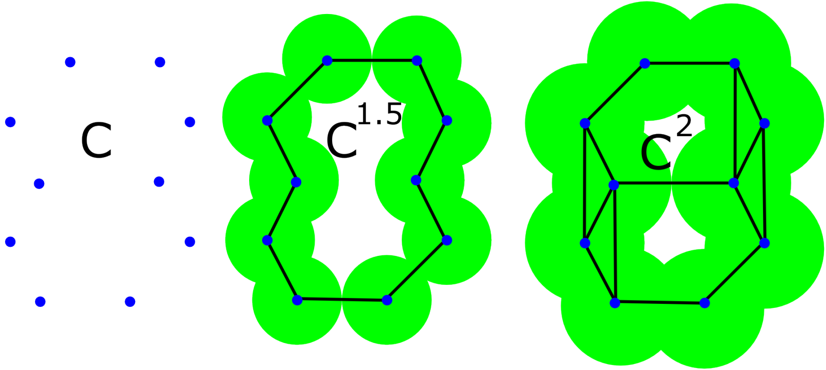
Filtration $C^0 \subset \cdots \subset C^\alpha \subset \ldots$ in a metric space.

Key idea: topology evolution

When α (discrete or continuous) is increasing,

we study how the topology of C^{α} changes:

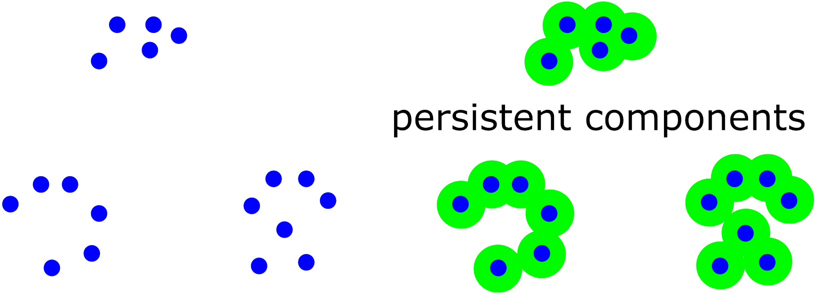
components in 0D, cycles in 1D, surfaces in 2D.



Single edge clustering

A manual choice of the scale α is needed:

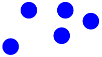
all points with $d(p, q) \leq 2\alpha$ are in one cluster.

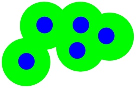


If α is increasing, clusters merge. Choose α?

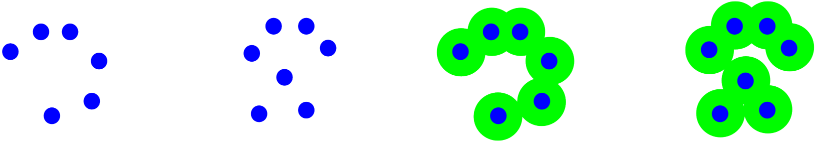
0D homology = con. components

Choosing a scale α might not be possible for high-dimensional data, hard to visualise.





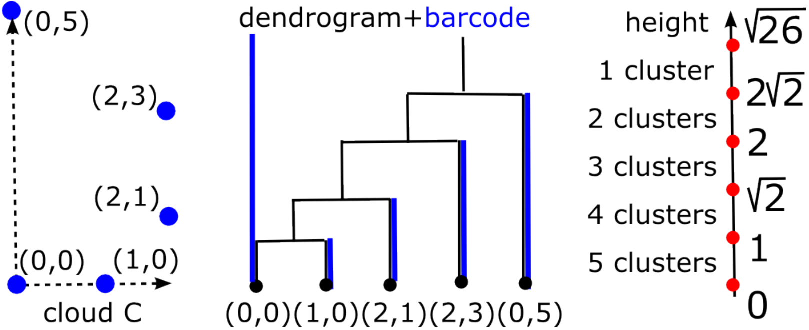
persistent components



Persistent components of C^α living over a long interval of α are more *natural clusters* of C.

Dendrogram of clustering

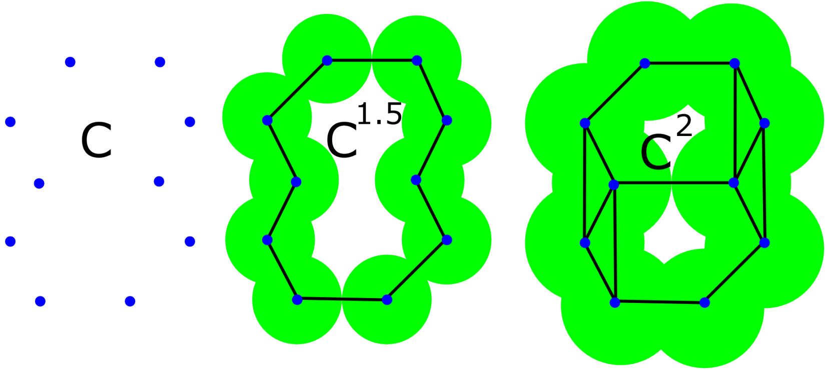
Each internal node is a cluster merged from 2 or more smaller clusters at the children nodes.



Red dots form a *persistence diagram* in 0D, so **TDA extends** clustering to *high-dim structures*.

1D homology = holes in 2D shapes

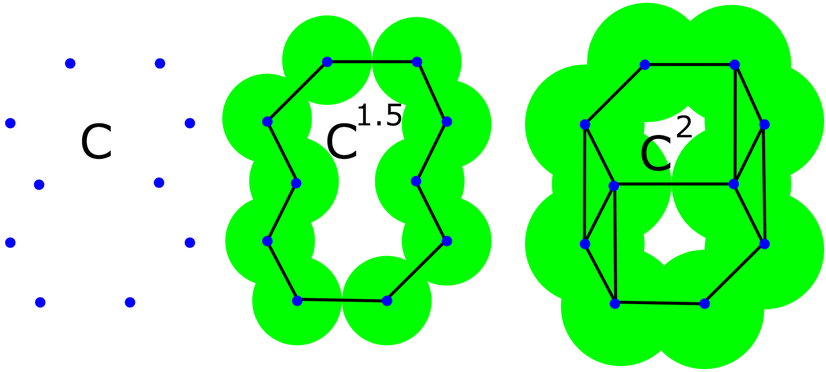
A *hole* is a bounded component of $\mathbb{R}^2 - C^\alpha$ enclosed by a 1D cycle represented in $H_1(C^\alpha)$.



$C^{1.5}$ has 1 hole, C^2 has 2 holes, C^3 has 0 holes.

Life spans of holes in 2D shapes

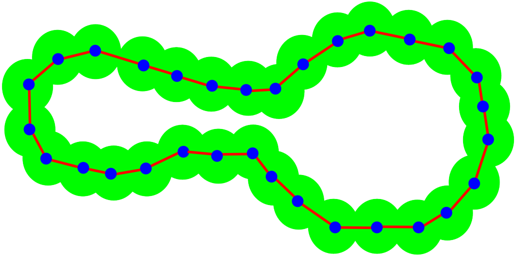
A hole is *born* at a scale $\alpha = \text{birth}$ and *dies* later at $\alpha = \text{death}$, so has a *life span* $[\text{birth}, \text{death})$.



A hole is born at 1.5, splits at 2, dies at ≈ 2.6.

Homology and its instability

Homology $H_k(S)$ counts *k-dimensional holes*: a vector space of combinations of simplices of S.



$H_k(S)$ is *unstable* under perturbations of data.

$f : X \to Y$ induces linear $f_k : H_k(X) \to H_k(Y)$, e.g. long cycle above $\to$ sum of 2 short cycles.

Persistent homology of data

Any filtration $S(\alpha_1) \subset S(\alpha_2) \subset \cdots \subset S(\alpha_m)$ of complexes induces linear maps in homology:

$$H_k(S(\alpha_1)) \to H_k(S(\alpha_2)) \to \cdots \to H_k(S(\alpha_m)),$$

which splits as a sum of basic sequences over $\mathbb{Z}_2$ from α_i to α_j, i.e. $0 \to \mathbb{Z}_2 \xrightarrow{\text{id}} \cdots \xrightarrow{\text{id}} \mathbb{Z}_2 \to 0$ by a classification of finitely generated modules.

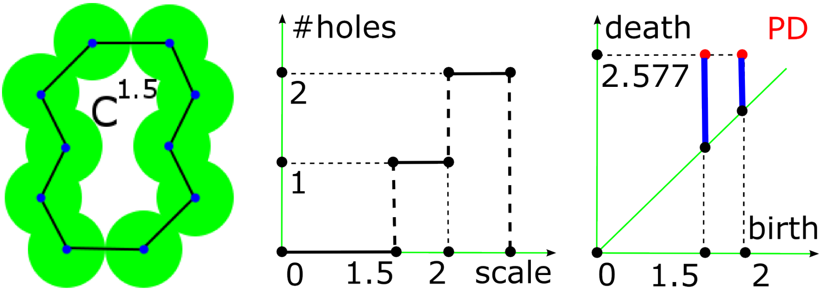
The evolution of homology *across all scales* is summarised by bars $[\alpha_i, \alpha_j)$ that form a barcode.

Output of TDA: all life spans

The evolution of all holes is summarised by

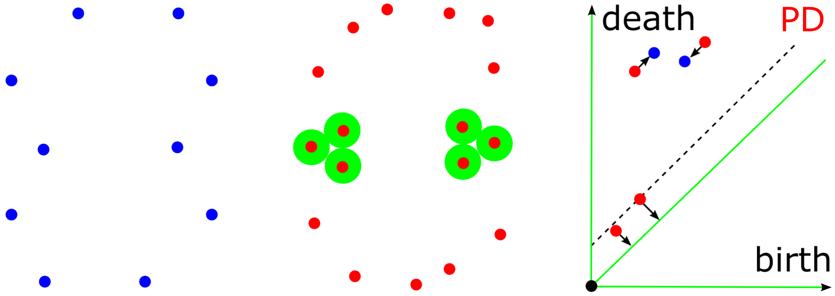
bars $[\mathrm{birth}, \mathrm{death})$ in the barcode or by

dots $(\mathrm{birth}, \mathrm{death})$ in the persistence diagram.



Stability of persistence

Th (Cohen-Steiner, Edelsbrunner, Harer, 2007)

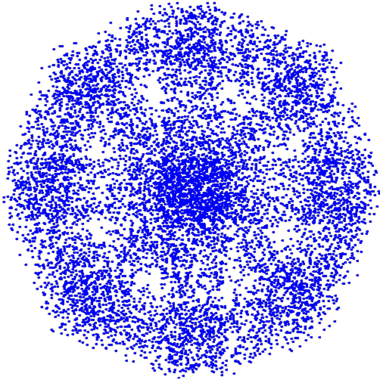


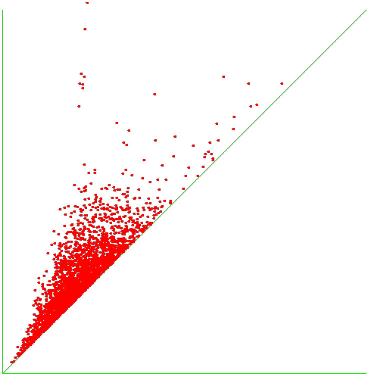
If a data cloud C is *perturbed by* ε (in the ε-offset C^ε), the persistence diagram is *perturbed by* ε, namely there is an ε*-matching* of all dots in PDs.

Guessing holes from a sample

Dots with a *high persistence* $\leftrightarrow$ ‘true’ holes.

Red dots near the diagonal $\leftrightarrow$ ‘noisy’ holes.



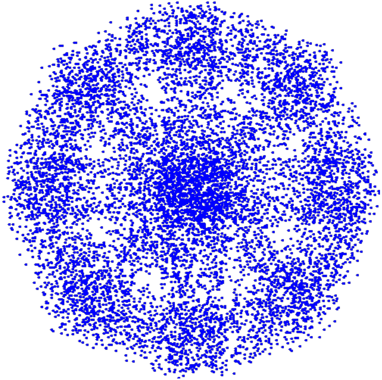


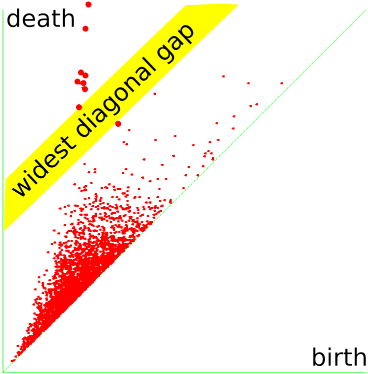
How many holes does the sampled graph have?

Counting holes in noisy clouds

$O(n \log n)$ algorithm, theoretical guarantees in

VK. CVPR'14: Computer Vision & Pattern Recognition

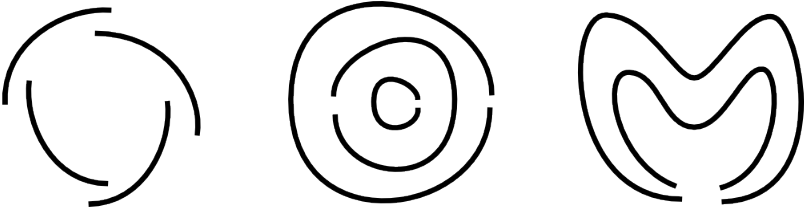




Where are these holes? No structure on data yet.

Computer Graphics application

Problem: complete all closed contours or paint all regions that they enclose (a *segmentation*).

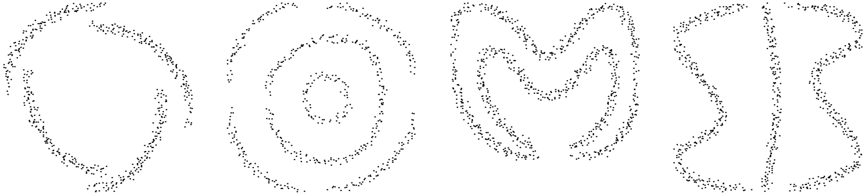


A user drawing a sketch on a tablet might be happy with our fast automatic ‘best guess’:

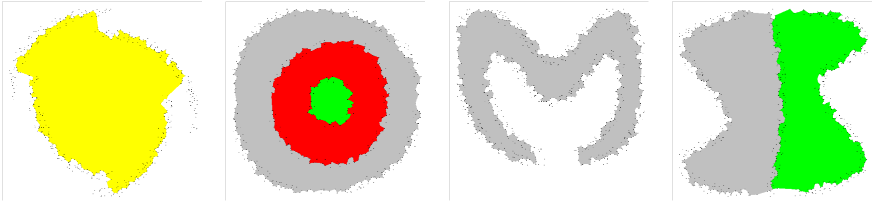
make contours closed so that I can paint areas

(a scale is easy to find, but we can’t ask for it).

Input & output of auto-completion

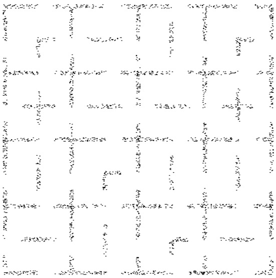


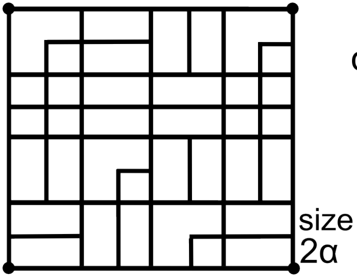
Required output: most ‘persistent’ contours.

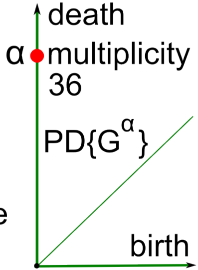


Counting holes in C may be easy

The graph G has H_1 of rank 36, hence any ε-sample C of G will probably have 36 holes.

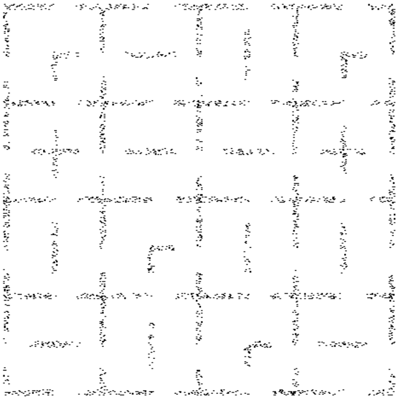


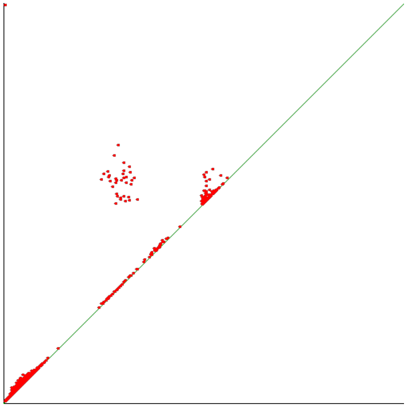




How can we see that there are 36 holes in C?

Using stability of persistence

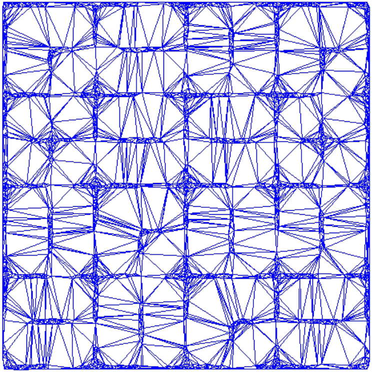


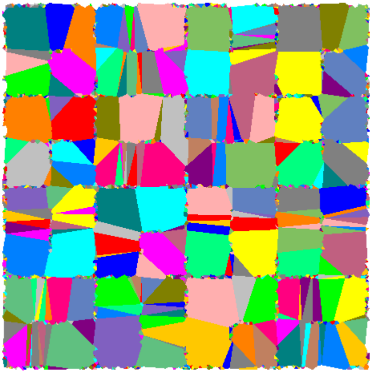


We can find the *widest diagonal gap* separating 36 points from the rest of persistence diagram.

An initial segmentation of C

Acute Delaunay triangle is a ‘center of gravity’.

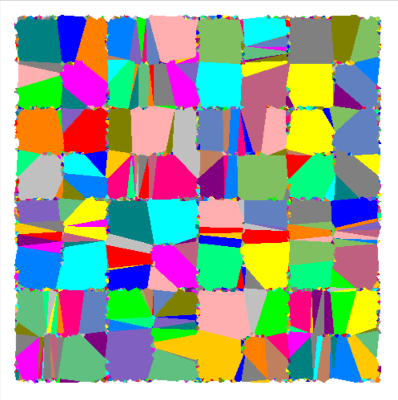


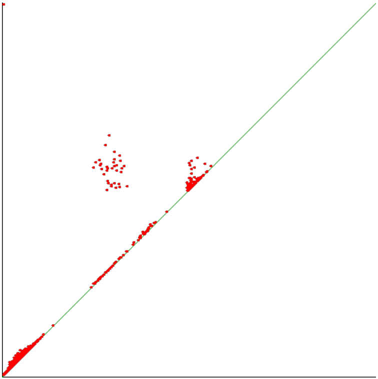


We attach all adjacent non-acute triangles to get an initial segmentation on the right hand side.

Harder than counting cycles

Initial regions $\leftrightarrow$ red dots in PD (too many).

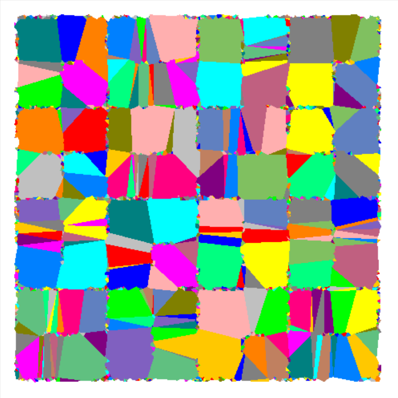


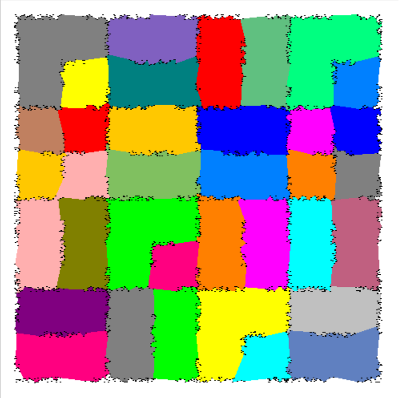


We should merge 36 regions of high persistence with all remaining regions of lower persistence.

Merging initial regions

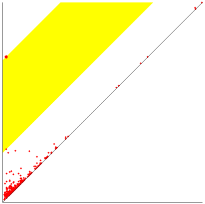
Building $\mathrm{PD}\{C^\alpha\}$, we keep adjacency relations of merged regions to enrich persistence info.

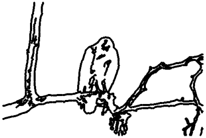


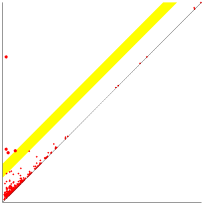


Hierarchy of segmentations

A user can prefer to get exactly *m* regions by choosing 2nd widest diagonal gap in PD1 etc.

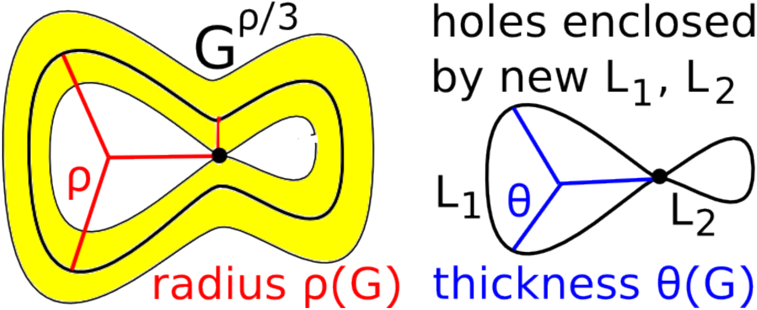






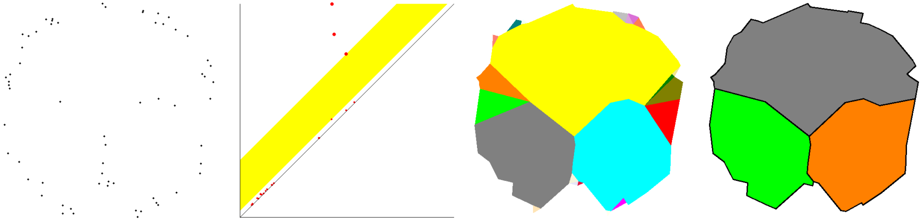
Radii and thickness of a graph

A contour $L \subset \mathbb{R}^2$ has $\rho(L) = \min \alpha$ when $L^\alpha \sim \cdot$



A graph $G \subset \mathbb{R}^2$ has $\theta(G) = \min \rho(L_i)$ over the contours enclosing all newborn holes in G^α.

Theoretical guarantees



Th (VK) : if C is an ε-sample of a graph $G \subset \mathbb{R}^2$ whose basic cycles have radii $\rho_1 \leq \cdots \leq \rho_m$ and $\rho_1 > 7\varepsilon + \theta(G) + \max\{\rho_{i+1} - \rho_i\}$, the output segmentation has *m contours 2ε-close to G*.

Pattern Recognition Letters, 2016, v. 83, p. 3-12.

TDA for learning a shape of data

Example questions for a point cloud C: does it look like a circle, graph, higher-dim manifold?

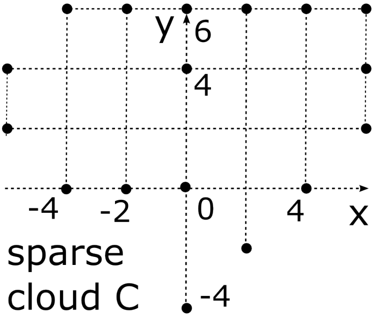
Many skeletonisation algorithms are *iterative*, use *parameters* (scale or weights of criteria).

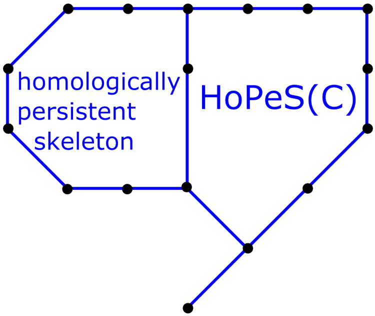
Key idea: analyse the data across all scales.

TDA provides a *quick and simple approximation* to data, e.g. a 1-dimensional skeleton whose parameters can be *refined by optimisation*.

Parameterless skeletonisation

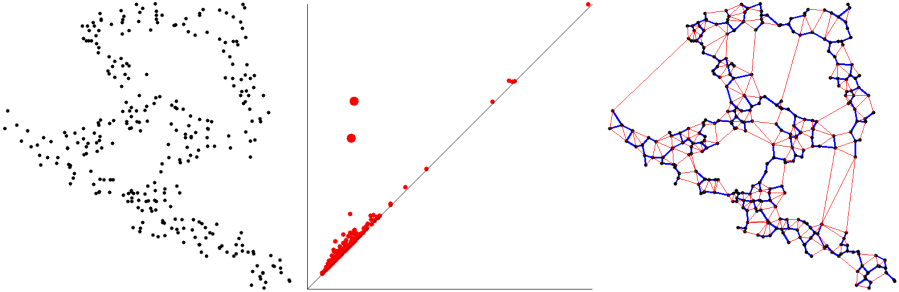
Homologically Persistent Skeleton $\mathrm{HoPeS}(C)$ is the first *universal structure* on a cloud C that optimally captures all 1D persistence on C.





$\mathrm{HoPeS}(C) = \mathrm{MST}(C) \cup$ critical edges

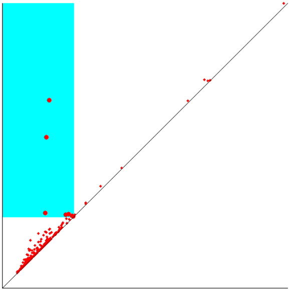
Def: each critical edge gives birth to a class for $\mathrm{birth} \le \alpha < \mathrm{death}$ in 1D persistence of $\{C^\alpha\}$.

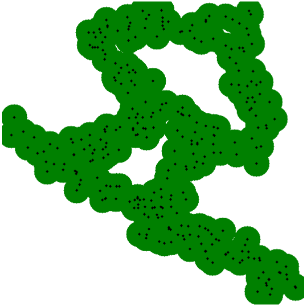


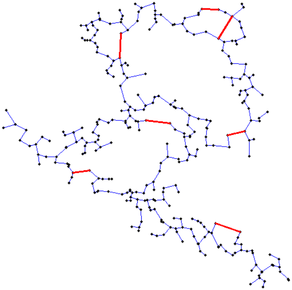
$\mathrm{HoPeS}(C)$ is a *rotation-and-scale invariant* structure on C, encodes all 1D persistence.

Optimality of $\mathrm{HoPeS}(C;\alpha)$

Th (VK'15). $\mathrm{HoPeS}(C;\alpha)$ for any scale α has the *minimum length* among all graphs $G \subset C^{\alpha}$ with the same homology H_0, H_1 as C^{α}, so $\mathrm{HoPeS}(C)$ 'captures' homology of the cloud C at all scales.

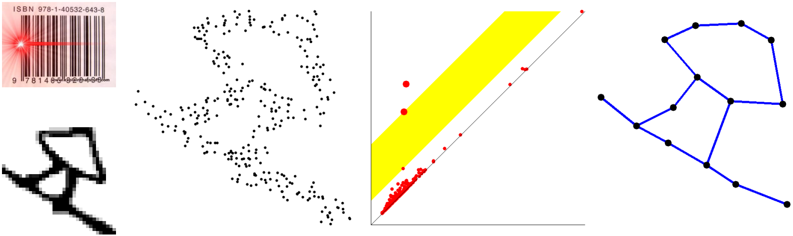






Graph reconstruction problem

Shop barcodes are not readable by humans.



We can make *visual markers* like Egyptian hieroglyphs readable by *humans and robots*.

VK, CAIP'15: Computer Analysis of Images and Patterns.

Global stability of $\mathrm{HoPeS}'(C)$

Cor (VK): derived skeleton $\mathrm{HoPeS}'(C)$ stays in a small offset under perturbations of a cloud C.

$\mathrm{HoPeS}(C)$ is extended to any finite metric space C and to any filtration of complexes on C.

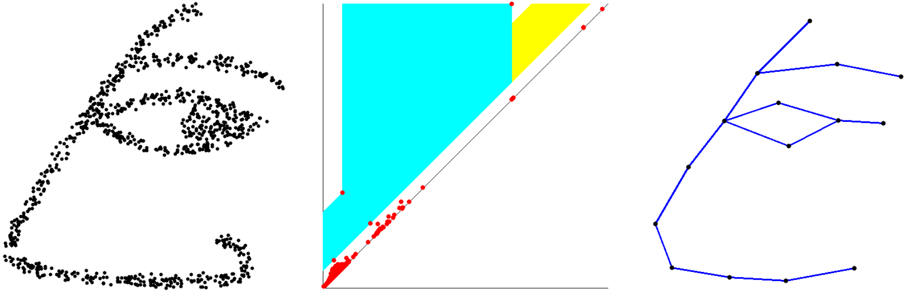
SGP 2015, Computer Graphics Forum 34-5.

Limitations: $G \subset \mathbb{R}^2$ must have $\mathrm{PD}_1\{G^\alpha\}$ with a wide diagonal gap, not for trees like T and C.

Next: better results by a deeper analysis of PD_1.

Another challenging example

The (noisy version of a) true cycle of G has a lower persistence than a fake cycle in $\mathrm{PD}_1\{C^\alpha\}$, but the optimal pipe separates the correct dot.



The reconstructed graph has a correct cycle.

Computing and using persistence

Cloud $\to$ filtration of complexes $\to$ persistence

Obstacle: a big number of simplices $u = O(n^k)$ in dimension k for n points in a given cloud C.

Faster: a near linear time in dimension $k = 0$, approximate persistence $u = O(n)$ for $k > 0$.

Pipeline: t-SNE reduces dimension to $m \approx 4$ preserving geometry, TDA approximates a 1D skeleton for a further optimisation/visualisation.

Summary: TDA needs Statistics

- TDA quantifies geometric properties of *topological features* (cycles, holes, voids)
- the persistence diagram is stable under any bounded noise in unorganised data
- $\mathrm{HoPeS}(C)$ is a *1D persistent structure* giving a provably correct reconstruction of a graph

Wanted : *statistics expertise* and open minds including PhDs and postdocs with C++ skills.