DSC 214
Topological Data Analysis

Topic 7: Persistence in Practice

Instructor: Zhengchao Wan

Persistence-based Framework

- Persistence-based feature representation

Input data:
A collection of point sets, graphs, 2D/3D images, time-series , etc

→

Persistence feature space:
A collection of persistence-based feature representations (e.g, vectors)

→

Further data analysis:
Feed to data analysis pipelines, say SVM, CNN, etc

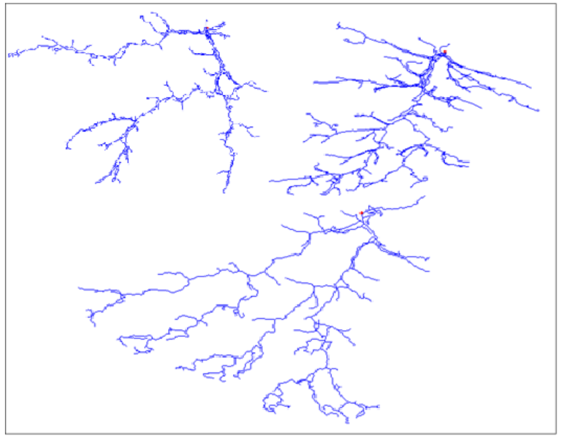
Persistence-based Framework

- Persistence-based feature representation

Input data:
A collection of point sets, graphs, 2D/3D images, time-series , etc

Persistence feature space:
A collection of persistence-based feature representations (e.g, vectors)

Further data analysis:
Feed to data analysis pipelines, say SVM, CNN, etc

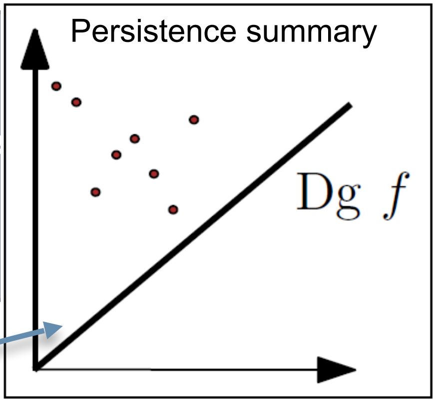


[Li, et al, PLOS One 2017]

Persistence-based Framework

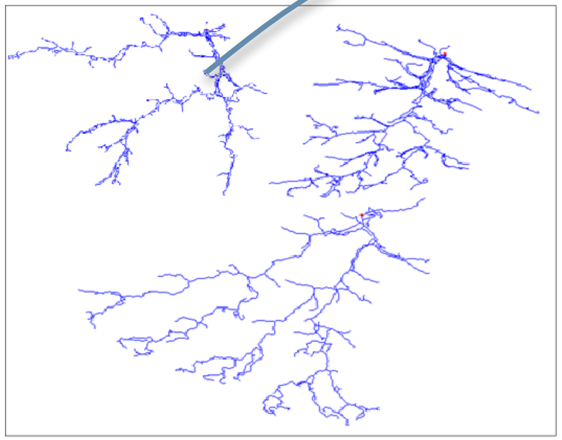
- Persistence-based feature representation

Input data:
A collection of point se
graphs, 2D/3D image
time-series , etc



nce feature
ace:
of persistence-
l feature
tations (e.g,
ctors)

Further data analysis:
Feed to data analysis
pipelines, say SVM,
CNN, etc

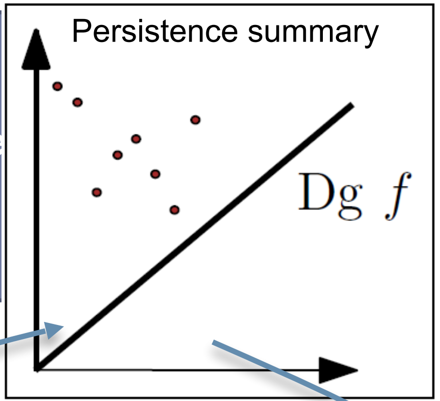


[Li, et al, PLOS One 2017]

Persistence-based Framework

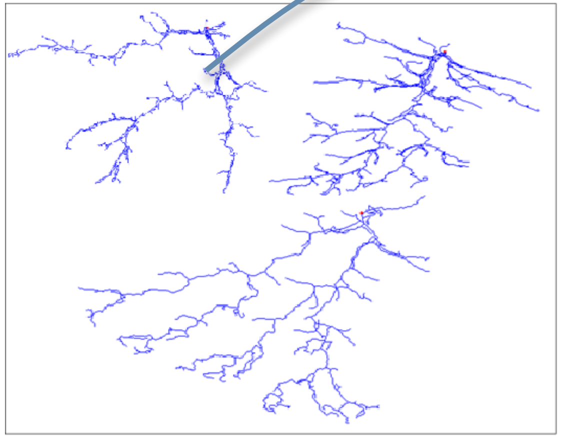
- Persistence-based feature representation

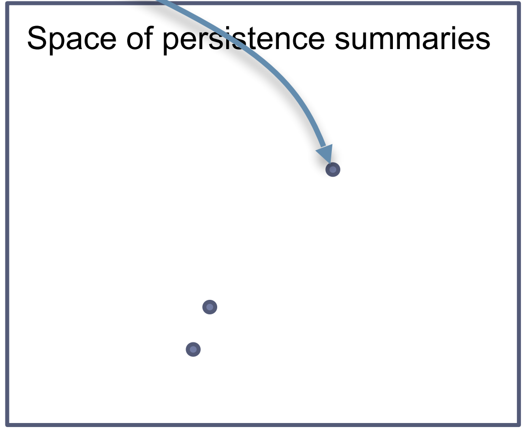
Input data:
A collection of point se
graphs, 2D/3D image
time-series , etc



nce feature
ace:
of persistence-
l feature
tations (e.g,
ctors)

Further data analysis:
Feed to data analysis pipelines, say SVM, CNN, etc





[Li, et al, PLOS One 2017]

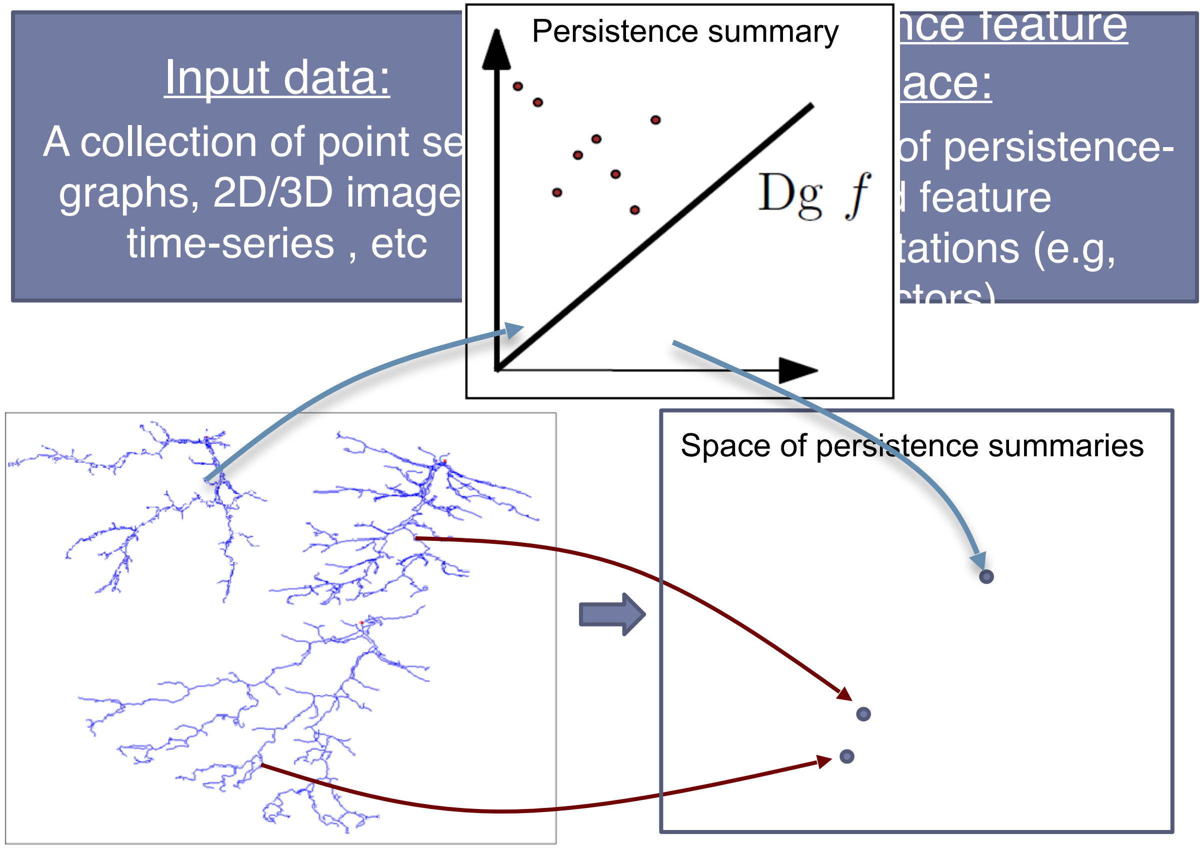
Persistence-based Framework

- Persistence-based feature representation

Input data:
A collection of point se
graphs, 2D/3D image
time-series , etc

nce feature
ace:
of persistence-
l feature
tations (e.g,
ctors)

Further data analysis:
Feed to data analysis pipelines, say SVM, CNN, etc

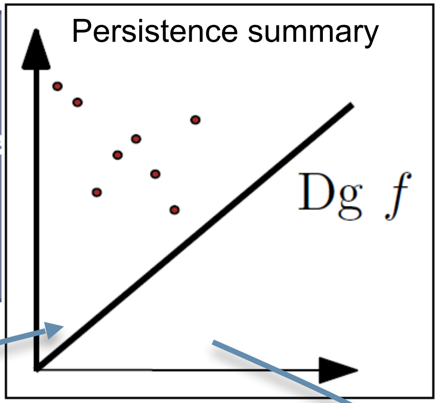


[Li, et al, PLOS One 2017]

Persistence-based Framework

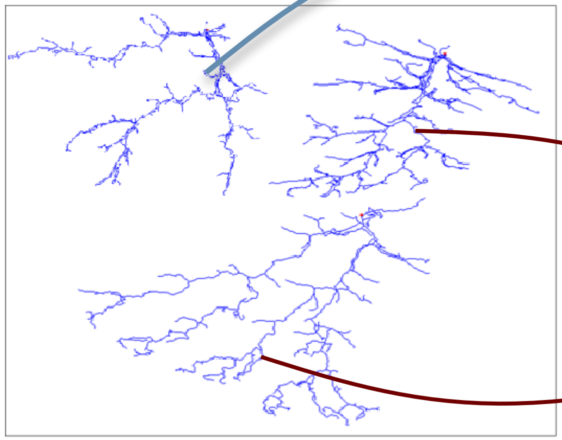
- Persistence-based feature representation

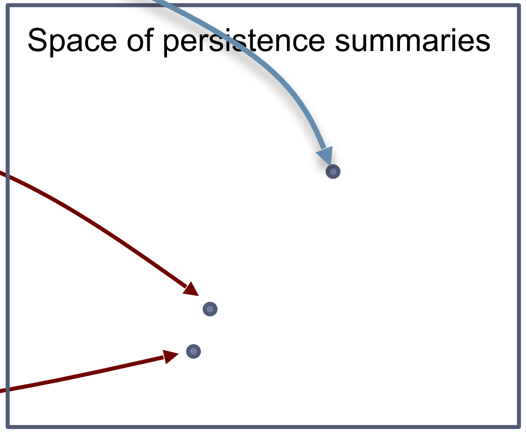
Input data:
A collection of point se
graphs, 2D/3D image
time-series , etc

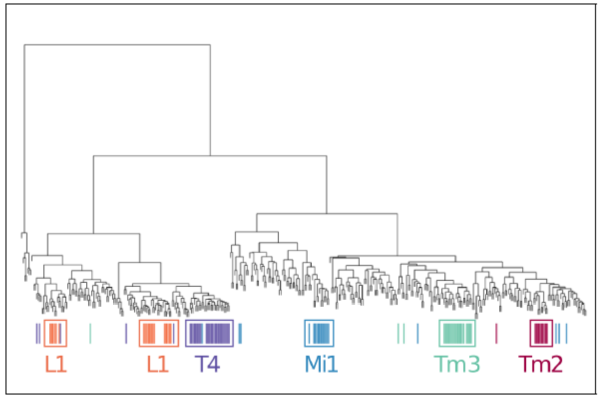


nce feature
ace:
of persistence-
l feature
tations (e.g,
ctors)

Further data analysis:
Feed to data analysis pipelines, say SVM, CNN, etc







[Li, et al, PLOS One 2017]

- Use PH for different types of data

- Vectorization / kernels for persistence diagrams

Section 1 :
Use PH summaries for different types of data

Point clouds data

- Use Rips or Cech filtrations
  - as approximation of distance field filtration

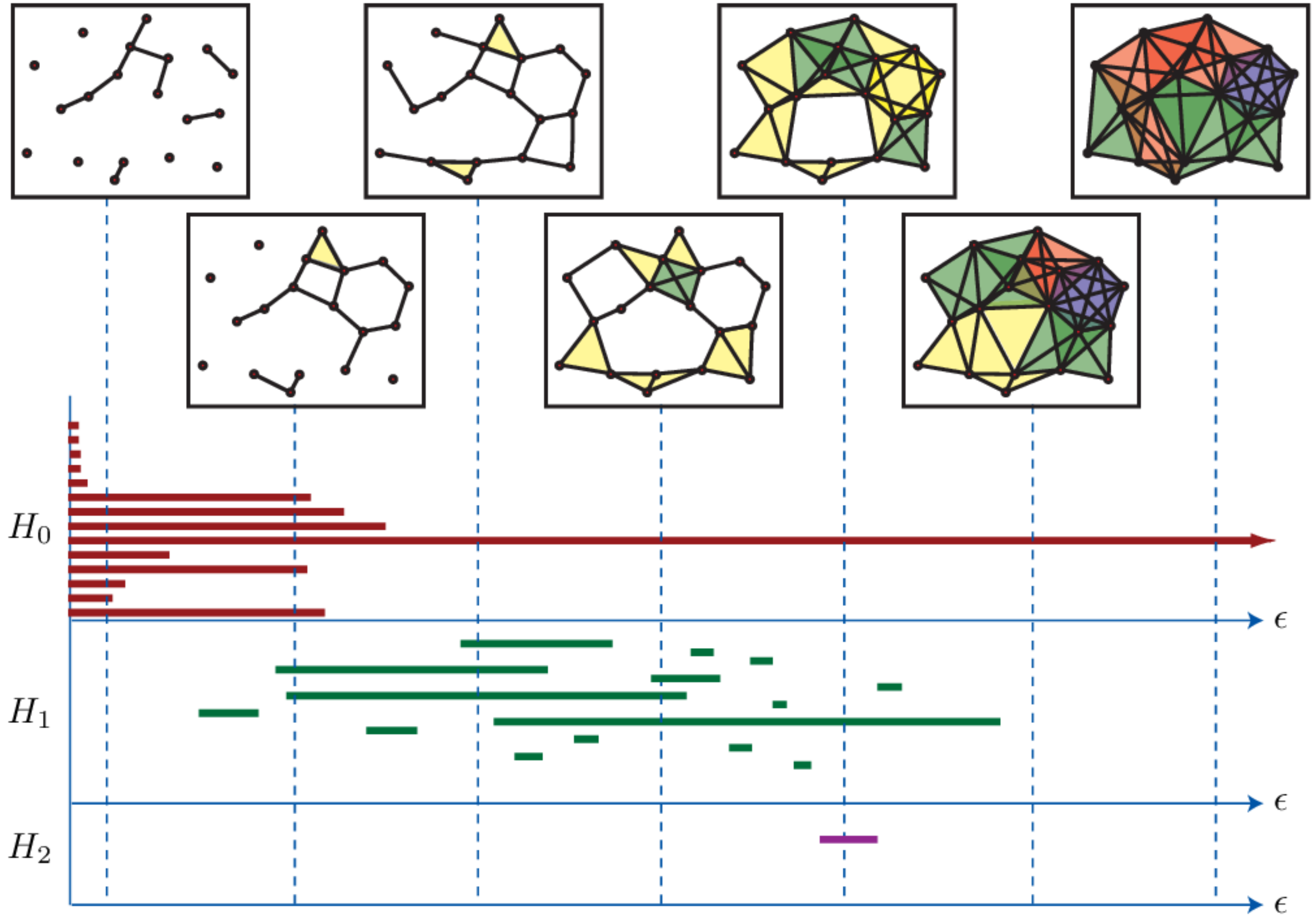


Image courtesy of [Ghrist 05]

General metric spaces

- Input is $P \subseteq (Z,\ d_Z)$, or a special case is that we are given a discrete metric space $(P,\ d_P)$

General metric spaces

$$\mathbb{C}^r_P(P) = \{\sigma = \{p_0, \ldots, p_k\} \mid \cap_{i\in[0,k]} B_P(p; r) \neq \emptyset\}$$

$$\mathbb{VR}^r(P) = \{\sigma = \{p_0, \ldots, p_k\} \mid \mathsf{d}_Z(p_i, p_j) \leq 2r \text{ for any } i, j \in [0, k]\}.$$

General metric spaces

- Input is $P \subseteq (Z,\ d_Z)$, or a special case is that we are given a discrete metric space $(P,\ d_P)$
- Intrinsic Čech complex:

$$\mathbb{C}_P^r(P) = \{\sigma = \{p_0, \ldots, p_k\} \mid \cap_{i\in[0,k]} B_P(p;r) \neq \emptyset\}$$

- Intrinsic Vietoris-Rips complex:

$$\mathbb{VR}^r(P) = \{\sigma = \{p_0, \ldots, p_k\} \mid \mathsf{d}_Z(p_i, p_j) \leq 2r \text{ for any } i, j \in [0, k]\}.$$

- We then can use the persistence diagram induced by intrinsic Cech filtration or intrinsic Rips filtration (for increasing r's)

Hausdorff distance, and Gromov-Hausdorff distance

- Hausdorff distance between two sets $A,\ B \subset \left(Z,\ d_Z\right)$
  - $d_H(A,\ B) = \max\{\max_{a\in A}\min_{b\in B} d_Z(a, b), \max_{b\in B}\min_{a\in A} d_Z(a, b)\}$

Hausdorff distance, and Gromov-Hausdorff distance

- Hausdorff distance between two sets $A,\ B \subset \left(Z,\ d_Z\right)$
  - $d_H(A,\ B) = \max\{\max_{a\in A}\min_{b\in B} d_Z(a, b), \max_{b\in B}\min_{a\in A} d_Z(a, b)\}$

Definition 6.3 (Gromov-Hausdorff distance). Given two metric spaces (X, d_X) and (Y, d_Y), a *correspondence* C is a subset $C \subseteq X \times Y$ so that (i) for every $x \in X$, there exists some $(x, y) \in C$; and (ii) for every $y' \in Y$, there exists some $(x', y') \in C$. The *distortion induced by* C is

$$\mathrm{distort}_C(X, Y) := \frac{1}{2} \sup_{(x,y),(x',y')\in C} |\mathsf{d}_X(x, x') - \mathsf{d}_Y(y, y')|.$$

The *Gromov-Hausdorff distance between* (X, d_X) *and* (Y, d_Y) is the smallest distortion possible by any correspondence; that is,

$$\mathsf{d}_{GH}(X, Y) := \inf_{C\subseteq X\times Y} \mathrm{distort}_C(X, Y).$$

Stability

Theorem 6.3. *Čech- and Rips-distances satisfy the following stability statements:*

1. *Given two finite sets* $P, Q \subseteq (Z, \mathsf{d}_Z)$, *we have*

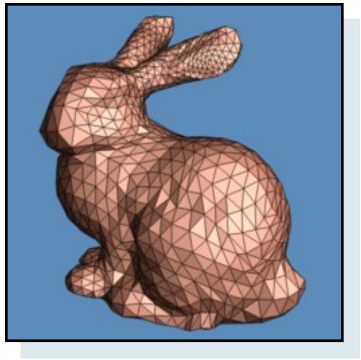
$$\mathsf{d}_{Cech}(P, Q) \leq \mathsf{d}_H(P, Q); \quad \textit{and} \quad \mathsf{d}_{Rips}(P, Q) \leq \mathsf{d}_H(P, Q).$$

2. *Given two finite metric spaces* (P, d_P) *and* (Q, d_Q), *we have*

$$\mathsf{d}_{Cech}(P, Q) \leq 2\mathsf{d}_{GH}((P, \mathsf{d}_P), (Q, \mathsf{d}_Q)), \quad \textit{and} \ \mathsf{d}_{Rips}(P, Q) \leq \mathsf{d}_{GH}((P, \mathsf{d}_P), (Q, \mathsf{d}_Q)).$$

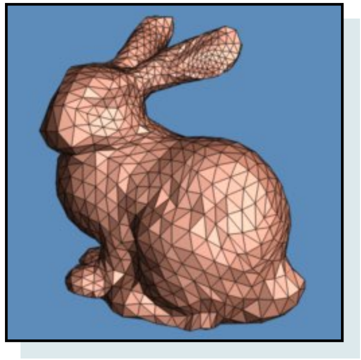
Mesh data

- For example, a triangulation K of surface models



Mesh data

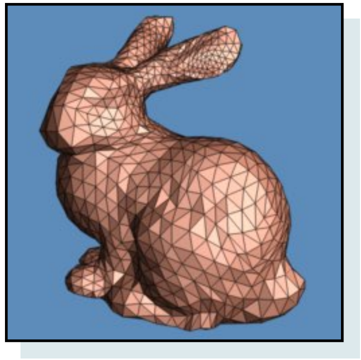
- For example, a triangulation K of surface models
- Idea:
  - Find a meaningful descriptor function $f: K \to R$
  - Compute $dgm_* f$ as its topological summary



Mesh data

- For example, a triangulation K of surface models
- Idea:
  - Find a meaningful descriptor function $f: K \to R$
  - Compute $dgm_* f$ as its topological summary

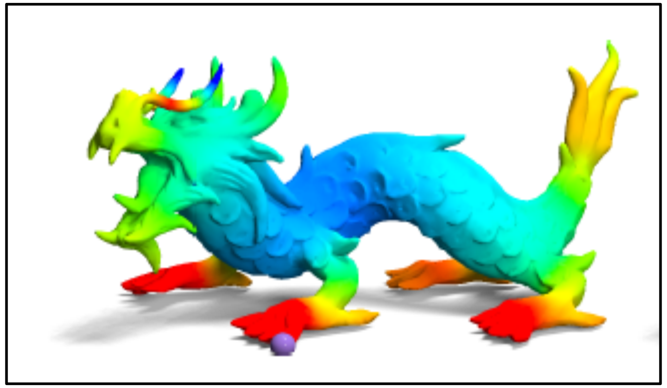
- Examples of choices of descriptor functions:
  - Gaussian curvature
  - Mean curvature
  - Heat-kernel signature function
  - …

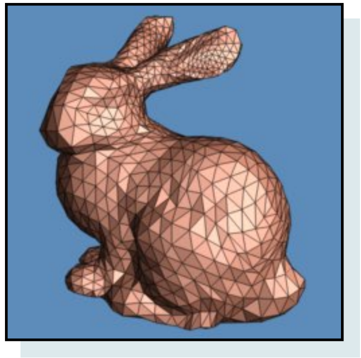


Mesh data

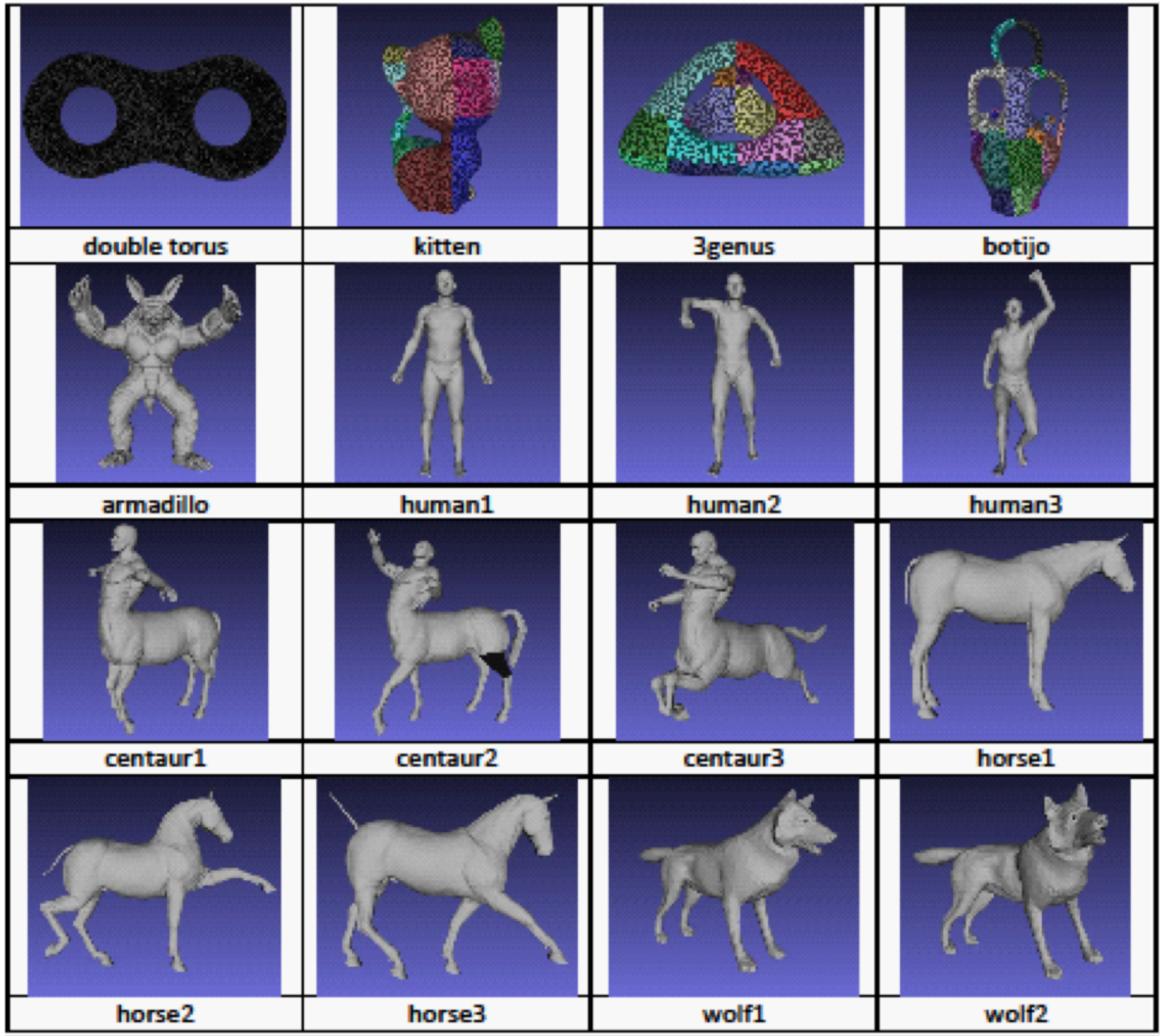
- For example, a triangulation K of surface models
- Idea:
  - Find a meaningful descriptor function $f: K \to R$
  - Compute $dgm_* f$ as its topological summary

- Examples of choices of descriptor functions:
  - Gaussian curvature
  - Mean curvature
  - Heat-kernel signature function
  - …





Example


double torus
kitten
3genus
botijo
armadillo
human1
human2
human3
centaur1
centaur2
centaur3
horse1
horse2
horse3
wolf1
wolf2

PD Distance	double torus	kitten	3genus	botijo	armadillo	human1	human2	human3	centaur1	centaur2	centaur3	horse1	horse2	horse3	wolf1	wolf2
double torus	**0.024**	0.235	0.299	0.264	0.307	0.364	0.358	0.340	0.269	0.253	0.245	0.247	0.240	0.250	0.248	0.243
kitten	0.235	**0.037**	0.319	0.268	0.259	0.274	0.268	0.250	0.266	0.249	0.245	0.241	0.242	0.239	0.238	0.238
3genus	0.299	0.319	**0.047**	0.256	0.394	0.388	0.393	0.394	0.394	0.392	0.399	0.401	0.399	0.392	0.386	0.401
botijo	0.264	0.268	0.256	**0.039**	0.276	0.276	0.277	0.278	0.279	0.279	0.277	0.278	0.279	0.278	0.278	0.280
armadillo	0.307	0.259	0.394	0.276	**0.049**	0.159	0.203	0.190	0.127	0.122	0.152	0.114	0.108	0.114	0.125	0.127
human1	0.364	0.274	0.388	0.276	0.159	**0.038**	0.052	0.078	0.207	0.189	0.183	0.195	0.190	0.205	0.176	0.165
human2	0.358	0.268	0.393	0.277	0.203	0.052	**0.038**	0.061	0.213	0.189	0.181	0.200	0.189	0.204	0.176	0.170
human3	0.340	0.250	0.394	0.278	0.190	0.078	0.061	**0.041**	0.210	0.189	0.184	0.199	0.188	0.197	0.176	0.166
centaur1	0.269	0.266	0.394	0.279	0.127	0.207	0.213	0.210	**0.044**	0.057	0.102	0.142	0.161	0.126	0.162	0.180
centaur2	0.253	0.249	0.392	0.279	0.122	0.189	0.189	0.189	0.057	**0.049**	0.085	0.142	0.133	0.117	0.160	0.160
centaur3	0.245	0.245	0.399	0.277	0.152	0.183	0.181	0.184	0.102	0.085	**0.042**	0.127	0.132	0.118	0.139	0.152
horse1	0.247	0.241	0.401	0.278	0.114	0.195	0.200	0.199	0.142	0.142	0.127	**0.042**	0.097	0.079	0.122	0.111
horse2	0.240	0.242	0.399	0.279	0.108	0.190	0.189	0.188	0.161	0.133	0.132	0.097	**0.048**	0.069	0.106	0.105
horse3	0.250	0.239	0.392	0.278	0.114	0.205	0.204	0.197	0.126	0.117	0.118	0.079	0.069	**0.046**	0.117	0.111
wolf1	0.248	0.238	0.386	0.278	0.125	0.176	0.176	0.176	0.162	0.160	0.139	0.122	0.106	0.117	**0.045**	0.066
wolf2	0.243	0.238	0.401	0.280	0.127	0.165	0.170	0.166	0.180	0.160	0.152	0.111	0.105	0.111	0.066	**0.045**

pairwise persistence-based distance between models

- Demo on teeth clustering

Graph Data

- There are different ways to look at graphs
  - Combinatorial graph based
  - Geometric graph based
  - Descriptor function based
  - …

Simplicial homology induced from weighted graphs

- Suppose we are given a weighted graph $G = (V, \; E; w)$
  - This gives us a sequence of graphs $G^a := \left\{ e \in E \mid w(e) \leq a \right\}$
- We can construct simplicial complexes from G
- E.g. 1:
  - clique complex $R^a(G) = \{ \left(v_{i_0}, \ldots, \; v_{i_s} \right) \mid \; v_{i_0}, \ldots, \; v_{i_s}$ forms a clique in $G^a\}$
  - Increasing a then gives us a filtration
    - $R^{a_0}(G) \subseteq R^{a_1}(G) \subseteq \cdots \subseteq R^{a_n}(G)$
  - We can then take the persistence diagram of it as its topological summaries
  - This, for example, has been used in the literature to compare brain networks

Simplicial homology induced from weighted graphs

- Suppose we are given a weighted graph $G = (V,\ E; w)$
  - we can treat this as a metric graph, with shortest path distance d_G induced by w
- E.g 2: intrinsic Čech or Rips complexes induced by d_G

Simplicial homology induced from weighted graphs

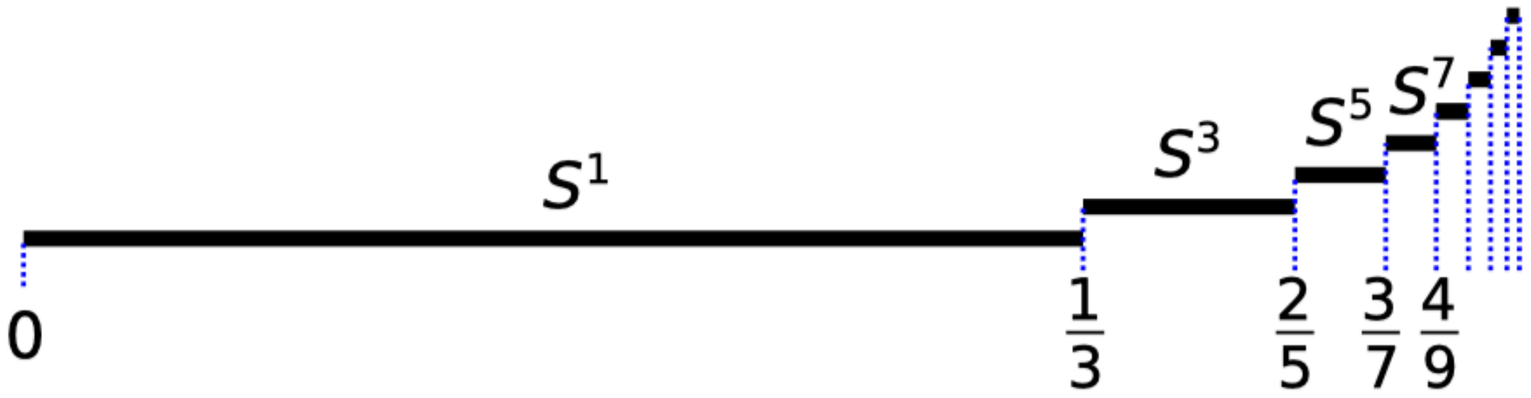
- Suppose we are given a weighted graph $G = (V,\ E; w)$
  - we can treat this as a metric graph, with shortest path distance d_G induced by w
- **Theorem 8.1.** *Let* $0 < r < \frac{1}{2}$. *There are homotopy equivalences: for* $\ell = 0, 1, \ldots,$

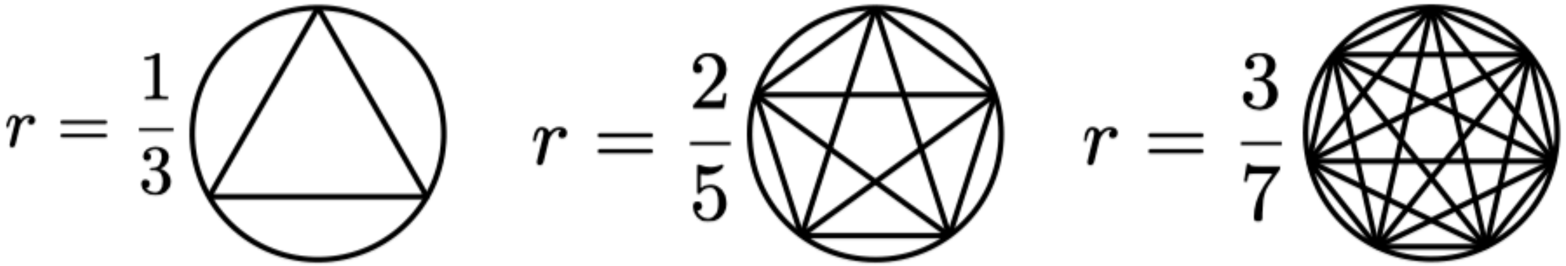
$$\mathbb{C}^r(\mathbb{S}^1) \simeq \mathbb{S}^{2\ell+1} \quad \text{if } \frac{\ell}{2(\ell+1)} < r \leq \frac{\ell+1}{2(\ell+2)}; \quad \text{and}$$

$$\mathbb{VR}^{r/2}(\mathbb{S}^1) \simeq \mathbb{S}^{2\ell+1} \quad \text{if } \frac{\ell}{2\ell+1} < \frac{r}{2} \leq \frac{\ell+1}{2\ell+3}.$$

VR complex of $\mathbb{S}^1$ - circle with unit circumference

- $Rips^r(\mathbb{S}^1) \simeq \mathbb{S}^{2l+1}, \dfrac{l}{2l+1} < r \leq \dfrac{l+1}{2l+3}$





Courtesy of Henry Adams

Simplicial homology induced from weighted graphs

- Suppose we are given a weighted graph $G = (V,\ E; w)$
  - we can treat this as a metric graph, with shortest path distance d_G induced by w

Theorem 8.3. *Let $G = (V, E, \omega)$ be a finite graph with positive weight function $\omega : E \to \mathbb{R}$. Let $\{\gamma_1, \ldots, \gamma_g\}$ be a shortest cycle basis of G where $g = \mathrm{rank}\,(\mathsf{Z}_1(G))$, and for each $i = 1, \ldots, g$, let $\ell_i = length(\gamma_i)$. Then, the 1-st persistence diagram $\mathrm{Dgm}_1\mathcal{C}$ induced by the intrinsic Čech filtration $\mathcal{C} := \{\mathbb{C}^r(|G|)\}_{r\in\mathbb{R}}$ on the metric graph $(|G|, \mathsf{d}_G)$ consists of the following set of points on the y-axis:*

$$\mathrm{Dgm}_1\mathcal{C} = \{(0, \frac{\ell_i}{4}) \mid 1 \le i \le g\}.$$

Descriptor function based

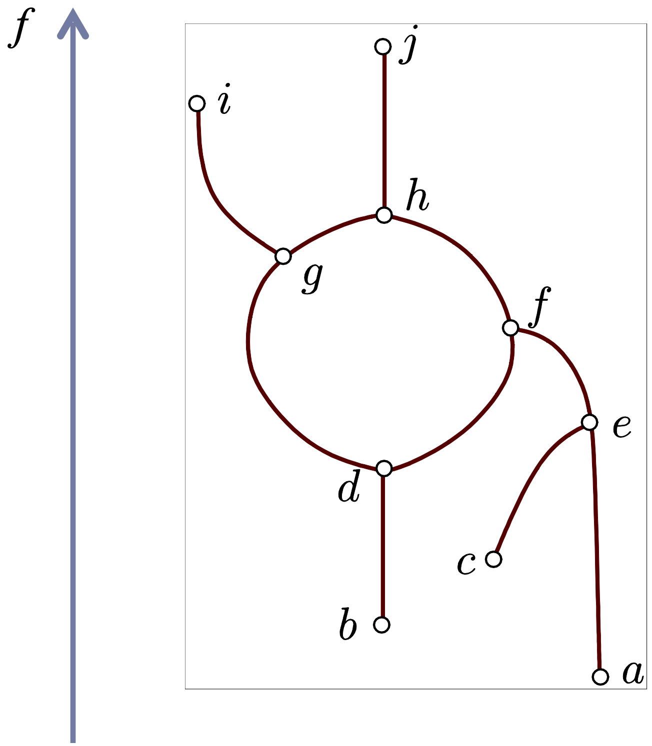
- A graph $G = (V, E)$ can be thought of as a 1-dimensional simplicial complex
- We can thus put a descriptor function on it, and compute its resulting (sub level set) persistence diagram representation

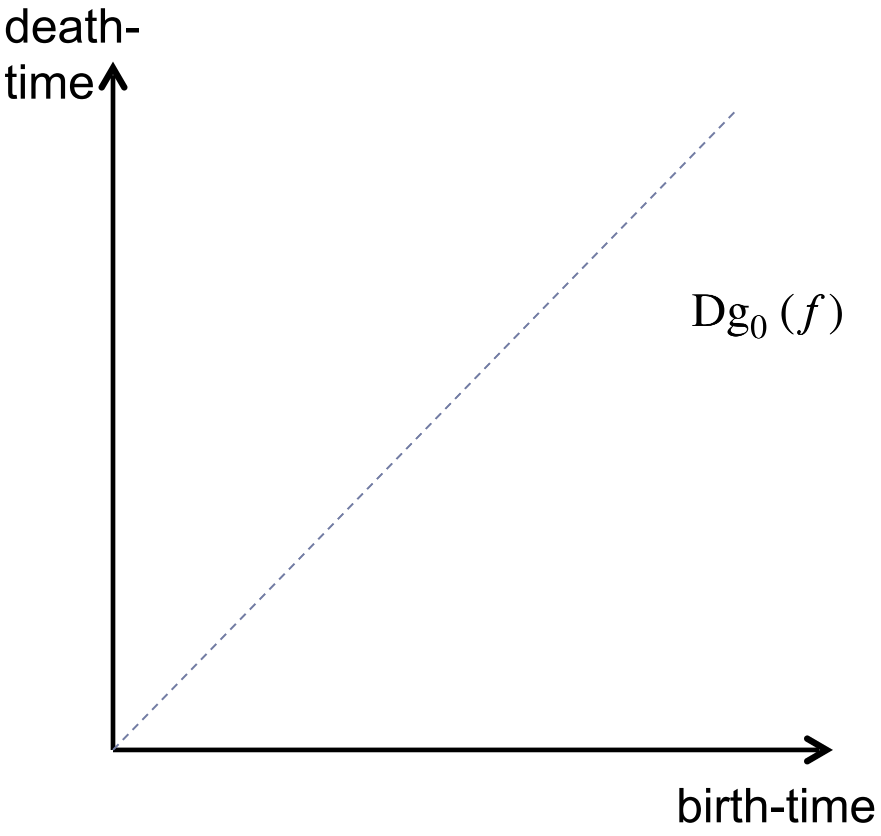
- Some choices of descriptor functions:
  - Discrete Ricci curvature
  - Degree function
  - Heat-kernel signature
  - …

One issue

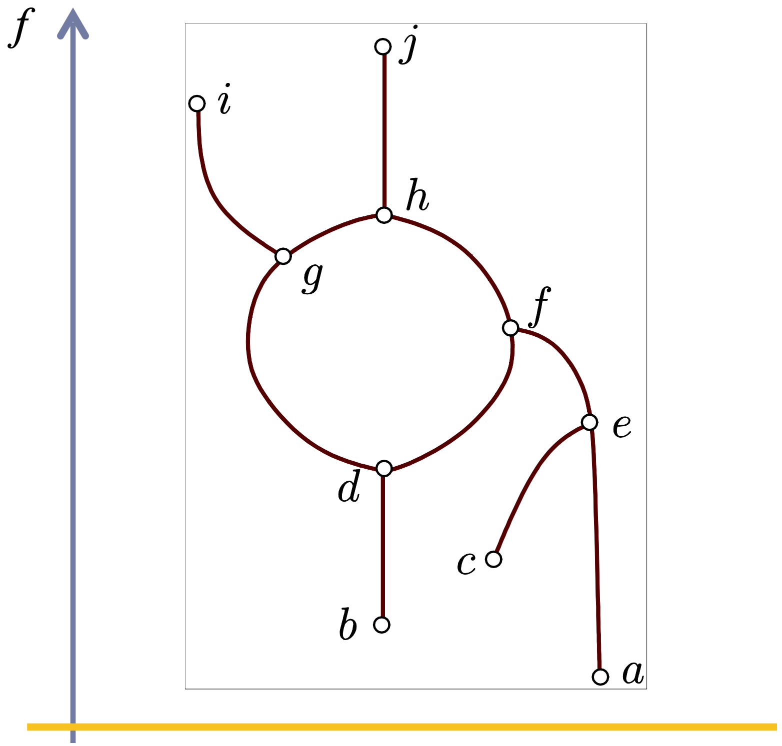
- There is only 0-th persistent homology for sub-level set filtration
- For 1-dim, cycles created will never die (as there are no triangles)
- Example

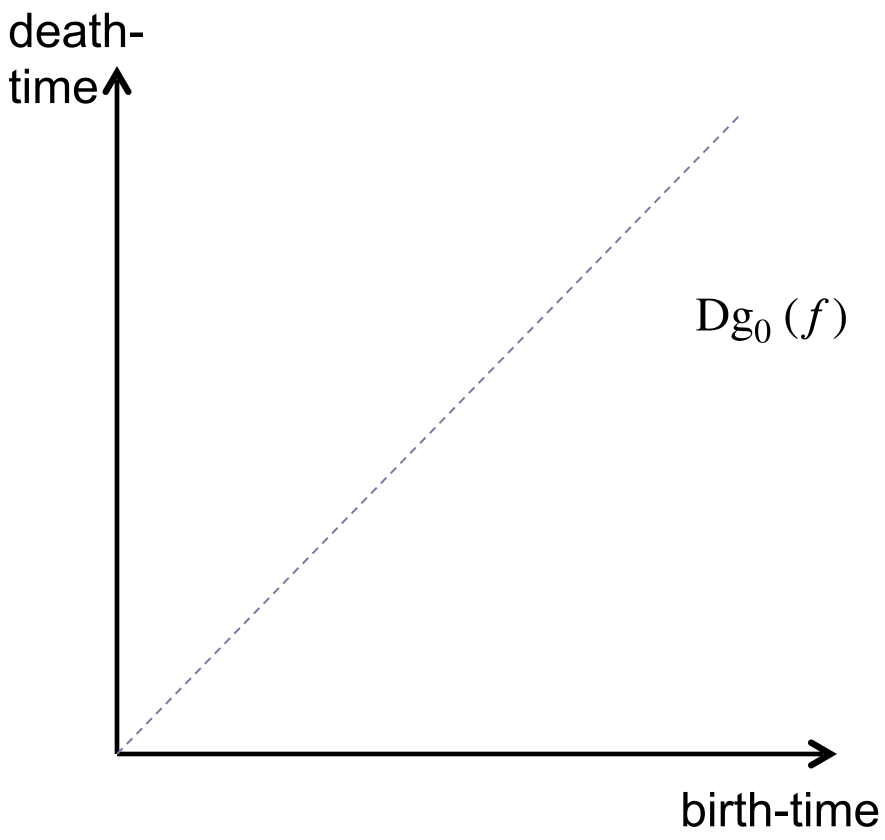
A (Very) Simplified Illustration



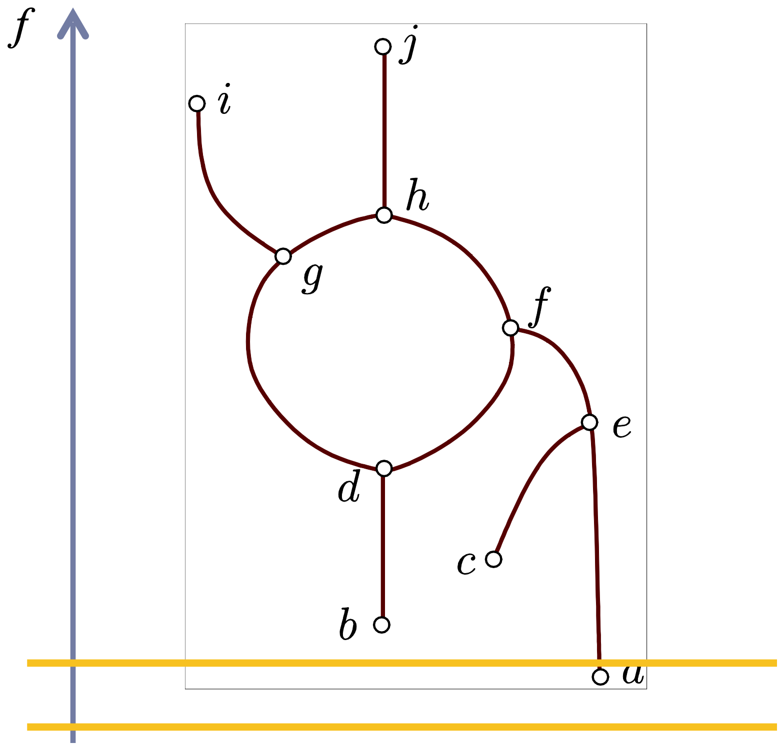


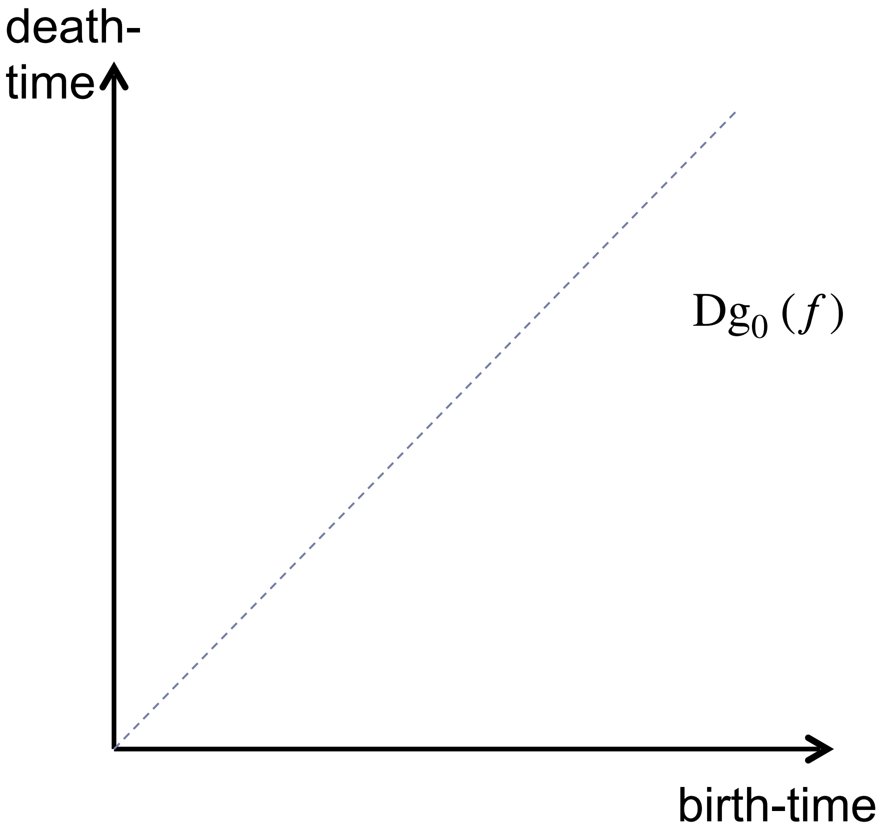
A (Very) Simplified Illustration



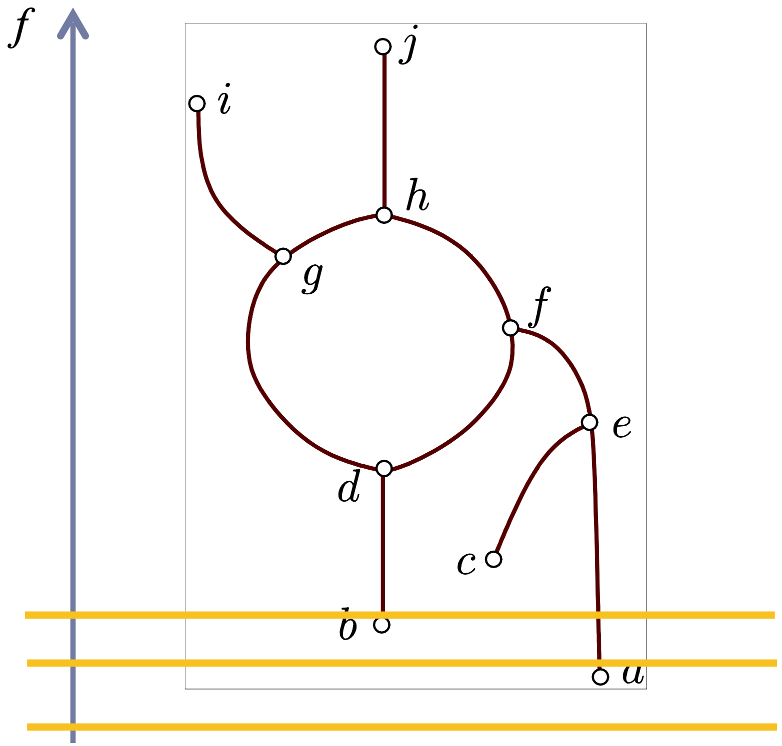


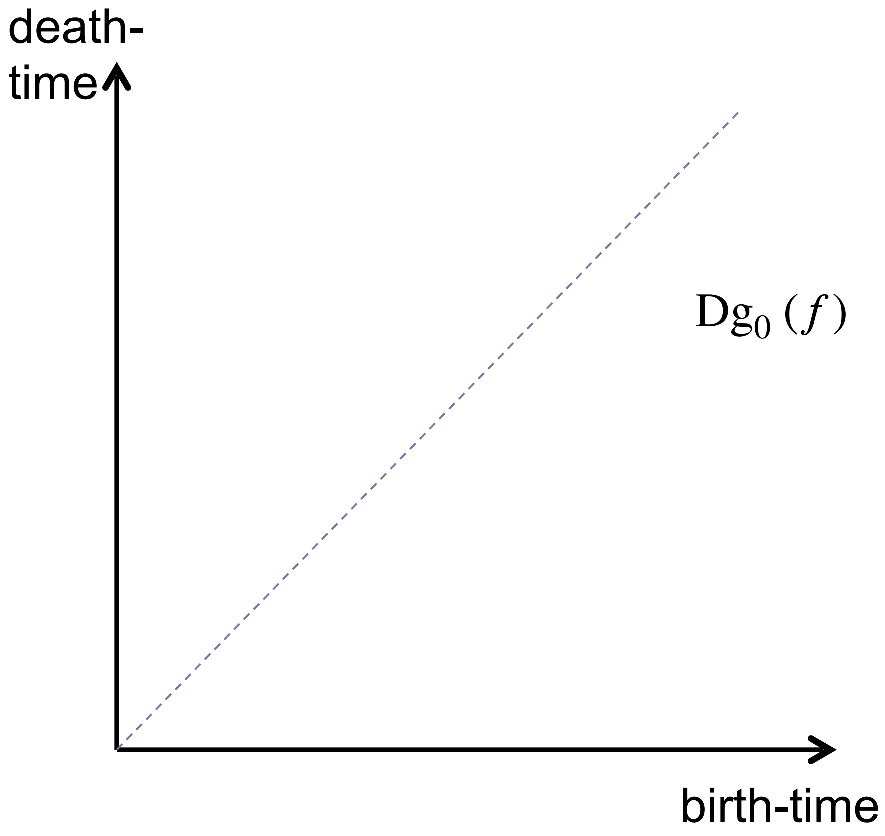
A (Very) Simplified Illustration



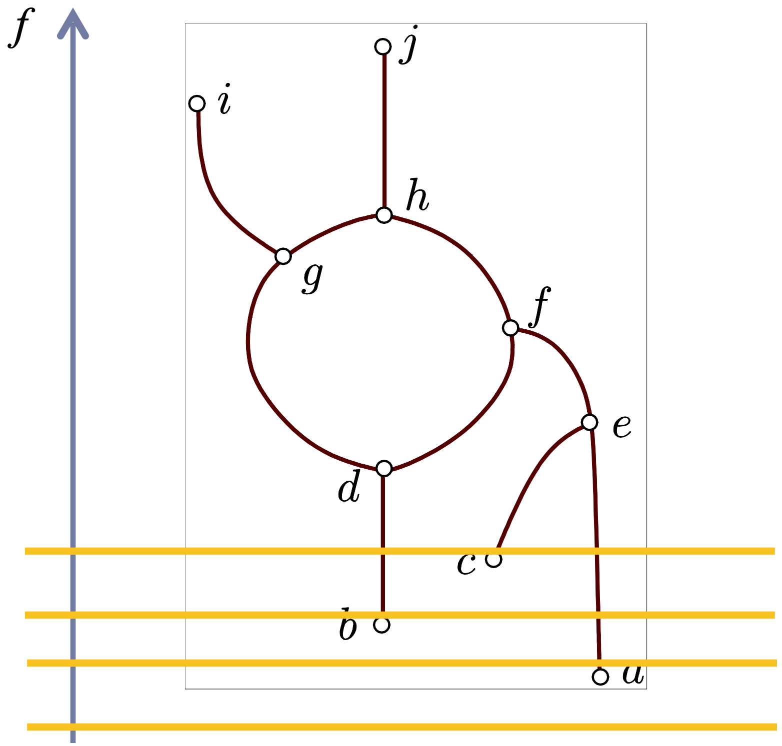


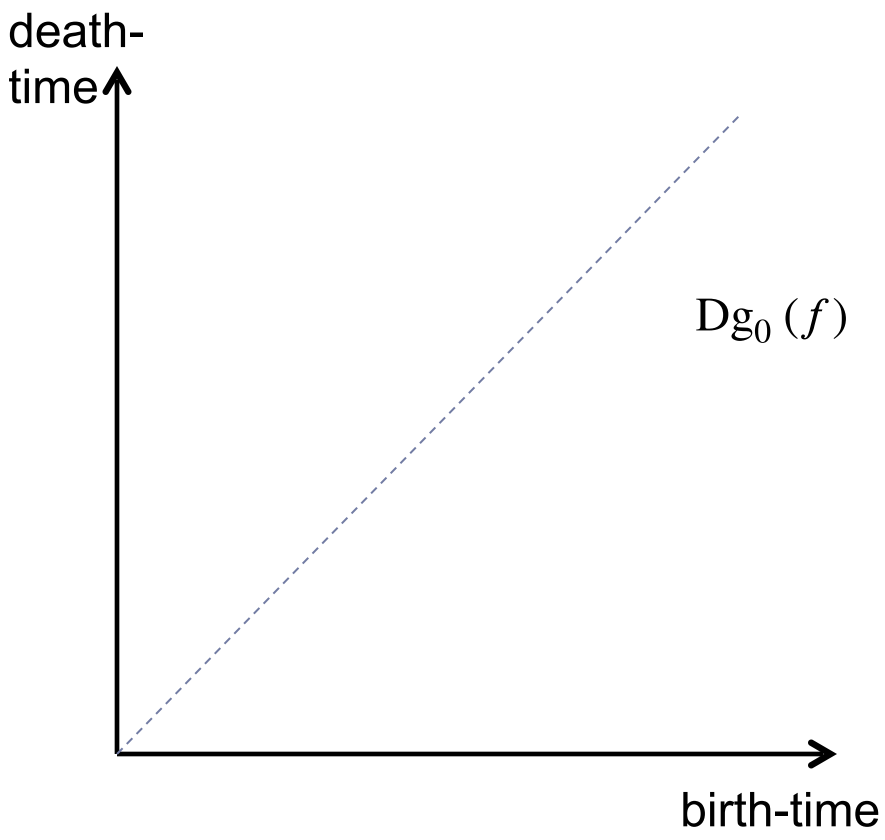
A (Very) Simplified Illustration



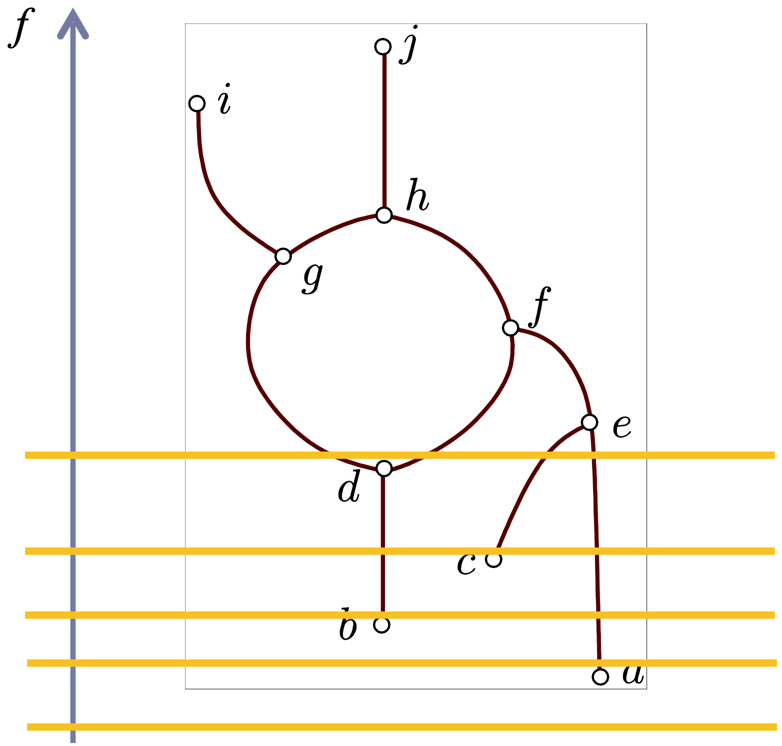


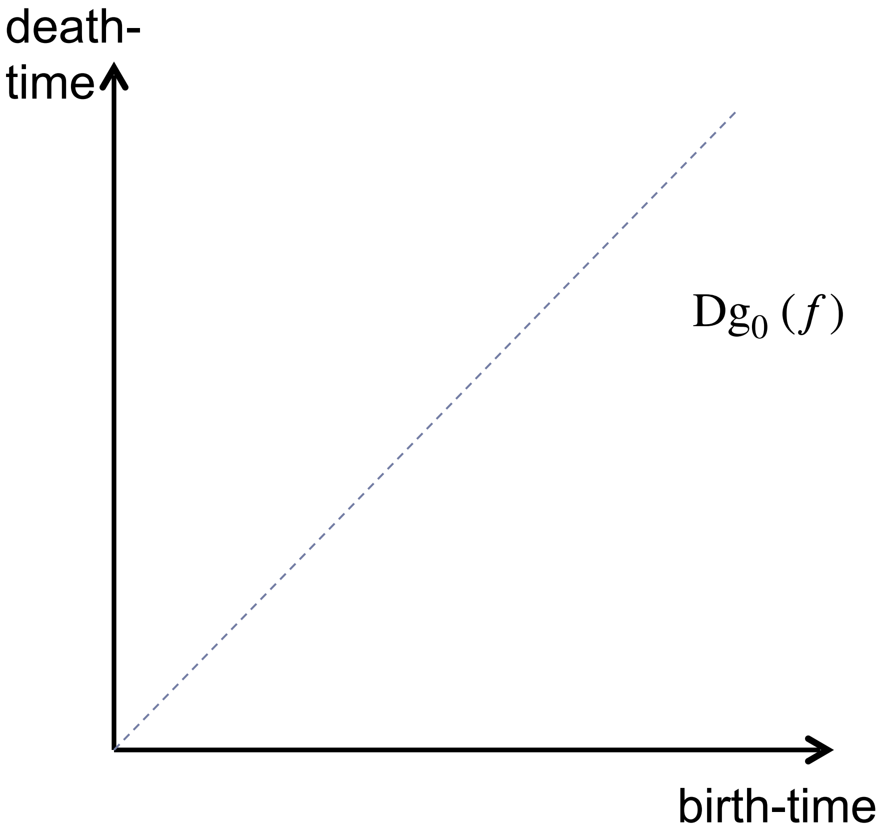
A (Very) Simplified Illustration



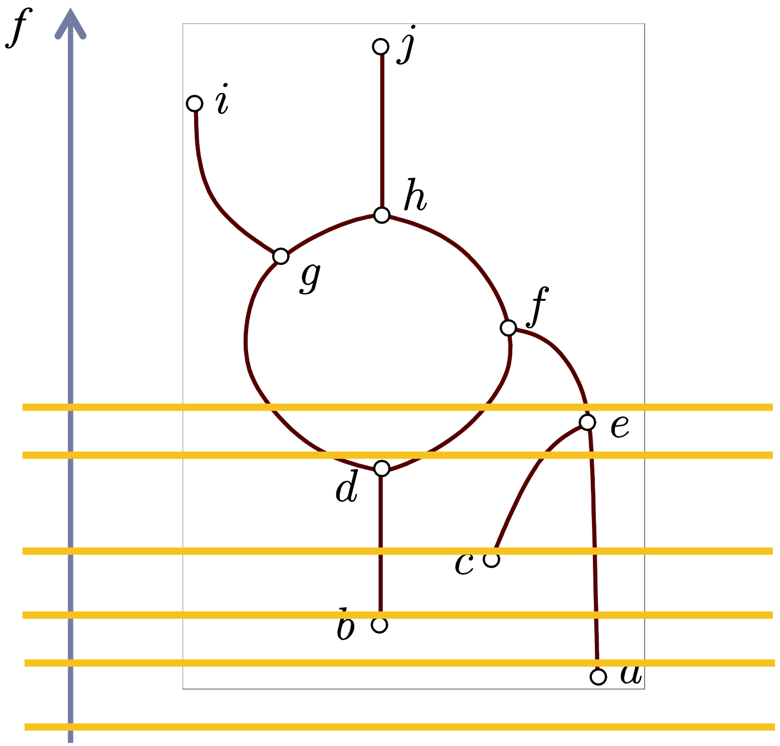


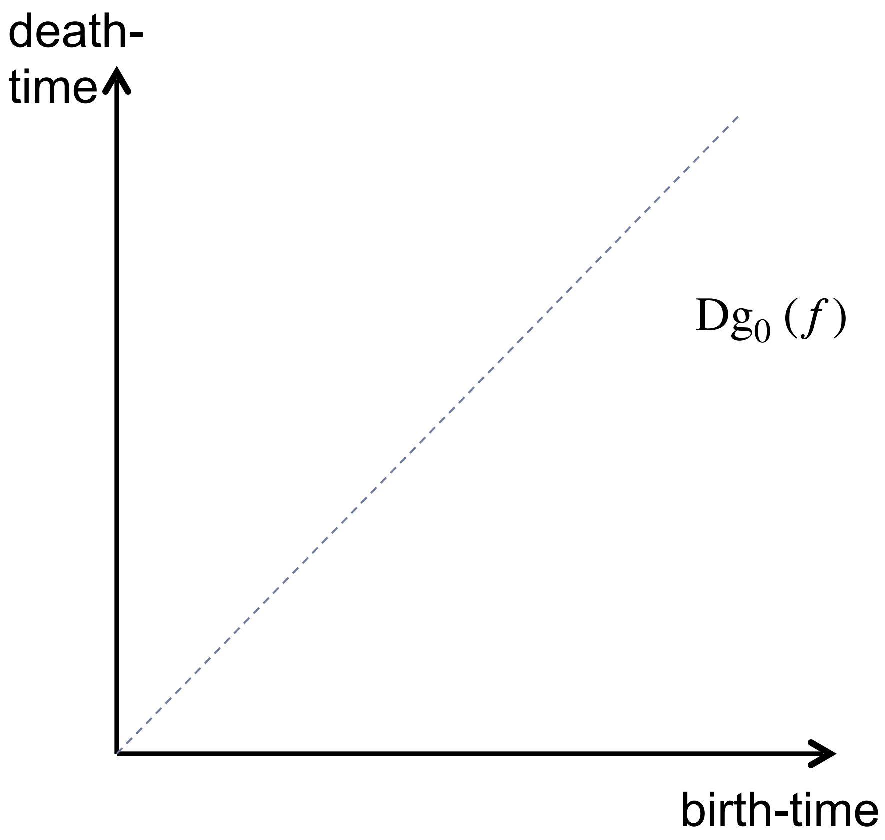
A (Very) Simplified Illustration



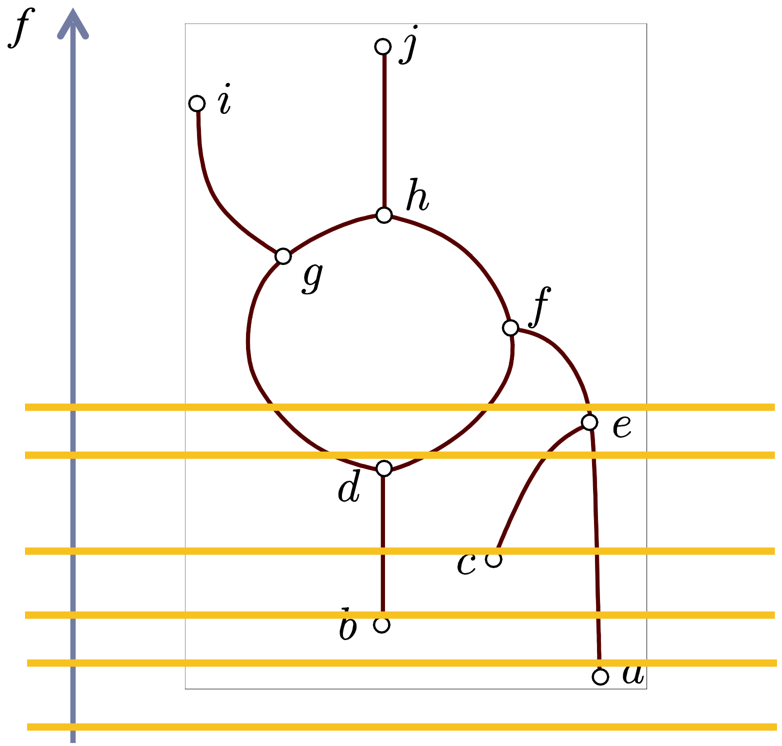


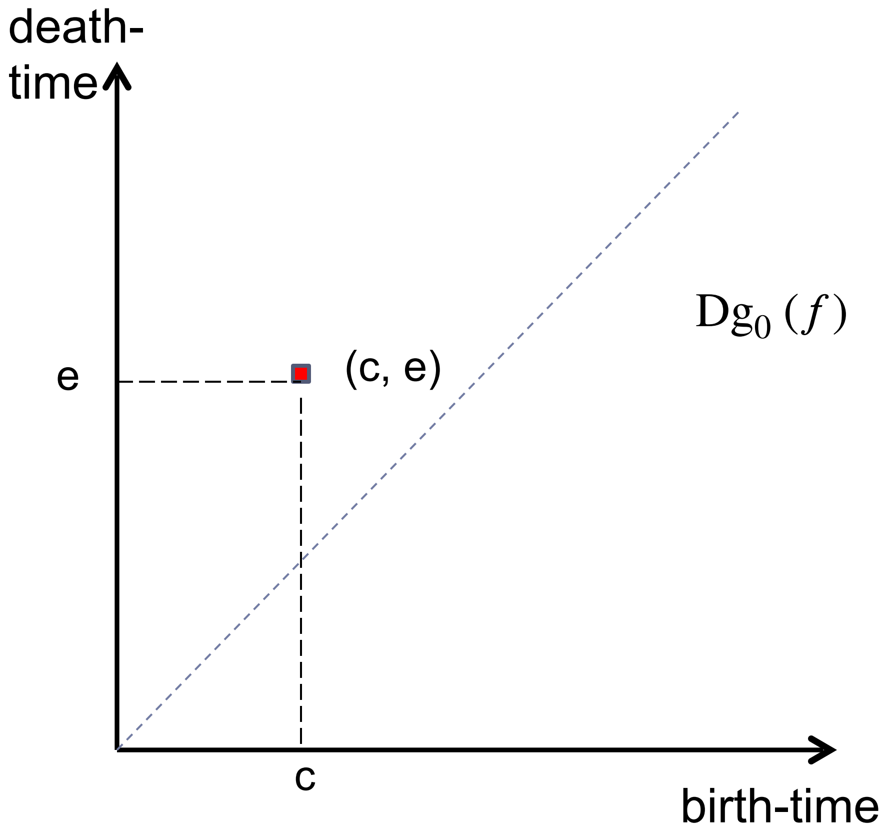
A (Very) Simplified Illustration



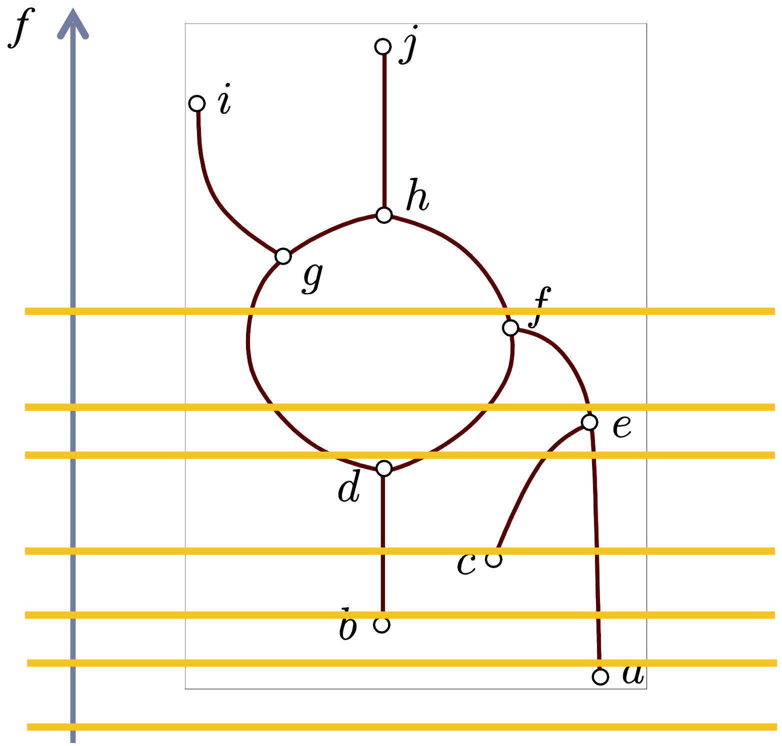


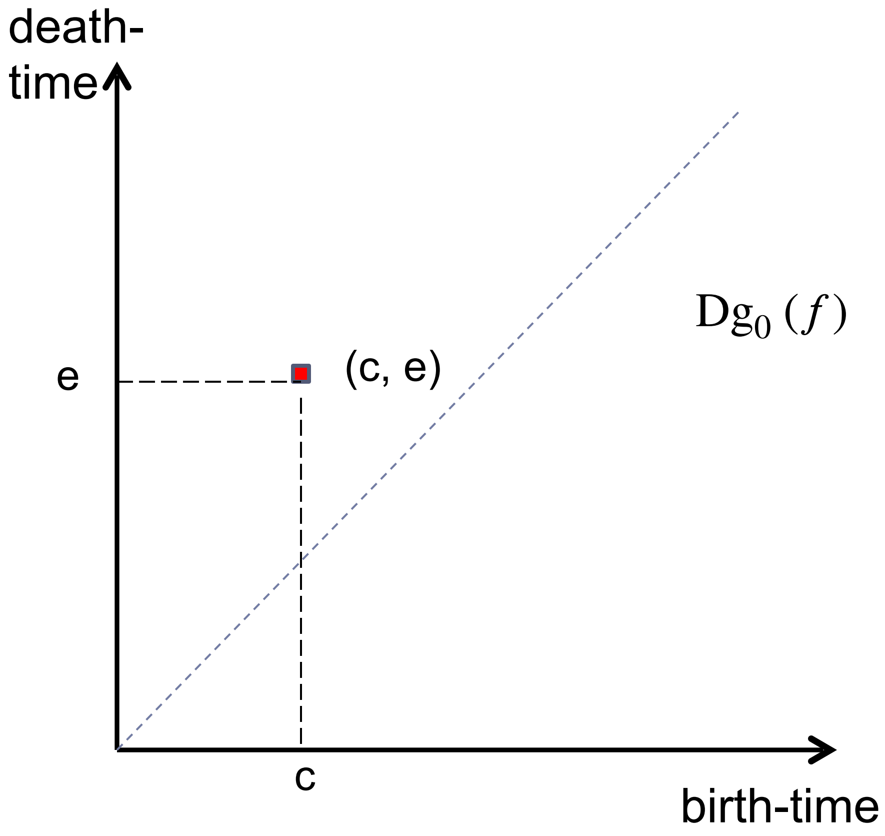
A (Very) Simplified Illustration



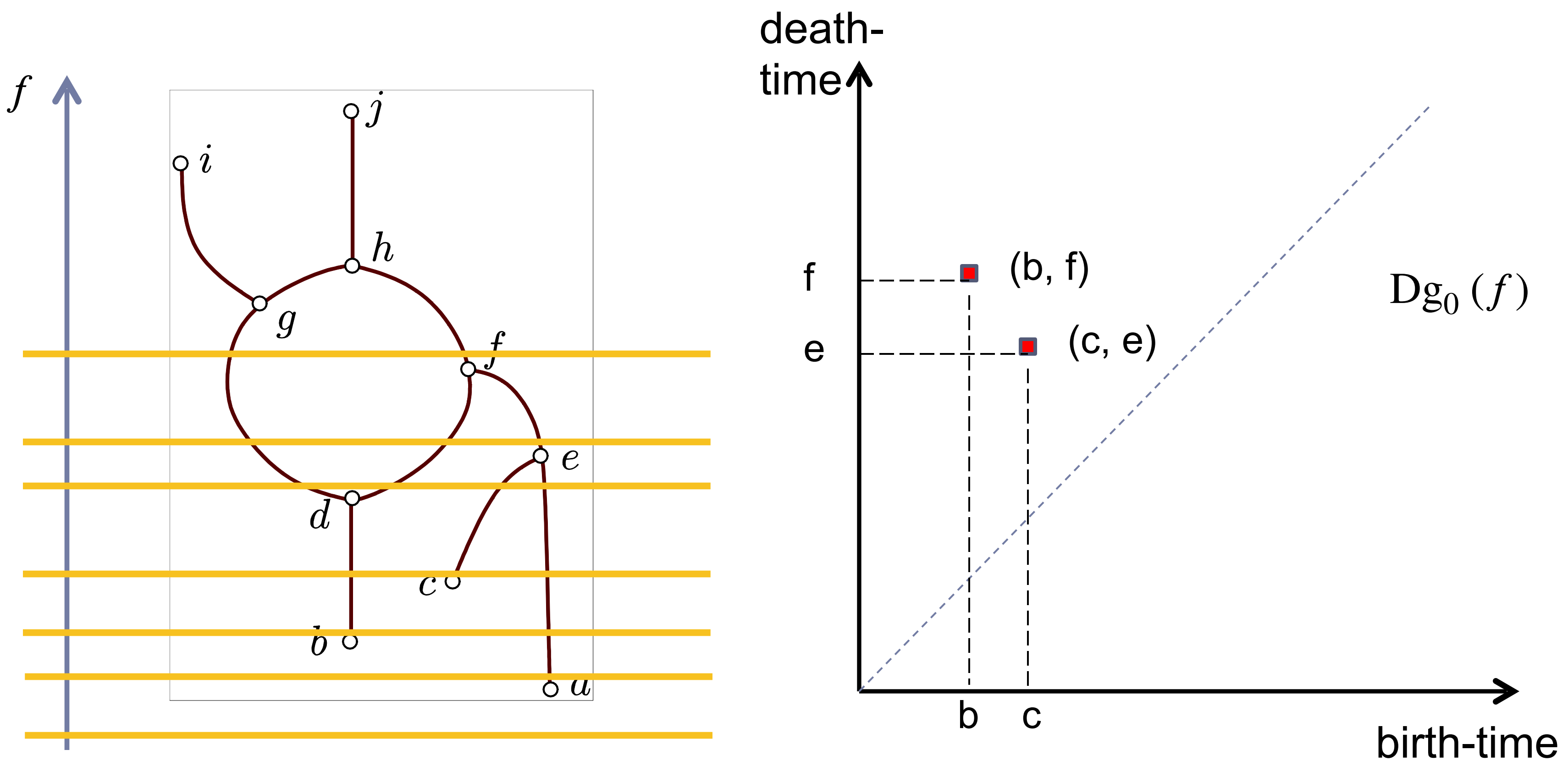


A (Very) Simplified Illustration

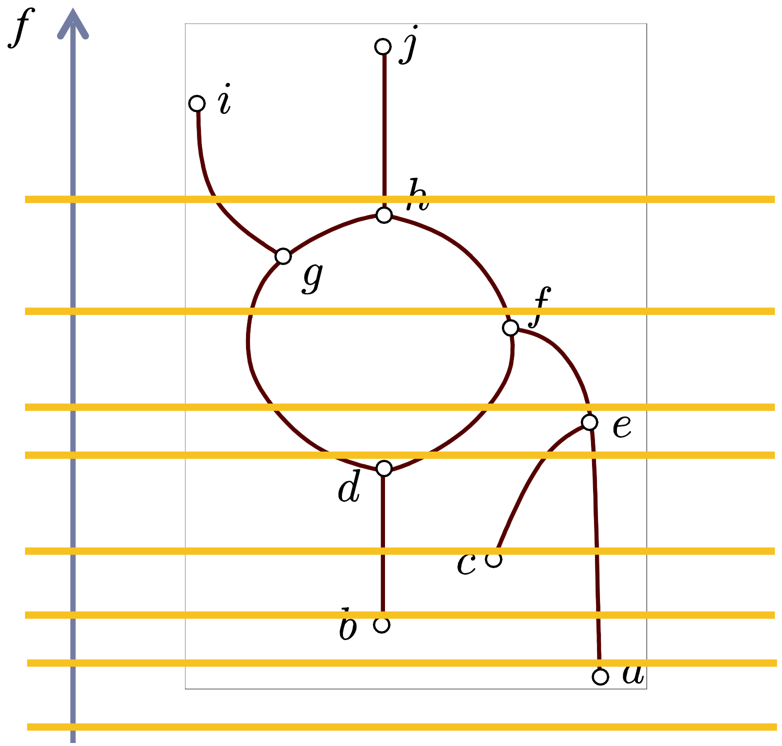


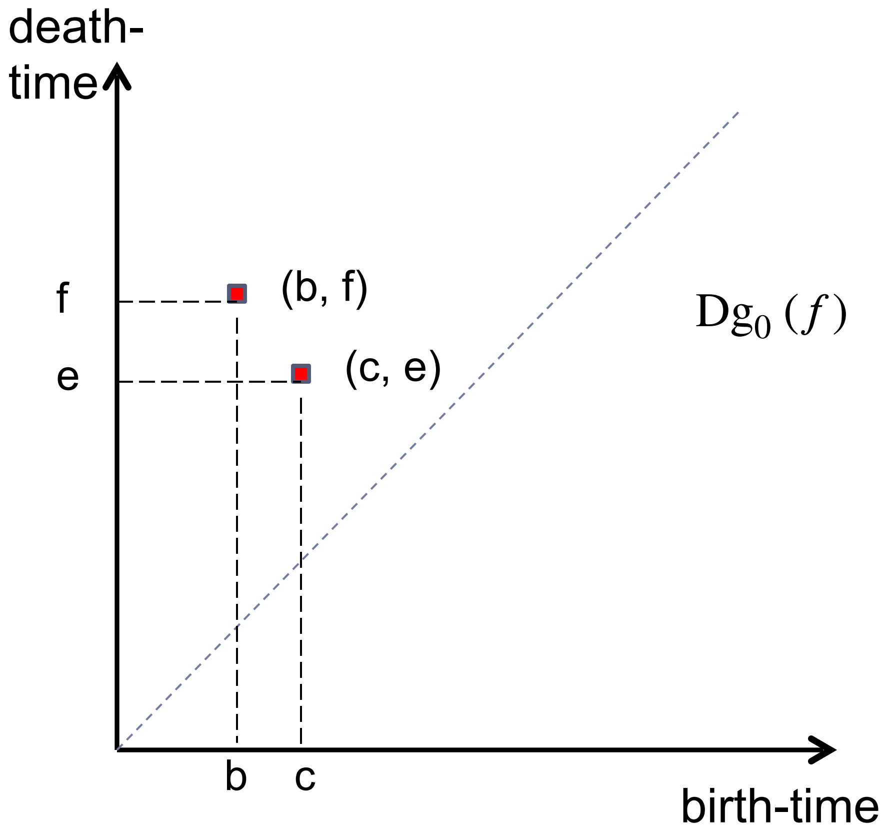


A (Very) Simplified Illustration

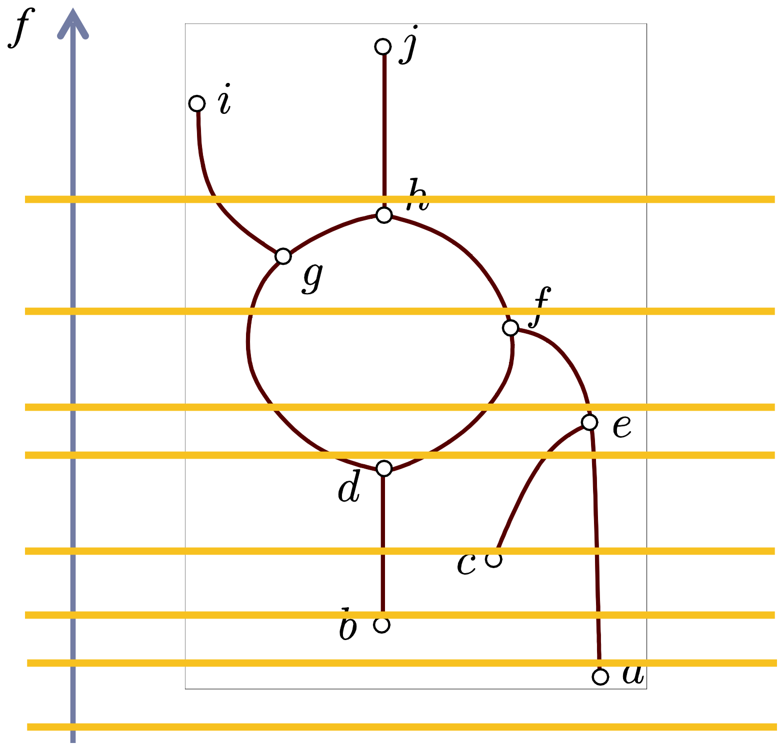


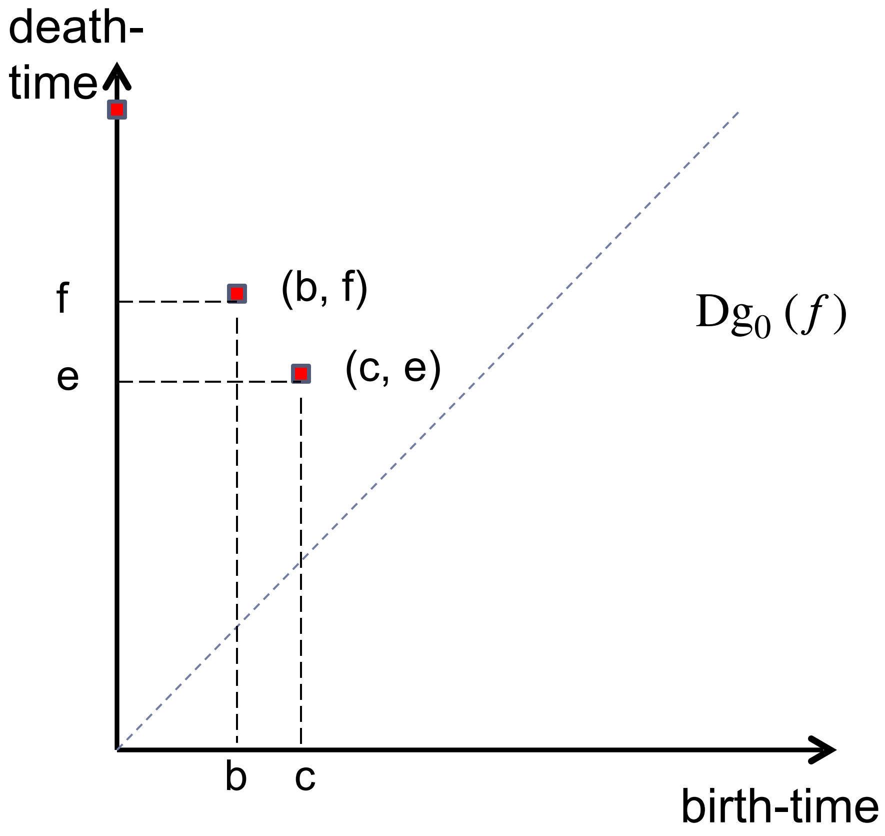
A (Very) Simplified Illustration





A (Very) Simplified Illustration



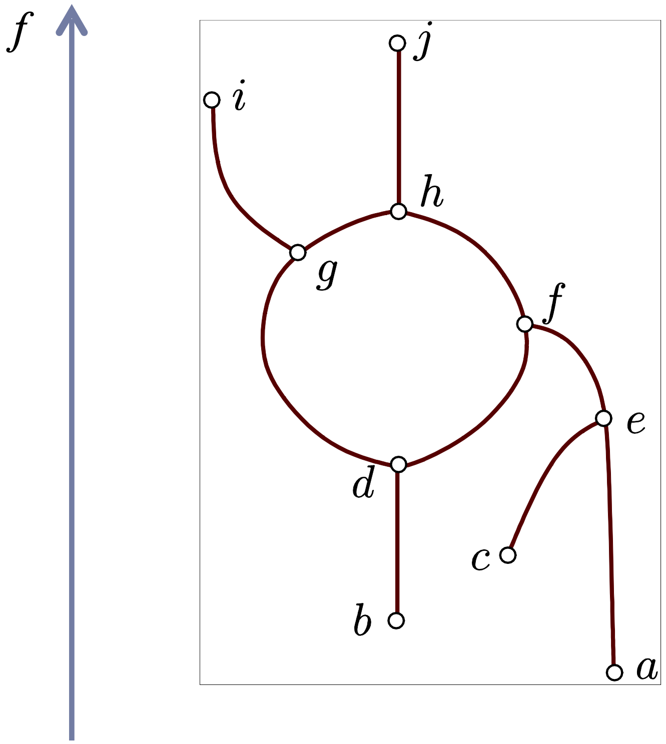


Loop Features

- Captured by Extended persistence:
  - *[Cohen-Steiner, Edelsbrunner and Harer, FoCM 2009]*

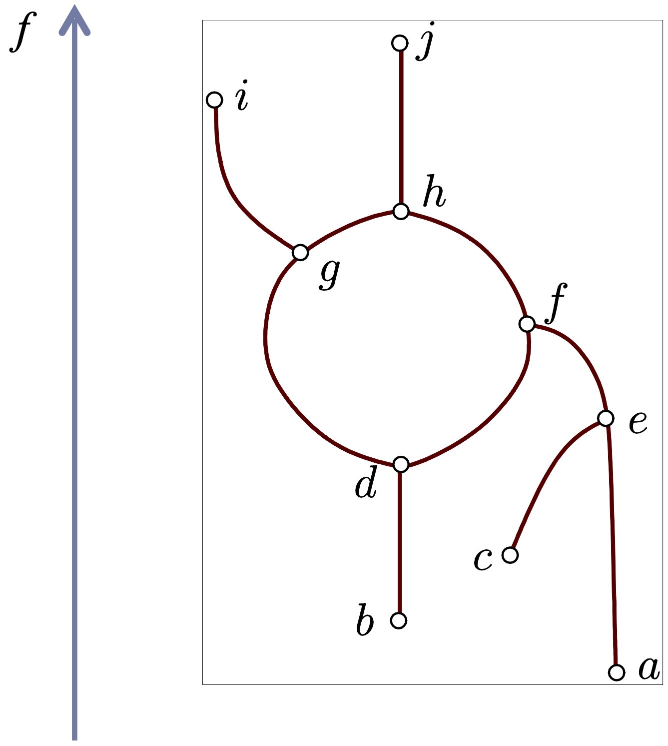
Loop Features

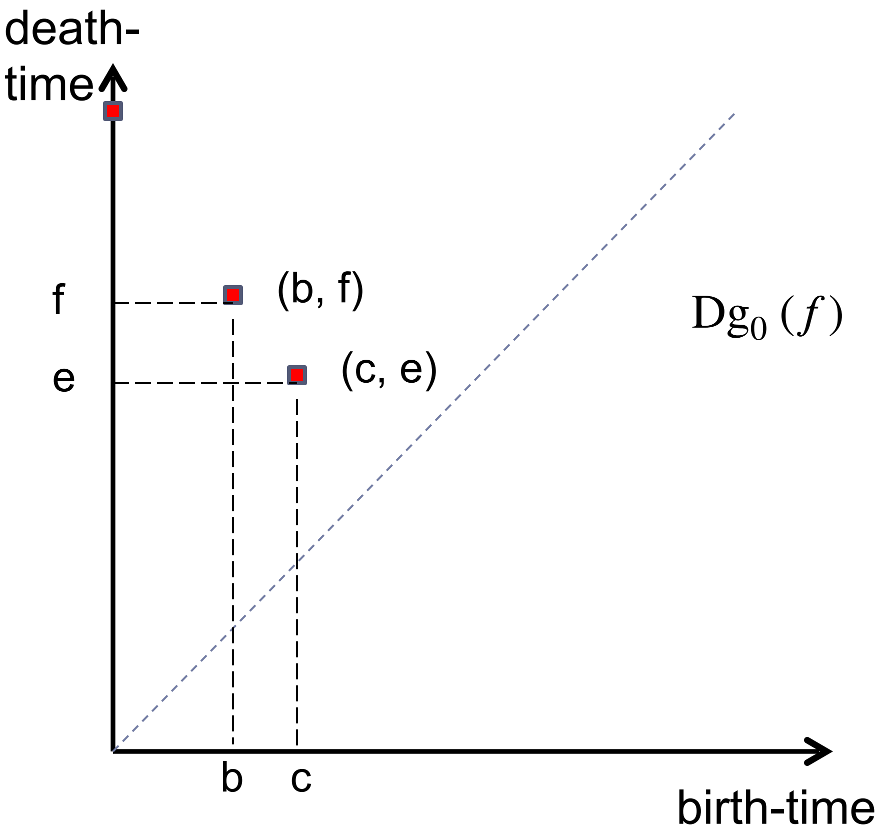
- Captured by Extended persistence:
  - *[Cohen-Steiner, Edelsbrunner and Harer, FoCM 2009]*



Loop Features

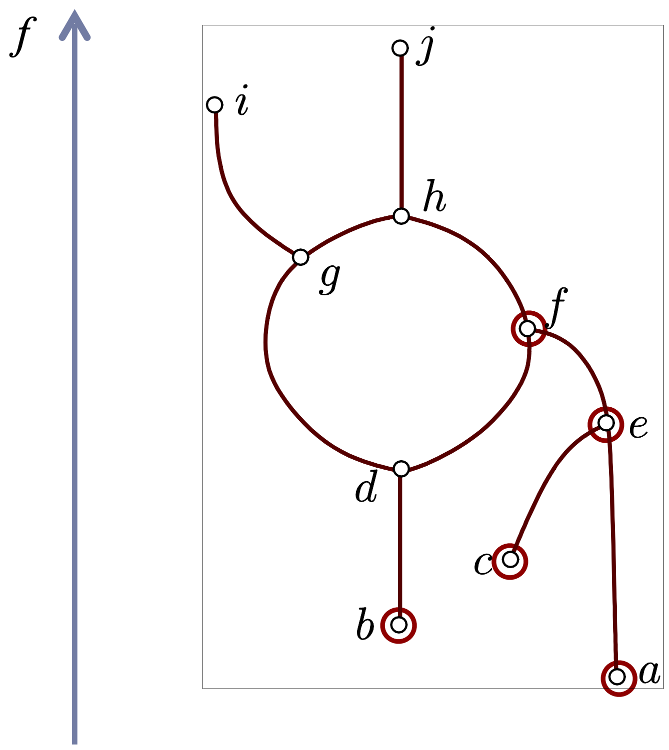
- Captured by Extended persistence:
  - *[Cohen-Steiner, Edelsbrunner and Harer, FoCM 2009]*

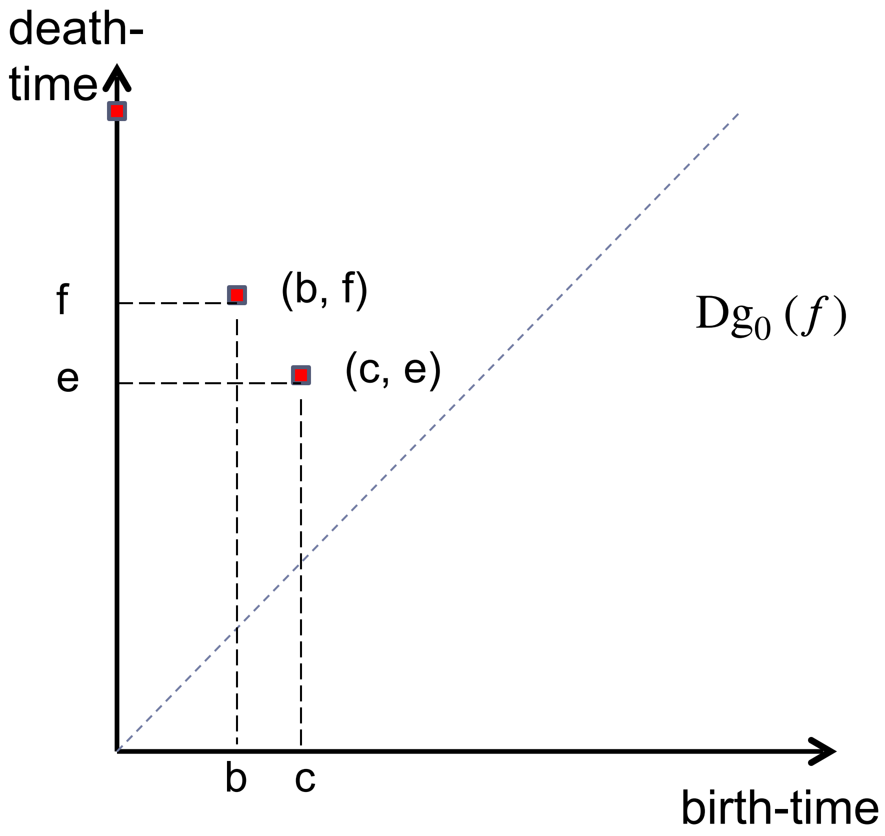




Loop Features

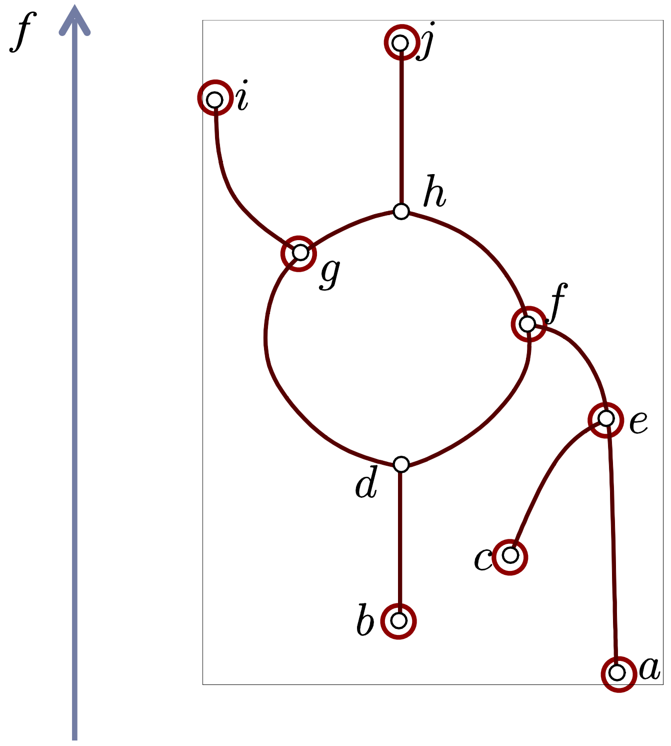
- Captured by Extended persistence:
  - *[Cohen-Steiner, Edelsbrunner and Harer, FoCM 2009]*

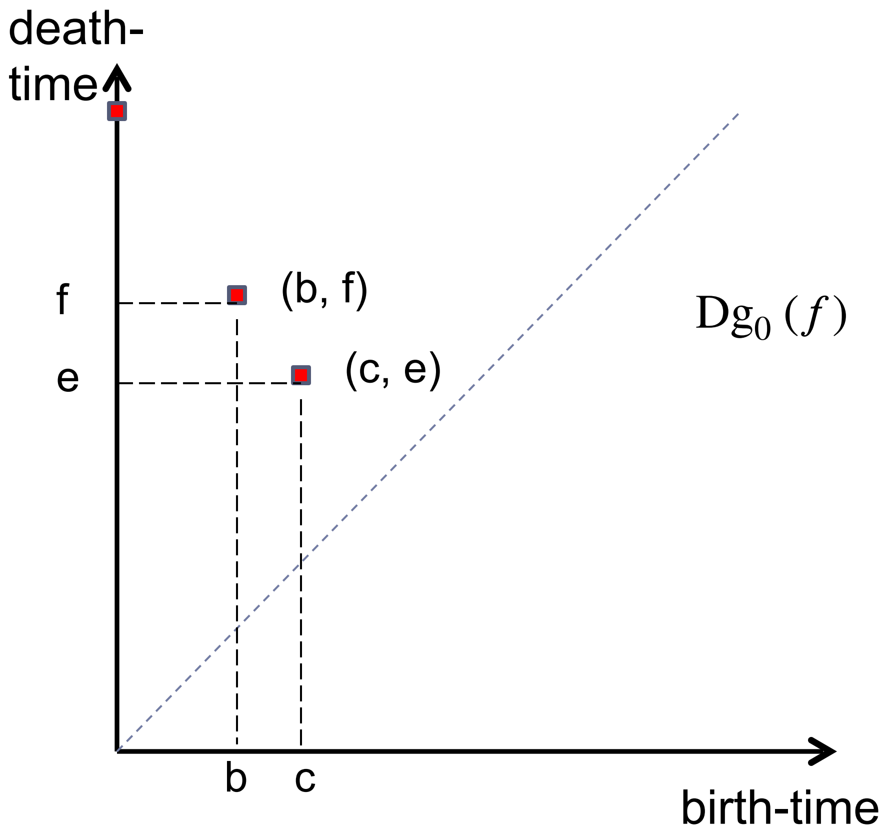




Loop Features

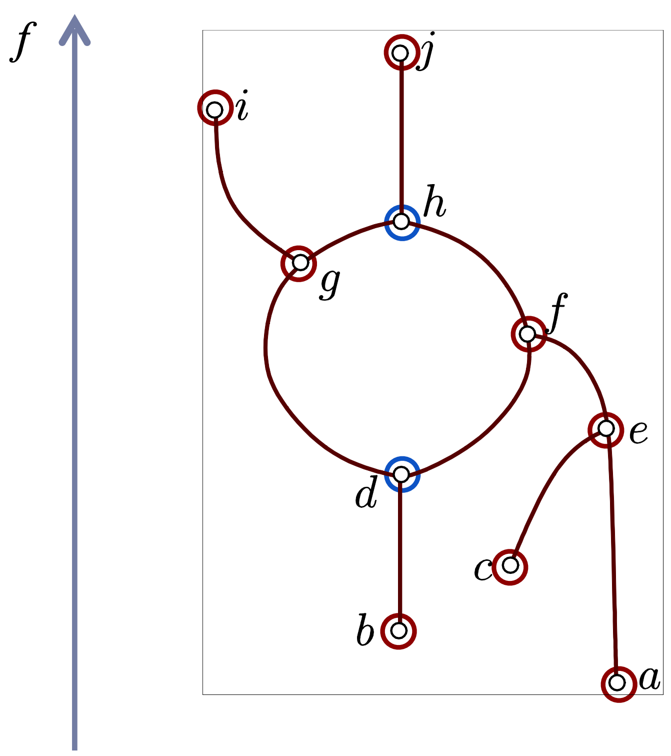
- Captured by Extended persistence:
  - *[Cohen-Steiner, Edelsbrunner and Harer, FoCM 2009]*

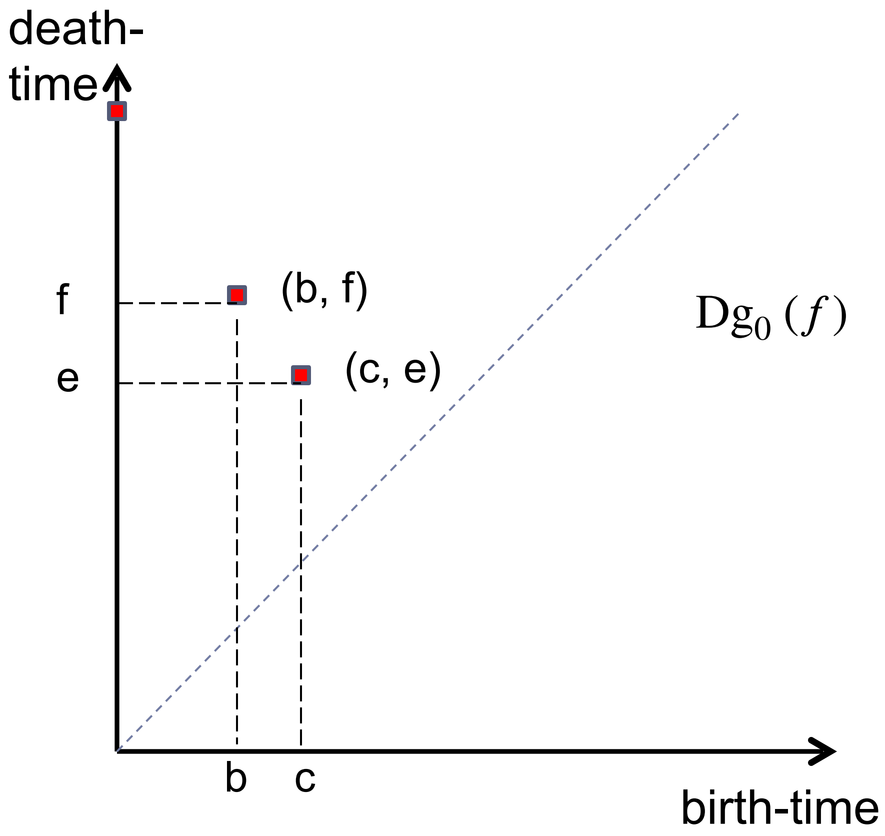




Loop Features

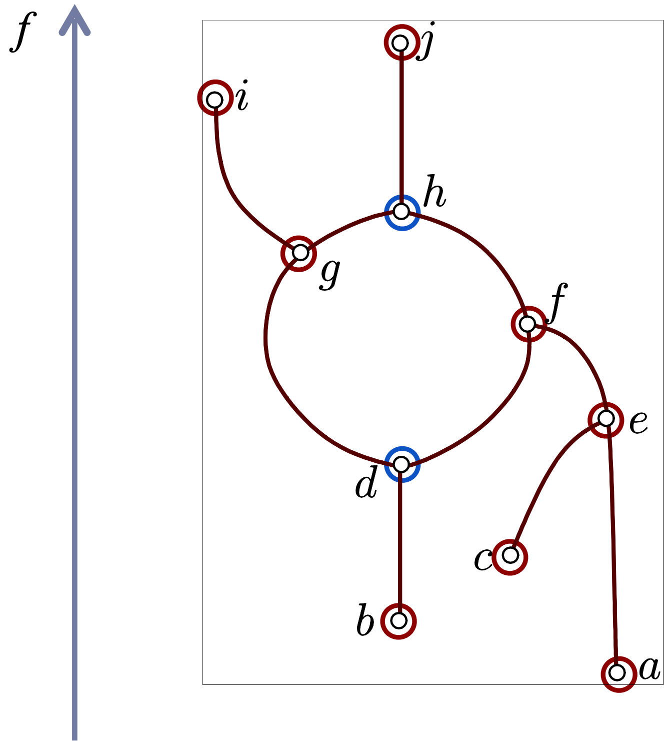
- Captured by Extended persistence:
  - *[Cohen-Steiner, Edelsbrunner and Harer, FoCM 2009]*

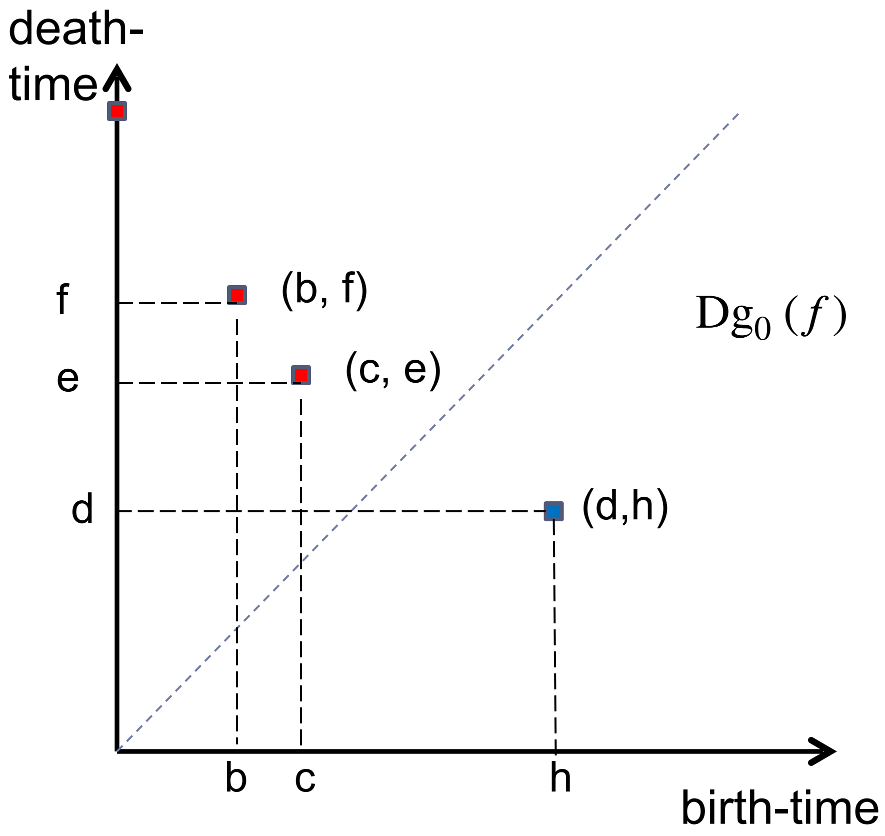


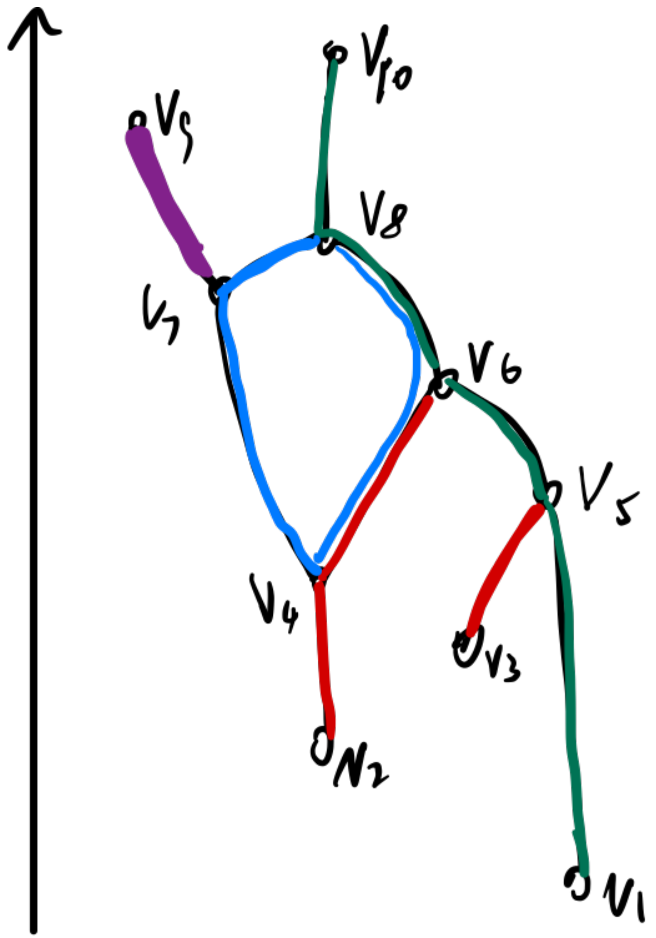


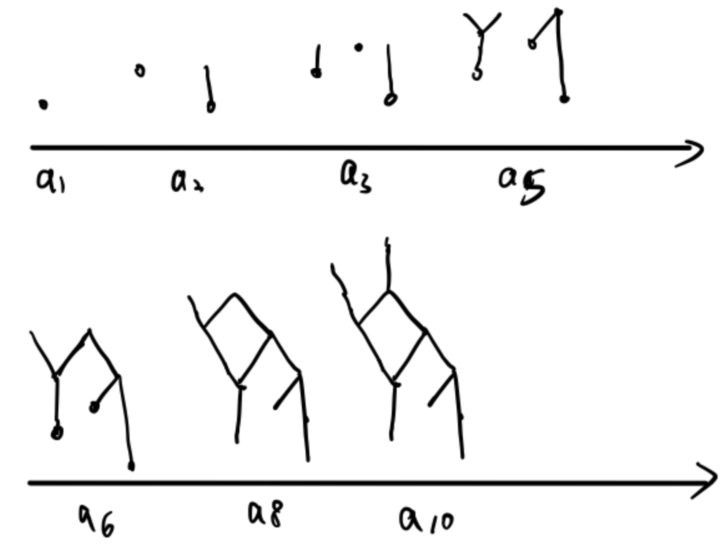
Loop Features

- Captured by Extended persistence:
  - *[Cohen-Steiner, Edelsbrunner and Harer, FoCM 2009]*









0-dim

(a_0, a_{10})

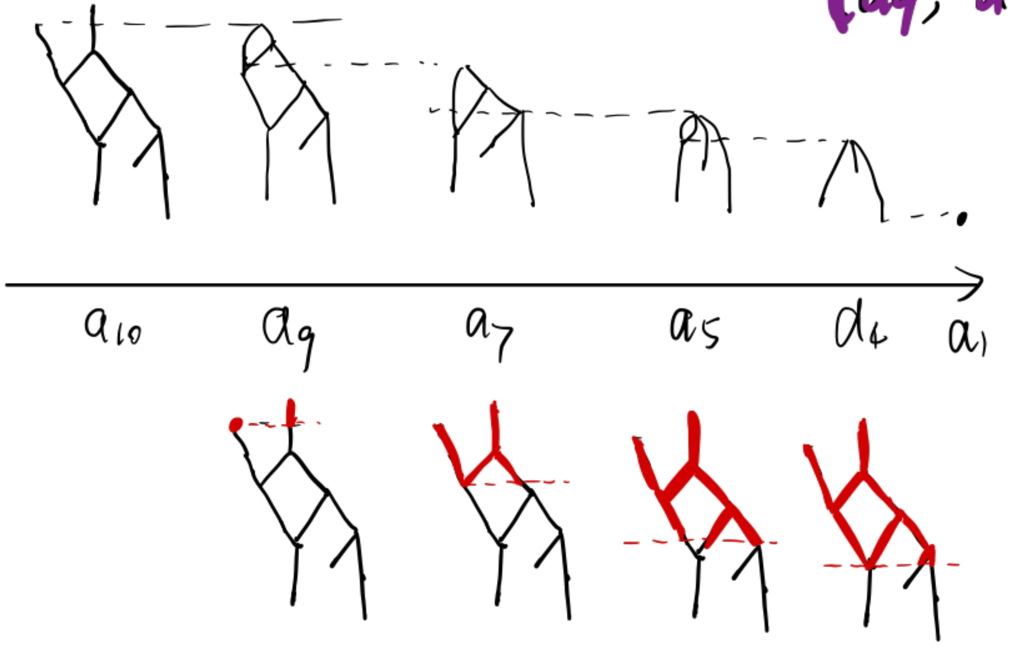
(a_2, a_6)

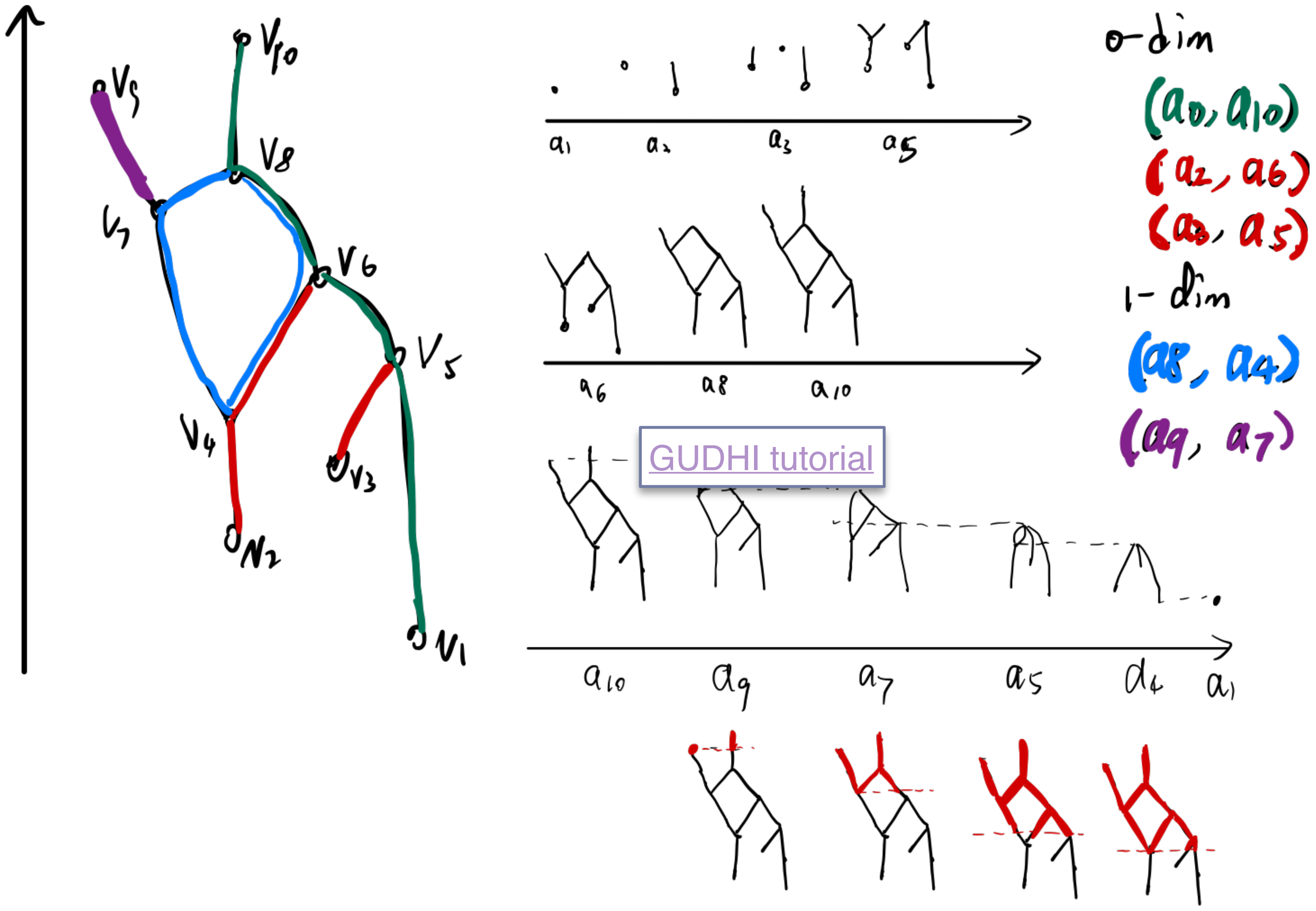
(a_3, a_5)

1-dim

(a_8, a_4)

(a_9, a_7)





V_{10}
V_9
V_8
V_7
V_6
V_5
V_4
V_3
V_2
V_1
a_1 a_2 a_3 a_5
a_6 a_8 a_{10}
GUDHI tutorial
a_{10} a_9 a_7 a_5 a_4 a_1
0-dim
(a_0, a_{10})
(a_2, a_6)
(a_3, a_5)
1-dim
(a_8, a_4)
(a_9, a_7)

A simple application: neuron classification

- For each neuron cell:
  - its morphology is represented by a rooted tree T

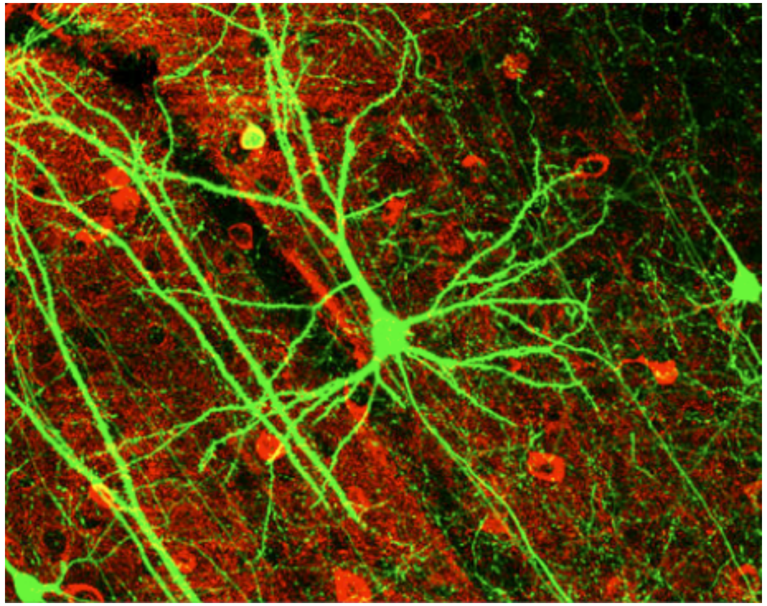


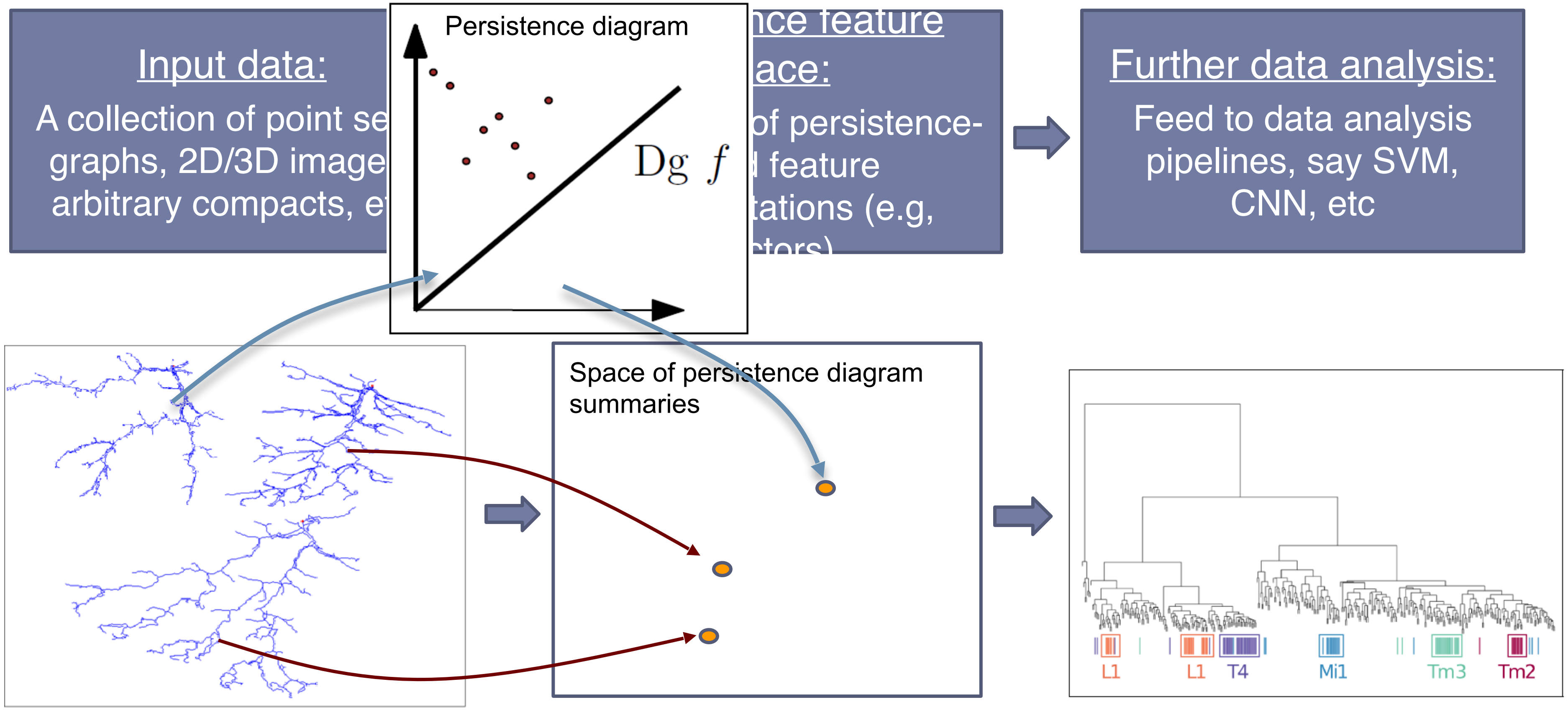
Image taken from Wikipedia

Neuron classification

- For each neuron cell:
  - its morphology is represented by a rooted tree T
- For a neuron tree T, choices of descriptor function $f: T \to R$:
  - Euclidean distance function to root; i.e, $f(x) = \left|\left|x - r\right|\right|_2$
  - Geodesic distance function to root

Persistence-based Framework

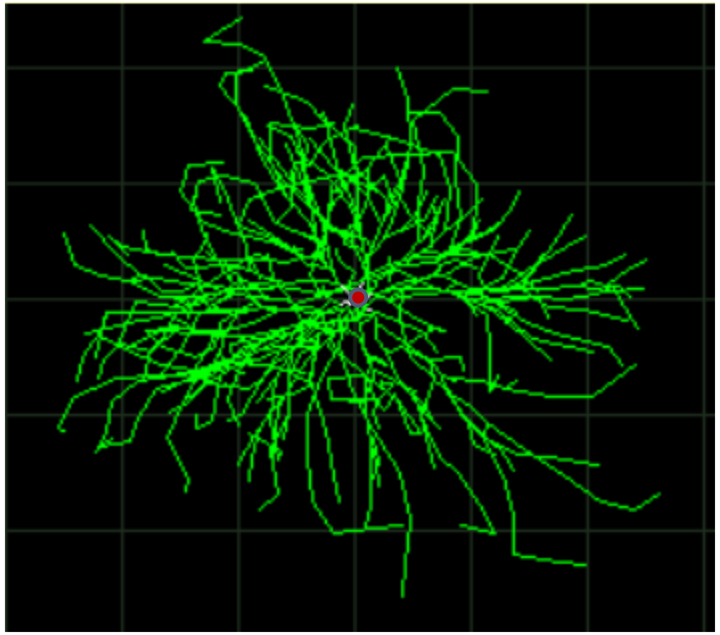
- Persistence-based feature vectorization



[Li, et al, PLOS One 2017]

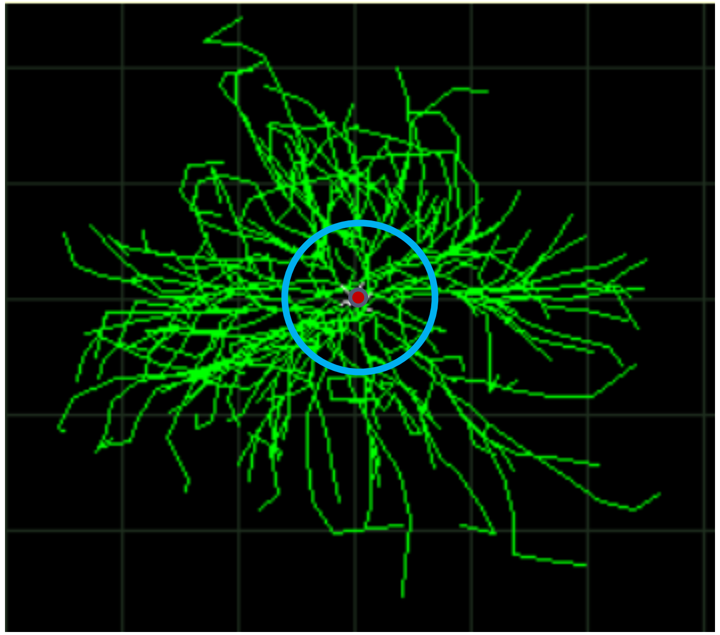
Connection to Sholl-like Analysis

- Sholl function $N: R^+ \to R^+$
  - $N(\lambda) :=$ number of intersection of T with a circle (sphere) centered at the root r with radius λ



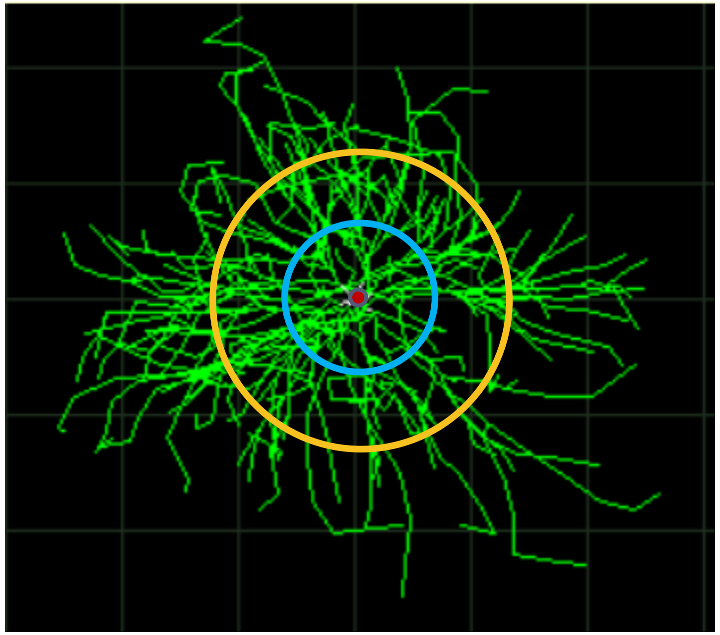
Connection to Sholl-like Analysis

- Sholl function $N\colon R^+ \to R^+$
  - $N(\lambda) :=$ number of intersection of T with a circle (sphere) centered at the root r with radius λ



Connection to Sholl-like Analysis

- Sholl function $N: R^+ \to R^+$
  - $N(\lambda) :=$ number of intersection of T with a circle (sphere) centered at the root r with radius λ

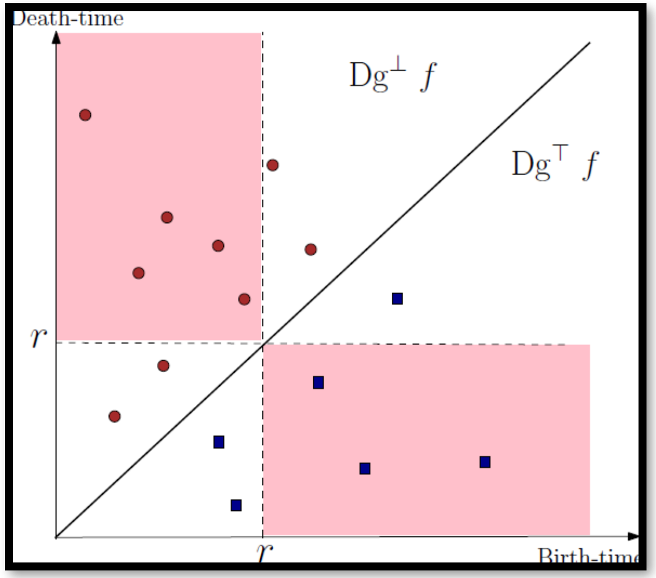


Connection to Sholl-like Analysis

- Sholl function $N: R^+ \to R^+$
  - $N(\lambda) :=$ number of intersection of T with a circle (sphere) centered at the root r with radius λ
- One can recover full Sholl function from persistence diagrams induced by Euclidean distance function

Connection to Sholl-like Analysis

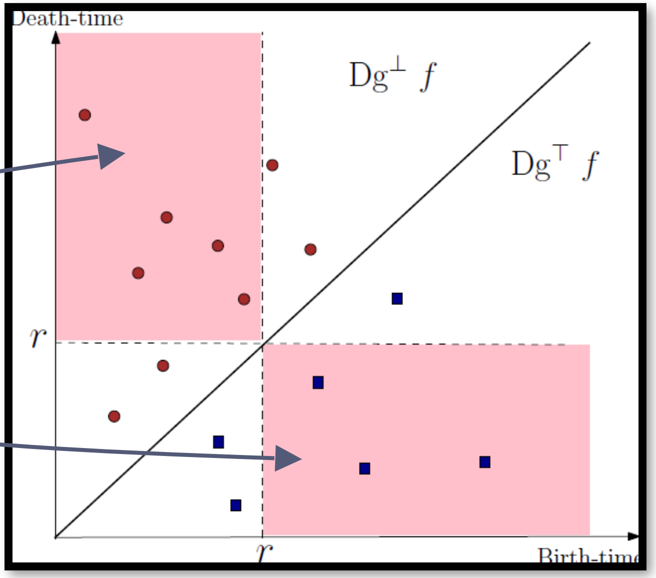
- Sholl function $N: R^+ \to R^+$
  - $N(\lambda) :=$ number of intersection of T with a circle (sphere) centered at the root r with radius λ
- One can recover full Sholl function from persistence diagrams induced by Euclidean distance function

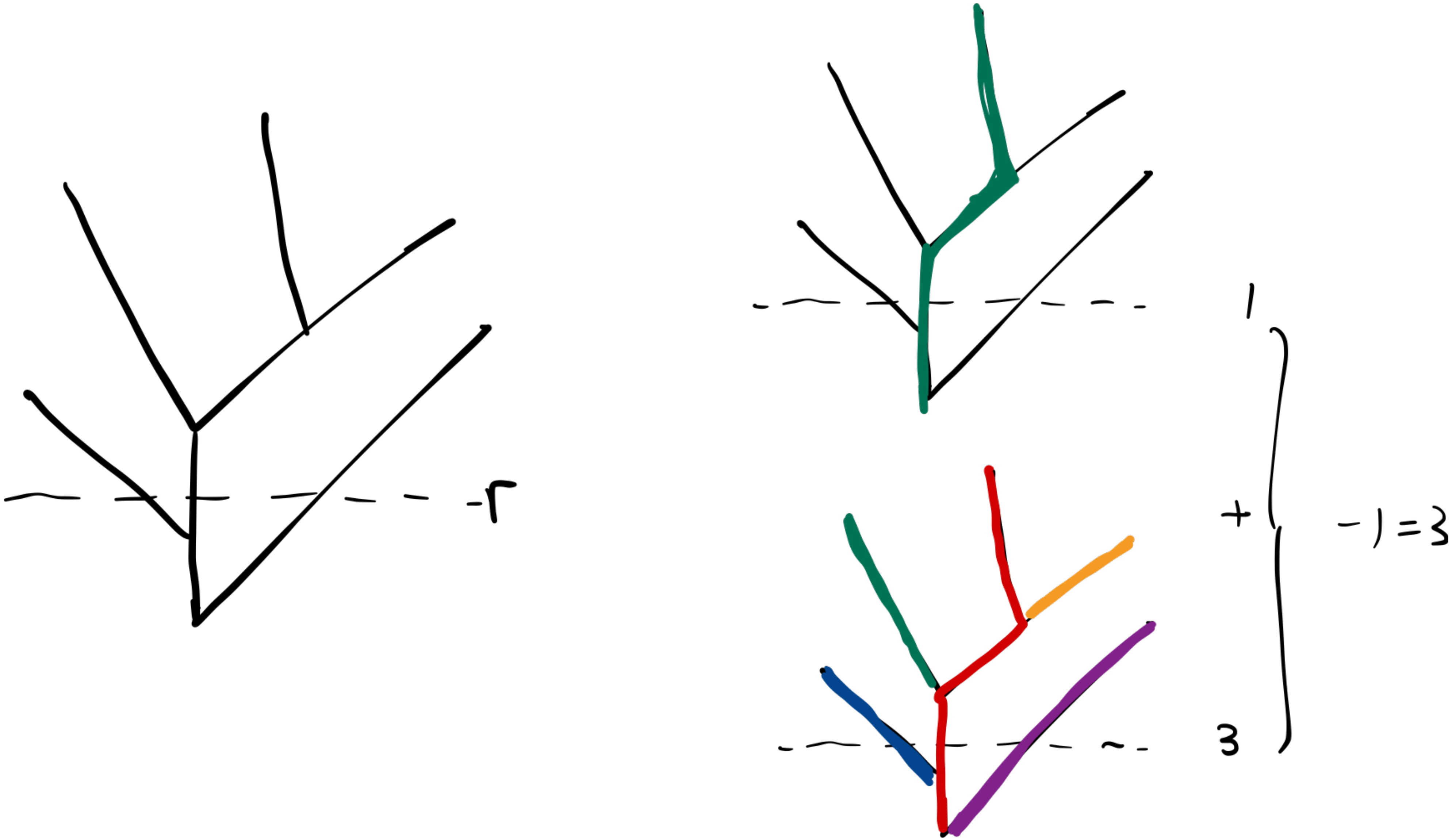


Connection to Sholl-like Analysis

- Sholl function $N: R^+ \to R^+$
  - $N(\lambda) :=$ number of intersection of T with a circle (sphere) centered at the root r with radius λ
- One can recover full Sholl function from persistence diagrams induced by Euclidean distance function

$N(r) =$ total number of points in these two regions - 1



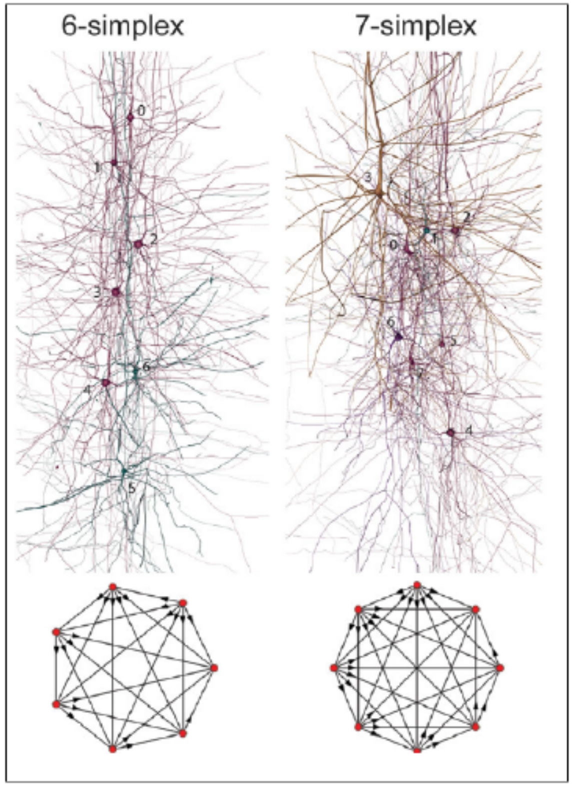

r
1
+
3
-1 = 3

Directed graphs / asymmetric networks

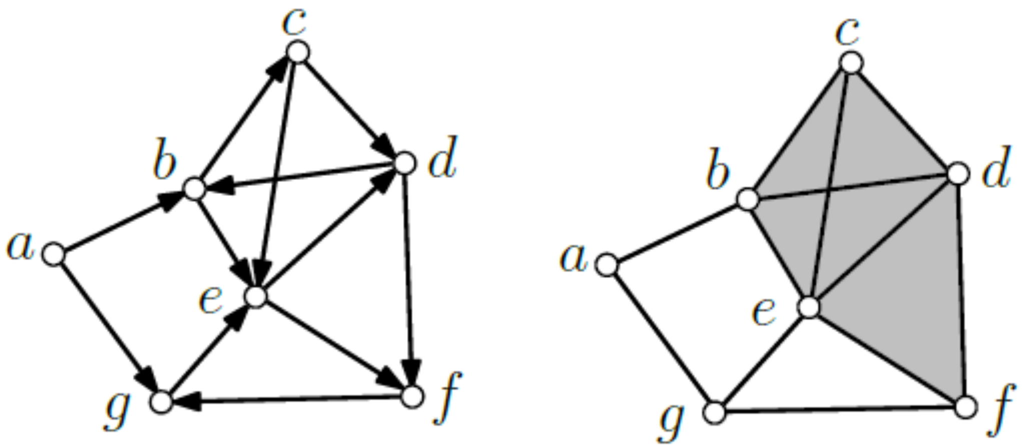
- Directed clique complex
- Path homology

Directed graphs / asymmetric networks

- Directed clique complex
- Path homology

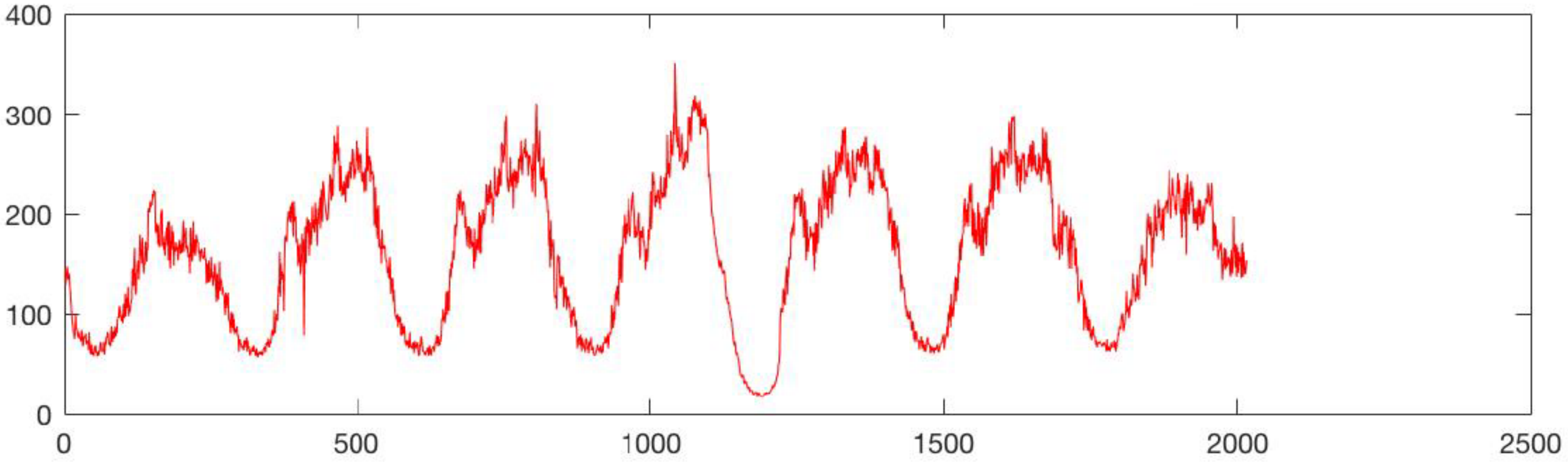


[Reimann et al., 2017]



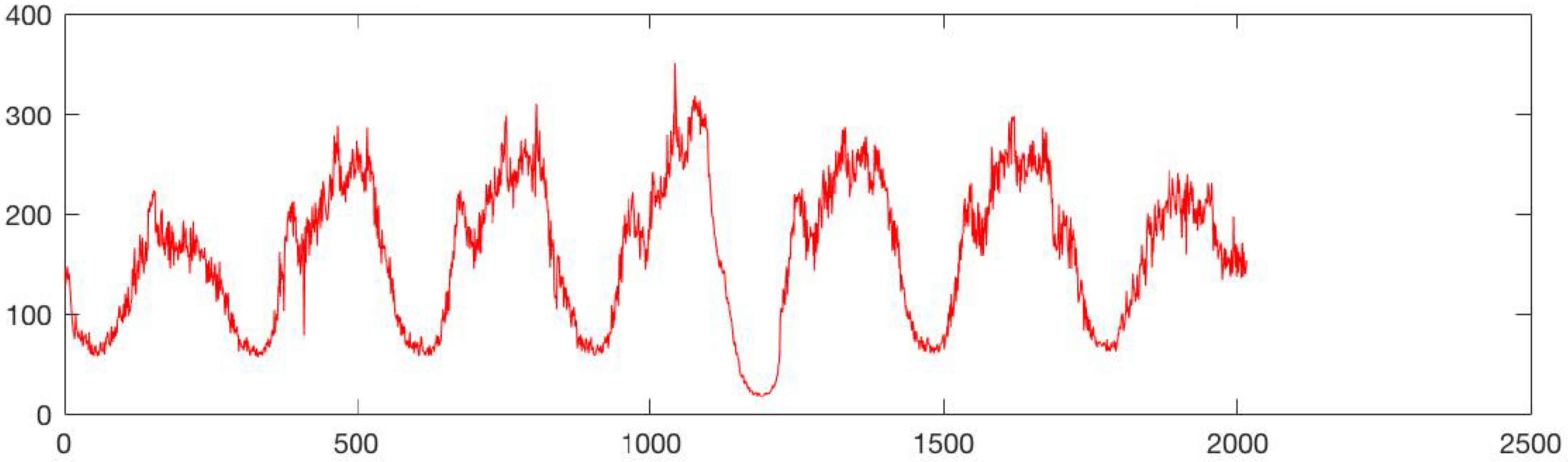
Time series data

- How do we create a filtration on top of a time series data?



Time series data

- How do we create a filtration on top of a time series data?
- Use some sort of "sliding window" embedding
  - Often one aims to capture quasi-periodic features



Time-delay embedding

- Map a time-series data $f: R \to R$ to a point cloud data

Given a time-series $f: R \to R$ and a parameter τ, a time-delay embedding is a lift to a time-series $\phi: R \to R^{M+1}$ defined by

$$\phi(t) = (f(t),\ f(t+\tau),\ \ldots,\ f(t+M\tau))$$

Time-delay embedding

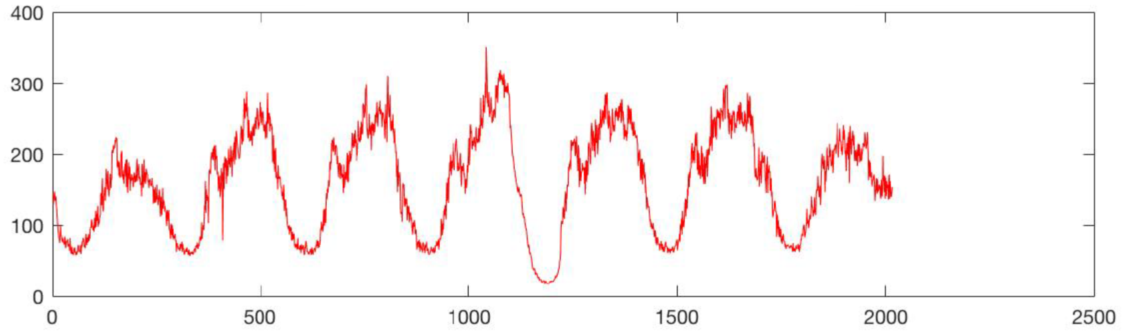
- Map a time-series data $f: R \to R$ to a point cloud data

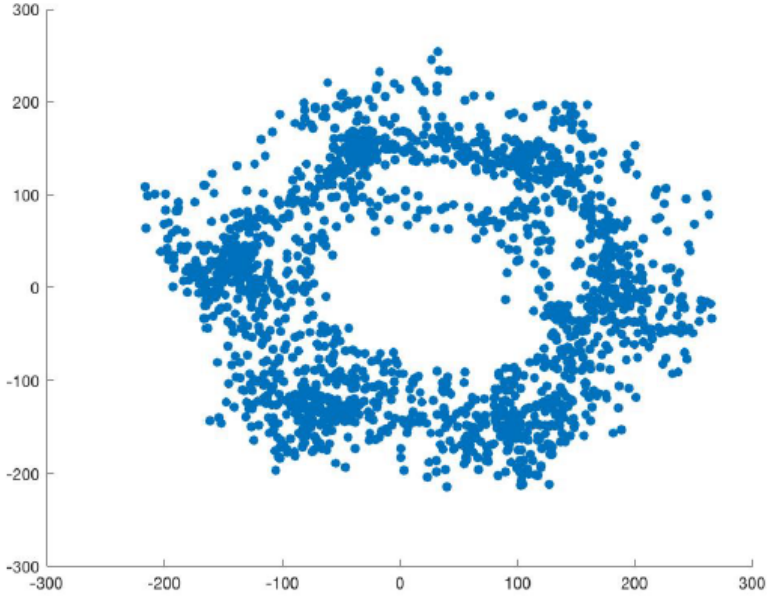
Given a time-series $f: R \to R$ and a parameter τ, a time-delay embedding is a lift to a time-series $\phi: R \to R^{M+1}$ defined by

$$\phi(t) = (f(t),\ f(t+\tau),\ \ldots,\ f(t+M\tau))$$

If the time-series is discrete time, then this will give a point cloud data.

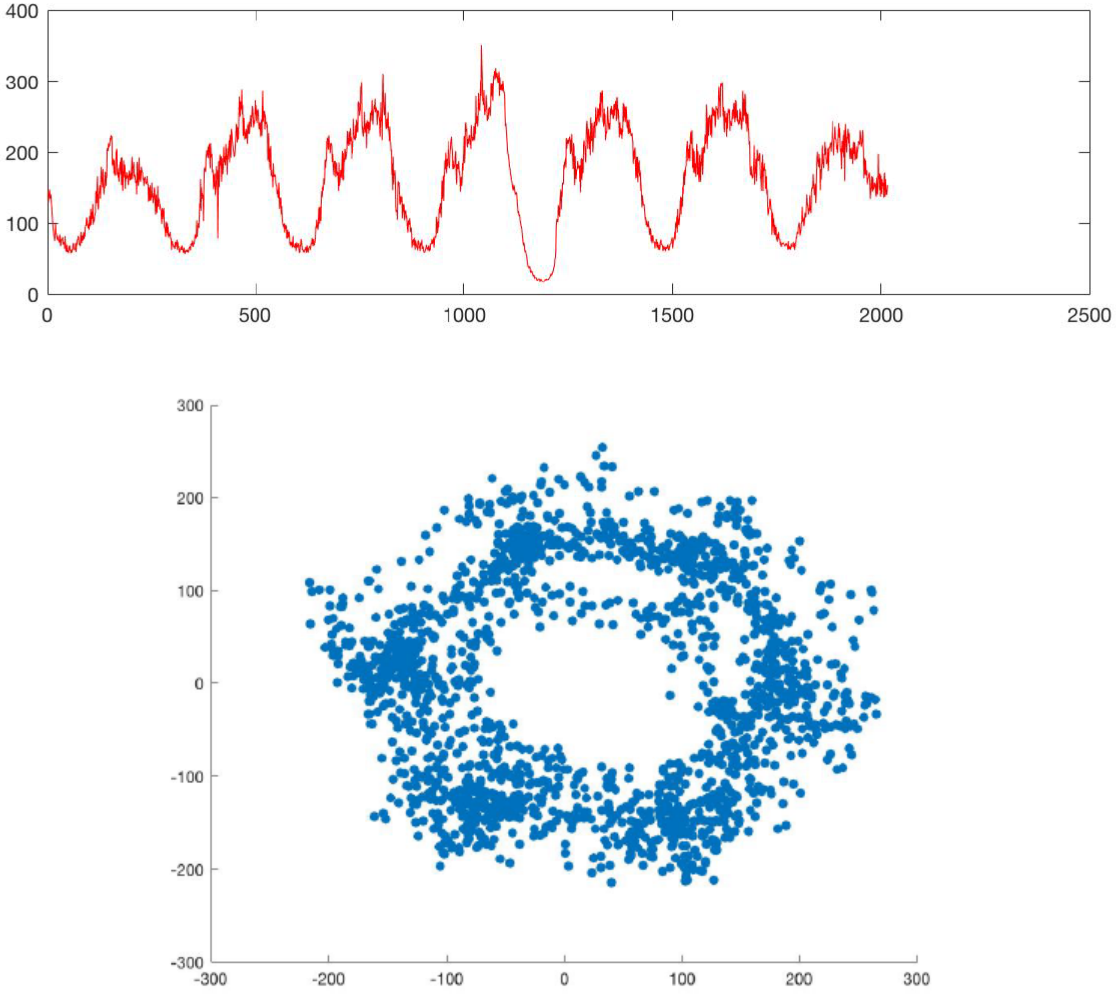
Examples



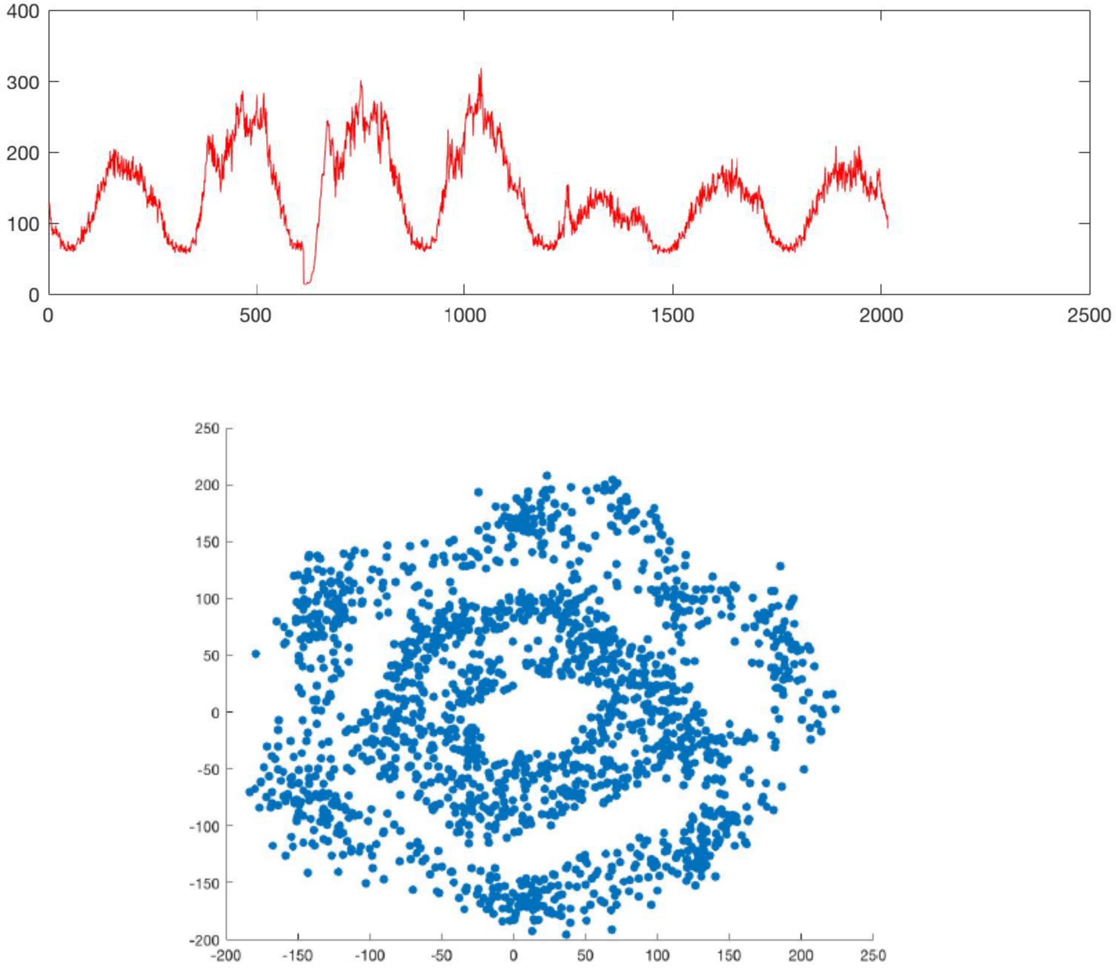


(a) Week 1

Examples

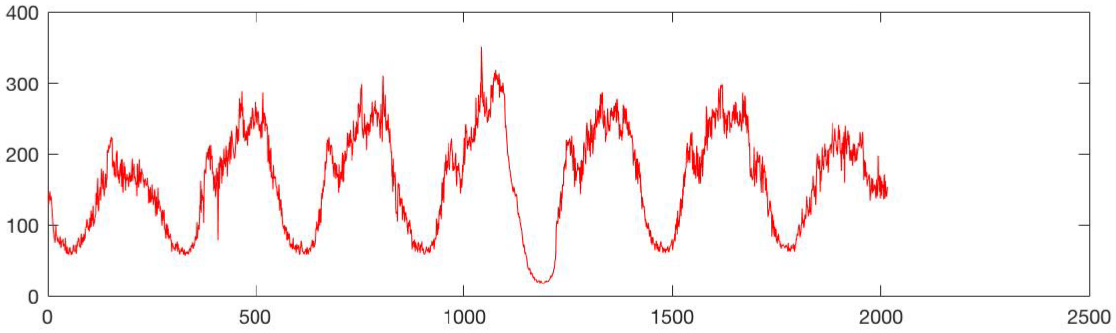


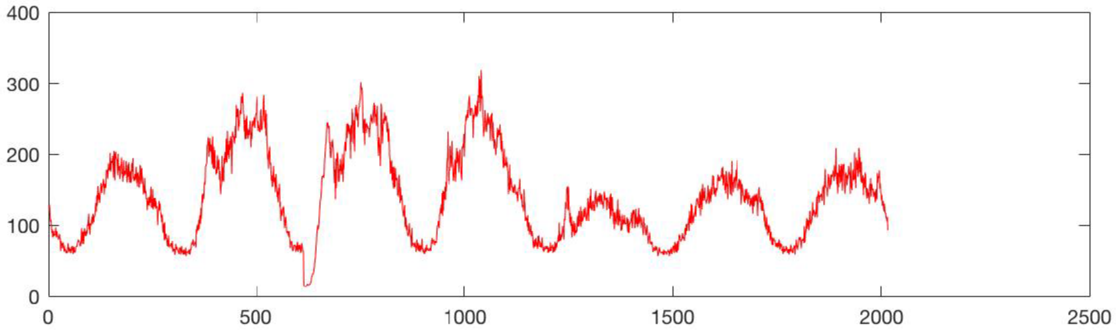
(a) Week 1

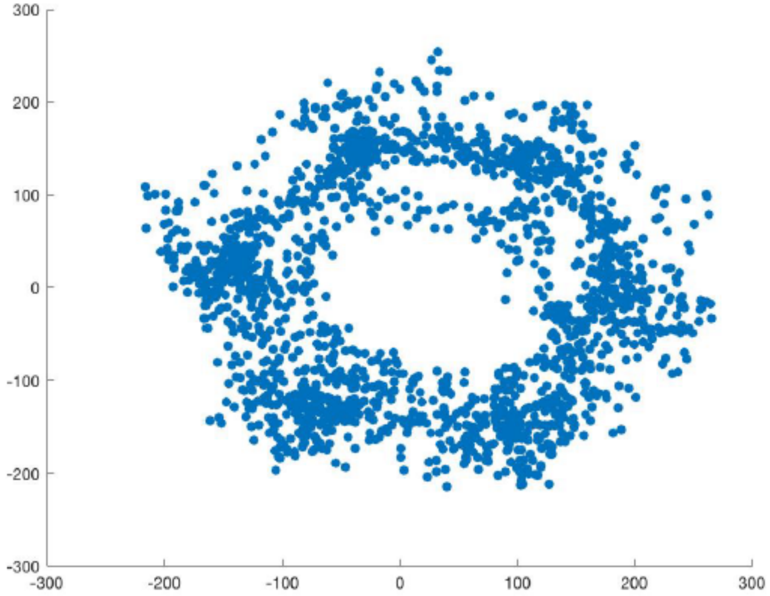


(b) Week 8

Examples

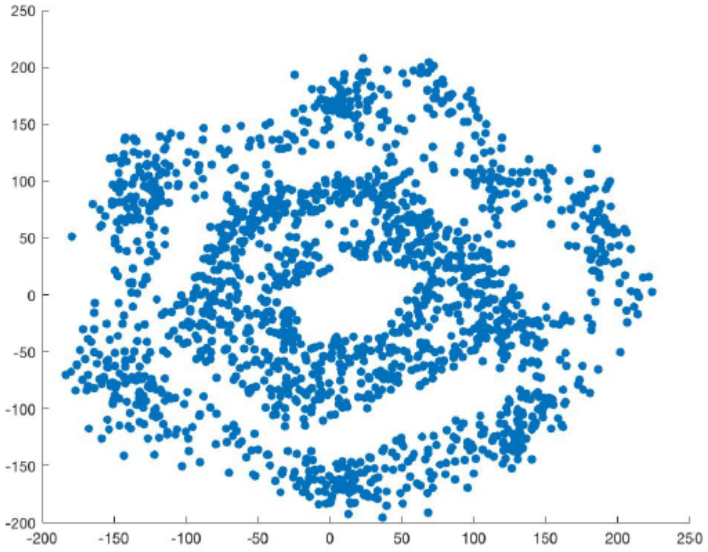






(a) Week 1

See a demo



(b) Week 8

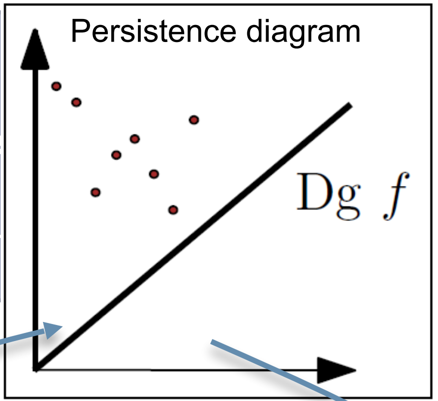
Section 2 :
Vectorization of persistence diagram summaries

Persistence-based Framework

- Persistence-based feature vectorization

Input data:

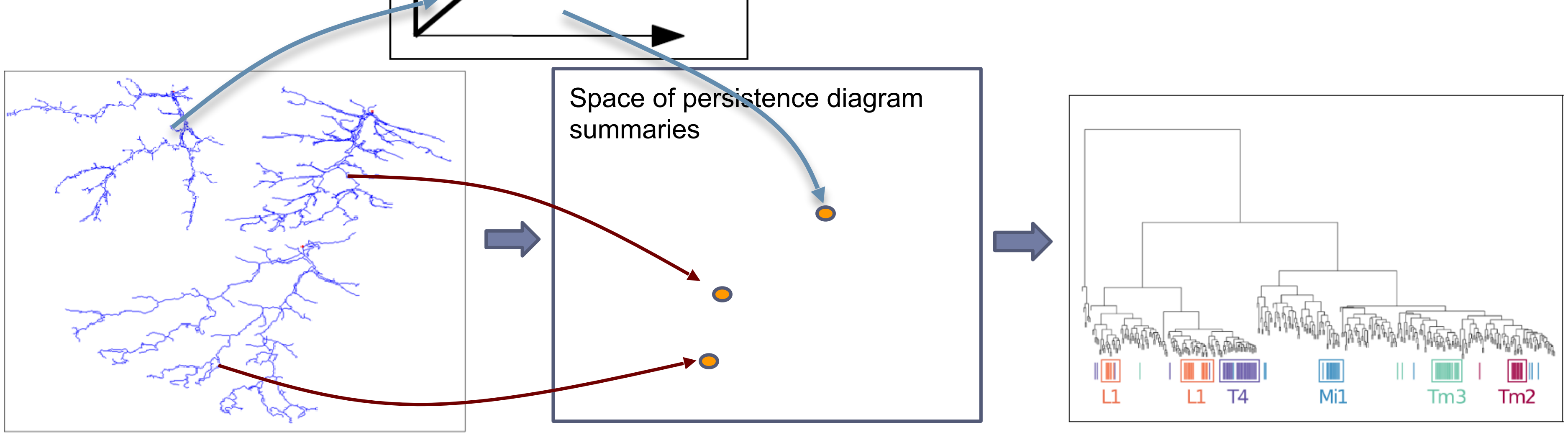
A collection of point se
graphs, 2D/3D image
arbitrary compacts, e



nce feature
ace:
of persistence-
l feature
tations (e.g,
ctors)

Further data analysis:

Feed to data analysis pipelines, say SVM, CNN, etc



[Li, et al., PLOS One 2017]

- Classical distance measure for persistence
  - e.g, bottleneck distance and Wasserstein distance, has stability properties,
  - but does not have nice (e.g, inner product) structure

- Map the space of persistence diagrams to yet another (finite or infinite dimensional) vector space

Persistence feature representation

- Persistence landscapes
  - [Bubenik 2012]
- Persistence scale space kernel
  - [Reininghause et al., 2014]
- Persistence images
  - [Adams et al., 2015, 2017]
- Persistence weighted Gaussian kernel
  - [Kusano et al., 2017]
- Sliced Wasserstein kernel
  - [Carriere et al., 2017]
- Persistence Fisher kernel
  - [Le and Yamada 2018]
- ....

Persistence feature representation

- Persistence landscapes
  - [Bubenik 2012]
- Persistence scale space kernel
  - [Reininghause et al., 2014]
- Persistence images
  - [Adams et al., 2015, 2017]
- Persistence weighted Gaussian kernel
  - [Kusano et al., 2017]
- Sliced Wasserstein kernel
  - [Carriere et al., 2017]
- Persistence Fisher kernel
  - [Le and Yamada 2018]
- ….

Gudhi package contains most of these

- Wonderful slides

Persistence landscapes

- Map persistence diagrams to a function space [Bubenik 2012]

Definition 13.1 (Persistence landscape). Given a finite persistence diagram $D = \{(b_i, d_i)\}_{i\in[1,n]}$ from $\mathbb{D}$, the *persistence landscape w.r.t. D* is a function $\lambda_D : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$ where

$$\lambda_D(k, t) := k\text{-th largest value of } [\min\{t - b_i, d_i - t\}]_+ \text{ for } i \in [1, n].$$

Here, $[c]_+ = max(c, 0)$.

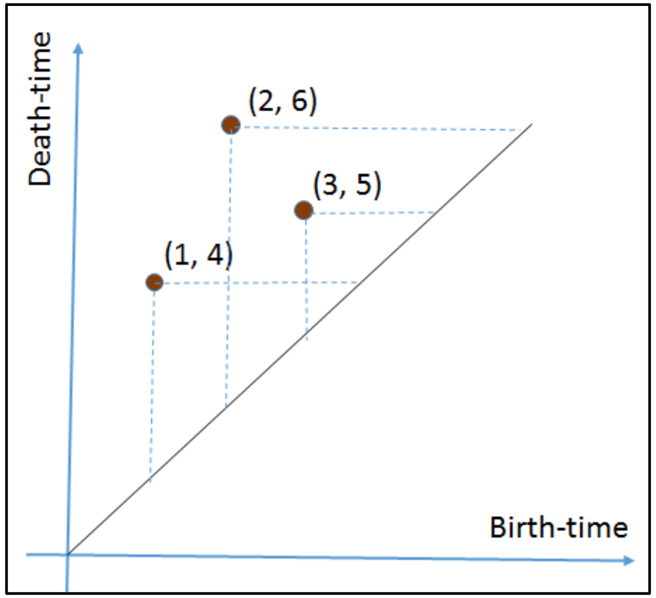
Persistence landscapes

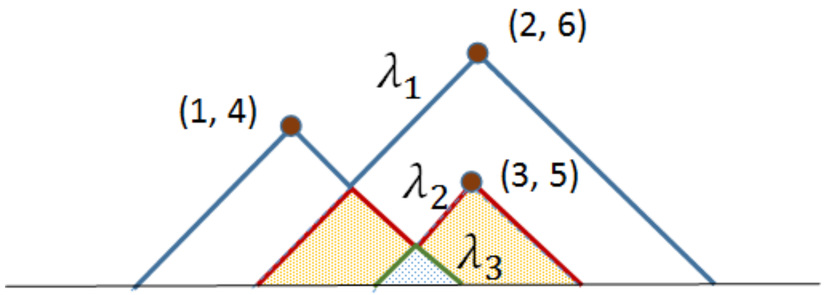
- Map persistence diagrams to a function space [Bubenik 2012]

Definition 13.1 (Persistence landscape). Given a finite persistence diagram $D = \{(b_i, d_i)\}_{i \in [1,n]}$ from $\mathbb{D}$, the *persistence landscape w.r.t.* D is a function $\lambda_D : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$ where

$$\lambda_D(k, t) := k\text{-th largest value of } [\min\{t - b_i, d_i - t\}]_+ \text{ for } i \in [1, n].$$

Here, $[c]_+ = max(c, 0)$.





Metric for persistence landscapes

- p-norm: $$\|\lambda_D\|_p^p = \sum_{k=1}^{\infty} \|\lambda_D(k,\cdot)\|_p^p.$$

- p-landscape distance:

Metric for persistence landscapes

- p-norm: $$\|\lambda_D\|_p^p = \sum_{k=1}^{\infty} \|\lambda_D(k,\cdot)\|_p^p.$$
- p-landscape distance:

$$\Lambda_p(D_1, D_2) = \|\lambda_{D_1} - \lambda_{D_2}\|_p$$

Some results

Claim *Given a persistence diagram D, let λ_D be its persistence landscape. Then from λ_D one can uniquely recover the persistence diagram D.*

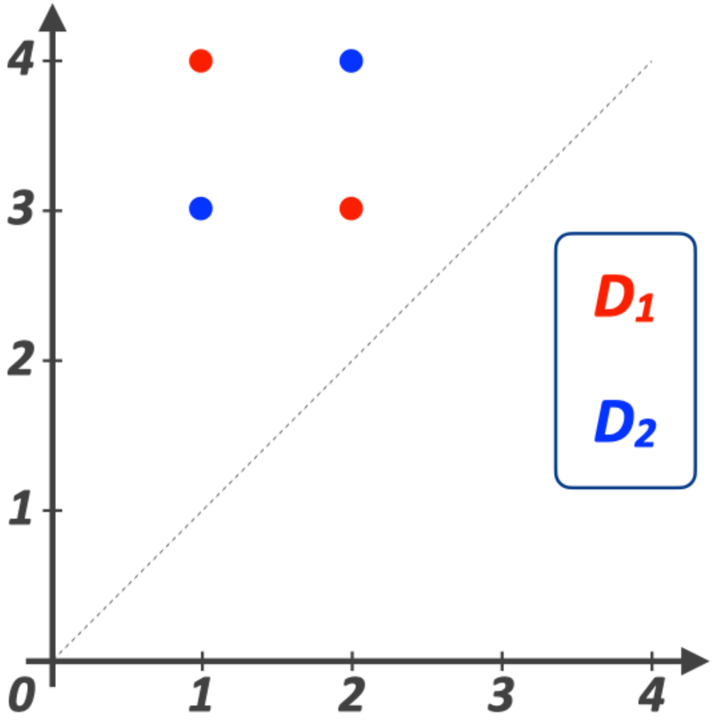
Some results

Claim *Given a persistence diagram D, let λ_D be its persistence landscape. Then from λ_D one can uniquely recover the persistence diagram D.*

Theorem *For persistence diagrams D and D', $\Lambda_\infty(D, D') \leq d_B(D, D')$.*

Statistics for TDA

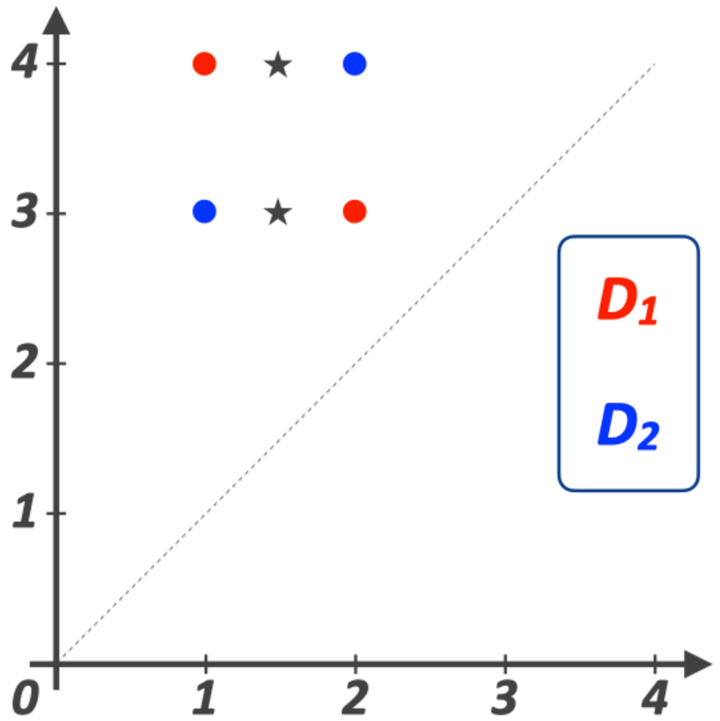
- There is no well-defined notion of mean for persistence diagrams



Courtesy of Fugacci

Statistics for TDA

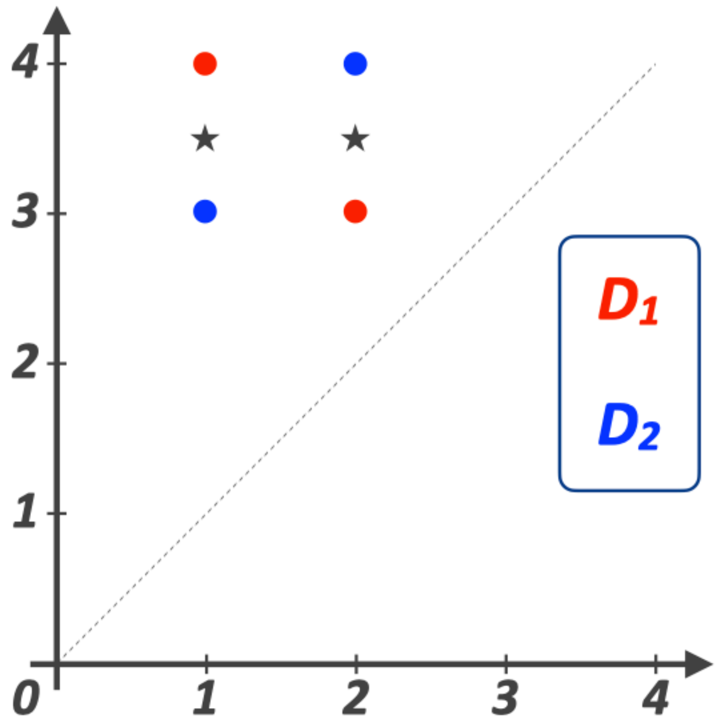
- There is no well-defined notion of mean for persistence diagrams



Courtesy of Fugacci

Statistics for TDA

- There is no well-defined notion of mean for persistence diagrams



Courtesy of Fugacci

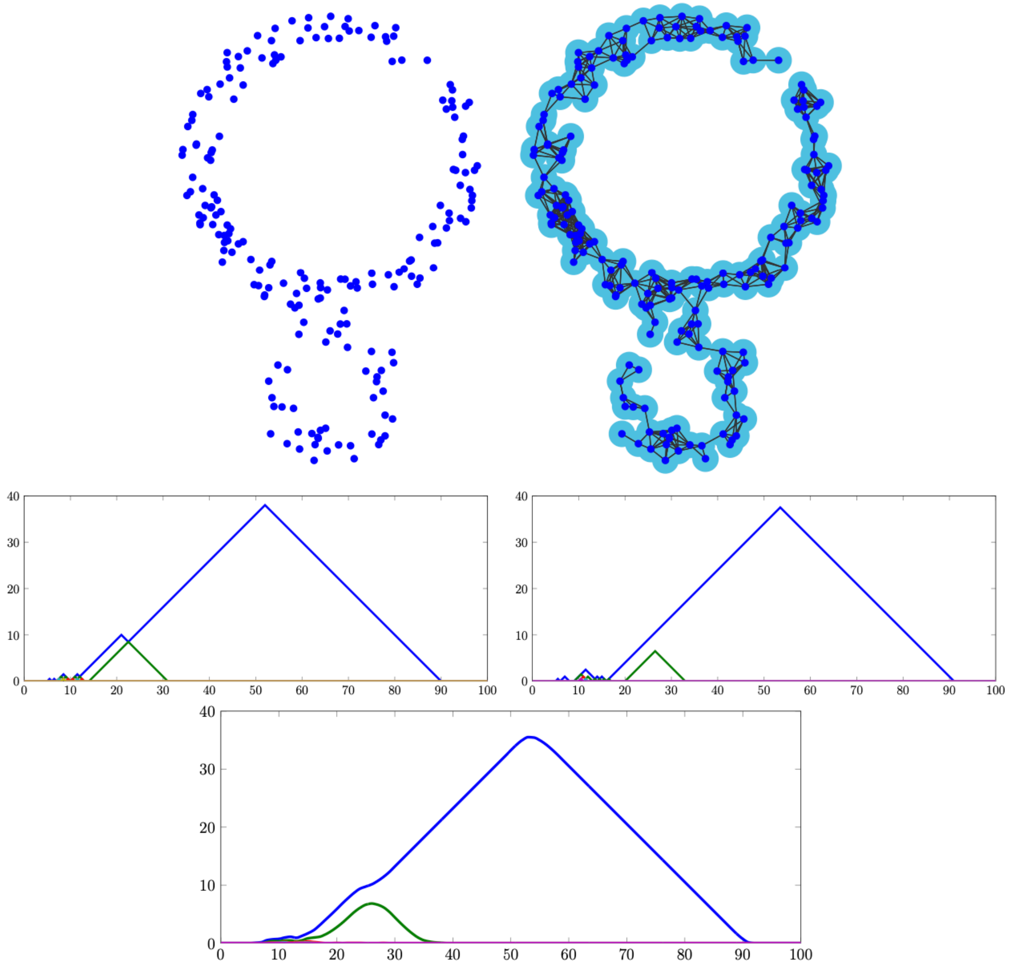
Statistics for TDA

- There is no well-defined notion of mean for persistence diagrams

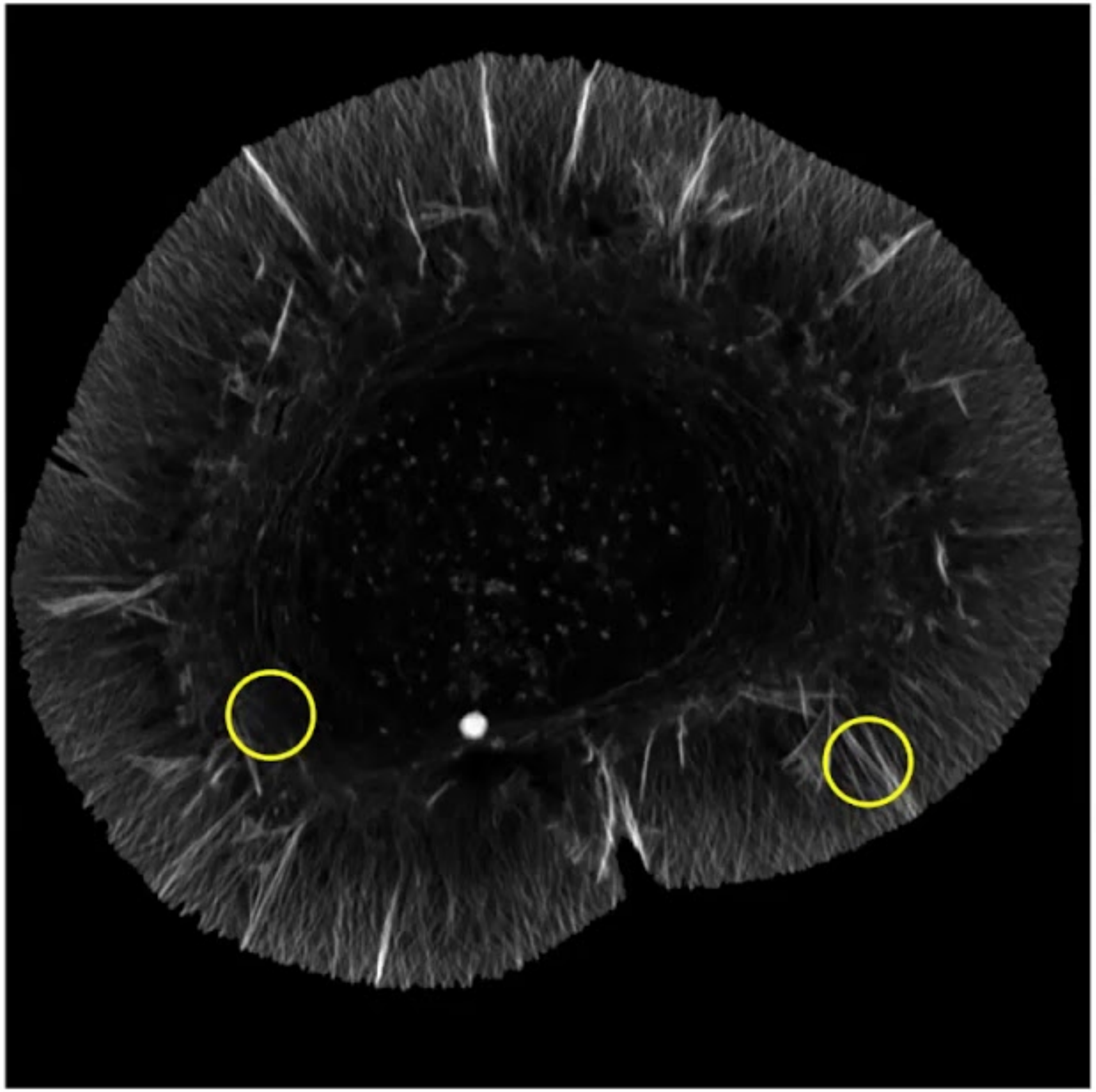
- mean landscape of $\lambda_{D_1}, \lambda_{D_2}, \ldots, \lambda_{D_\ell}$

$$\overline{\lambda}(k, t) = \frac{1}{n} \sum_{i=1}^{\ell} \lambda_{D_i}(k, t).$$

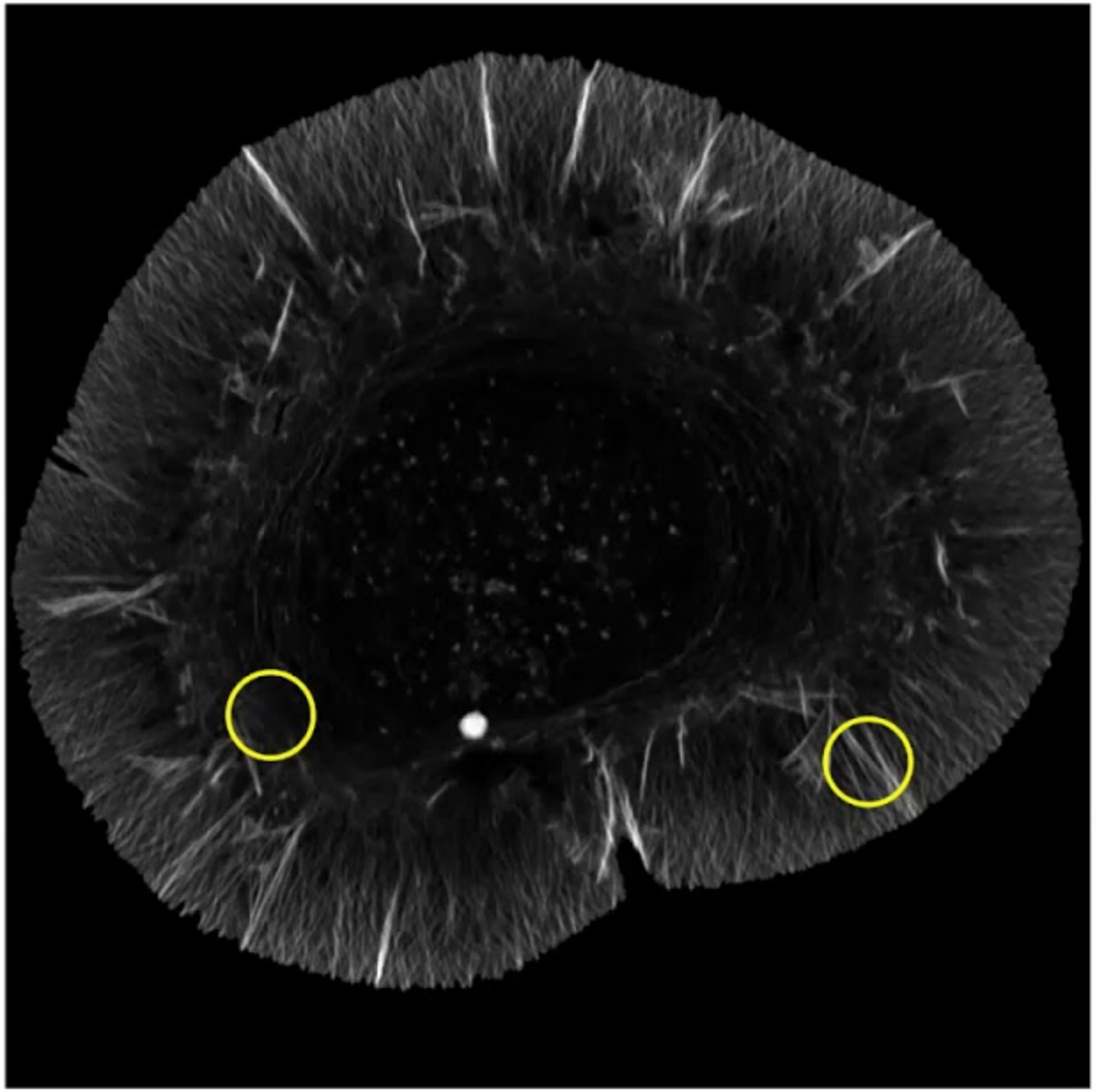
Average of persistence landscapes



e biological images



e biological images



Next

- Kernel-based approach
- SVM for classification
- PCA for dimension reduction, visualization

Kernels

- Given a topological space X, a kernel is simply a function $k: X \times X \rightarrow R$
  - i.e., given any $x, \ y \in X$, it sends them to a real value $k\left(x, y\right)$
- In practice, we often use positive (semi-)definite kernels:
- A Hilbert space is intuitively a vector space (over real or complex number) equipped with an inner product, and it is a complete metric space w.r.t. the distance induced by the inner product.

Kernels

- Given a topological space X, a kernel is simply a function $k: X \times X \to R$
  - i.e., given any $x,\ y \in X$, it sends them to a real value $k(x, y)$
- In practice, we often use positive (semi-)definite kernels:

Definition 13.2 (Positive, negative semi-definite kernel)**.** Given a topological space X, a function $\mathsf{k} : X \times X \to \mathbb{R}$ is a *positive semi-definite kernel* if it is symmetric and for any integer $n > 0$, any $x_1, \ldots, x_n \in X$, and any $a_1, \ldots, a_n \in \mathbb{R}$ with $\sum_i a_i = 0$, it holds that $\sum_{i,j} a_i a_j \mathsf{k}(x_i, x_j) \geq 0$. Analogously, k is a *negative semi-definite kernel* if it is symmetric and any integer $n > 0$, any $x_1, \ldots, x_n \in X$ and any $a_1, \ldots, a_n \in \mathbb{R}$ with $\sum_i a_i = 0$, it holds that $\sum_{i,j} a_i a_j \mathsf{k}(x_i, x_j) \leq 0$.

- A Hilbert space is intuitively a vector space (over real or complex number) equipped with an inner product, and it is a complete metric space w.r.t. the distance induced by the inner product.

- A positive semi-definite kernel can be thought of the inner product $k\left(x, y\right) = \langle\Phi(x),\ \Phi(y)\rangle_{\mathscr{H}}$ after mapping X to some Hilbert space $\mathscr{H}$ via a feature map $\Phi: X \to\ \mathscr{H}$
- Hence given a positive semi-definite kernel, we can induce a (pseudo-)metric:
  - $d_k^2(x,\ y) = \left\langle x - y, x - y \right\rangle = k(x,\ x) + k\left(y,\ y\right)\ - 2k\left(x, y\right)$
- In what follows, we will define kernel (distance) for persistence diagrams by either explicitly specifying the kernel; or specifying the feature map.

PSSK (Persistence scale space kernel)

- [Reininghaus et al. 2015]

Definition 13.3 (Persistence scale space kernel (PSSK)). Define the feature map $\Phi_\sigma : \mathbb{D} \to \mathcal{L}^2(\Omega)$ at scale $\sigma > 0$ as follows: for a persistence diagram $D \in \mathbb{D}$ and $x \in \Omega$, set:

$$\Phi_\sigma(D)(x) = \frac{1}{4\pi\sigma} \sum_{y \in D} [e^{-\frac{\|x-y\|^2}{4\sigma}} - e^{-\frac{\|x-\bar{y}\|^2}{4\sigma}}],$$

where $\bar{y} = (y_2, y_1)$ if $y = (y_1, y_2)$ (i.e, $\bar{y}$ is the reflection of y across the diagonal). This feature map induces the following *persistence scale space kernel (PSSK)* $k_\sigma : \mathbb{D} \times \mathbb{D} \to \mathbb{R}$ using the inner product structure on $\mathcal{L}^2(\Omega)$: given two diagrams $D, E \in \mathbb{D}$,

$$k_\sigma(D, E) = \langle \Phi_\sigma(D), \Phi_\sigma(E) \rangle_{\mathcal{L}^2(\Omega)} = \frac{1}{8\pi\sigma} \sum_{y \in D;\, z \in E} [e^{-\frac{\|y-z\|^2}{8\sigma}} - e^{-\frac{\|y-\bar{z}\|^2}{8\sigma}}].$$

PSSK (Persistence scale space kernel)

- [Reininghaus et al. 2015]

Definition 13.3 (Persistence scale space kernel (PSSK)). Define the feature map $\Phi_\sigma : \mathbb{D} \to \mathcal{L}^2(\Omega)$ at scale $\sigma > 0$ as follows: for a persistence diagram $D \in \mathbb{D}$ and $x \in \Omega$, set:

$$\Phi_\sigma(D)(x) = \frac{1}{4\pi\sigma} \sum_{y \in D} [e^{-\frac{\|x-y\|^2}{4\sigma}} - e^{-\frac{\|x-\bar{y}\|^2}{4\sigma}}],$$

where $\bar{y} = (y_2, y_1)$ if $y = (y_1, y_2)$ (i.e, $\bar{y}$ is the reflection of y across the diagonal). This feature map induces the following *persistence scale space kernel (PSSK)* $k_\sigma : \mathbb{D} \times \mathbb{D} \to \mathbb{R}$ using the inner product structure on $\mathcal{L}^2(\Omega)$: given two diagrams $D, E \in \mathbb{D}$,

$$k_\sigma(D, E) = \langle \Phi_\sigma(D), \Phi_\sigma(E) \rangle_{\mathcal{L}^2(\Omega)} = \frac{1}{8\pi\sigma} \sum_{y \in D;\ z \in E} [e^{-\frac{\|y-z\|^2}{8\sigma}} - e^{-\frac{\|y-\bar{z}\|^2}{8\sigma}}].$$

$$\|\Phi_\sigma(D) - \Phi_\sigma(E)\|_{\mathcal{L}^2(\Omega)} = \sqrt{k_\sigma(D, D) + k_\sigma(E, E) - 2k_\sigma(D, E)}.$$

- Hence a persistence diagram D is mapped to a function $\Phi_\sigma(D)$ under the feature map Φ_σ
- Induced metric:

$$\|\Phi_\sigma(D) - \Phi_\sigma(E)\|_{\mathcal{L}^2(\Omega)} = \sqrt{k_\sigma(D, D) + k_\sigma(E, E) - 2k_\sigma(D, E)}.$$

- Stability:

- Hence a persistence diagram D is mapped to a function $\Phi_\sigma(D)$ under the feature map Φ_σ
- Induced metric:

$$\|\Phi_\sigma(D) - \Phi_\sigma(E)\|_{\mathcal{L}^2(\Omega)} = \sqrt{k_\sigma(D,D) + k_\sigma(E,E) - 2k_\sigma(D,E)}.$$

- Stability:

Theorem 13.2. *Given two persistence diagrams $D, E \in \mathbb{D}$, we have*

$$\|\Phi_\sigma(D) - \Phi_\sigma(E)\|_{\mathcal{L}^2(\Omega)} \le \frac{1}{2\pi\sigma} \mathrm{d}_{W,1}(D,E).$$

PWGK (Persistence weighted Gaussian kernel)

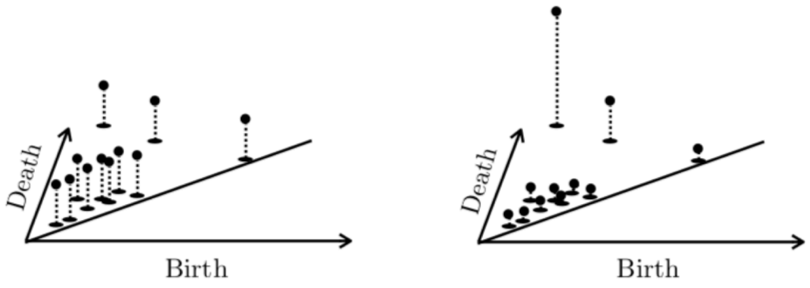
- [Kusano et al. 2017]

Definition 13.5 (Persistence weighted kernel)**.** Let $k : \Omega \to \mathbb{R}$ be a C_0-universal kernel on Ω (e.g, a Gaussian), and $\omega : \Omega \to \mathbb{R}^+$ a strictly positive (weight) function on Ω. The following feature map $\Psi_{k,\omega} : \mathbb{D} \to \mathcal{H}_k$ maps each persistence diagram $D \in \mathbb{D}$ to the RKHS $\mathcal{H}_k$ associated to k:

$$\Psi_{k,\omega}(D) = \sum_{x \in D} \omega(x) k(\cdot, x).$$

This feature map induces the following *persistence weighted kernel (PWK)* $K_{k,\omega} : \mathbb{D} \times \mathbb{D} \to \mathbb{R}$:

$$K_{k,\omega}(D, E) = \langle \Psi_{k,\omega}(D), \Psi_{k,\omega}(E) \rangle_{\mathcal{H}_k} = \sum_{x \in D;\ y \in E} \omega(x)\omega(y)k(x, y). \tag{13.4}$$



PWGK (Persistence weighted Gaussian kernel)

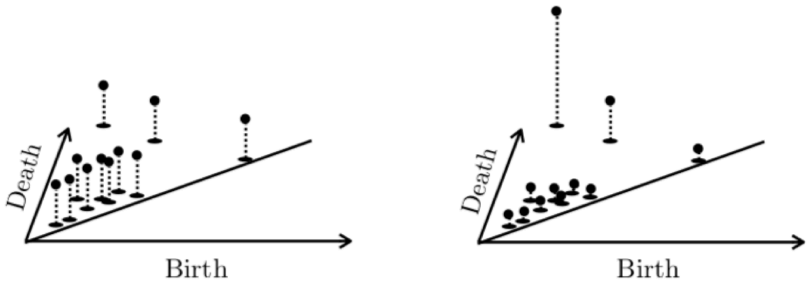
- [Kusano et al. 2017]

Definition 13.5 (Persistence weighted kernel). Let $k : \Omega \to \mathbb{R}$ be a C_0-universal kernel on Ω (e.g, a Gaussian), and $\omega : \Omega \to \mathbb{R}^+$ a strictly positive (weight) function on Ω. The following feature map $\Psi_{k,\omega} : \mathbb{D} \to \mathcal{H}_k$ maps each persistence diagram $D \in \mathbb{D}$ to the RKHS $\mathcal{H}_k$ associated to k:

$$\Psi_{k,\omega}(D) = \sum_{x \in D} \omega(x) k(\cdot, x).$$

This feature map induces the following *persistence weighted kernel (PWK)* $K_{k,\omega} : \mathbb{D} \times \mathbb{D} \to \mathbb{R}$:

$$K_{k,\omega}(D, E) = \langle \Psi_{k,\omega}(D), \Psi_{k,\omega}(E) \rangle_{\mathcal{H}_k} = \sum_{x \in D;\ y \in E} \omega(x)\omega(y)k(x, y). \tag{13.4}$$



PWGK (Persistence weighted Gaussian kernel)

- Specific choice of the weight function ω:
  - given a persistence point $x = (b, d)$, let $pers(x) = \left| d - b \right|$
  - then, we define $\omega_{arc}(x) = \arctan\left(C \cdot pers(x)^p\right)$, where C is some constant and p a positive integer.

- Using the 2D Gaussian kernel $k_G\left(x, y\right) = e^{-\frac{||x-y||^2}{2\tau^2}}$, and weight function ω_{arc}, the resulting kernel for persistence diagrams is called PWGK
  - i.e, $k_G^{\omega_{arc}}\left(x, y\right) = \omega_{arc}(x)\omega_{arc}\left(y\right)k_G(x, y)$

Other kernels

- Sliced Wasserstein kernel = OT + TDA
- Persistence Fisher kernel = Information Geometry + TDA

Sliced Wasserstein kernel (SW)

- [Carriere et al., 2017] define a kernel like $k(x, y) = \exp(-\frac{f(x, y)}{2\sigma^2})$
- Here it constructs a positive kernel via exponentiating a negative kernel!

Sliced Wasserstein kernel (SW)

- [Carriere et al., 2017] define a kernel like $k(x, y) = \exp(-\frac{f(x, y)}{2\sigma^2})$
- Here it constructs a positive kernel via exponentiating a negative kernel!

Theorem 13.4. *Given X and $\phi : X \times X \to \mathbb{R}$, the kernel ϕ is negative semi-definite if and only if $e^{-t\phi}$ is positive semi-definite for all $t > 0$.*

Sliced Wasserstein kernel (SW)

- [Carriere et al., 2017] define a kernel like $k(x,y) = \exp(-\frac{f(x,y)}{2\sigma^2})$
- Here it constructs a positive kernel via exponentiating a negative kernel!

Theorem 13.4. *Given X and $\phi : X \times X \to \mathbb{R}$, the kernel ϕ is negative semi-definite if and only if $e^{-t\phi}$ is positive semi-definite for all $t > 0$.*

- First construct the so-called Sliced Wasserstein distance d_{SW}, which turns out to be negative (semi-)definite.
- Then use d_{SW} to construct SW kernel via the above theorem.

Wasserstein Distance

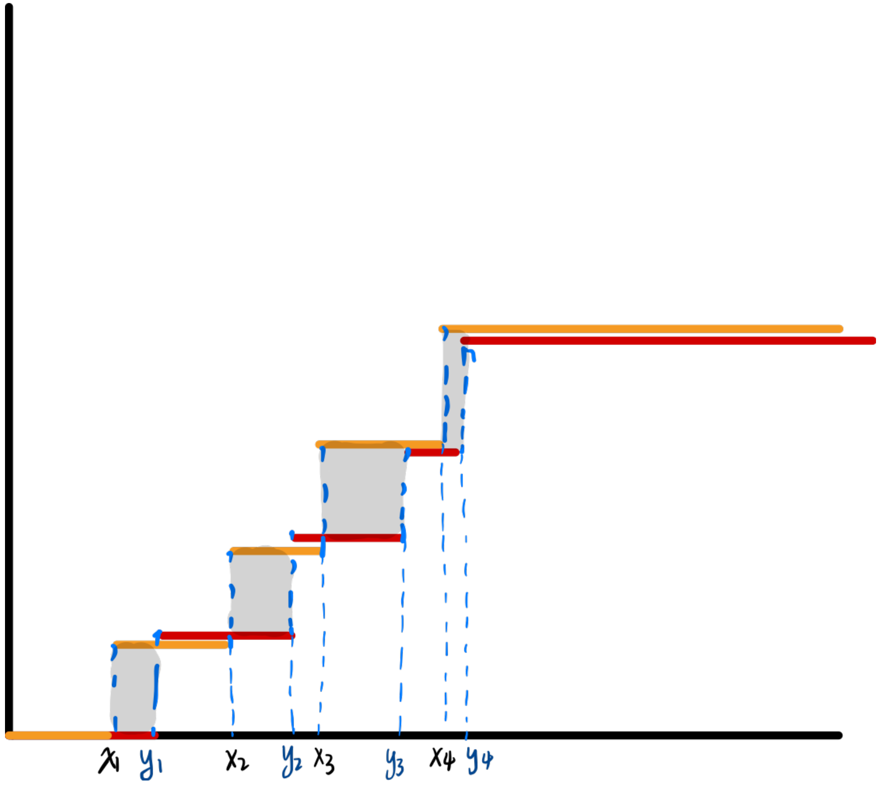
- A *transport plan* between two probability measures $\mu, \ \nu$ on $\mathbb{R}^d$ is a probability measure π on $\mathbb{R}^d \times \mathbb{R}^d$ such that for every $A, B \subseteq \mathbb{R}^d$, $\pi(A \times \mathbb{R}^d) = \mu(A)$ and $\pi(\mathbb{R}^d \times B) = \mu(B)$.
- The p-cost of a transport plan π is:

$$C_p(\pi) = \left(\int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^p \pi(dx \times dy) \right)^{1/p}$$

- The *Wasserstein distance* of order p between $\mu, \ \nu$ on $\mathbb{R}^d$ with finite p -moment
  - $W_p(\mu, \ \nu) =$ the minimum p-cost $C_p(\pi)$ of any transport plan π between μ and ν .

Wasserstein distance on real line

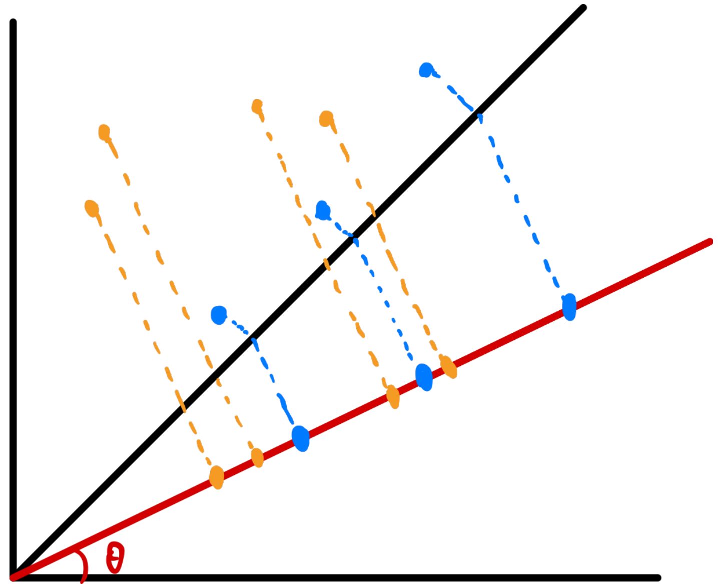
Remark 2.2. *For two unnormalized uniform empirical measures* $\mu = \sum_{i=1}^{n} \delta_{x_i}$ *and* $\nu = \sum_{i=1}^{n} \delta_{y_i}$ *of the same size, with ordered* $x_1 \leq \cdots \leq x_n$ *and* $y_1 \leq \cdots \leq y_n$*, one has:* $\mathcal{W}(\mu, \nu) = \sum_{i=1}^{n} |x_i - y_i| = \|X - Y\|_1$*, where* $X = (x_1, \cdots, x_n) \in \mathbb{R}^n$ *and* $Y = (y_1, \cdots, y_n) \in \mathbb{R}^n$*.*



Sliced-Wasserstein distance

Definition 13.6 (Sliced-Wasserstein distance)**.** Given a unit vector $\theta \in \mathbb{S}^1 \subseteq \mathbb{R}^2$, let $L(\theta)$ denote the line $\{\lambda\theta \mid \lambda \in \mathbb{R}\}$. Let $\pi_\theta : \mathbb{R}^2 \to L(\theta)$ be the orthogonal projection of the plane onto $L(\theta)$. Given two persistence diagrams D and E, set $\mu_D^\theta := \sum_{p \in D} \delta_{\pi_\theta(p)}$ and $\overline{\mu}_D^\theta := \sum_{p \in D} \delta_{\pi_\theta \circ \pi_\Delta(p)}$, where $\pi_\Delta : \mathbb{R}^2 \to \Delta$ is the orthogonal projection onto the diagonal $\Delta = \{(x, x) \mid x \in \mathbb{R}\}$. Set μ_E^θ and $\overline{\mu}_E^\theta$ in a symmetric manner. Then the *Sliced Wasserstein distance* between D and E is defined as:

$$\mathsf{d}_{SW}(D, E) := \frac{1}{2\pi} \int_{\mathbb{S}^1} \mathcal{W}(\mu_D^\theta + \overline{\mu}_E^\theta,\ \mu_E^\theta + \overline{\mu}_D^\theta) d\theta.$$



Sliced-Wasserstein distance

Definition 13.6 (Sliced-Wasserstein distance). Given a unit vector $\theta \in \mathbb{S}^1 \subseteq \mathbb{R}^2$, let $L(\theta)$ denote the line $\{\lambda\theta \mid \lambda \in \mathbb{R}\}$. Let $\pi_\theta : \mathbb{R}^2 \to L(\theta)$ be the orthogonal projection of the plane onto $L(\theta)$. Given two persistence diagrams D and E, set $\mu_D^\theta := \sum_{p \in D} \delta_{\pi_\theta(p)}$ and $\overline{\mu}_D^\theta := \sum_{p \in D} \delta_{\pi_\theta \circ \pi_\Delta(p)}$, where $\pi_\Delta : \mathbb{R}^2 \to \Delta$ is the orthogonal projection onto the diagonal $\Delta = \{(x, x) \mid x \in \mathbb{R}\}$. Set μ_E^θ and $\overline{\mu}_E^\theta$ in a symmetric manner. Then the *Sliced Wasserstein distance* between D and E is defined as:

$$\mathsf{d}_{SW}(D, E) := \frac{1}{2\pi} \int_{\mathbb{S}^1} \mathcal{W}(\mu_D^\theta + \overline{\mu}_E^\theta,\ \mu_E^\theta + \overline{\mu}_D^\theta)\, d\theta.$$

Sliced-Wasserstein distance

Definition 13.6 (Sliced-Wasserstein distance). Given a unit vector $\theta \in \mathbb{S}^1 \subseteq \mathbb{R}^2$, let $L(\theta)$ denote the line $\{\lambda\theta \mid \lambda \in \mathbb{R}\}$. Let $\pi_\theta : \mathbb{R}^2 \to L(\theta)$ be the orthogonal projection of the plane onto $L(\theta)$. Given two persistence diagrams D and E, set $\mu_D^\theta := \sum_{p\in D} \delta_{\pi_\theta(p)}$ and $\overline{\mu}_D^\theta := \sum_{p\in D} \delta_{\pi_\theta\circ\pi_\Delta(p)}$, where $\pi_\Delta : \mathbb{R}^2 \to \Delta$ is the orthogonal projection onto the diagonal $\Delta = \{(x,x) \mid x \in \mathbb{R}\}$. Set μ_E^θ and $\overline{\mu}_E^\theta$ in a symmetric manner. Then the *Sliced Wasserstein distance* between D and E is defined as:

$$\mathrm{d}_{SW}(D,E) := \frac{1}{2\pi}\int_{\mathbb{S}^1} \mathcal{W}(\mu_D^\theta + \overline{\mu}_E^\theta,\ \mu_E^\theta + \overline{\mu}_D^\theta)d\theta.$$

- The positive semi-definite *Sliced Wasserstein kernel* k_{SW} is defined as:
  - $k_{SW}(D,\ E) := e^{-\frac{d_{SW}(D,\ E)}{2\tau^2}}$ for some $\tau > 0$

Persistence Images

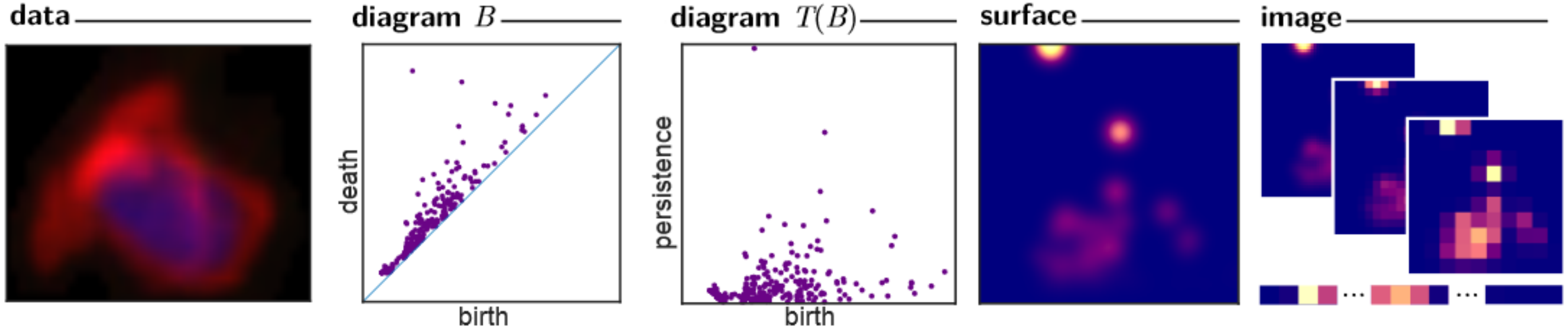


Figure 1: Algorithm pipeline to transform data into a persistence image.

Image taken from [Adams et al., 2017]

- Given a persistence diagram B
  - $T : \mathbb{R}^2 \to \mathbb{R}^2$ such that $T(x, y) = (x, y - x)$
  - Persistence surface ρ_B: $R^2 \to R$ where
    - $$\rho_B(z) = \sum_{u \in T(B)} f(u) \cdot \phi_u(z)$$

Persistence Images

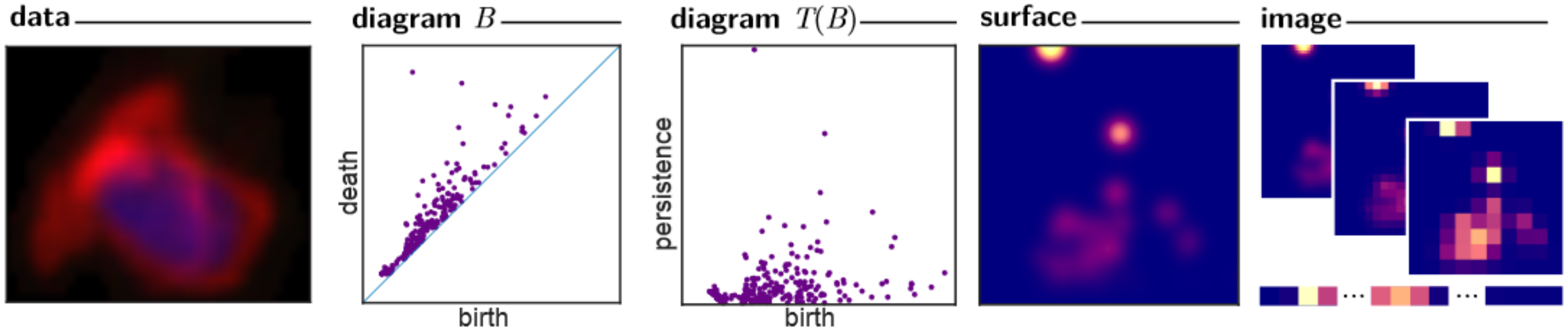


Figure 1: Algorithm pipeline to transform data into a persistence image.

Image taken from [Adams et al., 2017]

- Given a persistence diagram B
  - $T : \mathbb{R}^2 \to \mathbb{R}^2$ such that $T(x, y) = (x, y - x)$
  - Persistence surface $\rho_B : R^2 \to R$ where
    - $\rho_B(z) = \sum_{u \in T(B)} f(u) \cdot \phi_u(z)$

e.g, $\phi_u(z) = \frac{1}{2\pi\sigma^2} e^{-\frac{||z - u||^2}{2\sigma^2}}$

Persistence Images

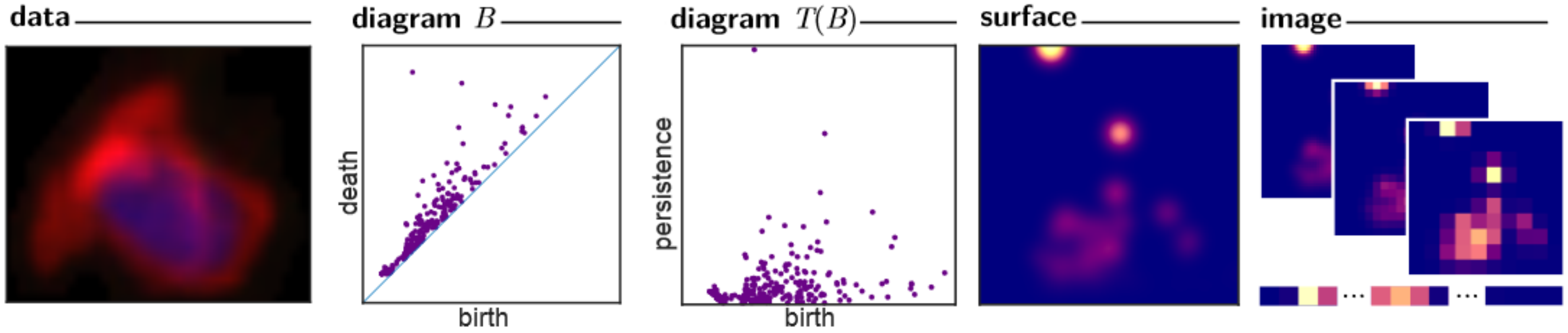


Figure 1: Algorithm pipeline to transform data into a persistence image.

Image taken from [Adams et al., 2017]

- Given a persistence diagram B
  - $T : \mathbb{R}^2 \to \mathbb{R}^2$ such that $T(x, y) = (x, y - x)$
  - Persistence surface $\rho_B : R^2 \to R$ where
    - $\rho_B(z) = \sum_{u \in T(B)} f(u) \cdot \phi_u(z)$

weight function

e.g, $\phi_u(z) = \frac{1}{2\pi\sigma^2} e^{-\frac{||z-u||^2}{2\sigma^2}}$

Persistence Images

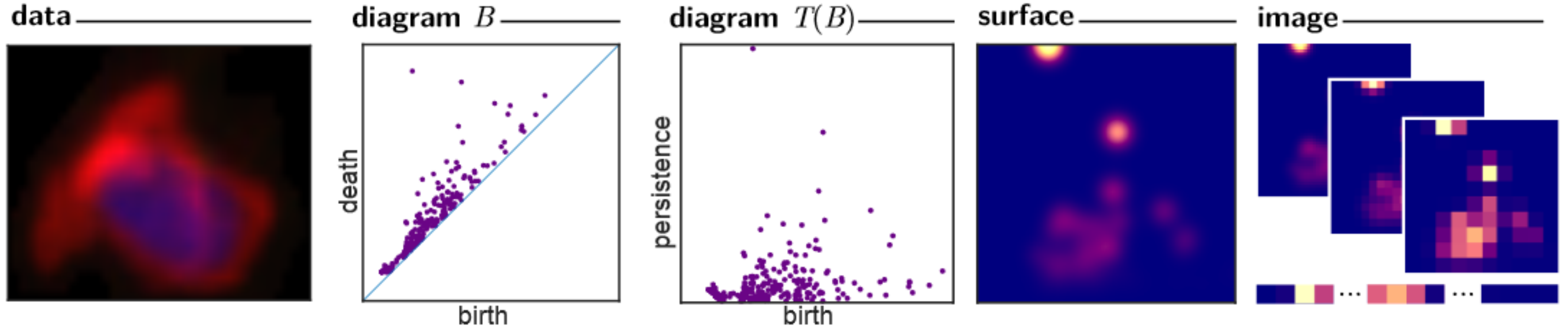


Figure 1: Algorithm pipeline to transform data into a persistence image.

Image taken from [Adams et al., 2017]

- Given a persistence diagram B
  - $T : \mathbb{R}^2 \to \mathbb{R}^2$ such that $T(x, y) = (x, y - x)$
  - Persistence surface ρ_B: $R^2 \to R$ where
    - $\rho_B(z) = \sum_{u \in T(B)} f(u) \cdot \phi_u(z)$
  - Persistence image:
    - I_B: a discretization of ρ_B: $I_B[p] = \int_p \rho_B dxdy$

- Metric:
  - We can simply use p-norms of vector difference $||I_D - I_E||_p$

- Metric:
  - We can simply use p-norms of vector difference $||I_D - I_E||_p$

Theorem 13.3. *Suppose persistence images are computed with the normalized Gaussian distribution with variance σ^2 and weight function $\omega : \mathbb{R}^2 \to \mathbb{R}$. Then the persistence images are stable w.r.t. the* 1*-Wasserstein distance between persistence diagrams. More precisely, given two finite and bounded persistence diagrams D and E, we have:*

$$|| \mathrm{I}_D - \mathrm{I}_E||_1 \leq \left(\sqrt{5}|\nabla\omega| + \sqrt{\frac{10}{\pi}} \frac{||\omega||_\infty}{\sigma} \right) \cdot \mathrm{d}_{W,1}(D, E).$$

Here, $\nabla\omega$ stands for the gradient of ω, and $|\nabla\omega| = \sup_{z\in\mathbb{R}^2} ||\nabla\omega||_2$ is the maximum norm of the gradient vector of ω at any point in $\mathbb{R}^2$. The same upper bound holds for $|| \mathrm{I}_D - \mathrm{I}_E||_2$ and $|| \mathrm{I}_D - \mathrm{I}_E||_\infty$ as well.

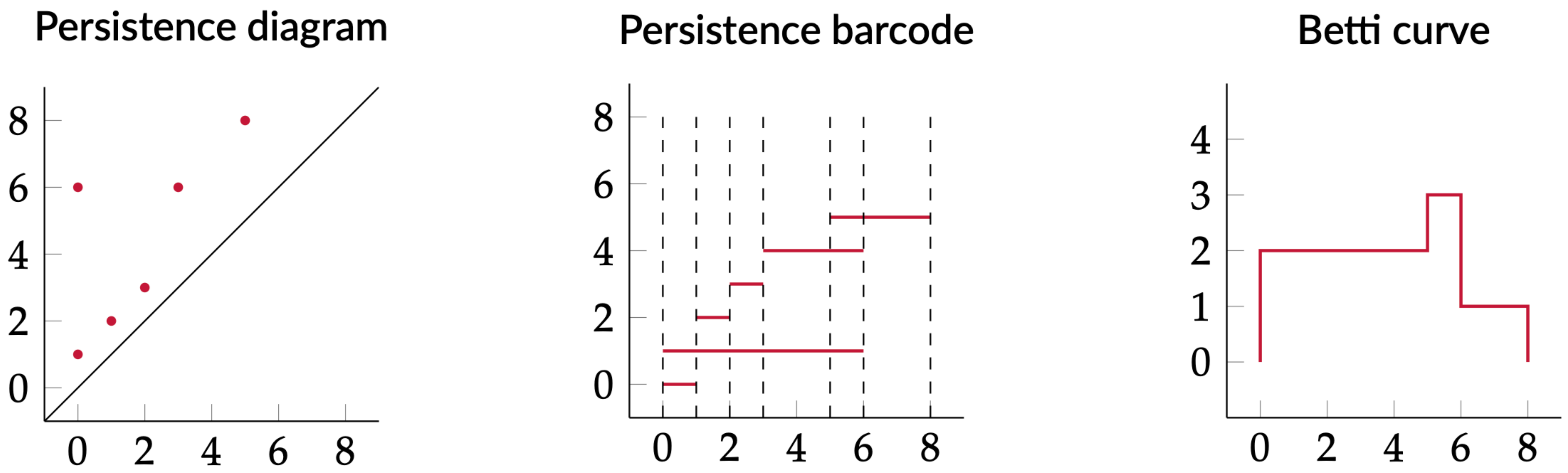
Packages

- scikit-tda persistence image
- Tutorial based on giotto-tda

Other methods

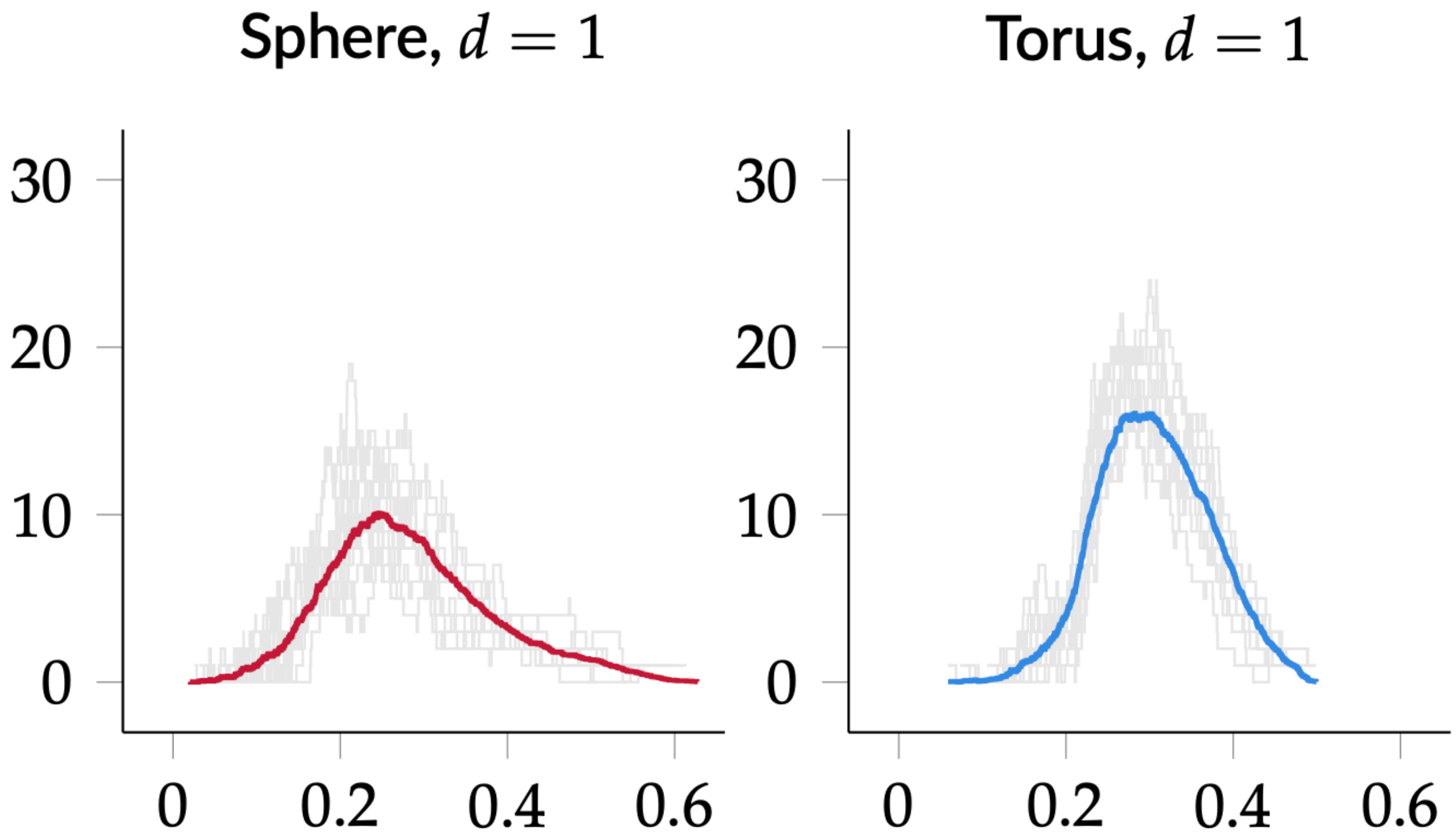
Betti curve

- [Rieck et al. 2019]



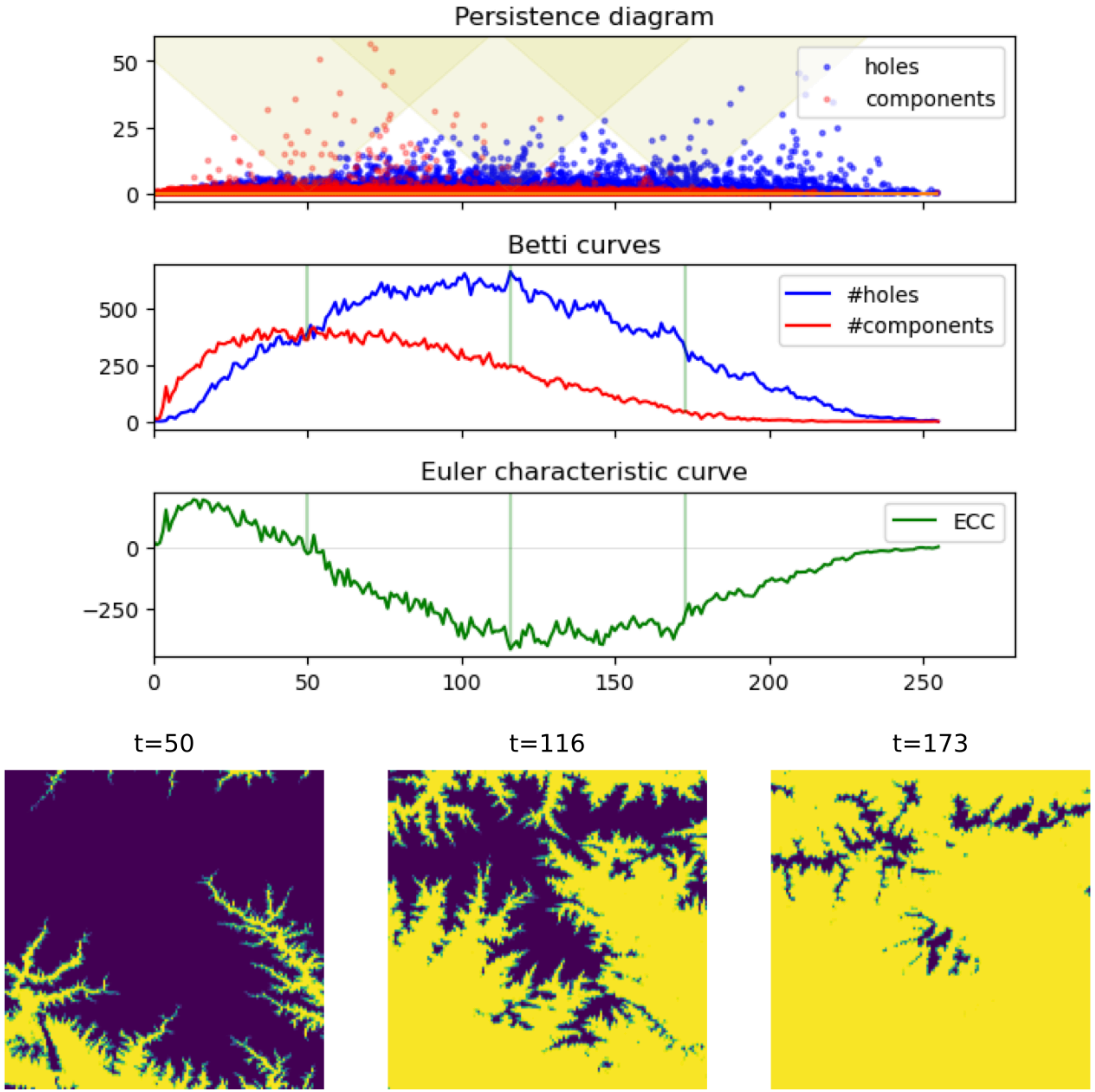
Courtesy of Bastian Rieck

Betti curve - bootstrap



Euler characteristic curve

- [Wang et al. 2023]
- $\chi(X) = \sum (-1)^i \beta_i(X)$
- GPU implementation
- Kernel available



Concluding remarks

- These kernels / vector representations make it easier to use ML methods on persistence-based features.
- Later we will talk about neural network architectures for handling PDs.

FIN