

PH-STAT

Moo K. Chung, Zijian Chen

University of Wisconsin-Madison
`{mkchung, zijian.chen}@wisc.edu`

Abstract. The PH-STAT toolbox performs various persistent homology based topological data analysis in MATLAB. The code and manual are distributed in <https://github.com/laplcebeltrami/ISBT2023TDA/tree/main/PH-STAT>.

1 Simplicial homology

A high dimensional object can be approximated by the point cloud data X consisting of p number of points. If we connect points of which distance satisfy a given criterion, the connected points start to recover the topology of the object. Hence, we can represent the underlying topology as a collection of the subsets of X that consists of nodes which are connected [5,4,3]. Given a point cloud data set X with a rule for connections, the topological space is a simplicial complex and its element is a simplex [16]. For point cloud data, the Delaunay triangulation is probably the most widely used method for connecting points. The Delaunay triangulation represents the collection of points in space as a graph whose face consists of triangles. Another way of connecting point cloud data is based on Rips complex often studied in persistent homology.

Homology is an algebraic formalism to associate a sequence of objects with a topological space [3]. In persistent homology, the algebraic formalism is usually built on top of objects that are hierarchically nested such as morse filtration, graph filtration and dendrograms. Formally homology usually refers to homology groups which are often built on top of a simplicial complex for point cloud and network data [8].

The k -simplex σ is the convex hull of $v + 1$ independent points v_0, \dots, v_k . A point is a 0-simplex, an edge is a 1-simplex, and a filled-in triangle is a 2-simplex. A *simplicial complex* is a finite collection of simplices such as points (0-simplex), lines (1-simplex), triangles (2-simplex) and higher dimensional counter parts [3]. A k -skeleton is a simplex complex of up to k simplices. Hence a graph is a 1-skeleton consisting of 0-simplices (nodes) and 1-simplices (edges). There are various simplicial complexes. The most often used simplicial complex in persistent homology is the Rips complex.

2 Rips complex

The Vietoris–Rips or Rips complex is the most often used simplicial complex in persistent homology. Let $X = \{x_0, \dots, x_p\}$ be the set of n points in \mathbb{R}^d . The distance matrix between points in X is given by $w = (w_{ij})$ where w_{ij} is the distance between points x_i and x_j . Then the Rips complex $R_\epsilon(X)$ is defined as follows [2,6]. The Rips complex is a collection of simplicial complexes parameterized by ϵ . The complex $R_\epsilon(X)$ captures the topology of the point set X at a scale of ϵ or less.

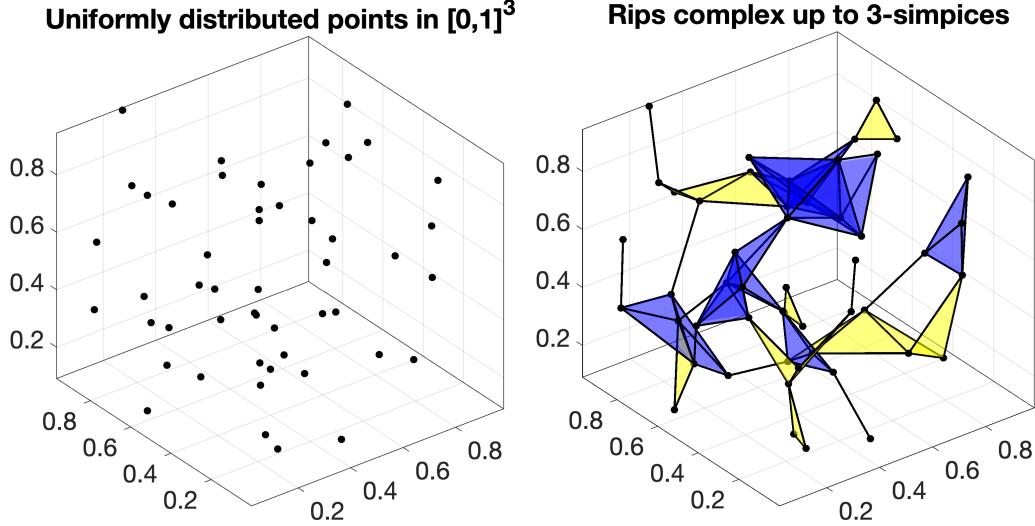


Fig. 1. Left: 50 randomly distributed points X in $[0,1]^3$. Right: Rips complex $R_{0.3}(X)$ within radius 0.3 containing 106 1-simplices, 75 2-simplices (yellow) and 22 3-simplices (blue).

- The vertices of $R_\epsilon(X)$ are the points in X .
- If the distance w_{ij} is less than or equal to ϵ , then there is an edge connecting points x_i and x_j in $R_\epsilon(X)$.
- If the distance between any two points in $x_{i_0}, x_{i_1}, \dots, x_{i_k}$ is less than or equal to ϵ , then there is a k -simplex in $R_\epsilon(X)$ whose vertices are $x_{i_0}, x_{i_1}, \dots, x_{i_k}$.

While a graph has at most 1-simplices, the Rips complex has at most k -simplices. In practice, the Rips complex is usually computed following the above definition iteratively adding simplices of increasing dimension. Given $p+1$ number of points, there are potentially up to $\binom{p+1}{k}$ k -simplices making the data representation extremely inefficient as the radius ϵ increases. Thus, we restrict simplices of dimension up to k in practice. Such a simplicial complex is called the k -skeleton. It is implemented as `PH_rips.m`, which inputs the matrix X of size $p \times d$, dimension k and radius e . Then outputs the structured array S containing the collection of nodes, edges, faces up to k -simplices. For instance, the Rips complex up to 3-simplices in Figure 1 is created using

```

p=50; d=3;
X = rand(p, d);
S= PH_rips(X, 3, 0.3)
PH_rips_display(X,S);

S =
4×1 cell array

```

```
{ 50×1 double}
{106×2 double}
{ 75×3 double}
{ 22×4 double}
```

The Rips complex is then displayed using `PH_rips_display.m` which inputs node coordinates \mathbf{X} and simplicial complex \mathbf{S} .

3 Boundary matrix

Given a simplicial complex K , the boundary matrices B_k represent the boundary operators between the simplices of dimension k and $k - 1$. Let C_k be the collection of k -simplices. Define the k -th boundary map

$$\partial_k : C_k \rightarrow C_{k-1}$$

as a linear map that sends each k -simplex σ to a linear combination of its $k - 1$ faces

$$\partial_k \sigma = \sum_{\tau \in F_k(\sigma)} (-1)^{\text{sgn}(\tau, \sigma)} \tau,$$

where $F_k(\sigma)$ is the set of $k - 1$ faces of σ , and $\text{sgn}(\tau, \sigma)$ is the sign of the permutation that sends the vertices of τ to the vertices of σ . This expression says that the boundary of a k -simplex σ is the sum of all its $(k - 1)$ -dimensional faces, with appropriate signs determined by the orientation of the faces. The signs alternate between positive and negative depending on the relative orientation of the faces, as determined by the permutation that maps the vertices of one face to the vertices of the other face. The k -th boundary map removes the filled-in interior of k -simplices. The vector spaces $C_k, C_{k-1}, C_{k-2}, \dots$ are then sequentially nested by boundary operator ∂_k [3]:

$$\dots \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} C_{k-2} \xrightarrow{\partial_{k-2}} \dots . \quad (1)$$

Such nested structure is called the *chain complex*.

Consider a filled-in triangle $\sigma = [v_1, v_2, v_3] \in C_2$ with three vertices v_1, v_2, v_3 in Figure 2. The boundary map ∂_k applied to σ resulted in the collection of three edges that forms the boundary of σ :

$$\partial_2 \sigma = [v_1, v_2] + [v_2, v_3] + [v_3, v_1] \in C_1. \quad (2)$$

If we give the direction or orientation to edges such that

$$[v_3, v_1] = -[v_1, v_3],$$

and use edge notation $e_{ij} = [v_i, v_j]$, we can write (2) as

$$\partial_2 \sigma = e_{12} + e_{23} + e_{31} = e_{12} + e_{23} - e_{13}.$$

The boundary map can be represented as a boundary matrix $\boldsymbol{\partial}_k$ with respect to a basis of the vector spaces C_k and C_{k-1} , where the rows of $\boldsymbol{\partial}_k$ correspond to the basis elements of C_k and the columns correspond to the basis elements of C_{k-1} . The (i, j) entry of $\boldsymbol{\partial}_k$ is given by the coefficient of the j th basis element in the linear combination of the $k - 1$ faces

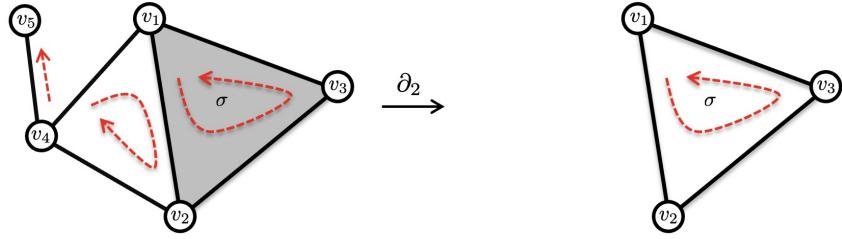


Fig. 2. A simplicial complex with 5 vertices and 2-simplex $\sigma = [v_1, v_2, v_3]$ with a filled-in face (colored gray). After boundary operation ∂_2 , we are only left with 1-simplices $[v_1, v_2] + [v_2, v_3] + [v_3, v_1]$, which is the boundary of the filled in triangle. The complex has a single connected component ($\beta_0 = 1$) and a single 1-cycle. The dotted red arrows are the orientation of simplices.

of the i th basis element in C_k . The boundary matrix is the higher dimensional version of the incidence matrix in graphs [10,9,15] showing how $(k-1)$ -dimensional simplices are forming k -dimensional simplex. The (i, j) entry of ∂_k is one if τ is a face of σ otherwise zero. The entry can be -1 depending on the orientation of τ . For the simplicial complex in Figure 2, the boundary matrices are given by

$$\partial_2 = \begin{pmatrix} \sigma \\ e_{12} & 1 \\ e_{23} & 1 \\ e_{31} & 1 \\ e_{24} & 0 \\ e_{41} & 0 \\ e_{45} & 0 \end{pmatrix}$$

$$\partial_1 = \begin{pmatrix} e_{12} & e_{23} & e_{31} & e_{24} & e_{41} & e_{45} \\ v_1 & -1 & 0 & 1 & 0 & 1 & 0 \\ v_2 & 1 & -1 & 0 & -1 & 0 & 0 \\ v_3 & 0 & 1 & -1 & 0 & 0 & 0 \\ v_4 & 0 & 0 & 0 & 1 & -1 & -1 \\ v_5 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\partial_0 = \begin{pmatrix} v_1 & v_2 & v_3 & v_4 & v_5 \\ 0 & (0 & 0 & 0 & 0 & 0) \end{pmatrix}.$$

In example in Figure 3-left, `PH_rips(X,3,0.5)` gives

>> S{1}

```
1
2
3
4
5
```

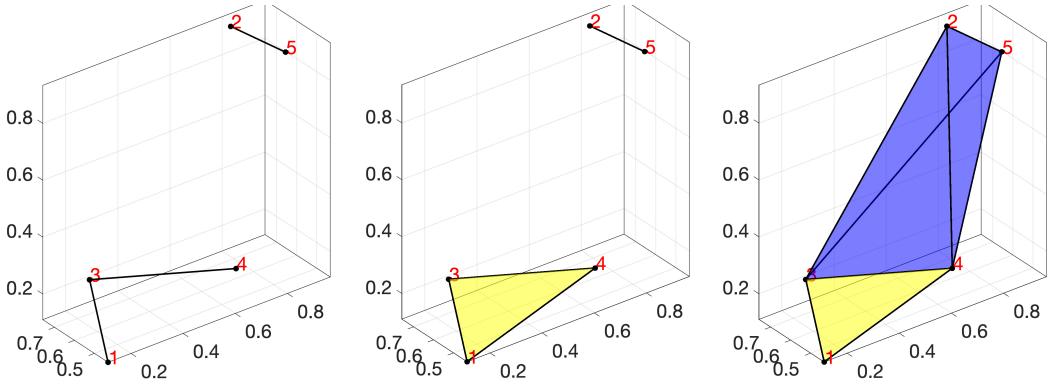


Fig. 3. Examples of boundary matrix computation. From the left to right, the radius is changed to 0.5, 0.6 and 1.0.

```
>> S{2}
```

1	3
2	5
3	4

`PH_boundary.m` only use node set `S{1}` and edge set `S{2}` in building boundary matrix `B1` saving computer memory.

```
>> B{1}
```

-1	0	0
0	-1	0
1	0	-1
0	0	1
0	1	0

The columns of boundary matrix `B{1}` is indexed with the edge set in `S{2}` such that the first column corresponds to edge [1,3]. Any other potential edges [2,3] that are not connected is simply ignored to save computer memory.

When we increase the filtration value and compute `PH_rips(X, 3, 0.6)`, a triangle is formed (yellow colored) and `S{3}` is created (Figure 3-middle).

```
>> S{2}
```

1	3
1	4
2	5
3	4

```
>> S{3}
```

```
1      3      4
```

Correspondingly, the boundary matrices change to

```
>> B{1}
```

```
-1    -1    0    0
 0     0   -1    0
 1     0    0   -1
 0     1    0    1
 0     0    1    0
```

```
>> B{2}
```

```
1
-1
 0
 1
```

From the edge set $S\{2\}$ that forms the row index for boundary matrix $B\{2\}$, we have $[1,3]$ - $[1,4]$ + $[3,4]$ that forms the triangle $[1,3,4]$.

When we increase the filtration value further and compute `PH_rips(X,3,1)`, a tetrahedron is formed (blue colored) and $S\{4\}$ is created (Figure 3-right).

```
>> S{3}
```

```
1      3      4
 2      3      4
 2      3      5
 2      4      5
 3      4      5
```

```
>> S{4}
```

```
2      3      4      5
```

Correspondingly, the boundary matrix $B\{3\}$ is created

```
>> B{3}
```

```
0
-1
 1
-1
 1
```

The easiest way to check the computation is correct is looking at the sign of triangles in - $[2,3,4]$ + $[2,3,5]$ - $[2,4,5]$ + $[3,4,5]$. Using the right hand thumb rule, which puts the orientation of triangle $[3,4,5]$ toward the center of the tetrahedron, the orientation of all the triangles are toward the center of the tetrahedron. Thus, the signs are correctly

assigned. Since computer algorithms are built inductively, the method should work correctly in higher dimensional simplices.

4 Homology group

The image of boundary map is defined as

$$\text{im}\partial_{k+1} = \{\partial_{k+1}\sigma | \sigma \in C_{k+1}\} \subset C_k,$$

which is a collection of boundaries. The elements of the image of ∂_{k+1} are called k -boundaries, and they represent k -dimensional features that can be filled in by $(k+1)$ -dimensional simplices. For instance, if we take the boundary ∂_2 of the triangle σ in Figure 2, we obtain a 1-cycle with edges e_{12}, e_{23}, e_{31} . The image of the boundary matrix B_{k+1} is the subspace spanned by its columns. The column space can be found by the Gaussian elimination or singular value decomposition.

The kernel of boundary map is defined as

$$\ker\partial_k = \{\sigma \in C_k | \partial_k\sigma = 0\},$$

which is a collection of cycles. The elements of the kernel of ∂_k are called cycles, since they form closed loops or cycles in the simplicial complex. The kernel of the boundary matrix B_k is spanned by eigenvectors v corresponding to zero eigenvalues of B_k .

The boundary map satisfy the property that the composition of any two consecutive boundary maps is the zero map, i.e.,

$$\partial_{k-1} \circ \partial_k = 0. \quad (3)$$

This reflect the fact that the boundary of a boundary is always empty. We can apply the boundary operation ∂_1 further to $\partial_2\sigma$ in Figure 2 example and obtain

$$\begin{aligned} \partial_1\partial_2\sigma &= \partial_1e_{12} + \partial_1e_{23} + \partial_1e_{31} \\ &= v_2 - v_1 + v_3 - v_2 + v_1 - v_3 = 0. \end{aligned}$$

This property (3) implies that the image of ∂_k is contained in the kernel of ∂_{k-1} , i.e.,

$$\text{im}\partial_{k+1} \subset \ker\partial_k.$$

Further, the boundaries $\text{im}\partial_{k+1}$ form subgroups of the cycles $\ker\partial_k$. We can partition $\ker\partial_k$ into cycles that differ from each other by boundaries through the quotient space

$$H_k(K) = \ker\partial_k / \text{im}\partial_{k+1},$$

which is called the k -th homology group. $H_k(K)$ is a vector space that captures the k th topological feature or cycles in K . The elements of the k -th homology group are often referred to as k -dimensional cycles or k -cycles. Intuitively, it measures the presence of k -dimensional loops in the simplicial complex.¹

¹ To do list: Need a MATLAB function that represent $H_k(K)$ as a vector in relation to $\text{im}\partial_{k+1}$ and $\ker\partial_k$.

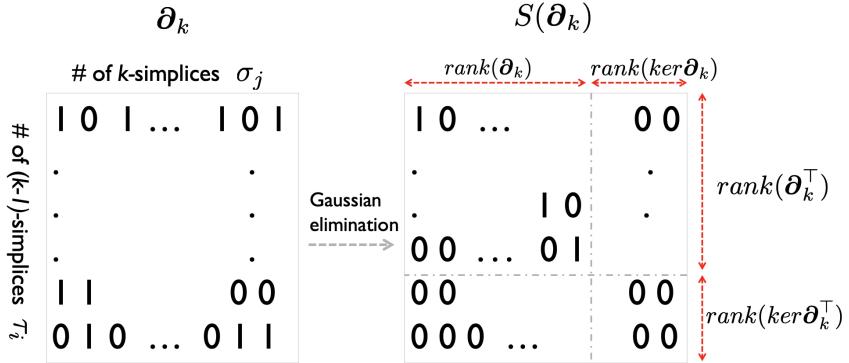


Fig. 4. The rank-nullity theorem for boundary matrix ∂_k , which states the dimension of the domain of ∂_k is the sum of the dimension of its image and the dimension of its kernel (nullity).

The rank of $H_k(K)$ is the k th Betti number of K , which is an algebraic invariant that captures the topological features of the complex K . Although we put direction in the boundary matrices by adding sign, the Betti number computation will be invariant of how we orient simplices. The k -th Betti number β_k is then computed as

$$\beta_k = rank(H_k) = rank(ker\partial_k) - rank(im\partial_{k+1}). \quad (4)$$

The 0-th Betti number is the number of connected components while the 1-st Betti number is the number of cycles. The Betti numbers β_k are usually algebraically computed by reducing the boundary matrix ∂_k to the Smith normal form $S(\partial_k)$, which has a block diagonal matrix as a submatrix in the upper left, via Gaussian elimination [3]. In the Smith normal form, we have the rank-nullity theorem for boundary matrix ∂_k , which states the dimension of the domain of ∂_k is the sum of the dimension of its image and the dimension of its kernel (nullity) (Figure 4). In $S(\partial_k)$, the number of columns containing only zeros is $rank(ker\partial_k)$, the number of k -cycles while the number of columns containing one is $rank(\partial_k)$, the number of k -cycles that are boundaries. Thus

$$\beta_k = rank(ker\partial_k) - rank(\partial_{k+1}). \quad (5)$$

The computation starts with initial rank computation

$$rank\partial_0 = 0, \quad rank(ker\partial_0) = p.$$

Example 1. The boundary matrices ∂_k in Figure 2 is transformed to the Smith normal form $S(\partial_k)$ after Gaussian elimination as

$$S(\partial_1) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad S(\partial_2) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

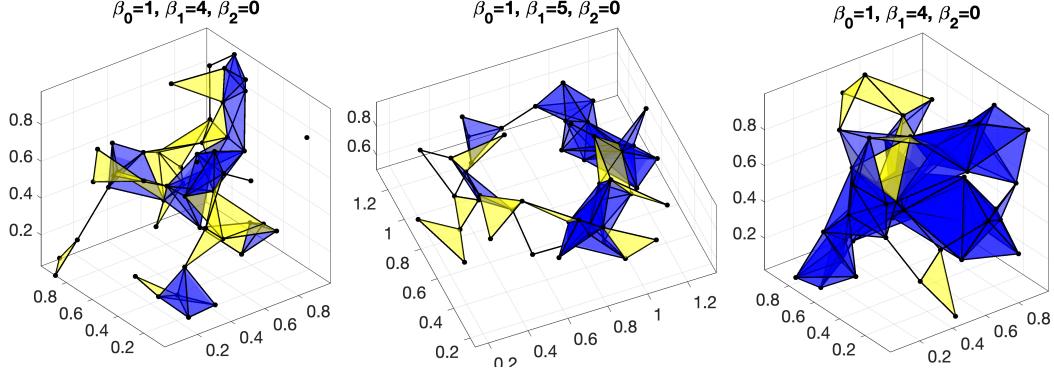


Fig. 5. Betti number computation on simplicial complex using `PH_betti.m` function.

From (5), the Betti number computation involves the rank computation of two boundary matrices. $\text{rank}(\partial_0) = 5$ is trivially the number of nodes in the simplicial complex. There are $\text{rank}(\ker \partial_1) = 2$ zero columns and $\text{rank}(\partial_1) = 4$ non-zero row columns. $\text{rank}(\partial_2) = 1$. Thus, we have

$$\begin{aligned}\beta_0 &= \text{rank}(\ker \partial_0) - \text{rank}(\partial_1) = 5 - 4 = 1, \\ \beta_1 &= \text{rank}(\ker \partial_1) - \text{rank}(\partial_2) = 2 - 1 = 1.\end{aligned}$$

Following the above worked out example, the Betti number computation is implemented as `betti = PH_betti.m` which inputs the boundary matrices generated by `PH_boundary.m`. The function outputs β_1, β_2, \dots . The function computes the $(d-1)$ -th Betti number as

```
betti(d)= rank(null(B{d-1})) - rank(B{d}).
```

Figure 5 displays few examples of Betti number computation on Rips complexes. The rank computation in most computational packages is through the singular value decomposition (SVD).

5 Hodge Laplacian

6 Rips filtrations

The Rips complex has the property that as the radius parameter value ϵ increases, the complex grows by adding new simplices. The simplices in the Rips complex at one radius parameter value are a subset of the simplices in the Rips complex at a larger radius parameter value. This nesting property is captured by the inclusion relation

$$\mathcal{R}\epsilon_0 \subset \mathcal{R}\epsilon_1 \subset \mathcal{R}\epsilon_2 \subset \dots$$

for $0 = \epsilon_0 \leq \epsilon_1 \leq \epsilon_2 \leq \dots$. This nested sequence of Rips complexes is called the *Rips filtration*, which is the main object of interest in persistent homology (Figure 6). The filtration

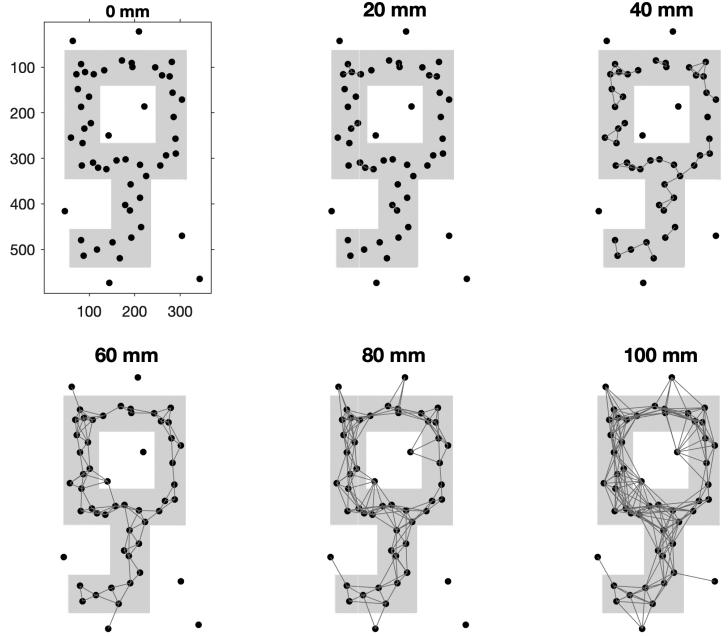


Fig. 6. Rips filtration on 1-skeleton of the point cloud data that was sampled along the underlying key shaped data. If two points are within the given radius, we connect them with an edge but do not form any other dimensional simplex. Such sparsity in Rips filtration can be more effective in practice. `PH_rips.m` limit the dimension of skeleton we build Rips filtrations and do not build every possible simplicial complexes.

values $\epsilon_0, \epsilon_1, \epsilon_2, \dots$ represent the different scales at which we are studying the topological structure of the point cloud. By increasing the filtration value ϵ , we are connecting more points, and therefore the size of the edge set, face set, and so on, increases.

The exponential growth in the number of simplices in the Rips complex as the number of vertices p increases can quickly become a computational bottleneck when working with large point clouds. For a fixed dimension k , the number of k -simplices in the Rips complex grows as $\mathcal{O}(p^k)$, which can make computations and memory usage impractical for large values of p . Furthermore, as the filtration value ϵ increases, the Rips complex becomes increasingly dense, with edges between every pair of vertices and filled triangles between every triple of vertices. Even for moderately sized point clouds, the Rips filtration can become very ineffective as a representation of the underlying data at higher filtration values. The complex becomes too dense to provide meaningful insights into the underlying topological structure of the data. To address these issues, various methods have been proposed to sparsify the Rips complex. One such method is the graph filtration first proposed in [12,11], which constructs a filtration based on a weighted graph representation of the data. The graph filtration can be more effective than the Rips filtration especially when the topological features of interest are related to the graph structure of the data.

7 Persistent diagrams

As the radius ϵ increases, the Rips complex $R_\epsilon(X)$ grows and contains a higher-dimensional simplex that merges two lower-dimensional simplices representing the death of the two lower-dimensional features and the birth of a new higher-dimensional feature. The persistent diagram is a plot of the birth and death times of features. We start by computing the homology groups of each of the simplicial complexes in the filtration.

Let $H_k(K_i)$ denote the k^{th} homology group of the simplicial complex K_i . We then track the appearance and disappearance of each homology class across the different simplicial complexes in the filtration. The birth time of a homology class is defined as the smallest radius ϵ_b for which the class appears in the filtration, and the death time is the largest radius ϵ_d for which the class is present. We then plot each homology class as a point in the two-dimensional plane as $(\epsilon_b, \epsilon_d) \in \mathbb{R}^2$. The collection of all these points is the persistence diagram for k -th homology group.

To track the birth and death times of homology classes, we need to identify when a new homology class is born or an existing homology class dies as the radius ϵ increases. We can do this by tracking the changes in the ranks of the boundary matrices. Specifically, a k -dimensional cycle is born when it appears as a new element in the kernel of ∂_k in a simplicial complex K_i that did not have it before, and it dies when it becomes a boundary in K_j for some $j > i$. Thus, we can compute the birth time ϵ_b of a k -dimensional homology class as the smallest radius for which it appears as a new element in the kernel of ∂_k . Similarly, we can compute the death time ϵ_d of the same class as the largest radius for which it is still a cycle in the simplicial complex K_j for some $j > i$. By tracking the changes in the ranks of boundary matrices, we can compute the birth and death times of homology classes and plot them in the persistence diagram for the k -th homology group. However, the computation is fairly demanding and not scale well.

³

Example 2. The example came from

8 Sampling persistent diagrams

Since there is no ground truth or common database for persistent diagrams, simulation based method can be often used to establish the ground truth. We will use a Dirichlet mixture in establishing the parametric model and sample points from the distribution. The persistent diagrams are scatter points in the unbounded domain

$$\mathcal{T}_\infty = \{(x_1, x_2) : x_2 \geq x_1\} \subset \mathbb{R}^2.$$

But this is not very convenient so we often constrain the domain as the bounded upper triangle

$$\mathcal{T} = \{(x_1, x_2) : x_2 \geq x_1, 0 \leq x_1, x_2 \leq 1\} \subset \mathbb{R}^2.$$

² Need a MATLAB function that performs Rips filtration in a scalable fashion.

³ Need a MATLAB function that computes the persistent diagram from Rips filtrations.

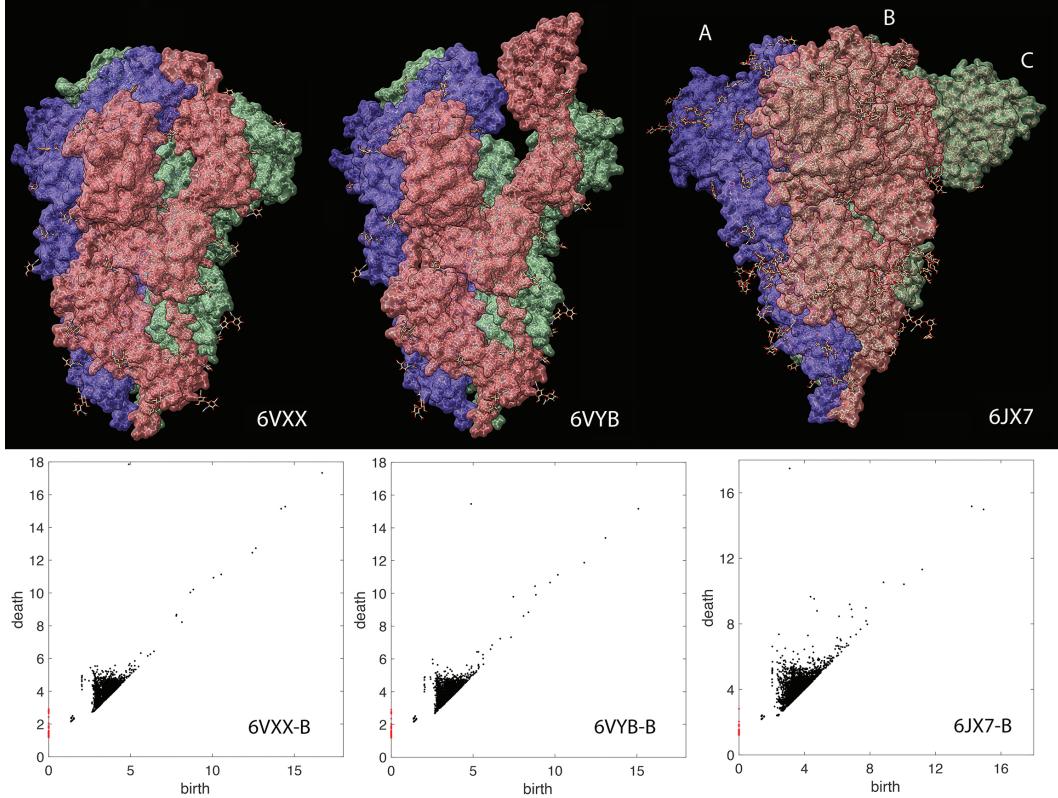


Fig. 7. Top: Spike proteins of the three different corona viruses. The spike proteins consist of three similarly shaped interwinding substructures identified as A (blue), B (red) and C (green) domains. Bottom: The persistent diagrams of spike proteins. The red dots are 0D homology and the black dots are 1D homology.

by scaling or thresholding data. We are interested in sampling points in \mathcal{T} using the Dirichlet distribution somehow. The Dirichlet-like distribution defined on this domain is given by

$$f_{\alpha}^{\mathcal{T}}(x_1, x_2) = \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)} x_1^{\alpha_1-1} (1-x_2)^{\alpha_2-1} (x_2 - x_1)^{\alpha_3-1}, \quad (6)$$

where $\alpha = (\alpha_1, \alpha_2, \alpha_3)$ are positive parameters [7]. To generate samples from $f_{\alpha}^{\mathcal{T}}$, we first need to sample from Dirichlet distribution defined on

$$\mathcal{T}_1 = \{(x_1, x_2, x_3) : x_1 + x_2 + x_3 = 1, x_1, x_2, x_3 \geq 0\} :$$

$$f_{\alpha}^{\mathcal{T}_1}(x_1, x_2, x_3) = \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)} x_1^{\alpha_1-1} x_2^{\alpha_2-1} x_3^{\alpha_3-1}.$$

This can be achieved using the Gamma distribution:

Theorem 1. [1,14] Let $Y_i \sim \text{Gamma}(\alpha_i, 1), i = 1, 2, 3$ be independent Gamma random variables with $\alpha_i > 0$, and let

$$X_i = \frac{Y_i}{\sum_{j=1}^3 Y_j}, \quad i = 1, \dots, 3.$$

Then (X_1, X_2, X_3) follows Dirichlet distribution on \mathcal{T}_1 with parameters $(\alpha_1, \alpha_2, \alpha_3)$.

Theorem 1 gives the Dirichlet distribution on \mathcal{T}_1 . Then through the change of coordinates, we transform the distribution from \mathcal{T}_1 to \mathcal{T} .

Theorem 2. Let $Y_i \sim \text{Gamma}(\alpha_i, 1), i = 1, 2, 3$ be independent Gamma random variables with $\alpha_i > 0$, and let

$$X_i = \frac{Y_i}{\sum_{j=1}^3 Y_j}, \quad i = 1, 2, 3.$$

Then $(X_1, 1 - X_2)$ follows the Dirichlet distribution on \mathcal{T} with parameters $(\alpha_1, \alpha_2, \alpha_3)$.

Proof. From Theorem 1, the random vector (X_1, X_2, X_3) follows the Dirichlet distribution on \mathcal{T}_1 with parameters $(\alpha_1, \alpha_2, \alpha_3)$. We then project this random vector onto \mathcal{T}_2 given by

$$\mathcal{T}_2 = \{(x_1, x_2) : x_1, x_2 \geq 0, x_1 + x_2 \leq 1\}.$$

For any point $(x_1, x_2, x_3) \in \mathcal{T}_1$, x_3 is uniquely determined as $x_3 = 1 - x_1 - x_2$. Thus, the Dirichlet distribution on \mathcal{T}_2 is simply given by

$$f_\alpha^{\mathcal{T}_2}(x_1, x_2) = \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)} x_1^{\alpha_1-1} x_2^{\alpha_2-1} (1 - x_1 - x_2)^{\alpha_3-1},$$

for $(x_1, x_2) \in \mathcal{T}_2$. We then transform \mathcal{T}_2 to \mathcal{T} by rotation $(x_1, x_2) \mapsto (x_1, 1 - x_2)$. Since the Jacobian of the rotation is 1, the density function \mathcal{T} is given by

$$f_\alpha^{\mathcal{T}}(x_1, x_2) = f_\alpha^{\mathcal{T}_2}(x_1, 1 - x_2) = \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)} x_1^{\alpha_1-1} (1 - x_2)^{\alpha_2-1} (x_2 - x_1)^{\alpha_3-1},$$

□

Figure 8 displays the transformation of sampled points from \mathcal{T}_1 to \mathcal{T}_2 and finally the target domain \mathcal{T} .

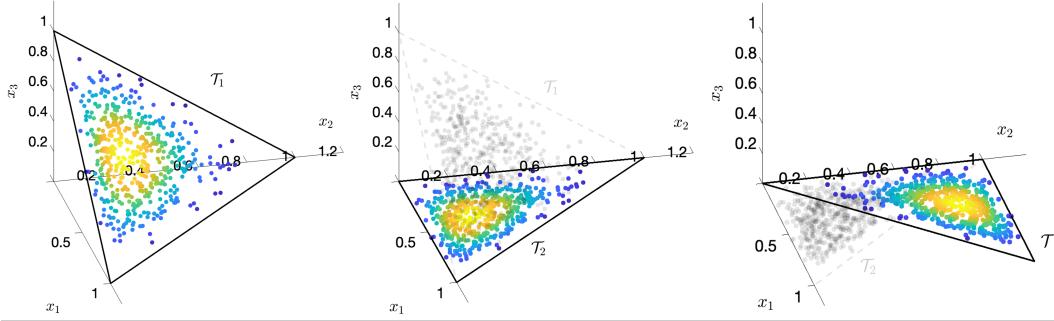


Fig. 8. Transformation of sampled points from \mathcal{T}_1 to \mathcal{T}_2 and finally the target domain \mathcal{T} . Points are sampled following the Dirichlet distribution in \mathcal{T}_1 with $\alpha = (3, 2, 3)$. If the problem is not explicitly obtaining the density in \mathcal{T} , equivalently we can simply sample in \mathcal{T}_1 and do the projection of the sampled points to \mathcal{T}_2 and then rotation to \mathcal{T} .

The codes are packaged into the function `Dirichlet_sample.m`, which inputs `alpha`, a 3×1 column vector and `n`, the number of samples. The output is a $n \times 2$ matrix. Each row of the output corresponds to one sample. To generate 500 scatter points from f_α^T with $\alpha = (3, 2, 3)$ (Figure 9-left), we run

```
alpha = [3,2,3]';
samples = Dirichlet_sample(alpha,1,500);
```

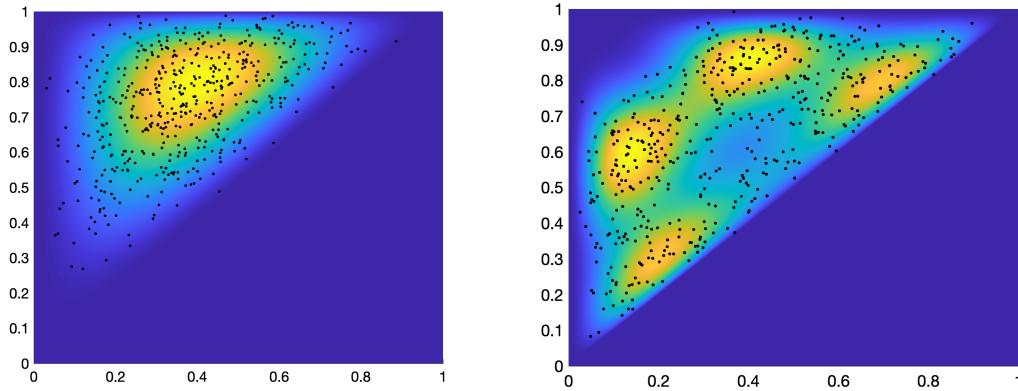


Fig. 9. Left: 500 samples from f_α^T with $\alpha = (3, 2, 3)$. The sampled points are overlaid on the contour plot of the theoretical densities. Right: 500 samples from mixture distribution f^T in (7). The sampled points are overlaid on the contour plot of the theoretical densities.

9 Sampling from a mixture of Dirichlet distributions

Sampling from a single Dirichlet distribution provides a concentric pattern with unimodal peak somewhere inside the triangle. This may not be realistic for various data. We propose sample from a mixture of Dirichlet distributions. Suppose $f_{\alpha_i}^{\mathcal{T}}$ are Dirichlet distributions with parameters α_i . A more realistic model is to sample from a mixture $\sum_{i=1}^k w_i f_{\alpha_i}^{\mathcal{T}}$ with $\sum_{i=1}^k w_i = 1$. This is a proper distribution in \mathcal{T} . This will provide multiple concentrated regions. To implement the sampling procedure, we first need to generate a uniform sample $U \sim \text{Unif}(0, 1)$. If $U \in (\sum_{i=1}^l w_i, \sum_{i=1}^{l+1} w_i)$, then we generate a sample from the distribution of the l -th component (i.e., $f_{\alpha_l}^{\mathcal{T}}$). This process is repeated until the desired number of samples from the mixture distribution is obtained [13]. To generate 500 scatter points from the mixture

$$f^{\mathcal{T}} = 0.25 f_{\alpha_1}^{\mathcal{T}} + 0.25 f_{\alpha_2}^{\mathcal{T}} + 0.25 f_{\alpha_3}^{\mathcal{T}} + 0.25 f_{\alpha_4}^{\mathcal{T}}, \quad (7)$$

with parameters $\alpha_1 = (3, 8, 2)$, $\alpha_2 = (8, 3, 2)$, $\alpha_3 = (7, 3, 8)$, $\alpha_4 = (3, 7, 8)$. The sampling is done through by extending the functionality of the previous MATLAB function `Dirichlet_sample.m`. The result is displayed in Figure 9-right.

```
alpha = [3,8,2;8,3,2;7,3,8;3,7,8]';
weight = [0.25,0.25,0.25,0.25];
samples = Dirichlet_sample(alpha, weight, 500);
```

References

1. Devroye, L.: Non-Uniform Random Variate Generation. Springer New York (2013)
2. Edelsbrunner, H., Harer, J.: Persistent homology - a survey. *Contemporary Mathematics* **453**, 257–282 (2008)
3. Edelsbrunner, H., Harer, J.: Computational topology: An introduction. American Mathematical Society (2010)
4. Ghrist, R.: Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society* **45**, 61–75 (2008)
5. Hart, J.: Computational topology for shape modeling. In: Proceedings of the International Conference on Shape Modeling and Applications. pp. 36–43 (1999)
6. Hatcher, A.: Algebraic topology. Cambridge University Press (2002)
7. Kotz, S., Balakrishnan, N., Johnson, N.: Continuous Multivariate Distributions, Volume 1: Models and Applications. Continuous Multivariate Distributions (2004)
8. Lee, H., Chung, M.K., K.H., Lee, D.: Hole detection in metabolic connectivity of Alzheimer’s disease using k-Laplacian. *MICCAI, Lecture Notes in Computer Science* **8675**, 297–304 (2014)
9. Lee, H., Chung, M., Choi, H., K., H., Ha, S., Kim, Y., Lee, D.: Harmonic holes as the submodules of brain network and network dissimilarity. *International Workshop on Computational Topology in Image Context, Lecture Notes in Computer Science* pp. 110–122 (2019)
10. Lee, H., Chung, M., Kang, H., Choi, H., Kim, Y., Lee, D.: Abnormal hole detection in brain connectivity by kernel density of persistence diagram and Hodge Laplacian. In: *IEEE International Symposium on Biomedical Imaging (ISBI)*. pp. 20–23 (2018)

⁴ Figure 9 should be changed to Figure 8 style

11. Lee, H., Chung, M., Kang, H., Kim, B.N., Lee, D.: Computing the shape of brain networks using graph filtration and Gromov-Hausdorff metric. *MICCAI, Lecture Notes in Computer Science* **6892**, 302–309 (2011)
12. Lee, H., Chung, M., Kang, H., Kim, B.N., Lee, D.: Discriminative persistent homology of brain networks. In: *IEEE International Symposium on Biomedical Imaging (ISBI)*. pp. 841–844 (2011)
13. McLachlan, G., Peel, D.: *Finite Mixture Models*. Wiley Series in Probability and Statistics, Wiley (2004)
14. Ng, K., Tian, G., Tang, M.: *Dirichlet and Related Distributions: Theory, Methods and Applications*. Wiley-Blackwell, United States (2011)
15. Schaub, M., Benson, A., Horn, P., Lippner, G., Jadbabaie, A.: Random walks on simplicial complexes and the normalized hodge laplacian. *arXiv preprint arXiv:1807.05044* (2018)
16. Zomorodian, A.: *Topology for computing*. Cambridge University Press, Cambridge (2009)