PH-STAT

Moo K. Chung, Zijian Chen

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

Abstract. The PH-STAT toolbox performs the various statistical inference on persistent homology in MATLAB. The code and manual are distributed in `https://github.com/laplcebeltrami/PH-STAT`.

1 Morse filtrations

In many applications, 1D functional data $f(t)$ is modeled as [40]

$$f(t) = \mu(t) + \epsilon(t),\ t \in \mathbb{R}, \tag{1}$$

where μ is the unknown mean signal to be estimated and ϵ is noise. In the usual statistical parametric mapping framework [22,29,62], inference on the model (1) proceeds as follows. If we denote an estimate of the signal by $\widehat{\mu}$, the residual $f-\widehat{\mu}$ gives an estimate of the noise. One then constructs a test statistic $T(t)$, corresponding to a given hypothesis about the signal. As a way to account for temporal correlation of the statistic $T(t)$, the global maximum of the test statistic over the search space $\mathcal{M}$ is taken as the subsequent test statistic. Hence a great deal of the signal processing and statistical literature has been devoted to determining the distribution of $\sup_{t\in\mathbb{M}} T(t)$ using the random field theory [56,62], permutation tests [45] and the Hotelling–Weyl volume of tubes calculation [43]. The use of the mean signal is one way of performing data reduction, however, this may not necessarily be the best way to characterize complex multivariate imaging data. Thus instead of using the mean signal, we can use persistent homology, which pairs local critical values [19,21,66]. It is intuitive that local critical values of $\widehat{\mu}$ approximately characterizes the shape of the continuous signal μ using only a finite number of scalar values. By pairing these local critical values in a nonlinear fashion and plotting them, one constructs the persistence diagram [17,19,42,64].

The function μ is called a *Morse function* if all critical values are distinct and non-degenerate, i.e., the Hessian does not vanish [41]. For a 1D Morse function $y = \mu(t)$, define sublevel set $R(y)$ as

$$R_y = \{t \in \mathbb{R} : \mu(t) \le y\}.$$

As we increase height $y_1 \le y_2 \le y_3 \le \cdots$, the sublevel set gets bigger such that

$$R_{y_1} \subset R_{y_2} \subset R_{y_3} \subset \cdots .$$

The sequence of the sublevel sets form a *Morse filtration* with filtration values $y_1, y_2, y_3, \cdots$. Let $\beta_0(R_y)$ be the 0-th Betti number of R_y, which counts the number of connected components in R_y. The number of connected components is the most often used topological invariant in applications [19]. $\beta_0(R_y)$ only changes its value as it passes through critical values (Figure 1). The birth and death of connected components in the Morse filtration is

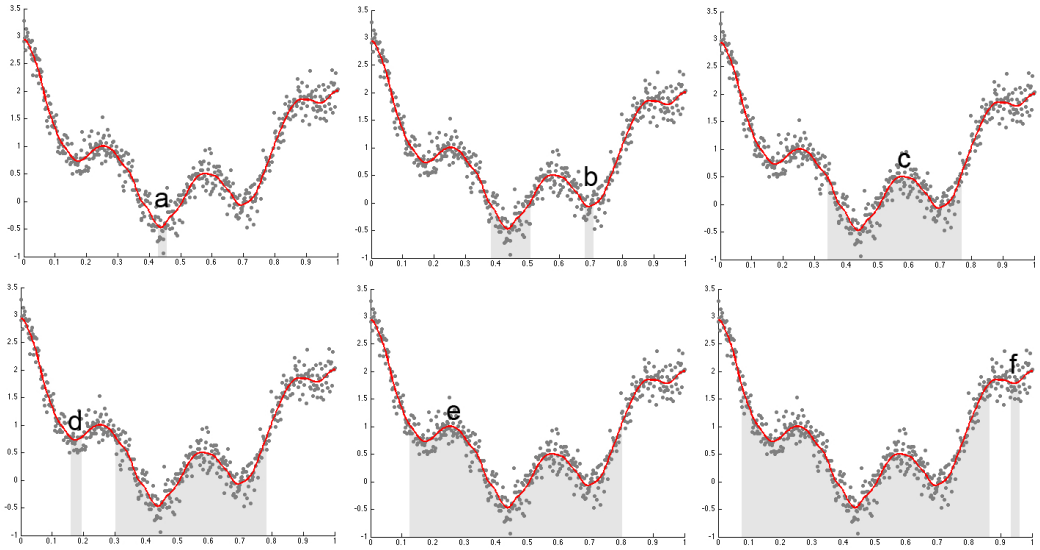

Fig. 1. The births and deaths of connected components in the sublevel sets in a Morse filtration [8]. We have local minimums $a < b < d < f$ and local maximums $c < e$. At $y = a$, we have a single connected component (gray area). As we increase the filtration value to $y = b$, we have the birth of a new component (second gray area). At the local maximum $y = c$, the two sublevel sets merge together to form a single component. This is viewed as the death of a component. The process continues till we exhaust all the critical values. Following the Elder rule, we pair birth to death: (b, c) and (d, e). Other critical values are paired similarly. These paired points form the persistent diagram.

characterized by the pairing of local minimums and maximums. For 1D Morse functions, we do not have higher dimensional topological features beyond the connected components.

Let us denote the local minimums as $g_1, \cdots, g_m$ and the local maximums as $h_1, \cdots, h_n$. Since the critical values of a Morse function are all distinct, we can combine all minimums and maximums and reorder them from the smallest to the largest: We further order all critical values together and let

$$g_1 = z_{(1)} < z_{(2)} < \cdots < z_{(m+n)} = h_n,$$

where z_i is either h_i or g_i and $z_{(i)}$ denotes the i-th largest number in $z_1, \cdots, z_{m+n}$. In a Morse function, g_1 is smaller than h_1 and g_m is smaller than h_n in the unbounded domain $\mathbb{R}$ [8].

By keeping track of the birth and death of components, it is possible to compute topological invariants of sublevel sets such as the 0-th Betti number β_0 [19]. As we move y from $-\infty$ to ∞, at a local minimum, the sublevel set adds a new component so that

$$\beta_0(R_{g_i - \epsilon}) = \beta_0(R_{g_i}) + 1$$

for sufficiently small ϵ. This process is called the *birth* of the component. The newly born component is identified with the local minimum g_i.

Similarly for at a local maximum, two components are merged as one so that

$$\beta_0(R_{h_i-\epsilon}) = \beta_0(R_{h_i}) - 1.$$

This process is called the *death* of the component. Since the number of connected components will only change if we pass through critical points and we can iteratively compute β_0 at each critical value as

$$\beta_0(R_{z_{(i+1)}}) = \beta_0(R_{z_{(i)}}) \pm 1.$$

The sign depends on if $z_{(i)}$ is maximum (-1) or minimum $(+1)$. This is the basis of the Morse theory [41] that states that the topological characteristics of the sublevel set of Morse function are completely characterized by critical values.

To reduce the effect of low signal-to-noise ratio and to obtain smooth Morse function, either spatial or temporal smoothing have been often applied to brain imaging data before persistent homology is applied. In [11,37], Gaussian kernel smoothing was applied to 3D volumetric images. In [60], diffusion was applied to temporally smooth data.

Example 1. As an example, elder's rule is illustrated in Figure 1, where the gray dots are simulated with Gaussian noise with mean 0 and variance 0.2^2 as

$$f(x) = \mu(x) + N(0, 0.2^2)$$

with signal $\mu(t) = t + 7(t - 1/2)^2 + \cos(7\pi t)/2$. The signal μ is estimated using heat kernel smoothing [9] using degree $k = 100$ and kernel bandwidth $\sigma = 0.0001$ using `WFS_COS.m`. Now we apply Morse filtration for filtration values y from $-\infty$ to ∞. When we hit the first critical value $y = a$, the sublevel set consists of a single point. When we hit the minimum at $y = b$, we have the birth of a new component at b. When we hit the maximum at $y = c$, the two components identified by a and b are merged together to form a single component. When we pass through a maximum and merge two components, we pair the maximum with the higher of the two minimums of the two components [19]. Doing so we are pairing the birth of a component to its death. Obviously the paired extremes do not have to be adjacent to each other. If there is a boundary, the function value evaluated at the boundary is treated as a critical value. In the example, we need to pair (b, c) and (d, e). Other critical values are paired similarly. The persistence diagram is then the scatter plot of these pairings computed using `PH_morse1D.m`. This is implemented as

```
t=[0:0.002:1]';
s= t + 7*(t - 0.5).^2 + cos(8*pi*t)/2;
e=normrnd(0,0.2,length(x),1);
Y=s+e;

k=100; sigma=0.0001;
[wfs, beta]=WFS_COS(Y,x,k,sigma);

pairs=PH_morse1D(x,wfs);
```

2 Simplical Complex

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

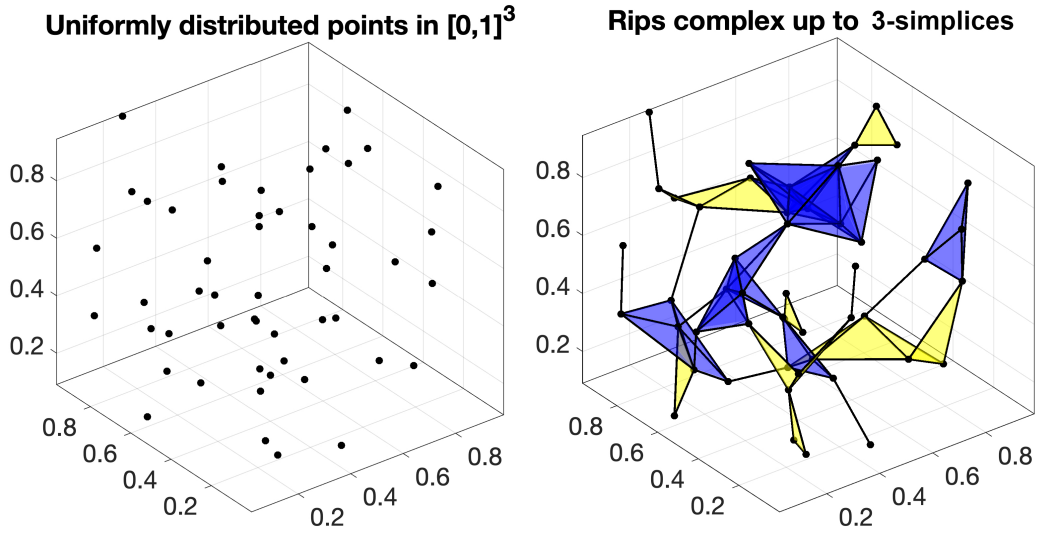

Fig. 2. Left: 50 randomly distributed points X in $[0,1]^3$. Right: Rips complex $R_{0.3}(X)$ within radius 0.3 containing 106 1-simplices, 75 2-simplices (yellow) and 22 3-simplices (blue).

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 [25,24,20]. 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 [65]. 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 [20]. 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 simplical complex for point cloud and network data [31].

The k-simplex σ is the convex hull of $v+1$ independent points $v_0, \cdots, v_k$. 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 [20]. 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.

2.1 Rips complex

The Vietoris–Rips or Rips complex is the most often used simplicial complex in persistent homology. Let $X = \{x_0, \cdots, x_p\}$ 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 [19,26]. 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.

- 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 $x_{i_0}, x_{i_1}, \ldots, x_{i_k}$ is less than or equal to ϵ, then there is a k-simplex in $R_\epsilon(X)$ whose vertices are $x_{i_0}, x_{i_1}, \ldots, x_{i_k}$.

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 simplical 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 2 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 `X` and simplical complex `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

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

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

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

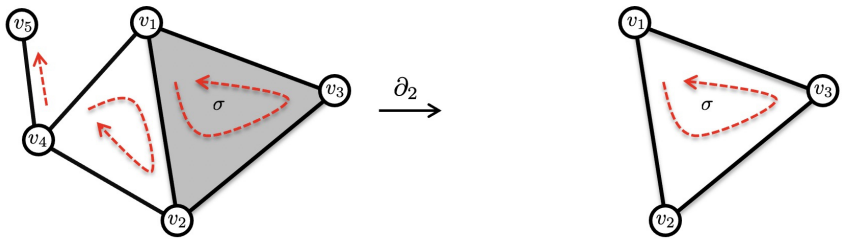

Fig. 3. A simplicial complex with 5 vertices and 2-simplex $\sigma = [v_1, v_2, v_3]$ with a filled-in face (colored gray). After boundary operation ∂_2, we are only left with 1-simplices $[v_1, v_2]+[v_2, v_3]+[v_3, v_1]$, which is the boundary of the filled in triangle. The complex has a single connected component ($\beta_0 = 1$) and a single 1-cycle. The dotted red arrows are the orientation of simplices.

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, $C_{k-1}, C_{k-2}, \cdots$ are then sequentially nested by boundary operator ∂_k [20]:

$$\cdots \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} C_{k-2} \xrightarrow{\partial_{k-2}} \cdots . \tag{2}$$

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 3. The boundary map ∂_k applied to σ resulted in the collection of three edges that forms the boundary of σ:

$$\partial_2 \sigma = [v_1, v_2] + [v_2, v_3] + [v_3, v_1] \in C_1. \tag{3}$$

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 (3) 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 jth basis element in the linear combination of the $k-1$ faces of the ith basis element in C_k. The boundary matrix is the higher dimensional version of the incidence matrix in graphs [33,32,51] showing how $(k-1)$-dimensional simplices are forming k-dimensional simplex. The (i, j) entry of $\boldsymbol{\partial}_k$ is one if τ is a face of σ otherwise zero. The

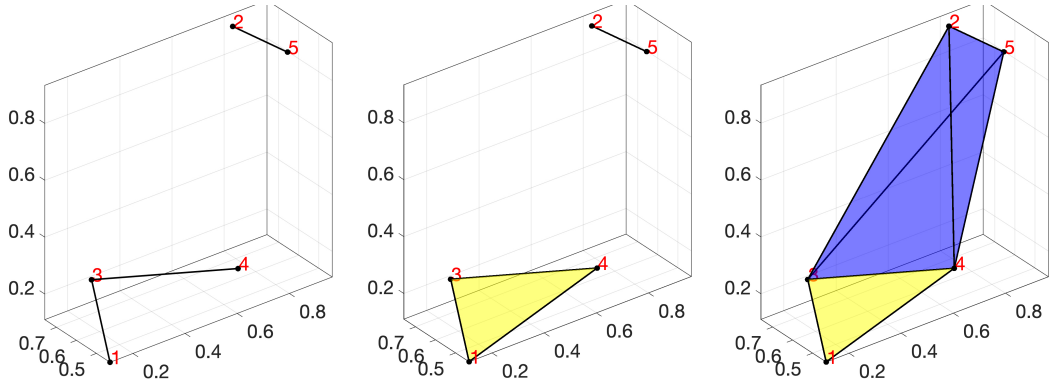

Fig. 4. Examples of boundary matrix computation. From the left to right, the radius is changed to 0.5, 0.6 and 1.0.

entry can be -1 depending on the orientation of τ. For the simplicial complex in Figure 3, the boundary matrices are given by

$$\boldsymbol{\partial}_2 = \begin{matrix} \\ e_{12} \\ e_{23} \\ e_{31} \\ e_{24} \\ e_{41} \\ e_{45} \end{matrix} \begin{matrix} \sigma \\ \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{matrix}$$

$$\boldsymbol{\partial}_1 = \begin{matrix} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} \begin{matrix} \begin{matrix} e_{12} & e_{23} & e_{31} & e_{24} & e_{41} & e_{45} \end{matrix} \\ \begin{pmatrix} -1 & 0 & 1 & 0 & 1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

$$\boldsymbol{\partial}_0 = \begin{matrix} \\ 0 \end{matrix} \begin{matrix} \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}.$$

In example in Figure 4-left, `PH_rips(X,3,0.5)` gives

```
>> S{1}

     1
     2
     3
     4
     5
```

```
>> 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 4-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 4-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

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

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 3, 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.,

$$\partial_{k-1} \circ \partial_k = 0. \tag{4}$$

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 3 example and obtain

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

This property (4) implies that the image of ∂_k is contained in the kernel of ∂_{k-1}, i.e.,

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

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

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

which is called the k-th homology group. $H_k(K)$ is a vector space that captures the kth 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.

The rank of $H_k(K)$ is the kth 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

$$\beta_k = rank(H_k) = rank(\mathrm{ker}\partial_k) - rank(\mathrm{im}\partial_{k+1}). \tag{5}$$

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 $\boldsymbol{\partial}_k$ to the Smith normal form $\mathcal{S}(\boldsymbol{\partial}_k)$, which has a block diagonal matrix as a submatrix in the upper left, via Gaussian elimination [20]. In the Smith normal form, we have the rank-nullity theorem for boundary matrix $\boldsymbol{\partial}_k$, which states the dimension of the domain of $\boldsymbol{\partial}_k$ is the sum of the dimension of its image and the dimension of its kernel (nullity) (Figure 5). In $\mathcal{S}(\boldsymbol{\partial}_k)$, the number of columns containing only zeros is $rank(\mathrm{ker}\boldsymbol{\partial}_k)$, the number of k-cycles while the number of columns containing one is $rank(\boldsymbol{\partial}_k)$, the number of k-cycles that are boundaries. Thus

$$\beta_k = rank(ker\boldsymbol{\partial}_k) - rank(\boldsymbol{\partial}_{k+1}). \tag{6}$$

The computation starts with initial rank computation

$$rank\boldsymbol{\partial}_0 = 0, \quad rank(ker\boldsymbol{\partial}_0) = p.$$

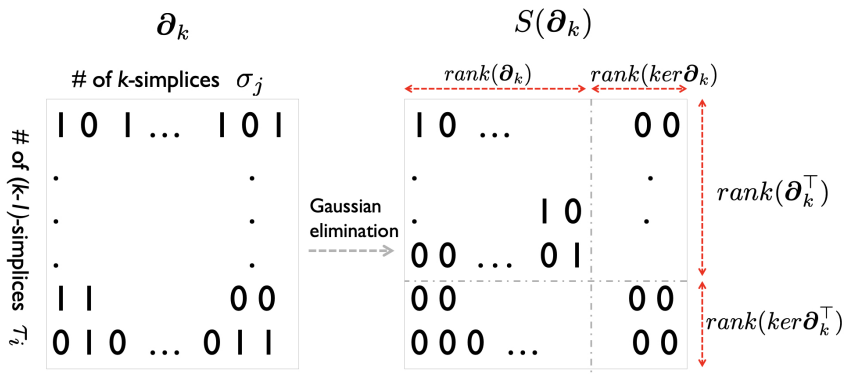

Fig. 5. The rank-nullity theorem for boundary matrix $\boldsymbol{\partial}_k$, which states the dimension of the domain of $\boldsymbol{\partial}_k$ is the sum of the dimension of its image and the dimension of its kernel (nullity).

Example 2. The boundary matrices $\boldsymbol{\partial}_k$ in Figure 3 is transformed to the Smith normal form $\mathcal{S}(\boldsymbol{\partial}_k)$ after Gaussian elimination as

$$\mathcal{S}(\boldsymbol{\partial}_1) = \begin{pmatrix} 1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0 \end{pmatrix}, \quad \mathcal{S}(\boldsymbol{\partial}_2) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

From (6), the Betti number computation involves the rank computation of two boundary matrices. $rank(\boldsymbol{\partial}_0) = 5$ is trivially the number of nodes in the simplicial complex. There are $rank(\ker\boldsymbol{\partial}_1) = 2$ zero columns and $rank(\boldsymbol{\partial}_1) = 4$ non-zero row columns. $rank(\boldsymbol{\partial}_2) = 1$. Thus, we have

$$\beta_0 = rank(\ker\boldsymbol{\partial}_0) - rank(\boldsymbol{\partial}_1) = 5 - 4 = 1,$$
$$\beta_1 = rank(\ker\boldsymbol{\partial}_1) - rank(\boldsymbol{\partial}_2) = 2 - 1 = 1.$$

Following the above worked out example, the Betti number computation is implemented in `PH_boundary_betti.m` which inputs the boundary matrices generated by `PH_boundary.m`. The function outputs $\beta_1, \beta_2, \cdots$. The function computes the $(d-1)$-th Betti number as

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

Figure 6 displays few examples of Betti number computation on Rips complexes. The rank computation in most computational packages is through the singular value decomposition (SVD).

4.1 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

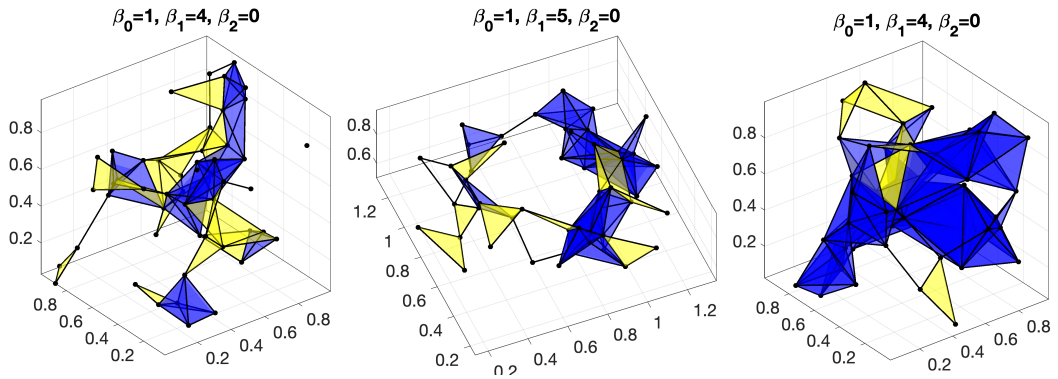

Fig. 6. Betti number computation on simplical complex using `PH_betti.m` function.

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

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

for $0 = \epsilon_0 \leq \epsilon_1 \leq \epsilon_2 \leq \cdots$. This nested sequence of Rips complexes is called the *Rips filtration*, which is the main object of interest in persistent homology (Figure 7). The filtration values $\epsilon_0, \epsilon_1, \epsilon_2, \ldots$ 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 [35,34], 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.

4.2 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.

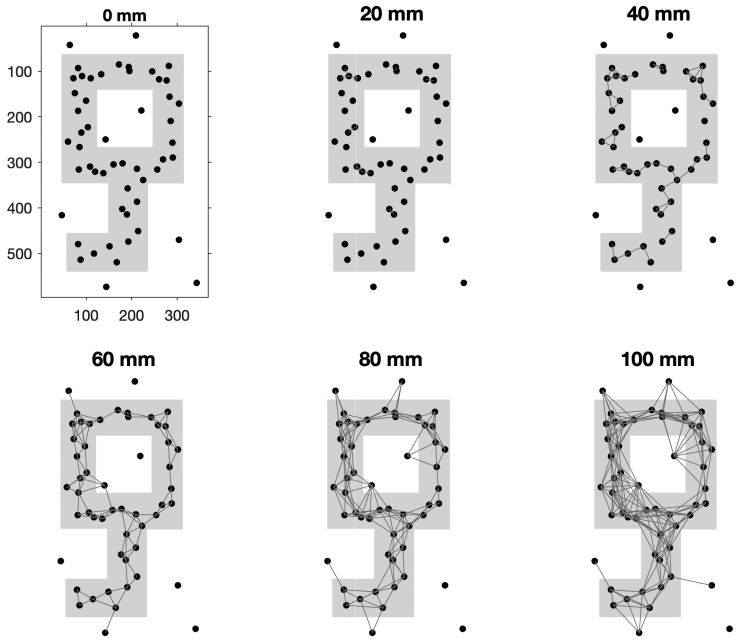

Fig. 7. Rips filtration on 1-skeleton of the point cloud data that was sampled along the underlying key shaped data. If two points are within the given radius, we connect them with an edge but do not form any other dimensional simplex. Such sparsity in Rips filtration can be more effective in practice. `PH_rips.m` limit the dimension of skeleton we build Rips filtrations and do not build every possible simplical complexes.

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.

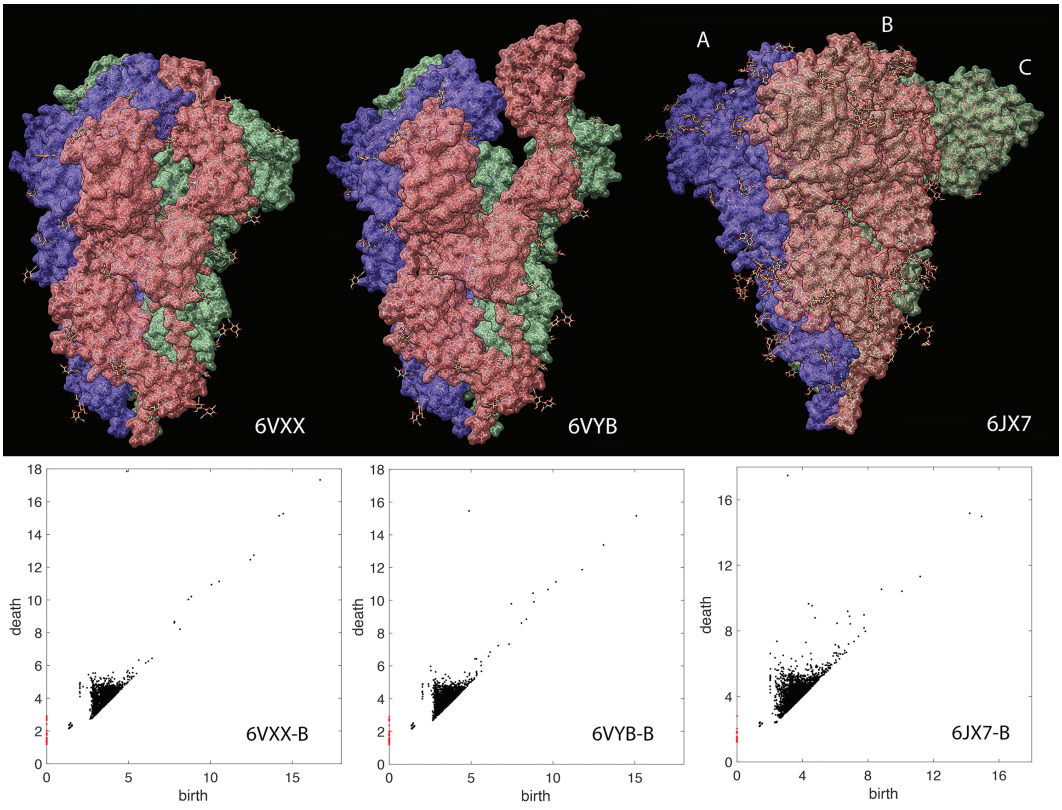

Fig. 8. Top: Spike proteins of the three different corona viruses. The spike proteins consist of three similarly shaped interwinding substructures identified as A (blue), B (red) and C (green) domains. Bottom: The persistent diagrams of spike proteins. The red dots are 0D homology and the black dots are 1D homology.

Example 3. The example came from [14]. The atomic structure of spike proteins of corona virus can be determined through the cryogenic electron microscopy (cryo-EM) [4,58]. Figure 8-top displays a spike consists of three similarly shaped protein molecules with rotational symmetry often identified as A, B and C domains. The 6VXX and 6VYB are respectively the closed and open states of SARS-Cov-2 from human [58] while 6JX7 is feline coronavirus [63]. We used the atomic distances in building Rips filtrations in computing persistent diagrams. The persistent diagrams of both closed and open states are almost identical in smaller birth and death values below 6 Å (angstrom) (Figure 8-bottom). The major difference is in the scatter points with larger brith and death values. However, we need a quantitative measures for comparing the topology of closed and open states.

5 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.

5.1 Sampling from the Dirichlet 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

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

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

$$f_\alpha^{\mathcal{T}}(x_1, x_2) = \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)} x_1^{\alpha_1 - 1} (1 - x_2)^{\alpha_2 - 1} (x_2 - x_1)^{\alpha_3 - 1}, \tag{7}$$

where $\alpha = (\alpha_1, \alpha_2, \alpha_3)$ are positive parameters [30]. 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. *[18,44] Let $Y_i \sim \text{Gamma}(\alpha_i, 1), i = 1, 2, 3$ be independent Gamma random variables with $\alpha_i > 0$, and let*

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

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 $(\alpha_1, \alpha_2, \alpha_3)$.

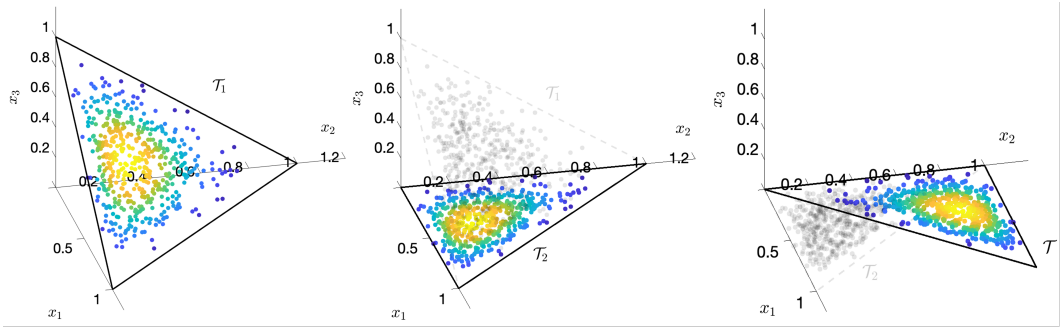

Fig. 9. Transformation of sampled points from $\mathcal{T}_1$ to $\mathcal{T}_2$ and finally the target domain $\mathcal{T}$. Points are sampled following the Dirichlet distribution in $\mathcal{T}_1$ with $\alpha = (3, 2, 3)$. If the problem is not explicitly obtaining the of density in $\mathcal{T}$, 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 $\mathcal{T}$.

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 than transform $\mathcal{T}_2$ to $\mathcal{T}$ by rotation $(x_1, x_2) \longmapsto (x_1, 1 - x_2)$. 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 9 displays the transformation of sampled points from $\mathcal{T}_1$ to $\mathcal{T}_2$ and finally the target domain $\mathcal{T}$.

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 `x`. `x(:,1)` is the birth values and `x(:,2)` is the death values. Each row of the output corresponds to one sample point. To generate 500 scatter points from $f_\alpha^{\mathcal{T}}$ with $\alpha = (3, 2, 3)$ (Figure 10-left), we run

```
alpha = [3,2,3]';
x = Dirichlet_sample(alpha,1,500);
f = Dirichlet_density(alpha, 1, x);
PH_PD_display(x1, f);
```

where `Dirichlet_density.m` computes the theoretical density at given points `x`. `PH_PD_display.m` displays the persistent diagrams of the sampled points.

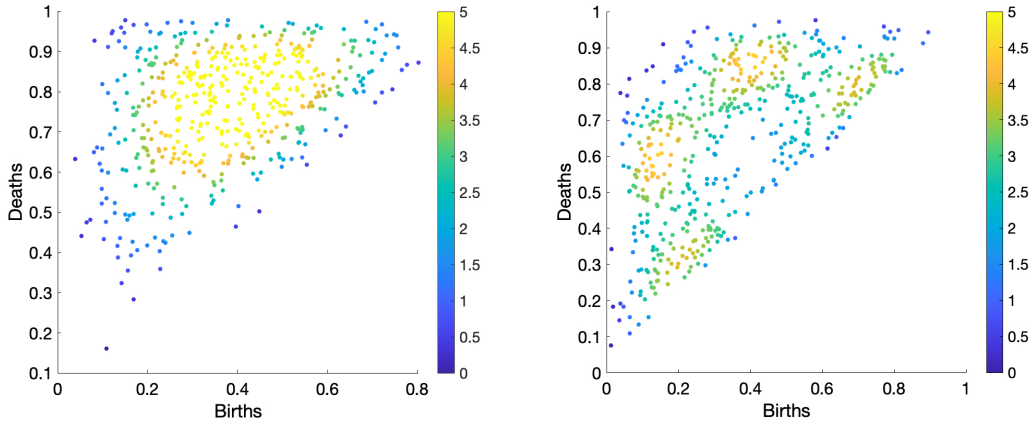

Fig. 10. Left: 500 samples from $f_{\alpha}^{\mathcal{T}}$ with $\alpha = (3, 2, 3)$. Right: 500 samples from mixture distribution $f^{\mathcal{T}}$ in (8). The sampled points are colored with the theoretical densities.

5.2 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 [39]. To generate 500 scatter points from the mixture

$$f^{\mathcal{T}} = 0.25 f_{\alpha_1}^{\mathcal{T}} + 0.25 f_{\alpha_2}^{\mathcal{T}} + 0.25 f_{\alpha_3}^{\mathcal{T}} + 0.25 f_{\alpha_4}^{\mathcal{T}}, \tag{8}$$

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 10-right.

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

6 Graph filtrations

6.1 Graph filtration

The graph filtration has been the first type of filtrations applied in brain networks and it is now considered as the baseline filtrations in brain network data [35,34,36]. Euclidean distance

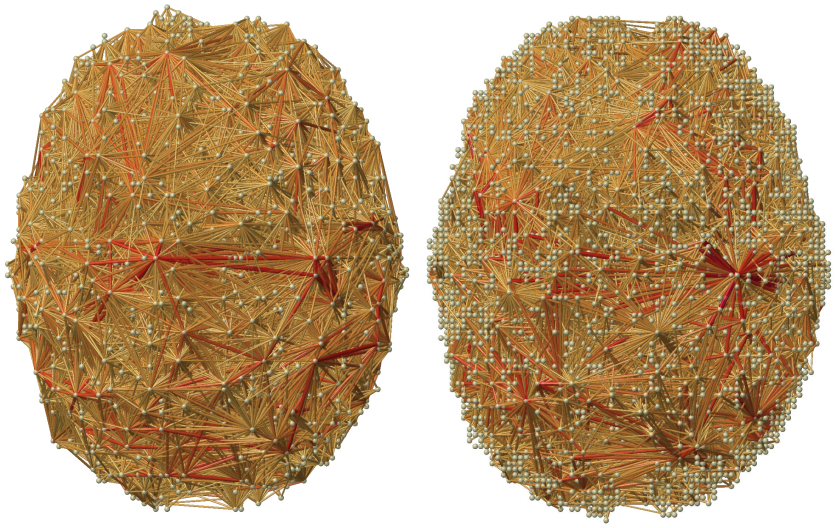

Fig. 11. rs-fMRI correlation network of two subjects from HCP with more than 25000 nodes. Identifying cycles and computing the number of cycles can be computationally demanding in this type of dense correlation network since persistent homology computations are not very scalable.

is often used metric in building filtrations in persistent homology [20]. Most brain network studies also use the Euclidean distances for building graph filtrations [48,28,6,61,2,47]. Given weighted network $\mathcal{X} = (V, w)$ with edge weight $w = (w_{ij})$, the binary network $\mathcal{X}_\epsilon = (V, w_\epsilon)$ is a graph consisting of the node set V and the binary edge weights $w_\epsilon = (w_{\epsilon,ij})$ given by

$$w_{\epsilon,ij} = \begin{cases} 1 & \text{if } w_{ij} > \epsilon; \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Note [34,36] defines the binary graphs by thresholding above such that $w_{\epsilon,ij} = 1$ if $w_{ij} < \epsilon$ which is consistent with the definition of the Rips filtration. However, in brain connectivity, higher value w_{ij} indicates stronger connectivity so we usually thresholds below [11].

Note w_ϵ is the adjacency matrix of $\mathcal{X}_\epsilon$, which is a simplicial complex consisting of 0-simplices (nodes) and 1-simplices (edges) [24]. In the metric space $\mathcal{X} = (V, w)$, the Rips complex $\mathcal{R}_\epsilon(X)$ is a simplicial complex whose $(p-1)$-simplices correspond to unordered p-tuples of points that satisfy $w_{ij} \leq \epsilon$ in a pairwise fashion [24]. While the binary network $\mathcal{X}_\epsilon$ has at most 1-simplices, the Rips complex can have at most $(p-1)$-simplices . Thus, $\mathcal{X}_\epsilon \subset \mathcal{R}_\epsilon(\mathcal{X})$ and its compliment $\mathcal{X}_\epsilon^c \subset \mathcal{R}_\epsilon(\mathcal{X})$. Since a binary network is a special case of the Rips complex, we also have

$$\mathcal{X}_{\epsilon_0} \supset \mathcal{X}_{\epsilon_1} \supset \mathcal{X}_{\epsilon_2} \supset \cdots$$

and equivalently

$$\mathcal{X}_{\epsilon_0}^c \subset \mathcal{X}_{\epsilon_1}^c \subset \mathcal{X}_{\epsilon_2}^c \subset \cdots$$

for $0 = \epsilon_0 \leq \epsilon_1 \leq \epsilon_2 \cdots$. The sequence of such nested multiscale graphs is defined as the *graph filtration* [34,36]. Figure 12 illustrates a graph filtration in a 4-nodes example while

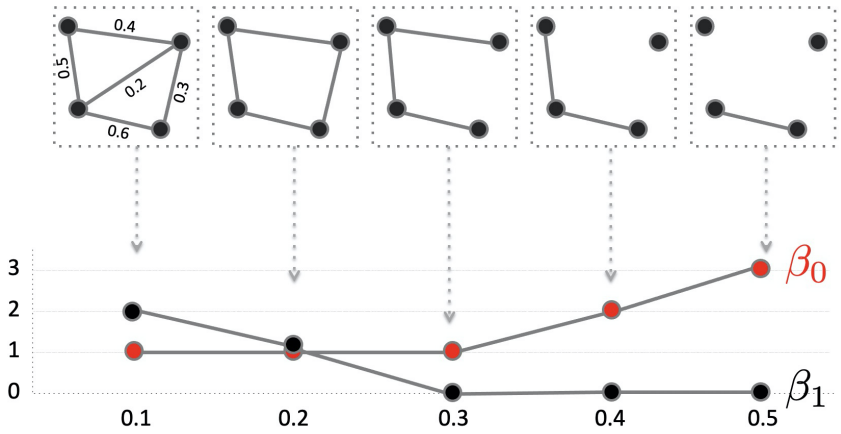

Fig. 12. Schematic of graph filtration and Betti curves. We sort the edge weights in an increasing order. We threshold the graph at filtration values and obtain binary graphs. The thresholding is performed sequentially by increasing the filtration values. The 0-th Betti number β_0, which counts the number of connected components, and the first Betti number β_1, which counts the number of cycles, is then plotted over the filtration. The Betti plots curves monotone in graph filtrations.

Figure 13 shows the graph filtration on structural covariates on maltreated children on 116 parcellated brain regions.

Note that $\mathcal{X}_0$ is the complete weighted graph while $\mathcal{X}_\infty$ is the node set V. By increasing the threshold value, we are thresholding at higher connectivity so more edges are removed. Given a weighted graph, there are infinitely many different filtrations. This makes the comparisons between two different graph filtrations difficult. For a graph with p nodes, the maximum number of edges is $(p^2-p)/2$, which is obtained in a complete graph. If we order the edge weights in the increasing order, we have the sorted edge weights:

$$0 = w_{(0)} < \min_{j,k} w_{jk} = w_{(1)} < w_{(2)} < \cdots < w_{(q)} = \max_{j,k} w_{jk},$$

where $q \leq (p^2-p)/2$. The subscript $_{(\,)}$ denotes the order statistic. For all $\lambda < w_{(1)}$, $\mathcal{X}_\lambda = \mathcal{X}_0$ is the complete graph of V. For all $w_{(r)} \leq \lambda < w_{(r+1)}$ $(r = 1, \cdots, q-1)$, $\mathcal{X}_\lambda = \mathcal{X}_{w_{(r)}}$. For all $w_{(q)} \leq \lambda$, $\mathcal{X}_\lambda = \mathcal{X}_{\rho_{(q)}} = V$, the vertex set. Hence, the filtration given by

$$\mathcal{X}_0 \supset \mathcal{X}_{w_{(1)}} \supset \mathcal{X}_{w_{(2)}} \supset \cdots \supset \mathcal{X}_{w_{(q)}}$$

is *maximal* in a sense that we cannot have any additional filtration $\mathcal{X}_\epsilon$ that is not one of the above filtrations. Thus, graph filtrations are usually given at edge weights [11].

The condition of having unique edge weights is not restrictive in practice. Assuming edge weights to follow some continuous distribution, the probability of any two edges being equal is zero. For discrete distribution, it may be possible to have identical edge weights. Then simply add Gaussian noise or add extremely small increasing sequence of numbers to q number of edges.

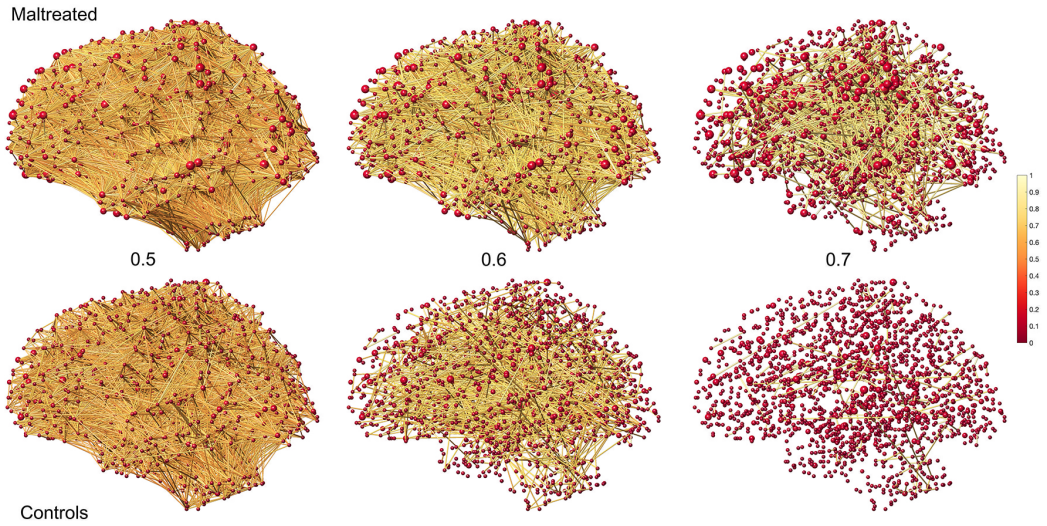

Fig. 13. Graph filtrations of maltreated children vs. normal control subjects on FA-values [11]. The Pearson correlation is used as filtration values at 0.5, 0.6 and 0.7. maltreated subjects show much higher correlation of FA-values indicating more homogeneous and less varied structural covariate relationship.

6.2 Monotone Betti curves

The graph filtration can be quantified using monotonic function f satisfying

$$f(\mathcal{X}_{\epsilon_0}) \geq f(\mathcal{X}_{\epsilon_1}) \geq f(\mathcal{X}_{\epsilon_2}) \geq \cdots \tag{10}$$

or

$$f(\mathcal{X}_{\epsilon_0}) \leq f(\mathcal{X}_{\epsilon_1}) \leq f(\mathcal{X}_{\epsilon_2}) \leq \cdots \tag{11}$$

The number of connected components (zeroth Betti number β_0) and the number of cycles (first Betti number β_1) satisfy the monotonicity (Figures 12 and 14). The size of the largest cluster also satisfies a similar but opposite relation of monotonic increase. There are numerous monotone graph theory features [11,16].

For graphs, β_1 can be computed easily as a function of β_0. Note that the Euler characteristic χ can be computed in two different ways

$$\begin{aligned}\chi &= \beta_0 - \beta_1 + \beta_2 - \cdots \\ &= \#nodes - \#edges + \#faces - \cdots,\end{aligned}$$

where $\#nodes, \#edges, \#faces$ are the number of nodes, edges and faces. However, graphs do not have filled faces and Betti numbers higher than β_0 and β_1 can be ignored. Thus, a graph with p nodes and q edges is given by [1]

$$\chi = \beta_0 - \beta_1 = p - q.$$

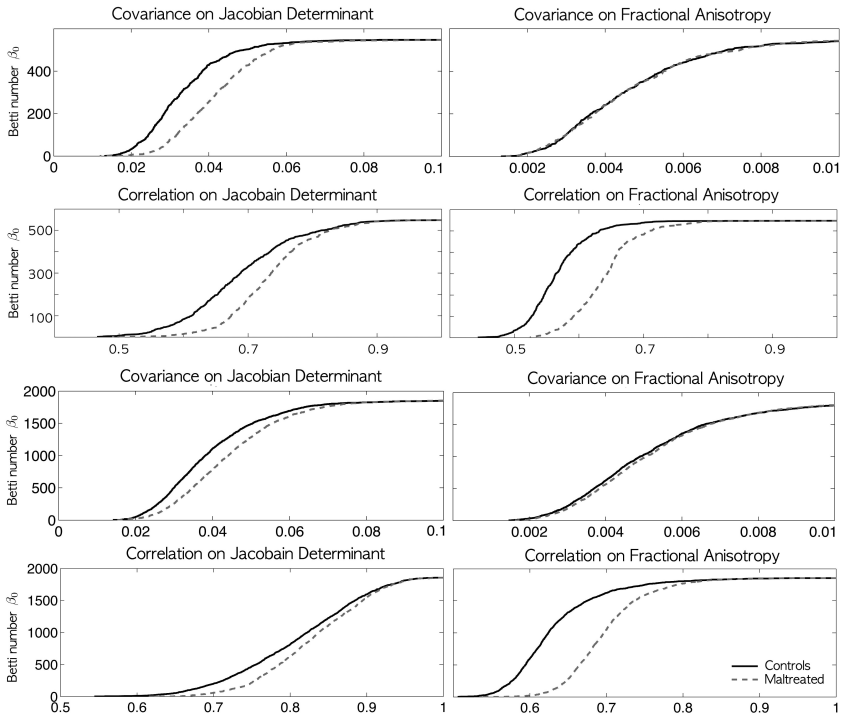

Fig. 14. The Betti curves on the covariance correlation matrices for Jacobian determinant (left column) and fractional anisotrophy (right column) on 548 (top two rows) and 1856 (bottom two rows) nodes [11]. Unlike the covariance, the correlation seems to shows huge group separation between normal and maltreated children visually. However, in all 7 cases except top right (548 nodes covariance for FA), statistically significant differences were detected using the rank-sum test on the areas under the Betti-plots (p-value < 0.001). The shapes of Betti-plots are consistent between the studies with different node sizes indicating the robustness of the proposed method over changing number of nodes.

Thus,

$$\beta_1 = p - q - \beta_0.$$

In a graph, Betti numbers β_0 and β_1 are monotone over filtration on edge weights [12,13]. When we do filtration on the maximal filtration in (10), edges are deleted one at a time. Since an edge has only two end points, the deletion of an edge disconnect the graph into at most two. Thus, the number of connected components (β_0) always increases and the increase is at most by one. Note p is fixed over the filtration but q is decreasing by one while β_0 increases at most by one. Hence, β_1 always decreases and the decrease is at most by one. Further, the length of the largest cycles, as measured by the number of nodes, also decreases monotonically (Figure 15).

Identifying connected components in a network is important to understand in decomposing the network into disjoint subnetworks. The number of connected components (0-th Betti number) of a graph is a topological invariant that measures the number of structurally independent or disjoint subnetworks. There are many available existing algorithms, which

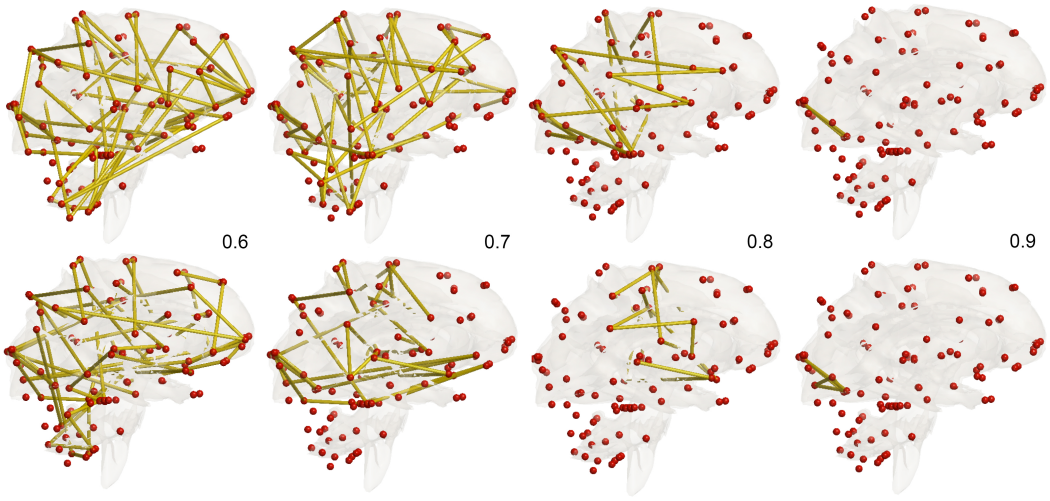

Fig. 15. The largest cycle at given correlation thresholds on rs-fMRI. Two representative subjects in HCP were used [12]. As the threshold increases, the length of cycles decreases monotonically.

are not related to persistent homology, for computing the number of connected components including the Dulmage-Mendelsohn decomposition [50], which has been widely used for decomposing sparse matrices into block triangular forms in speeding up matrix operations.

In graph filtrations, the number of cycles increase or decreases as the filtration value increases. The pattern of monotone increase or decrease can visually show how the topology of the graph changes over filtration values. The overall pattern of *Betti curves* can be used as a summary measure of quantifying how the graph changes over increasing edge weights [10] (Figure 12). The Betti curves are related to barcodes. The Betti number is equal to the number of bars in the barcodes at the specific filtration value.

Figure 16 displays an example of graph filtration constructed using random scatter points in a cube. Given scatter points `X`, the pairwise distance matrix is computed as `w = pdist2(X,X)`. The maximum distance is given by `maxw = max (w(:))`. Betti curves are computed using `PH_betti.m`, which inputs the pairwise distance `w` and the range of filtration values `[0:0.05:maxw]`. The function outputs β_0 and β_1 values as structured arrays `beta.zero` and `beta.one`. We display them using `PH_betti_display.m`.

```
p=50; d=3;
X = rand(p, d);
w = pdist2(X,X);
maxw = max(w(:));

thresholds=[0:0.05:maxw];
beta = PH_betti(w, thresholds);
PH_betti_display(beta,thresholds)
```

6.3 Rips filtration vs. graph filtration

Persistent homology does not scale well with increased data size (Figure 11). The computational complexity of persistent homology grows rapidly with the number of simplices [57]. With p number of nodes, the size of the k-skeleton grows as p^{k+1}. Homology calculations are often done by Gaussian elimination, and if there are N simplices, it takes $\mathcal{O}(N^3)$ time to perform. In $\mathbb{R}^d$, the computational complexity is $\mathcal{O}(p^{3k+3})$ [54]. Thus, the computation of Rips complex is exponentially costly. It can easily becomes infeasible when when one tries to use brain networks at the voxel level resolution. Thus, there have been many attempts in computing Rips complex approximately but fast for large-scale data such as alpha filtration based on Delaunay triangulation with $\mathcal{O}(n^2)$ for $k = 3$ and sparse Rips filtration with $\mathcal{O}(n)$ simplicies and $\mathcal{O}(n \log n)$ runtime [15,46,52]. However, all of these filtrations are all approximation to Rips filtration. To remedy the computational bottleneck caused by Rips filtrations, *graph filtration* was introduced particularly for network data [34,36].

The graph filtration is a special case of Rips filtration restricted to 1-simplices. If the Rips filtration up to 2-simplices is given by `PH_rips(X,2, e)`, the graph filtration is given by `PH_graph(X,1, maxw-e)`. In Figure 16 displays the comparison of two filtrations for randomly generated 50 nodes in a cube. In the both filtrations, β_0-curves are monotone. However, β_1-curve for the Rips filtration is not monotone. Further, the range of changes in β_1 is very narrow. In some randomly generated points, we can have multiple peaks in β_1 making the β_1-curve somewhat unstable. On the other hand, the β_1-curve for the graph filtration is monotone and gradually changing over the whole range of filtration values. This will give consistent to the β_1-curve that is required for increasing statistical power in the group level inference.

References

1. Adler, R., Bobrowski, O., Borman, M., Subag, E., Weinberger, S.: Persistent homology for random fields and complexes. In: Borrowing strength: theory powering applications–a Festschrift for Lawrence D. Brown, pp. 124–143. Institute of Mathematical Statistics (2010)
2. Anirudh, R., Thiagarajan, J., Kim, I., Polonik, W.: Autism spectrum disorder classification using graph kernels on multidimensional time series. arXiv preprint arXiv:1611.09897 (2016)
3. Bendich, P., Marron, J., Miller, E., Pieloch, A., Skwerer, S.: Persistent homology analysis of brain artery trees. The annals of applied statistics **10**, 198 (2016)
4. Cai, Y., Zhang, J., Xiao, T., Peng, H., Sterling, S., Walsh, R., Rawson, S., Rits-Volloch, S., Chen, B.: Distinct conformational states of SARS-CoV-2 spike protein. Science **369**, 1586–1592 (2020)
5. Carlsson, G., Memoli, F.: Persistent clustering and a theorem of J. Kleinberg. arXiv preprint arXiv:0808.2241 (2008)
6. Cassidy, B., Rae, C., Solo, V.: Brain activity: conditional dissimilarity and persistent homology. In: IEEE 12th International Symposium on Biomedical Imaging (ISBI). pp. 1356–1359 (2015)
7. Chung, M., Adluru, N., Dalton, K., Alexander, A., Davidson, R.: Scalable brain network construction on white matter fibers. In: Proc. of SPIE. vol. 7962, p. 79624G (2011)
8. Chung, M., Bubenik, P., Kim, P.: Persistence diagrams of cortical surface data. Proceedings of the 21st International Conference on Information Processing in Medical Imaging (IPMI), Lecture Notes in Computer Science (LNCS) **5636**, 386–397 (2009)
9. Chung, M., Dalton, K., Shen, L., Evans, A., Davidson, R.: Weighted Fourier representation and its application to quantifying the amount of gray matter. IEEE Transactions on Medical Imaging **26**, 566–581 (2007)

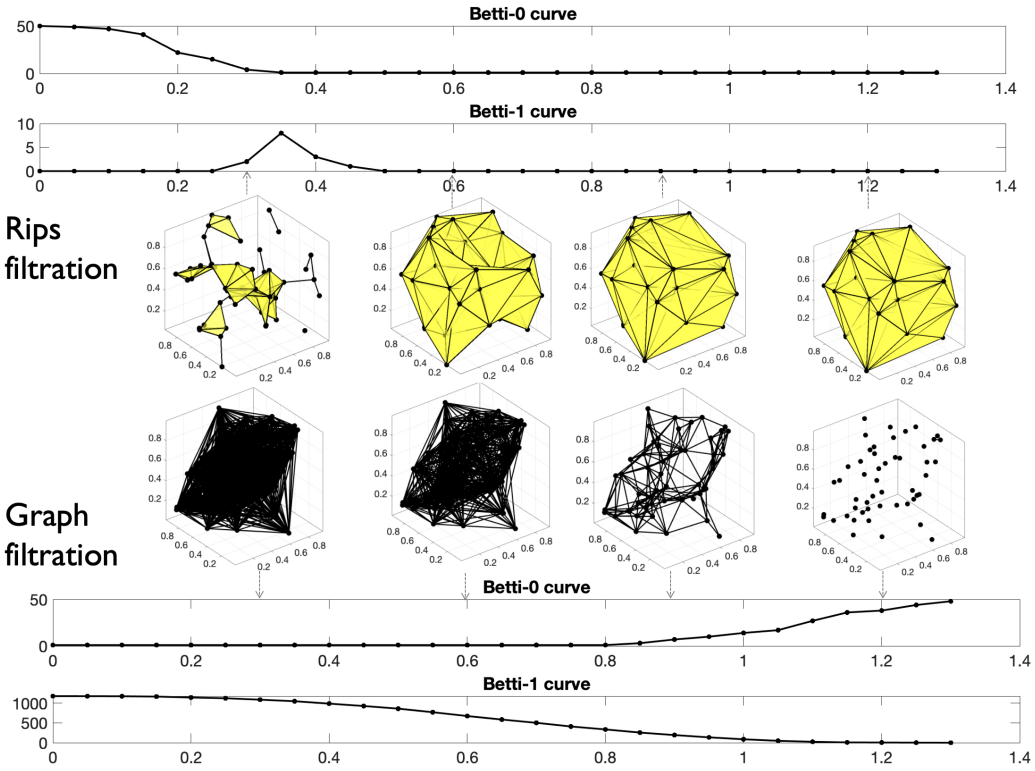

Fig. 16. The comparison between the Rips and graph filtrations.

10. Chung, M., Hanson, J., Lee, H., Adluru, N., Alexander, A.L., Davidson, R., Pollak, S.: Persistent homological sparse network approach to detecting white matter abnormality in maltreated children: MRI and DTI multimodal study. MICCAI, Lecture Notes in Computer Science (LNCS) **8149**, 300–307 (2013)
11. Chung, M., Hanson, J., Ye, J., Davidson, R., Pollak, S.: Persistent homology in sparse regression and its application to brain morphometry. IEEE Transactions on Medical Imaging **34**, 1928–1939 (2015)
12. Chung, M., Huang, S.G., Gritsenko, A., Shen, L., Lee, H.: Statistical inference on the number of cycles in brain networks. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). pp. 113–116. IEEE (2019)
13. Chung, M., Lee, H., DiChristofano, A., Ombao, H., Solo, V.: Exact topological inference of the resting-state brain networks in twins. Network Neuroscience **3**, 674–694 (2019)
14. Chung, M., Ombao, H.: Lattice paths for persistent diagrams. In: Interpretability of Machine Intelligence in Medical Image Computing, and Topological Data Analysis and Its Applications for Medical Data, LNCS 12929, pp. 77–86 (2021)
15. Chung, M., Smith, A., Shiu, G.: Reviews: Topological distances and losses for brain networks. arXiv e-prints pp. arXiv–2102.08623 (2020), `https://arxiv.org/pdf/2102.08623`
16. Chung, M., Vilalta-Gil, V., Lee, H., Rathouz, P., Lahey, B., Zald, D.: Exact topological inference for paired brain networks via persistent homology. Information Processing in Medical Imaging

(IPMI), Lecture Notes in Computer Science (LNCS) **10265**, 299–310 (2017)
17. Cohen-Steiner, D., Edelsbrunner, H., Harer, J.: Stability of persistence diagrams. Discrete and Computational Geometry **37**, 103–120 (2007)
18. Devroye, L.: Non-Uniform Random Variate Generation. Springer New York (2013)
19. Edelsbrunner, H., Harer, J.: Persistent homology - a survey. Contemporary Mathematics **453**, 257–282 (2008)
20. Edelsbrunner, H., Harer, J.: Computational topology: An introduction. American Mathematical Society (2010)
21. Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological persistence and simplification. Discrete and Computational Geometry **28**, 511–533 (2002)
22. Friston., K.: A short history of statistical parametric mapping in functional neuroimaging. Tech. Rep. Technical report, Wellcome Department of Imaging Neuroscience, ION, UCL., London, UK. (2002)
23. Garside, K., Gjoka, A., Henderson, R., Johnson, H., Makarenko, I.: Event history and topological data analysis. arXiv preprint arXiv:2012.08810 (2020)
24. Ghrist, R.: Barcodes: The persistent topology of data. Bulletin of the American Mathematical Society **45**, 61–75 (2008)
25. Hart, J.: Computational topology for shape modeling. In: Proceedings of the International Conference on Shape Modeling and Applications. pp. 36–43 (1999)
26. Hatcher, A.: Algebraic topology. Cambridge University Press (2002)
27. Hofer, C., Graf, F., Rieck, B., Niethammer, M., Kwitt, R.: Graph filtration learning. In: International Conference on Machine Learning. pp. 4314–4323. PMLR (2020)
28. Khalid, A., Kim, B., Chung, M., Ye, J., Jeon, D.: Tracing the evolution of multi-scale functional networks in a mouse model of depression using persistent brain network homology. NeuroImage **101**, 351–363 (2014)
29. Kiebel, S., Poline, J.P., Friston, K., Holmes, A., Worsley, K.: Robust smoothness estimation in statistical parametric maps using standardized residuals from the general linear model. NeuroImage **10**, 756–766 (1999)
30. Kotz, S., Balakrishnan, N., Johnson, N.: Continuous Multivariate Distributions, Volume 1: Models and Applications. Continuous Multivariate Distributions (2004)
31. 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)
32. 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)
33. 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)
34. 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)
35. 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)
36. Lee, H., Kang, H., Chung, M., Kim, B.N., Lee, D.: Persistent brain network homology from the perspective of dendrogram. IEEE Transactions on Medical Imaging **31**, 2267–2277 (2012)
37. Lee, H., Kang, H., Chung, M., Lim, S., Kim, B.N., Lee, D.: Integrated multimodal network approach to PET and MRI based on multidimensional persistent homology. Human Brain Mapping **38**, 1387–1402 (2017)

38. Li, Y., Wang, D., Ascoli, G., Mitra, P., Wang, Y.: Metrics for comparing neuronal tree shapes based on persistent homology. PloS one **12**(8), e0182184 (2017)
39. McLachlan, G., Peel, D.: Finite Mixture Models. Wiley Series in Probability and Statistics, Wiley (2004)
40. Miller, M., Banerjee, A., Christensen, G., Joshi, S., Khaneja, N., Grenander, U., Matejic, L.: Statistical methods in computational anatomy. Statistical Methods in Medical Research **6**, 267–299 (1997)
41. Milnor, J.: Morse Theory. Princeton University Press (1973)
42. Morozov, D.: Homological Illusions of Persistence and Stability. Ph.D. thesis, Duke University (2008)
43. Naiman, D.: volumes for tubular neighborhoods of spherical polyhedra and statistical inference. Ann. Statist. **18**, 685–716 (1990)
44. Ng, K., Tian, G., Tang, M.: Dirichlet and Related Distributions: Theory, Methods and Applications. Wiley-Blackwell, United States (2011)
45. Nichols, T., Hayasaka, S.: Controlling the familywise error rate in functional neuroimaging: a comparative review. Stat Methods Med. Res. **12**, 419–446 (2003)
46. Otter, N., Porter, M., Tillmann, U., Grindrod, P., Harrington, H.: A roadmap for the computation of persistent homology. EPJ Data Science **6**(1), 17 (2017)
47. Palande, S., Jose, V., Zielinski, B., Anderson, J., Fletcher, P., Wang, B.: Revisiting abnormalities in brain network architecture underlying autism using topology-inspired statistical inference (2017)
48. Petri, G., Expert, P., Turkheimer, F., Carhart-Harris, R., Nutt, D., Hellyer, P., Vaccarino, F.: Homological scaffolds of brain functional networks. Journal of The Royal Society Interface **11**, 20140873 (2014)
49. Petri, G., Scolamiero, M., Donato, I., Vaccarino, F.: Topological strata of weighted complex networks. PloS One **8**, e66506 (2013)
50. Pothen, A., Fan, C.: Computing the block triangular form of a sparse matrix. ACM Transactions on Mathematical Software (TOMS) **16**, 324 (1990)
51. 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)
52. Sheehy, D.: Linear-size approximations to the Vietoris–Rips filtration. Discrete & Computational Geometry **49**, 778–796 (2013)
53. de Silva, V., Ghrist, R.: Homological sensor networks. Notic Amer Math Soc **54**, 10–17 (2007)
54. Solo, V., Poline, J., Lindquist, M., Simpson, S., Bowman, D., C.M., Cassidy, B.: Connectivity in fMRI: a review and preview. IEEE Transactions on Medical Imaging p. in press (2018)
55. Stolz, B., Harrington, H., Porter, M.: Persistent homology of time-dependent functional networks constructed from coupled time series. Chaos: An Interdisciplinary Journal of Nonlinear Science **27**, 047410 (2017)
56. Taylor, J., Worsley, K.: Random fields of multivariate test statistics, with applications to shape analysis. Annals of Statistics **36**, 1–27 (2008)
57. Topaz, C., Ziegelmeier, L., Halverson, T.: Topological data analysis of biological aggregation models. PLoS One p. e0126383 (2015)
58. Walls, A., Park, Y.J., Tortorici, M., Wall, A., McGuire, A., Veesler, D.: Structure, function, and antigenicity of the SARS-CoV-2 spike glycoprotein. Cell **181**, 281–292 (2020)
59. Wang, Y., Chung, M., Dentico, D., Lutz, A., Davidson, R.: Topological network analysis of electroencephalographic power maps. In: International Workshop on Connectomics in NeuroImaging, Lecture Notes in Computer Science (LNCS). vol. 10511, pp. 134–142 (2017)
60. Wang, Y., Ombao, H., Chung, M.: Topological data analysis of single-trial electroencephalographic signals. Annals of Applied Statistics **12**, 1506–1534 (2018)

61. Wong, E., Palande, S., Wang, B., Zielinski, B., Anderson, J., Fletcher, P.: Kernel partial least squares regression for relating functional brain network topology to clinical measures of behavior. In: IEEE International Symposium on Biomedical Imaging (ISBI). pp. 1303–1306 (2016)
62. Worsley, K., Marrett, S., Neelin, P., Vandal, A., Friston, K., Evans, A.: A unified statistical approach for determining significant signals in images of cerebral activation. Human Brain Mapping **4**, 58–73 (1996)
63. Yang, Z., Wen, J., Davatzikos, C.: Smile-GANs: Semi-supervised clustering via GANs for dissecting brain disease heterogeneity from medical images. arXiv preprint **arXiv**, 2006.15255 (2020)
64. Zomorodian, A.: Computing and Comprehending Topology: Persistence and Hierarchical Morse Complexes. Ph.D. Thesis, University of Illinois, Urbana-Champaign (2001)
65. Zomorodian, A.: Topology for computing. Cambridge University Press, Cambridge (2009)
66. Zomorodian, A., Carlsson, G.: Computing persistent homology. Discrete and Computational Geometry **33**, 249–274 (2005)