Topological Data Analysis:
Concepts, Computation, and Applications in Chemical Engineering

Alexander D. Smith†, Paweł Dłotko‡, and Victor M. Zavala‖∗
†Department of Chemical and Biological Engineering
University of Wisconsin-Madison, 1415 Engineering Dr, Madison, WI 53706, USA
‡Mathematics Department
Swansea University, Singleton Park, Sketty, Swansea SA2 8PP, United Kingdom

Abstract
A primary hypothesis that drives scientific and engineering studies is that data has structure. The dominant paradigms for describing such structure are statistics (e.g., moments, correlation functions) and signal processing (e.g., convolutional neural nets, Fourier series). Topological Data Analysis (TDA) is a field of mathematics that analyzes data from a fundamentally different perspective. TDA represents datasets as geometric objects and provides dimensionality reduction techniques that project such objects onto low-dimensional spaces that are composed of elementary geometric objects. Key property of these elementary objects (also known as topological features) are that they persist at different scales and that they are stable under perturbations (e.g., noise, stretching, twisting, and bending). In this work, we review key mathematical concepts and methods of TDA and present different applications in chemical engineering.

Keywords: data, topology, space, time, geometry.

1 Introduction
Statistical and signal processing techniques are the dominant paradigms used to analyze data. Unfortunately, these techniques provide limited capabilities to analyze certain types of datasets. A couple of interesting examples that highlight this limitation are the anscombe quartet and the datasaurus dozen datasets [5, 40]. These datasets are visually distinct but they have the same descriptive statistics (e.g., mean, standard deviation and correlation). An illustration of this issue is provided in Figure 1; here, the datasets have the same mean and standard deviation along both dimensions and have the same correlation between dimensions. However, it is clear that the data sets define objects with different geometrical features (shape). Signal processing techniques such as convolutional neural nets can capture visual differences in data sets but they require an extremely large number of parameters and are limited to a small number of dimensions (no more than three).

The recent application of algebraic and computational topology to data science has led to the development of a new field known as Topological Data Analysis (TDA) [13]. TDA techniques are based on the observation that data (e.g., a set of points in a Euclidean space) can be interpreted as elements of a geometrical object. As the name suggests, TDA utilizes techniques from computational
topology to quantify the geometry of data [23]. Fundamentally, topology studies geometric and spatial relations that persist (are stable) in the face of continuous deformations of an object (e.g., stretching, twisting, and bending). This perspective brings a number of advantages over other data analysis techniques [13, 67]:

- Topology studies data in a manner that is independent of the chosen coordinates.
- Topology studies data in a way that minimizes sensitivity to the metric chosen.
- Topology generalizes to high-dimensional spaces.

The main focus of this paper is a technique in the field of TDA known as persistence homology [26, 14]. The goal of persistent homology is to identify and quantify topologically dominant features within the data in the form of basic (low-dimensional) topological features such as connected components, holes, loops, and voids. This feature information can then be used by statistical and machine learning techniques to perform regression, classification, hypothesis testing, and clustering tasks [9, 10, 11, 7, 1, 53]. The TDA methodology is summarized in Figure 2. It is important to emphasize that TDA is a dimensionality reduction technique that maps data from its original high-dimensional space to a low-dimensional space that it is easier to understand and visualize. This is similar in spirit to principal component analysis (PCA), which is a statistical technique that projects the data into a low-dimensional space by extracting latent variables (principal components) that contain maximum information in terms of variance. In TDA, the latent variables are homologies that contain information in terms of geometrical features.

Figure 1: Datasets with the same mean along the $x_1, x_2$ dimensions (zero), the same standard deviation along both dimensions ($1/2$) and the same correlation between dimensions (zero). While the statistical descriptors are identical (i.e., first and second moments of a 2D Gaussian ellipse), the geometric objects that they define are different.

TDA has been used in different scientific and engineering domains. In the medical imaging field, persistent homology has been used in studying brain dendrograms [38] and in identifying brain networks in children with ADHD and autism [37]. In material science, these techniques have been used to characterize complex craze formations [31], to analyze hierarchical structures in glasses [44], and in materials informatics [12]. TDA has been used in the analysis of dynamical systems and time series [54, 51] to study gene expression [50] and dynamics of Kolmogorov flows and Raleigh-Bernard convection [36]. TDA has also been used in the analysis of time-varying functional networks [59]. In chemistry and biochemistry, persistent homology has been used to characterize protein structure,
In this work, we provide a concise summary of relevant concepts and computational methods of TDA from the perspective of chemical engineering applications. We show how to apply persistent homology techniques to analyze datasets described by point clouds and functions in high dimensions and we discuss fundamental stability results of topological features in the face of perturbations. We present multiple case studies with complex synthetic and real experimental datasets to demonstrate the benefits of TDA. Specifically, we show that TDA extracts informative features from complex datasets that correlate strongly with emerging features of practical interest. For instance, we show that the topological features of a 3D solvent environment explains reactivity in such environment and that topological features a liquid crystal explains composition of its environment. Moreover, we show how to characterize topological features of scatter fields from flow cytometry experiments. Our work seeks to open new research directions and applications of TDA in chemical engineering.

2 Fundamental Concepts of TDA

We discuss fundamental concepts and computational methods of TDA. We first introduce the notion of simplicial and cubical complexes, which are the basic geometrical constructs used to represent data objects. The representation of data objects as a simplicial or cubical complex enables the use of methods from simplicial and cubical homology to quantify the shape of the data [30, 33]. In the following discussion, we use $\mathbb{R}$ to denote the set of real numbers and $\mathbb{Z}$ to denote the set of integer numbers.

2.1 Simplices and Simplicial Complexes

A simplex is a generalization of a triangle from 2D to higher dimensions (e.g., a pyramid is a simplex in 3D). Simplexes (also referred to as simplices) spanning dimension $k = 0$ to dimension $k = 3$ are shown in Figure 3. The formal definition of a simplex is as follows.
Figure 3: Examples of $k$-dimensional simplexes. A simplex is a generalization of a triangle in high dimensions. $k = 0$ simplexes are vertices (points), 1-simplices are edges, and 2-simplices are triangles.

**Definition 2.1. k-simplex:** Given set of points $v \in \mathbb{R}^m$, a $k$-simplex is a convex set spanned by $k+1$ affinely independent points and is denoted as:

$$\sigma = \{v_1, v_2, ..., v_n, v_{k+1}\}$$

Some interesting properties of simplexes are:

1. The $m$-face of simplex $\sigma$ is the convex hull of any of its nonempty subsets, where $m + 1$ is the rank of the $m$-face.
2. An $m$-face is a simplex.
3. A 0-face is a vertex.
4. A 1-face is an edge.

A simplicial complex (denoted as $\mathcal{K}$) is obtained by connecting simplexes, as shown in Figure 4a.

Figure 4: (a) A simplicial 2-complex created by connecting a 2-simplex (a triangle) and multiple 1-simplices (edges). This simplicial complex contains two cycles, where one cycle is a hole. (b) A geometrical object (a ring) and its simplicial complex representation using 2-simplexes. Both shapes contain an empty hole and are homotopy equivalent.

**Definition 2.2. Simplicial Complex:** A simplicial complex $\mathcal{K} \in R^m$ is a collection of simplexes that satisfies the following properties:

1. Every face of $\mathcal{K}$ is also in $\mathcal{K}$
2. A nonempty intersection of simplexes $\sigma_1, \sigma_2 \in \mathcal{K}$ is a face of $\sigma_1$ and $\sigma_2$
3. The dimension of $\mathcal{K}$ is the highest dimension of its simplexes: $\dim(\mathcal{K}) = \max(\dim(\sigma) : \sigma \in \mathcal{K})$
A simplicial complex is used to characterize topological characteristics of objects. This is often done through a process known as triangulation, which is a technique used in finite element analysis to represent complex domains over which partial differential equations are solved [21]. In Figure we represent a geometrical object (a ring) as a simplicial complex. We see that the central hole of the ring, which is its main topological feature, is preserved in its representation as the simplicial complex. These objects are thus said to be homotopy equivalent (represented by the notation $\simeq$). Homotopy equivalence identifies spaces which can be deformed continuously without cutting or tearing into one another (e.g., via stretching) [43]. The flexibility of simplicial complexes allows us to create a homotopically equivalent representation of any shape. As we will see, algebraic calculations can be applied to simplicial complexes to quantify the features of the original object [30].

![Figure 5: Possible orientations of a 2-simplex.](image)

### 2.2 Simplicial Homology

Simplicial homology provides computational techniques to study topological spaces that are represented as simplicial complexes. We make some basic definitions that are necessary to explain the working principles of these techniques.

**Definition 2.3. Simplex Orientation:** The orientation of an $k$-simplex is given by the ordering of the vertices in the simplex $\{v_1, v_2, ..., v_{k+1}\}$. Two orderings can define the same orientation if and only if they differ by an even permutation; thus, there are only two allowable orientations of a simplex.

An example of the possible orientations of a 2-simplex is shown in Figure 5. We can see that only two orientations are possible for this simplex. Here, each simplex is equal to the negative of the simplex with opposite orientation; mathematically, this is stated as $\{a, b, c\} = -\{b, a, c\}$.

### 2.3 Cycles, Holes, and Homology Groups

In simplicial homology, we want to identify cycles in a given object and we want to know whether a given cycle is bounding a higher dimensional simplex (if it is not then the cycle is a hole). An example of this concept is presented in Figure 6; we see that complex $K_1$ is a cycle that bounds an empty space which constitutes a hole, while the complex $K_2$ represents a line with no holes. We now proceed to explore concepts that will allow us to systematically identify the presence of holes and cycles.

A simplicial $k$-chain on a complex $K$ is used in the identification of holes in a simplicial complex, and is defined as follows.
Figure 6: Examples of 1-simplicial complexes with the same number of vertices; \( K_1 \) contains a hole while \( K_2 \) does not.

Figure 7: Visualization of the boundary operation \( (\partial_2) \) applied to a 2-simplex. The boundary operator maps the 2-simplex onto its bounding 1-simplices and retains the orientation of the simplices.

**Definition 2.4. Simplicial \( k \)-chains:** A \( k \)-chain is a finite weighted sum defined on all \( k \)-simplicies within a complex \( K \):

\[
\sum_{i=1}^{N} c_i \sigma_i \quad (2.2)
\]

where \( c_i \in \mathbb{Z} \) and \( N \) is the number of simplices. The set of \( k \)-chains on \( K \) is written as \( C_k \). Typically, the coefficients are given by \( c_i, \{ -1, 1 \} \) and we recall that a value of \(-1\) inverts the simplex.

The boundary operator is a linear operator that maps the \( k \)-simplexes of a complex to its boundaries. The boundaries are the associated \((k-1)\)-simplexes that make up the higher dimensional \( k \)-simplex. A visualization of the boundary operator is presented in Figure 7.

**Definition 2.5. Boundary Operator:** For a set of \( k \)-chains (denoted as \( C_k \)) we define the boundary operator \( \partial \) as the mapping:

\[
\partial : C_k \rightarrow C_{k-1} \quad (2.3)
\]

The boundary operation on a general simplex \( \sigma \) with vertices \([v_1, v_2, ..., v_{k+1}]\) is shown in (2.4), where the vertex \( \hat{v}_i \) is removed from the set of vertices in the summation. The boundary operation on an \( k \)-simplex maps the simplex to a summation of its \( k - 1 \) faces.

\[
\partial(C_k) = \sum_{i=0}^{k+1} (-1)^i[v_0, ..., \hat{v}_i, ..., v_{k+1}] \quad (2.4)
\]

We use the short-hand notation \( \partial(C_k) = \partial_k \) to represent a boundary operation. The boundary operator and the \( k \)-chain operation create a sequence of vector spaces and linear maps. We show
an example for a complex $\mathcal{K}$ with $\text{dim}(\mathcal{K}) = k$ in (2.5); here, we see that the boundary operator is a mapping from the chain complex of a higher dimension to that of a lower dimension. We also note that the $k$-chain for dimensions greater than $k$ and less than 0 are zero.

\[
0 \xrightarrow{\partial_{k+1}} C_k(\mathcal{K}) \xrightarrow{\partial_k} C_{k-1}(\mathcal{K}) \xrightarrow{\partial_{k-1}} \ldots \xrightarrow{\partial_1} C_0(\mathcal{K}) \xrightarrow{\partial_0} 0
\]  

(2.5)

As an example, we apply the boundary operator to the complexes in Figure 6. We note that these complexes are built from sets of 1-simplices; thus, when we apply the boundary operator $\partial_1$, we obtain a set of vertices (0-simplices). The result of the operation on complex $\mathcal{K}_1$ is:

\[
\partial_1(\mathcal{K}_1) = \partial([a, b]) + \partial([b, c]) - \partial([a, c]) = (a - b) + (b - c) - (a - c) = 0
\]  

(2.6)

and the operation on complex $\mathcal{K}_2$ is:

\[
\partial_1(\mathcal{K}_2) = \partial([a, b]) + \partial([b, c]) = (a - b) + (b - c) = a - c
\]  

(2.7)

This example illustrates how simplicial homology uses basic boundary operations to identify cycles in a complex. We can see that the cycle formed by $\mathcal{K}_1$ is mapped to zero. In simplicial homology, a cycle is defined as an element that is mapped to the null space or kernel of the boundary operator (denoted as $\text{ker}(\partial)$) and which is zero. With these basic definitions, we can now formally define cycles and boundaries.

**Definition 2.6. Cycles:** The cycles of a given $k$-chain $C_k$ are given by:

\[
Z_k = \text{ker}(\partial_k)
\]  

(2.8)

where $\text{ker}(\partial_k)$ is the kernel of the operator $\partial_k$.

**Definition 2.7. Boundaries:** The boundaries of simplexes contained within $C_{k+1}$ are given by:

\[
B_k = \text{im}(\partial_{k+1})
\]  

(2.9)

where $\text{im}(\partial_{k+1})$ is the image of the operator $\partial_{k+1}$.

In summary, the information contained in $Z_k$ gives us the number of cycles of dimension $k$ within a given complex and the information in $B_k$ tells us whether or not the cycle is the boundary of a higher dimensional simplex. It is also important to note that $B_k$ is a subset of $Z_k$. Also, if a cycle is enclosing an empty space then it is known as a hole. We can summarize this information for any complex by defining the $k$-homology group $H_k$ and the Betti number $\beta_k$. In simple terms, the $H_k$ group is a set that contains the unique $k$-holes within a complex and $\beta_k$ counts the number of unique $k$-holes in a complex.

**Definition 2.8. $k$-Homology Group:** The $k$-homology group $H_k$ is given by the quotient group:

\[
H_k = Z_k / B_k.
\]  

(2.10)

We illustrate $H_k$ in Figure 8; here, we have two simplicial representations $z$ and $b$, where $z$ is a cycle that does not bound a higher dimensional simplex (we have $z \in Z_1$ and $z \notin B_1$) while $b$ does bound a higher dimensional simplex (we have $b \in B_1$ and $b \in Z_1$ as $B_1 \subseteq Z_1$). We can also see that both $z$ and $z + b$ contain the same hole and thus they are equivalent from a topological standpoint.
Figure 8: An illustration of the homology group $H_1$. (a) $z$ represents a cycle $z \in Z_1$ that is not a boundary of a higher dimensional simplex ($z \notin B_1$), whereas $b \in Z_1$ is a cycle that bounds a 2-simplex ($b \in B_1$). (b) The illustration demonstrates that the cycle $z$ is homotopy equivalent to $z + b$ ($z \simeq z + b$) and should not be counted as a separate hole.

The homology group $H_k$ formally defines this concept and states that, if a cycle $z_1 = z_2 + B_k$ and $z_1, z_2 \in Z_k$, then the two cycles are homotopy equivalent $z_1 \simeq z_2$ and are not independent elements of the group. This logic prevents counting the same topological feature multiple times.

Before we can formally define Betti numbers, we must define the rank of a group.

**Definition 2.9. Group Rank:** We define the rank($Z$) of a group $Z$ as:

$$\text{rank}(Z) = \min\{|Y| : Y \subseteq Z, \langle Y \rangle = Z\}$$

where $|Y|$ represents the cardinality of set $Y$ and $\langle Y \rangle$ represents the smallest subgroup of $Z$ containing every element of $Y$.

We will see that the rank of a group is analogous to the notion of the rank of a matrix (or to the dimension of a vector space). Specifically, it identifies the number of independent basis elements (known as generators) of a group. With this in mind, we define the $k^{th}$-Betti number as follows.

**Definition 2.10. $k^{th}$-Betti Number:** The $k^{th}$-Betti number $\beta_k$ is the rank of $H_k$ and is given by:

$$\beta_k = \text{rank}(H_k) = \text{rank}(Z_k) - \text{rank}(B_k)$$

We illustrate these concepts using the complex $K_1$ presented in Figure 6. This complex presents a hole; performing the necessary computations to obtain $Z_1$ and $B_1$ for $K_1$ we find the that:

$$Z_1(K_1) = \{(a, b) + (b, c) - (a, c)\}$$

$$B_1(K_1) = \{0\}$$

We now perform the quotient operation to obtain $H_1$ and $\beta_1$ for $K_1$:

$$H_1(K_1) = \{(a, b) + (b, c) - (a, c)\}/\{0\} = \{(a, b) + (b, c) - (a, c)\}$$

$$\beta_1(K_1) = \text{rank}(\{(a, b) + (b, c) - (a, c)\}) = 1$$

We see that the $H_1$ group identifies the 1-dimensional holes within the complex, and $\beta_1$ counts the number of 1-dimensional holes in the complex. The same is true for all other dimensions $k \geq 0$ as long as $\dim(K) \geq k$ because $H_k = 0$ for all $k > \dim(K)$. 

2.3.1 The 0th Homology Group ($H_0$)

The 0th homology group $H_0$ plays an important role in topological analysis. $H_0$ is the measure of the number of connected components in a complex. A component (or subcomplex) is defined as follows.

**Definition 2.11. Subcomplex:** A subcomplex is a subset $S$ of a complex $K$ such that $S$ is also a complex.

**Definition 2.12. Connected Subcomplex:** A subcomplex is connected if there exists a path made of 1-simplices from any vertex of the subcomplex to any other vertex.

The group $H_0$ describes how many disconnected subcomplexes $S$ there are within a given complex $K$. A simple example is shown in Figure 9; here, we note that the first complex $K_3$ has two disjoint subcomplexes, and the second complex $K_4$ has a single connected component. The calculations for $K_3$ are given by:

$$H_0(K_3) = \{a, b, c, d\}/\{(a - b) + (c - d)\} \quad (2.15a)$$

$$\beta_0(K_3) = 2 \quad (2.15b)$$

and for $K_3$ are:

$$H_0(K_4) = \{a, b, c, d\}/\{(a - b) + (b - c) + (c - d)\} \quad (2.16a)$$

$$\beta_0(K_4) = 1 \quad (2.16b)$$

![Example complexes with different number of connected components. $K_3$ has two connected components and $K_4$ has a single connected component.](image)

3 Computational Methods for TDA

Now that we have developed a basic understanding of simplicial homology, we can begin to streamline computations through the tools of numerical linear algebra. In order to simplify our discussion, we will define a new $k$-chain where the coefficients $c_i \in \mathbb{Z}_2$ where $\mathbb{Z}_2$ is the set of binaries $\{0, 1\}$ (rather than $c_i \in \mathbb{Z}$). With this new definition, we can remove the need for defining an orientation on a simplicial complex. In (3.17a)-(3.17c), we see that computational results are the same as those for the example provided in (2.6).

$$C_1(K_1) = [a, b] + [b, c] + [a, c] \quad (3.17a)$$

$$\partial_1(K_1) = \partial([a, b]) + \partial([b, c]) + \partial([a, c]) = (a + b) + (b + c) + (a + c) \quad (3.17b)$$

$$\partial_1(K_1) = (1 + 1)a + (1 + 1)b + (1 + 1)c = 0 + 0 + 0 = 0 \quad (3.17c)$$
We can now couple the newly defined $k$-chain with the boundary operator to create what is known as a boundary matrix $B \in \mathbb{Z}_2^{n \times m}$ where $n$ represents the simplexes of $\text{dim}(k-1)$ in $\mathcal{K}$ and $m$ represents the simplexes of $\text{dim}(k)$ in $\mathcal{K}$. The boundary matrix is built based on the following rules:

- The face of a simplex precedes the simplex in column (row) index.
- An entry of one is placed in position $(j, i)$ if $\sigma_i$ is a face of $\sigma_j$, otherwise it is a zero.
- If two simplices are of the same dimension, lexicographic ordering is used (i.e., $a < b < ... < z$).

For example, we take complex in Figure 4a and compute the unique 1-holes. We show matrix $B$ for $\partial_0$ in Table 1, for $\partial_1$ in Table 2, and for $\partial_2$ in Table 3. Again, the boundary matrix associated with $\partial_1$ is mapping the 1-simplices of the complex to its boundaries and similarly for $\partial_2$.

### Table 1: Boundary matrix $B_0$ for $\partial_0$ of complex in Figure 4a

| $\partial_0$ | [a] | [b] | [c] | [d] | [e] |
|--------------|-----|-----|-----|-----|-----|
| [0]          | 0   | 0   | 0   | 0   | 0   |

### Table 2: Boundary matrix $B_1$ for $\partial_1$ of complex in Figure 4a

| $\partial_1$ | [a,b] | [a,c] | [a,d] | [b,c] | [b,d] | [d,e] |
|--------------|-------|-------|-------|-------|-------|-------|
| [a]          | 1     | 1     | 1     | 0     | 0     | 0     |
| [b]          | 1     | 0     | 0     | 1     | 1     | 0     |
| [c]          | 0     | 1     | 0     | 1     | 0     | 0     |
| [d]          | 0     | 0     | 1     | 0     | 1     | 1     |
| [e]          | 0     | 0     | 0     | 0     | 0     | 1     |

### Table 3: Boundary matrix $B_2$ for $\partial_2$ of complex in Figure 4a

| $\partial_2$ | [a,b,c] |
|--------------|---------|
| [a,b]        | 1       |
| [a,c]        | 1       |
| [b,c]        | 1       |
| [b,d]        | 0       |
| [d,e]        | 0       |

In order to compute $\text{rank}(Z_j)$ and $\text{rank}(B_j)$ and the number of $k$-holes in the complex (the Betti number $\beta_k$), we must first reduce the matrices to a canonical form known as the Smith normal form (SNF).

**Definition 3.1. Smith Normal Form:** A matrix $M \in \mathbb{Z}_2^{n \times m}$ is in Smith normal form if it is diagonal and if it can be obtained by multiplying $M$ by invertible matrices $S \in \mathbb{Z}_2^{n \times n}$ and $T \in \mathbb{Z}_2^{m \times m}$ as $M_{\text{SNF}} = SMT$. 

10
The reduced matrix $B_{1\text{SNF}}$ is shown in Table 4 and $B_{2\text{SNF}}$ is shown in Table 3. The $\partial_0$ matrix cannot be further reduced and thus it is not shown. The SNF matrices contain all the information required to find $\text{rank}(Z_k)$ and $\text{rank}(B_k)$ for $k \in \{1, 2\}$:

$$\text{rank}(Z_k) = m - \text{rank}(B_{(k)\text{SNF}})$$  \hspace{1cm} (3.18a)
$$\text{rank}(B_k) = \text{rank}(B_{(k)\text{SNF}})$$  \hspace{1cm} (3.18b)

These simple calculations show us that $\text{rank}(Z_1) = 2$ and $\text{rank}(B_1) = 1$. From this we can see that the number of 1-holes in our complex is $\beta_1 = \text{rank}(Z_1) - \text{rank}(B_1) = 1$, as expected.

| $\partial_1$ | [a,b] | [a,c] | [a,d] | [b,c] | [b,d] | [d,e] |
|--------------|-------|-------|-------|-------|-------|-------|
| [a]          | 1     | 0     | 0     | 0     | 0     | 0     |
| [b]          | 0     | 1     | 0     | 0     | 0     | 0     |
| [c]          | 0     | 0     | 1     | 0     | 0     | 0     |
| [d]          | 0     | 0     | 0     | 0     | 0     | 1     |
| [e]          | 0     | 0     | 0     | 0     | 0     | 0     |

Table 4: Reduced boundary matrix $B_{1\text{SNF}}$ for $\partial_1$ of the simplicial complex in Figure 4a

| $\partial_2$ | [a,b,c] |
|--------------|--------|
| [a,b]        | 1      |
| [a,c]        | 0      |
| [b,c]        | 0      |
| [b,d]        | 0      |
| [d,e]        | 0      |

Table 5: Reduced boundary matrix $B_{2\text{SNF}}$ for $\partial_2$ of the simplicial complex in Figure 4a

## 4 Persistent Homology

Persistent homology is a methodology originally proposed by Gunnar Carlsson and further developed by many others for extracting and quantifying topological information from data [13]. This methodology is discussed in detail in [68, 6, 17, 24, 19, 26]. The homology of data provides deep insight into the structure of the data and provides quantification capabilities of their geometrical features [26, 19].

### 4.1 Building Simplicial Complexes from Data

Directly computing the homology of an arbitrary object defined by the space $\mathcal{X} \in \mathbb{R}^n$ is a complex task. To simplify such task, we leverage simplicial homology; here, we identify a simplicial complex $\mathcal{K}$ such that its homology is the same or similar to that of $\mathcal{X}$. We can define such a complex $\mathcal{K}$ by creating a geometrical object known as the cover $\mathcal{U}$ of $\mathcal{X}$.

**Definition 4.1. Cover:** A cover $\mathcal{U}$ of a metric space $\mathcal{X}$ is the union of indexed sets $U_i$ where $\mathcal{X} \subseteq \bigcup_{i \in I} U_i$. 

Figure 10: A cover of points $x_i \in X$ is defined by a set of balls $B(x_i, \epsilon)$ expanded around each point.

If we define our metric space to be in $\mathbb{R}^n$, then we can imagine each set $U_i$ to be a ball $\{B(x_i, \epsilon) : x_i \in X, \epsilon \in \mathbb{R}\}$ centered around each point $x_i \in X$, where $\epsilon$ is the ball radius. An example of such a cover is presented in Figure 10. We now utilize this cover to develop a simplicial complex, known as a Čech complex.

**Definition 4.2. Čech Complex:** The Čech Complex $(\tilde{C})$ is a simplicial complex built from $k$-simplicies that are the non-empty intersection of $k+1$ sets of a cover $U$.

A Čech complex is also known as the “nerve” of the cover $U$.

**Definition 4.3. Nerve:** The nerve of collection $U = \cup\{U_{i \in I}\}$ is the simplicial complex with vertices $I$ and $k$-simplices built from $\{i_0, i_1, ..., i_k\}$ if and only if $U_{i_0} \cap U_{i_1} \cap ... \cap U_{i_k} \neq \emptyset$.

The so-called “Nerve Theorem” states that the nerve of the cover $U$ of $X$ has the same homology as that of $X$ [3, 27]. This is a powerful result as indicates that we can develop a simplicial complex $K$ for a space $X$ such that their homology is preserved. With this, we can apply the calculations and analysis of simplicial homology directly to our data. However, there is one caveat that is important to note, which is the selection of the distance $\epsilon$. An example of the nerve of a dataset with varying levels of $\epsilon$ is shown in Figure 11. We can see that, as we adjust $\epsilon$ of the cover $U$, we obtain different Čech complexes, each with a different homology. We want to ensure that the homology captures the most interesting features of the data. In the complex shown in Figure 11, these features are the two clusters of points; here, one cluster forms a loop and the other does not. It is easy to see in this example what range of $\epsilon$ values would be most effective at capturing this information. However, if our dataset is of much higher dimension, finding the correct $\epsilon$ is much more difficult. Consequently, we characterize the dataset for multiple values of $\epsilon$. This information is captured by a filtered simplicial complex [13].

**Definition 4.4. Filtered Simplicial Complex:** A filtered simplicial complex $K \in \mathbb{R}^m$ is a simplicial complex for which there is a series of nested simplicial subcomplexes $K_\epsilon \in \mathbb{R}^m$ such that:

$$K_{\epsilon_0} \subset K_{\epsilon_1} \subset ... \subset K_{\epsilon_n}$$

where $\epsilon_0 < \epsilon_1 < \cdots < \epsilon_n$.

Referring back to the example in which we define the cover $U$ as a set of balls $B(x_i, \epsilon)$ of radius $\epsilon$, we can view the filtered complex as the set of nerves that are formed as we expand $\epsilon$. Figure 11 demonstrates that $K_{\epsilon=0.1} \subset K_{\epsilon=0.3} \subset K_{\epsilon=0.5} \subset K_{\epsilon=1}$. Note that the interesting features of the data
Figure 11: Filtration of points \( x_i \in \mathcal{X} \) by a set of balls \( B(x_i, \epsilon) \) with expanding \( \epsilon \). As \( \epsilon \) is increased (filtered) the topology of the Czech complex shifts and this introduces holes and higher dimensional simplices. This filtration builds a filtered simplicial complex \( (K_{\epsilon=0} \subset K_{\epsilon=0.1} \subset K_{\epsilon=0.5} \subset K_{\epsilon=1}) \).

(the two clusters and one loop) are present in the homology of the subcomplexes (persist) during a large portion of the filtration. The main goal of this analysis is to identify persistence intervals for \( \epsilon \) over which topological features of the filtered complex are born (appear) and die (disappear).

**Definition 4.5. Birth:** For a filtered complex \( K \) and subcomplexes \( K_i, K_j \) where \( i < j \). A topological feature \( x \in H_p(K_j) \) is born at \( j \) if \( x \notin H_p(K_i) \).

**Definition 4.6. Death:** For a filtered complex \( K \) and subcomplexes \( K_i, K_j \) where \( i < j \). A topological feature \( x \in H_p(K_i) \) dies at \( j \) if \( x \notin H_p(K_j) \).

**Definition 4.7. Persistence Interval:** For a given topological feature \( x \) with birth point \( i \) and death point \( j \), the persistence interval \( \text{Int} \) for the feature is given by:

\[
\text{Int} = [i, j] : i, j \in \mathbb{R}
\]  

If \( j = \infty \) then the component does not die during the filtration (persists forever).

With a filtration we are identifying the appearance and disappearance of topologically interesting features in our dataset. The filtered complex in Figure 11 demonstrates this concept. We can see that the hole in \( H_1 \) of the dataset is born at \( \epsilon = 0.5 \) and is completely filled in at \( \epsilon = 1 \), which is a majority of the filtration. We can also see that the individual points become two connected components at \( \epsilon = 1 \) and then become a single component at \( \epsilon = 5 \). Thus, the longest persistence intervals in both \( H_1 \) and \( H_0 \) capture the defining topological characteristics of the filtered complex.

### 4.2 Persistence Diagrams

The topological feature information contained within a filtered complex is summarized into what is known as a persistence diagram (PD) [24]. This can be computed via an extension of the matrix methods presented in Section 3 [23]. The PD is a visual method that represents the birth \( (x) \) and death \( (y) \) of topological features as a point \( \{x, y\} \in \mathbb{R}^2 \). The persistence diagram associated with the filtration in Figure 11 is shown in Figure 12. This diagram represents the birth and death of the features of the \( H_0 \) and \( H_1 \) homology groups for the filtered complex and represents the persistence interval associated with each feature as the vertical distance from the diagonal. This information allows for a direct visual understanding of the topology of the dataset.
Figure 12: Filtration of points $x_i \in X$ by a set of balls $B(x_i, \epsilon)$ with expanding $\epsilon$ and its corresponding persistence diagram. The PD records the $\epsilon$ value at which topological features are born and the $\epsilon$ value of their death during the filtration. For example, the cycle born at $\epsilon = 3$ ($x = 3$) dies at $\epsilon = 5$ ($y = 5$) when it is filled in, with a total persistence of $5 - 3 = 2$, which is seen in the PD.

An important property of PDs is that they can be vectorized to enable quantification and these vectors can be used to perform tasks such as regression, classification, or clustering. For instance, as we will see in Section 5, one can apply PCA to the vectorized PDs to identify clusters defined by different topological features.

**Definition 4.8. Vectorization**: A persistence diagram $PD_{X_i}$ of a dataset $X_i$ is vectorized by mapping the $PD_{X_i}$ to a vector $\vec{PD}_{X_i} \in \mathbb{R}^q$ through a mapping of the form $\phi: PD_i \rightarrow \vec{PD}_{X_i}$.

There are multiple ways to vectorize a PD, the most widely used mappings are the persistence landscape [11, 9] and the persistence image [1] (we will focus on the latter). The persistence image is a smoothed representation of the points $\{x, y\}$ in a persistence diagram $\{x, y\} \in PD$. Typically, the smoothing is done by applying a Gaussian kernel (with mean $u$ and variance $\sigma^2$) to each of the points $\{x, y\} \in PD$:

$$\phi(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x-u_x)^2 + (y-u_y)^2}{\sigma^2}\right)$$  \hspace{1cm} (4.21)

**Definition 4.9. Persistence Surface**: The persistence surface is a scalar mapping $\rho : \mathbb{R}^2 \rightarrow \mathbb{R}$ with weighting function $w : \mathbb{R}^2 \rightarrow \mathbb{R}$ and is defined as:

$$\rho(z) = \sum_{u \in PD} w(u)\phi(z)$$  \hspace{1cm} (4.22)

A persistence image is the discretization of the persistence surface.

**Definition 4.10. Persistence Image**: For a given PD, the persistence image $\vec{\rho} \in \mathbb{R}^{n \times m}$ is the collection of pixels $\vec{\rho}(p) = \int p \rho dxdy$, where $p = \{x, y\} : x, y \in \mathbb{R}$.

The ultimate goal of vectorization methods is to create a stable representation of the PD.
**Definition 4.11. Stability**: A vector representation $\vec{PD}_{X_i}$ of a persistence diagram $PD_{X_i}$ is said to be **stable** if small perturbations in $PD_{X_i}$ (represented as $PD'_{X_i}$) results in a bounded changes in $\vec{PD}'_{X_i}$.

Mathematically, we establish the stability property as:

$$\text{dist}(\vec{PD}_{X_i}, \vec{PD}'_{X_i}) \leq L \cdot \text{dist}(PD_{X_i}, PD'_{X_i}) \quad (4.23)$$

where $\text{dist}(\cdot, \cdot)$ represents a distance metric and $L$ represents a scalar constant.

Notably, it has been proven that persistence landscapes and persistence images are stable under multiple distance metrics [11, 9, 1]. Intuitively, stability indicates that topology changes in a continuous manner under perturbations. This makes them excellent representations of PD and amenable to use in diverse tasks such as regression and classification, as we demonstrate in Section 5.

The distance between the vectorized PDs can be expressed in terms of $L_p$ norms. The distance between PD is measured using the Wasserstein distance or the Bottleneck distance.

**Definition 4.12. Wasserstein distance**: The $p^{th}$-Wasserstein distance between persistence diagrams $PD_1$ and $PD_2$ is defined as:

$$d_{W_p}(PD_1, PD_2) = \left( \inf_{\gamma} \sum_{x \in PD_1} \|x - \gamma(x)\|_p \right)^{1/p} \quad (4.24)$$

where $\gamma$ ranges over all possible bijections from $PD_1$ to $PD_2$.

**Definition 4.13. Bottleneck distance**: The bottleneck distance between persistence diagrams, $PD_1$ and $PD_2$, is defined as:

$$d_B(PD_1, PD_2) = \inf_{\gamma} \sup_{x} \|x - \gamma(x)\|_{\infty} \quad (4.25)$$

where $\gamma$ ranges over all possible bijections from $PD_1$ to $PD_2$.

### 4.3 Topology of Continuous Functions

In the previous sections we focus on understanding the topology of datasets that are made of cloud points. We now discuss how to quantify the topology of continuous functions. An example of this type of object is the scalar function shown Figure 13. Topologically, the interesting features of this function are its critical points (min and max points). In order to characterize these critical points we utilize a new form of filtration known as a Morse filtration or level set filtration [42]. The Morse filtration is derived from ideas of Morse theory, which is the study of the topology of manifolds through differential functions and the analysis of the critical points of these functions [41]. Here, we consider the graph of a continuous function as a differentiable manifold [52].

**Definition 4.14. Level Set**: Given a differentiable manifold $M$ and function $f : M \rightarrow \mathbb{R}$, the level set $M^a$ at a point $a$ is defined as the pre-image:

$$M^a = f^{-1}(a) = \{ x \in M : f(x) = a \}. \quad (4.26)$$

The level set contains all points of the manifold that have the same function value. We use the level set to filter through every function value and build a filtered complex within a sublevel set.
Figure 13: Morse filtration of a function $f(x)$; the filtration increases the value of the level set and records the number of connected components. The topology of a continuous function changes at its critical points (i.e. separate components become connected at a maximum). The persistence diagram records the appearance and disappearance of these topological features.

**Definition 4.15. Sublevel Set**: A sublevel set $M^{(a,b)}$, where $a, b \in \mathbb{R}$ and $a = -\infty$ is defined as:

$$M^{(-\infty,b)} = \{x \in M : x \leq f(b)\}. \quad (4.27)$$

As we pass through the Morse filtration and build the sublevel set of the function, we are creating a well-defined filtered complex. The topology of the function will change as the filtration passes through critical points in the manifold [42]. These topological changes are quantified in a persistence diagram and is vectorized for analysis. An example of this type of filtration and a persistence diagram are shown in Figure 13. The Morse filtration is a good choice for functions that are continuous or that can be approximated as piece-wise functions (e.g., a time series). The method can be expanded to $k$-dimensional functions [29]. This makes it a powerful approach to characterize complex surfaces (landscapes) that have many minima/maxima. We demonstrate this technique on 2D and 3D functions in Section 5.

### 4.3.1 Stability of Persistence Diagrams for Functions

Persistence diagrams of real valued functions are also stable representations of data. The following Theorem 1, established by Cohen-Steiner and co-workers, highlights this result [19].

**Theorem 1.** Given real valued functions $f, g$ with finitely many critical points and their corresponding persistence diagrams $PD_f, PD_g$, we have that:

$$d_B(PD_f, PD_g) \leq \|f - g\|_{\infty} \quad (4.28)$$

where $\| \cdot \|_{\infty}$ represents the $L_{\infty}$ norm between two functions over the space $X$:

$$\|f - g\|_{\infty} = \sup\{||f(x) - g(x)|| : x \in X\}. \quad (4.29)$$
An illustration of the stability of persistence diagrams is shown in Figure 14. Here, we can see that the persistence diagram of the functions is similar. The presence of the strong or highly persistent critical points are well-captured in both diagrams. We can also see that the persistence diagram captures the structure of the weak critical points (arising from noise), which have short persistence.

Figure 14: Filtration of functions \( f(x) \) and \( \hat{f}(x) \). The strong critical points of both functions are captured in the persistence diagram analysis while the weak critical points (arising from noise) remain close to the diagonal as they have minimal persistence.

4.4 Cubical Complexes and Images

We briefly discuss cubical complexes and cubical homology as these are important in understanding data over rectangular domains such as images (represented by pixels in 2D or voxels in 3D) [4, 45]. Cubical homology is similar to simplicial homology but uses a different basis shape. This highlights the fact that one can choose different basis shapes in topological analysis. The basis shapes for cubical complexes are \( k \)-cubes (hypercubes) or elementary cubes. In Figure 15a we show \( k \)-cubes of dimension 0 through 3. The example cubical complex in Figure 15b demonstrates that a cubical complex is built from a series of \( k \)-cubes of possibly varying dimension.

Filtered cubical complexes can be developed from data and persistence homology calculations on the filtered complexes, similar to filtered simplicial complexes. An important application of cubical analysis is the analysis of images. An image can be viewed as a 2D surface embedded in three dimensions where two dimensions are the coordinates of each pixel, and the third dimension is the scalar value.
value associated with each pixel (e.g., intensity). We can also view an image as an approximation of a continuous object over which a Morse filtration can be performed. We demonstrate the filtration on the image in Figure 16a. We filter through the level sets of this image and develop the filtered complex through the sublevel set. This filtration is demonstrated in Figure 16b, where \( C_{f=1} \subseteq C_{f=3} \subseteq C_{f=5} \). The persistence diagram is then represented in Figure 17. The PD is able to capture the dominant topological features of our system such as the presence of the two critical points in the image and the fact that there is only a single connected component. The persistence diagram generated from a cubical complex filtration and a simplicial complex filtration have the same properties and can be vectorized in the same way. Thus, we can extend our analysis from discrete point data to that of images or other high-dimensional continuous objects.

![Figure 16](attachment:image.png)

Figure 16: (a) Representation of an image; the values within each pixel represent intensity. (b) Filtration of the image. In (b) the top row represents filtration on the image itself and the corresponding sublevel sets, the bottom row demonstrates cubical complex representation during the filtration.

![Figure 17](attachment:image.png)

Figure 17: Persistence diagram for the filtration of the image in Figure 16a. The PD reveals the structure of the image, which changes at the critical points of the image (i.e., \( f = 3 \) and \( f = 5 \).)
4.5 Inverse Analysis

TDA provides the ability to identify features of a given dataset that drive persistence (inverse analysis). This information can be used, for instance, to uncover physical phenomena from the data. The mathematics of inverse analysis methods is complex and will not be covered in detail here; for details please refer to [46, 22]. We illustrate this method on the dataset and persistence diagram shown in Figure 12. Here, we want to identify the features of the point cloud that have the longest persistence. The inverse mapping is shown in Figure 18; here, the arrows represent the inverse mapping from the persistence diagram to the dataset features. The most persistent feature in $H_1$ is related to the ring formed by the four points highlighted in green. The most persistent feature in $H_0$ represents the distinct clusters of the data in red.

Inverse analysis can also be used to understand dominant features that drive classification and regression results. For instance, the defining characteristics of a given class within a dataset can be identified via regression/classification in the space of PDs. These defining characteristics can then be mapped back to the original data set which can be used to gain physical understanding of these differences. Applications of these methods in material science are found in [12, 44, 31]. Here, inverse analysis is used to identify fracture or degradation sites in materials and to identify pore configurations in granular crystallization.

![Inverse mapping of the features of the persistence diagram to the features dataset in Figure 12. The inverse mapping for $H_1$ identifies the loop in the dataset and the inverse mapping for $H_0$ identifies the distinct clusters of the data.](image)

5 Applications

We now proceed to demonstrate how these methods can be applied to different types of datasets. To do so, we use a couple of illustrative examples and real datasets derived from soft materials and molecular dynamics simulations. All the calculations presented were conducted in Python [48] using the TDA packages GUDHI [39] and Homcloud [47]. All scripts needed to reproduce the results can be found in https://github.com/zavalab/ML/tree/master/TDAPaper.
5.1 Topology of Point Clouds

We illustrate how to use TDA to analyze point clouds; specifically, we seek to extract topological features from the data to perform binary classification of point clouds. The clouds used here are collections of points in two dimensions $x_1, x_2$ (for visualization purposes). In actual applications, one can conduct analysis on a point cloud of any dimension. We represent each cloud as $X_i$ and each cloud can belong to two different types of classes (Class 1 or Class 2). Our goal is to take each point cloud $X_i$ as input, project this data to their respective $H_1$ persistence diagram $PD_{X_i}$ through an epsilon ball filtration (i.e., extract the topological features), vectorize the PDs ($\overrightarrow{PD_{X_i}} \in \mathbb{R}^q$), and perform classification of the cloud based on the vectorized topological features.

The point cloud classes are shown in Figure 19 and the $H_1$ PDs are found in Figure 20. Note how the point clouds of Class 1 define a simple object (ellipse) while those of Class 2 define a more complex object (overlapping ellipses). We can see that the persistent diagrams are visually distinct; specifically, point clouds of Class 2 have features that persist over a longer range of the filtration.

We utilize the persistence image method to vectorize the PDs. We apply principal component analysis (PCA) to the vectorized PDs verify if there is a separation between Class 1 and Class 2. We emphasize that the PCA projection is not applied to the original datasets $X_i$ but to the transformed datasets $\overrightarrow{PD_{X_i}}$ obtained from TDA. The PCA projection on first and second principal components is shown in Figure 20c. This shows an obvious separation between Class 1 and Class 2, which means that the topological features extracted from the data (contained in $H_1$) are informative. This also suggests that a simple linear classifier using vectorized PDs as features should work well. To test this hypothesis, we apply a linear support vector machine (SVM) classifier using the $\overrightarrow{PD_{X_i}}$ as features; we find that we can perfectly classify the datasets (we use a 5-fold cross validation scheme). This again indicates that the topological features extracted with TDA are highly informative.

An advantage of utilizing a linear classifier is the ability to extract which features are the ones driving classification. Specifically, the magnitude of the weights of the SVM classifier $w \in \mathbb{R}^q$ can be directly associated with the importance of each feature of the vectorized PDs [57]. Weights with a large negative value are characteristic of point cloud in Class 1 and weights with a large positive value are characteristic of point clouds in Class 2. A visualization of the weights is shown in Figure 21a. From this representation, we can see that Class 1 is characterized by 1-D holes that are born and die in the early stages of the filtration suggesting a higher number of small radius holes. Class 2 is characterized by 1-D holes that persist over multiple stages of the filtration, suggesting the presence of holes with large radius. These results highlight how one can exploit topological information obtained with TDA to perform statistical (PCA) or machine learning tasks (SVM classification).

Work conducted in [46] has led to the development a set of techniques that are useful in the interpretation of PDs. These techniques allow for the identification of volume-optimal cycles, which are cycles that correspond to the minimal representation of the topological features identified in a PD. This technique has been implemented in the Homcloud software; with this, we can identify the 1D holes that are responsible for differences between the classes. In Figure 21 we create masks for the areas of the PD that are the most important for identifying classes. The mask identifies which areas of the PD are important for the classification and inverse analysis identifies features of the data corresponding to the PD areas. The inverse analysis for a sample of Class 1 and Class 2 is shown in Figure 22. For Class 2 we see larger separation between points and larger holes that persist for a longer period of the filtration. In Class 1 we see that the separation between points is smaller, resulting in smaller loops that are formed early and die quickly.
Figure 19: Types of point clouds analyzed using epsilon ball filtration.

Figure 20: Persistence diagrams for Class 1 and Class 2 point clouds and the corresponding PCA analysis of this persistence diagram space. (a) $H_1$ persistence diagram for Class 1. (b) $H_1$ persistence diagram for Class 2. (c) Principal components of vectorized PDs for Class 1 and Class 2 datasets. It is clear that there is separation of the persistent diagrams.

Figure 21: Masks highlight the areas of the PD that are important in distinguishing Class 1 from Class 2. We perform inverse analysis on these areas to visualize what features of the original data distinguishing classes. (a) Weights from SVM classification in the space of PDs. The areas of the diagram that distinguish Class 2 are in red and the areas of the diagram that distinguish Class 1 are in blue. (b) PD mask for Class 1. (c) PD mask for Class 2.
Inverse $H_1$ analysis for Class 1.

Inverse $H_1$ analysis for Class 2.

Figure 22: Inverse analysis based on classification weights for Class 1 and Class 2. The analysis reveals that the classifier is separating the classes based on the presence of large cycles in Class 2 and the higher density of smaller cycles in Class 1.

5.2 Topology of Time-Series and Phase-Planes

Persistence homology has seen applications in the area of time series analysis [54, 60, 64, 49, 35]. A simple example of the application of persistence homology is the analysis of the topology of phase-planes generated by a dynamical system. An example for two state variables $f_1, f_2$ is shown in Figure 23. The phase plane for this periodic system defines an ellipse, which is easy to characterize (e.g., in terms of its axes). We can add complexity to the topology of the phase plane by perturbing the dynamical system. For example, by adding a perturbation, we change the phase-plane to that shown in Figure 24. The topology of the new plane cannot be fully characterized using simple ellipsoids.

The analysis of the phase plane topology through an epsilon ball filtration allows us to differentiate the dynamics of the perturbed and unperturbed systems. We compare their PDs in Figure 25 and 26; the unperturbed system contains a single highly persistent cycle while the perturbed system contains four cycles that are less persistent.

Figure 23: Phase plane for periodic orbit with two states. The plane is represented as cloud points from the edge of an ellipse and is ideal for a geometric analysis.

We can also use persistent homology within a windowing method to detect when a change has occurred in the dynamics of a system (e.g., for fault detection). This concept is illustrated in Figure 27; here we can see that the first, second, and fourth windows have similar topology while the third window has a different topology. We compare the PDs of each window in Figure 27; the PDs clearly reveal that that phase plane of the third window is different.
Another common perturbation of a system is noise; an important observation here is that noise usually has the effect of introducing local effects to a trajectory but does not distort the overall topology of the trajectory. This is illustrated in Figure 28, where we can see that the ellipse shape of the
Figure 27: Windowing method applied to time series. The first, second, and fourth window have phase planes with similar topologies while the third window has an obvious shift (which introduces a change in the topology of its plane). This change in topology is captured by the persistence diagrams.
phase plane is retained. In other words, the topology of the phase plane is stable in the phase of noise perturbations. In a real system, one may wish to characterize whether noise-free and noisy systems exhibit similar dynamics. One way to do this is to perform persistent homology calculations on both phase planes and to compare the resulting PDs. We show the results of this analysis in Figure 29; the PDs reveal that both phase planes contain a large cycle (which indicates that they have phase planes with similar topologies). This is seen from the point far away from the diagonal line, which has long persistence over the filtration. Features created by noise are shown at the bottom of the diagonal (these features have short persistence). These results demonstrate the stability of PDs [16], which is an important concept in TDA. These results also highlight that TDA is a powerful tool for the classification of time series [60], identification of periodic orbits and shifts [51], and for change point detection [28]. The paper of Perea and Edelsbrunner [49] provides an excellent overview of TDA in signal processing.

![Figure 28: Phase plane for noisy $f_1$ and $f_2$ shows a similar topology to the noiseless counterparts.](image)

(a) Time series with added Gaussian noise. 
(b) Phase plane representation.

![Figure 29: Persistence diagrams for noisy and noiseless functions. Note that the dominant feature (cycle) persists.](image)

(a) PD for functions $f_1$ and $f_2$ 
(b) PD for functions $f_1$ and $f_2$ with added noise

5.3 Topology of 2D Scalar Fields

In this example we use of TDA to analyze the topology of 2D scatter fields over a domain $U \in \mathbb{R}^2$. We generate data by applying propagating a random field \( \{u_{i,j} : u_{i,j} \in U(0,1)\} \) using the dynamic 2D diffusion equation (5.30) and we obtain the final steady-state field. We generate fields with different textures by using different diffusion coefficients $D$. The resulting fields are used as the datasets;
illustrative examples for different diffusion coefficients are shown in Figure 30. Here, the blue color are points of small intensity (small values of $u_{i,j}$) while the red color are points of high intensity. We see that small coefficients generate textures that are more granular while large coefficients generate smoother textures. Our goal is to characterize the geometry of the fields to investigate if their underlying structure can be correlated to the diffusion coefficient. In our analysis, we represent the scalar field as a function (in 3D), as shown in Figure 31. This functional representation of the field reveals that low and high intensity points define critical points of the function. This also reveals that the field has complex topological features (many critical points with no obvious patterns are present).

$$\frac{u_{i,j}^{(n+1)} - u_{i,j}^{(n)}}{\Delta t} = D \left[ \frac{u_{i+1,j}^{(n)} - 2u_{i,j}^{(n)} + u_{i-1,j}^{(n)}}{\Delta x^2} + \frac{u_{i,j+1}^{(n)} - 2u_{i,j}^{(n)} + u_{i,j-1}^{(n)}}{\Delta y^2} \right]$$

(5.30)

Figure 30: Samples from 2D scalar fields with their corresponding diffusion coefficient ($D$) value.

Figure 31: 3D-functional representation of a scatter field with diffusion coefficient $D = 0.6$. The function is treated as a cubical complex and the filtration is performed over the scalar value.

To illustrate the benefits of using TDA against other techniques, we investigate whether the structure of the dataset can be revealed by direct application of PCA [32] and diffusion maps [20]. This is a simple naive comparison but will demonstrate that the datasets have nontrivial structure. The projection of the data onto the first two principal components is shown in Figure 32a. Here, we highlight the points based on the associated diffusion coefficient. We also apply the diffusion maps, which is a
nonlinear dimensionality reduction technique, and we obtain similar results (see Figure 32b). From these results we see that the features extracted by PCA and diffusion maps do not correlate to the diffusion coefficient.

Figure 32: Dimensionality reduction for the 2D dimensional scalar fields using PCA and diffusion maps.

Figure 33: Evolution of PDs with the diffusion coefficient. A dependence of the topology with the diffusion coefficient emerges.

Figure 34: PCA performed on the PD for two-dimensional scalar fields. This reveals that the geometry of the dataset is directly related to the diffusion coefficient.
We now apply TDA to the 3D field functions and extract persistence diagrams. Example persistence diagrams for $H_0$ and $H_1$ are shown in Figure 33. We see that there is a visual shift in the PDs as we increase the diffusion coefficient. This seems to indicate that the PDs vary continuously with the diffusion coefficient. To verify this, we vectorize the PDs and apply PCA to the vectorized diagrams. The projection of these onto the dominant principal components is shown in Figure 34. It is clear that there is a continuous dependence of the PD on the diffusion coefficient (it forms a continuous manifold). This result provides another demonstration of the stability of persistence diagrams and on how topology varies continuously under perturbations. Specifically, stability indicates that small changes in a given function $(f, g)$ results in bounded changes in the associated persistence diagrams $(PD_f, PD_g)$. Thus, because our perturbations to each function are based on changes in $D$, we can guarantee that the distance between PDs is bounded by the size in the perturbation in the diffusion coefficient (i.e., the distance is not arbitrary).

5.4 Topology of Images

We now illustrate how to use TDA to analyze images. Specifically, we analyze the optical response of liquid crystal sensors to air contaminants, in particular dimethyl methylphosphonate (DMMP) [57, 55]. We analyze the response of a sensor in the presence of DMMP and in the presence of humid nitrogen (water). The sensor responds to both DMMP and water, but there are subtle spatial differences in the optical response of the sensor to these different environments. The working principle of these sensors relies on a change in orientation of liquid crystal molecules in a film when exposed to an air contaminant (analyte). The change in orientation results in optical fields with different spatial and color features (see Figure 35). We use TDA to investigate if the topological features of these patterns present a dependence on the air environment. Such information can be used to design sensors (i.e., we can calibrate the sensor by correlating the optical response to the presence of DMMP).

Figure 35: Optical patterns for a liquid crystal sensor when exposed to DMMP or Water.

Each optical micrograph is an image with three channels (Red, Green, and Blue). We project these three channels onto a single grayscale channel by computing the total intensity of each pixel in the image. The conversion of the image to a single channel allows us to treat the image as a cubical complex over which we can perform a simple Morse filtration (level sets defined in terms of intensity, as done in the previous diffusion field example). A grayscale image was used to simplify the computations; however, more complicated approaches could be taken to deal with the three color channels. In order to understand the important of the information contained in the Morse filtration analysis, we apply a linear SVM to the persistence diagrams associated to our images. We find that the topological features of the images gives us a classification accuracy of $85 \pm 2\%$ (for a dataset of more than
Figure 36: (a) Areas corresponding to optimal weights from linear SVM classification. The areas of the PD that distinguish DMMP are shown in red and the ones that distinguish water are shown in blue. Inverse analysis based on SVM weights for (b) DMMP and (c) Water responses. Note that the camera artifact in (c) has no highlighted areas, demonstrating that the extracted features are physically relevant.

1,000 images). In order to identify the characteristic features for the responses at high and low concentration, we utilize the classification weights of the linear SVM model. The classification weights are visualized in Figure 36. We apply a masking method to identify the portions of the persistence diagram that are critical for defining whether a response pattern is a result of high or low concentration. We can use these masked areas to identify the features of the images that separate the high concentration patterns from the low concentration patterns. To visualize the geometric differences between the patterns associated with DMMP and water, we again utilize the Homcloud software to perform the inverse analysis via volume optimal cycles. Here, we focus on inverse analysis for the $H_1$ homology group. The results of this analysis for a couple of sample images is found in Figure 36. Inverse analysis reveals that, when the sensors are exposed to water, the pattern exhibits many small distinct clusters; in contrast, when the sensor is exposed to DMMP, there are few large clusters. This shows how inverse analysis allows us to pinpoint topological features of the image that drive classification.

5.5 Topology of Probability Density Functions

The analysis of the shape of probability density functions is typically done using summarizing statistics (e.g., moments such as skewness and kurtosis) or through parametric techniques (fitting a parametric model such as a Gaussian mixture to the data) \[63\]. These models are powerful in their simplicity but might not be flexible enough to capture complex features of density functions (particularly in high dimensions). In this example we explore the shape of complex density functions by using topological analysis techniques. We use an experimental flow cytometry dataset to illustrate how this can be done. The flow cytometry dataset was obtained through the FlowRepository \[58\] (Repository ID: FR-FCM-ZZC9). This dataset represents a temporal study of the kinetics of gene transcription and protein translation within stimulated human blood mononuclear cells through the quantification of proteins (CD4 and IFN-γ) and mRNA (CD4 and IFN-γ) \[61\]. In our study, we focus on the evolution of the concentration of CD4 mRNA and IFN-γ mRNA in a given cell which is measured via a flow cytometer. At each time point in the study, a number of cells (∼ 15,000) are passed through the flow cytometer, each one of these cells provides a vector of scalar values $x \in \mathbb{R}^n$ corresponding to each
measurable variable. In this case we set $n = 2$ as we are only utilizing the scalar values that represent the measure of both CD4 mRNA and IFN-γ mRNA, samples of these distributions are found in Figure 37. From this we obtain a 2D scatter field. We have restricted ourselves to two dimensions for illustrative purposes, but this same analysis can be conducted for a point cloud of higher dimensions, accounting for all variables measured by a flow cytometer. The goal in this analysis is to use TDA to quantify the temporal evolution of the shape of the scatter field during the kinetic response of human blood mononuclear cells during stimulation. This approach provides an alternative to traditional parametric methods such as gating (which are difficult to tune).

![Deformation of a 2D scatter field over time.](image)

Figure 37: Deformation of a 2D scatter field over time.

![Processing of the flow cytometry scatter field.](image)

Figure 38: Processing of the flow cytometry scatter field. The raw data is first smoothed via a Gaussian kernel and then the smoothed diagram is processed via a Morse filtration.

In order to analyze the topology of the scatter fields, we utilize Gaussian kernel smoothing [56]. The work of [8] demonstrates that provided a large enough sample, the homology of the Gaussian kernel density estimate derived from a sample is equivalent to the homology of the true density. Figure 38 shows the Gaussian kernel smoothing of a flow cytometry scatter plot. An example persistence diagram for the function is shown in Figure 38. As in the case of the diffusion example, in 3D we can represent the scatter field as a continuous function (in this case a probability density function). This probability density function (Figure 39a) can be analyzed using Morse filtration. Our goal in this analysis is to quantify the time evolution of the probability density functions during stimulation. Our strategy consists of computing the Wasserstein distance ($w_d$) between the persistence diagram of a given time point to the persistence diagram of the sample at time zero [15]. Specifically, given flow cytometry samples $X_1, X_2$ and the time zero sample $X_0$ as well as their corresponding persistence diagrams $PD_1, PD_2, PD_0$ and time points $T_0 \leq T_1 \leq T_2$, we observe that:

$$w_d(PD_1, PD_0) \leq w_d(PD_2, PD_0)$$ (5.31)

From Figure 39b, we can see that the distance exhibits a strong dependence on time. This suggests
that there exists a continuous mapping between the persistent diagram and time (the topological deformation is continuous with respect to time). This again reveals the continuity of the persistence diagrams (of topology) to perturbations. These results highlight how topological data analysis provides a quantifiable approach to characterize complex probability density functions and their evolution over time.

5.6 Topology of 3D Fields

We now illustrate how to use TDA to analyze the topology induced by 3D point clouds. Specifically, we study datasets generated by molecular dynamics (MD) simulations [18, 62]. The dataset under study analyzes the influence of the 3D liquid-phase environment formed by molecules of a solvent, co-solvent, and a reactant on reactivity [18]. The reactivity is quantified via a kinetic solvent parameter $\sigma$ that is obtained from experiments. Experiments suggest that reactivity is influenced by hydrophilicity of the solvent. The main hypothesis is that, as the solvent concentration is increased, the water in the system is concentrated around the solvent molecule, and that molecules with high hydrophilicity are able to take advantage of this effect. In order to study this hypothesis, molecular dynamics computations were performed in [18]. The data output of an MD simulation has both spatial and temporal dimensions. Each simulation gives atomic positions $X_t \in \mathbb{R}^{M \times 3}$ ($M$ is the number of species) at multiple times $t$ (measured in nanoseconds). In our analysis, we utilize a 3D point cloud of water molecule positions that result from a time average of 100 nanoseconds. An example of this point cloud (visualized as a field) is provided in Figure 40. Each density field is labeled with a reactivity $\sigma$ obtained from experiments. Recent work by Chew and co-workers has analyzed the 3D point cloud by using 3D convolutional networks and has shown that the features extracted from the CNN are strongly correlated to reactivity [18]. CNNs are highly effective tools but require a large number of parameters and are difficult to interpret. Our goal is to study if the topology of the 3D point cloud can be characterized in a more straightforward manner and to explore whether such
topology changes as the concentration of the solvent changes.

![Visualization of 3D water density field generated by MD simulation.](http://zavalab.engr.wisc.edu)

> Figure 40: Visualization of 3D water density field generated by MD simulation.

![Slices of 3D water density field as filtration proceeds for different density values.](http://zavalab.engr.wisc.edu)

> Figure 41: Slices of 3D water density field as filtration proceeds for different density values. The filtration reveals the presence of voids in the data associated with high concentrations of water molecules.

To perform our analysis, we treat the 3D point cloud as a continuous field (function). Here, we perform Morse filtration and treat the data as a 3D cubical complex (a voxel). The filtration will be done by exploring levels sets for the water density. Note that this is a filtration in a higher dimension than our previous examples for the diffusion field and for liquid crystal sensors. The main focus of this approach is to capture the clustering of water near hydrophilic molecules, and the lack of clustering near non-hydrophilic molecules. We visualize the water density filtration in Figure 41 via a 2-dimensional slice. We can see that voids in the data are generated as we increase the filtration value. The voids represent areas of high water density which is precisely what we wish to quantify. From these filtrations, we produce persistence diagrams focused on $H_3$ (since this homology group quantifies these voids). The persistent diagrams are then vectorized we use PCA reduction to visualize them (see Figure 42a). We use SVM regression with a radial basis kernel to predict reactivity as a function of the persistence diagrams (see Figure 42b).

The PCA projection reveals that there is strong dependence the reactivity on the topology. This suggests that the information gained via persistent homology extracts informative features of the
Figure 42: (a) PCA analysis on persistence diagrams for MD simulations. The analysis reveals strong dependence of reactivity $\sigma$. (b) Regression plot for SVM with a radial basis function. The predictions over 5-fold cross validation yield an MSE of $0.07 \pm 0.003$.

3D field that explain reactivity. This also suggests that a simple regression method (as opposed to a complex neural net) would be effective at predicting the reactivity. In order to test this hypothesis, the experimental dataset with 70 points is split into train/test sets with 49/21, points to build the SVM regression model. A 5-fold cross validation is performed to estimate the performance of the SVM regression. We found that this model captures the trend of reactivity well; moreover, this simple model yields a mean square error of $0.07 \pm 0.002$. These results are relevant because they indicate that it is possible to predict experimental reactivity directly from MD simulations.

6 Conclusions

Topological data analysis (TDA) provides a set of powerful methods and tools for understanding the underlying geometry of data. These techniques represent data such as point clouds and functions as geometric objects and explores these objects in terms of basic geometrical features. We have shown that TDA offers a number of important theoretical properties (such as stability), offers flexibility to extract features from different types of dataset, and extracted features can be exploited using statistical and machine learning techniques. The TDA field is in rapid development from both a theoretical and applied perspective. Moreover, there are easily accessible software packages to conduct scalable computations (such as such as Gudhi [39], Homcloud [47], Javaplex-Matlab [2] and R-TDA [25]). For further reading on the mathematical foundations of these techniques, the reader is referred to [30, 23, 67]. We believe that TDA can bring new fundamental understanding to challenging problems in chemical engineering.

Acknowledgements

We acknowledge funding from the U.S. National Science Foundation (NSF) under BIGDATA grant IIS-1837812 and also acknowledge partial support from the NSF through the University of Wisconsin Materials Research Science and Engineering Center (DMR-1720415). We thank Alex Chew and Prof. Reid Van Lehn for providing datasets on MD simulations and thank Prof. Nicholas Abbott for providing the liquid crystal dataset.
References

[1] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *The Journal of Machine Learning Research*, 18(1):218–252, 2017.

[2] Henry Adams, Andrew Tausz, and Mikael Vejdemo-Johansson. Javaplex: A research software package for persistent (co) homology. In *International Congress on Mathematical Software*, pages 129–136. Springer, 2014.

[3] Paul Alexandroff. Über den allgemeinen dimensionsbegriff und seine beziehungen zur elementaren geometrischen anschauung. *Mathematische Annalen*, 98(1):617–635, 1928.

[4] Madjid Allili, Konstantin Mischaikow, and Allen Tannenbaum. Cubical homology and the topological classification of 2d and 3d imagery. In *Proceedings 2001 international conference on image processing (Cat. No. 01CH37205)*, volume 2, pages 173–176. IEEE, 2001.

[5] Francis J Anscombe. Graphs in statistical analysis. *The american statistician*, 27(1):17–21, 1973.

[6] Ulrich Bauer and Michael Lesnick. Induced matchings of barcodes and the algebraic stability of persistence. In *Proceedings of the thirtieth annual symposium on Computational geometry*, pages 355–364, 2014.

[7] Andrew J Blumberg, Itamar Gal, Michael A Mandell, and Matthew Pancia. Robust statistics, hypothesis testing, and confidence intervals for persistent homology on metric measure spaces. *Foundations of Computational Mathematics*, 14(4):745–789, 2014.

[8] Omer Bobrowski, Sayan Mukherjee, Jonathan E Taylor, et al. Topological consistency via kernel estimation. *Bernoulli*, 23(1):288–328, 2017.

[9] Peter Bubenik. Statistical topology using persistence landscapes. *arXiv preprint arXiv:1207.6437*, 3, 2012.

[10] Peter Bubenik, Gunnar Carlsson, Peter T Kim, and Zhi-Ming Luo. Statistical topology via morse theory persistence and nonparametric estimation. *Algebraic methods in statistics and probability II*, 516:75–92, 2010.

[11] Peter Bubenik and Paweł Dłotko. A persistence landscapes toolbox for topological statistics. *Journal of Symbolic Computation*, 78:91–114, 2017.

[12] Mickaël Buchet, Yasuaki Hiraoka, and Ippei Obayashi. Persistent homology and materials informatics. In *Nanoinformatics*, pages 75–95. Springer, Singapore, 2018.

[13] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[14] Gunnar Carlsson, Afra Zomorodian, Anne Collins, and Leonidas J Guibas. Persistence barcodes for shapes. *International Journal of Shape Modeling*, 11(02):149–187, 2005.
[15] Mathieu Carriere, Marco Cuturi, and Steve Oudot. Sliced wasserstein kernel for persistence diagrams. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 664–673. JMLR. org, 2017.

[16] Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J Guibas, and Steve Y Oudot. Proximity of persistence modules and their diagrams. In Proceedings of the twenty-fifth annual symposium on Computational geometry, pages 237–246, 2009.

[17] Frédéric Chazal, Vin De Silva, Marc Glisse, and Steve Oudot. The structure and stability of persistence modules. arXiv preprint arXiv:1207.3674, 21, 2012.

[18] Alex K Chew, Shengli Jiang, Weiqi Zhang, Victor M Zavala, and Reid C Van Lehn. Fast predictions of liquid-phase acid-catalyzed reaction rates using molecular dynamics simulations and convolutional neural networks. 2019.

[19] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. Discrete & Computational Geometry, 37(1):103–120, 2007.

[20] Ronald R Coifman and Stéphane Lafon. Diffusion maps. Applied and computational harmonic analysis, 21(1):5–30, 2006.

[21] Robert D Cook et al. Concepts and applications of finite element analysis. John wiley & sons, 2007.

[22] Tamal K Dey, Anil N Hirani, and Bala Krishnamoorthy. Optimal homologous cycles, total unimodularity, and linear programming. SIAM Journal on Computing, 40(4):1026–1044, 2011.

[23] Herbert Edelsbrunner and John Harer. Computational topology: an introduction. American Mathematical Soc., 2010.

[24] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In Proceedings 41st annual symposium on foundations of computer science, pages 454–463. IEEE, 2000.

[25] Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, and Clément Maria. Introduction to the r package tda. arXiv preprint arXiv:1411.1830, 2014.

[26] Robert Ghrist. Barcodes: the persistent topology of data. Bulletin of the American Mathematical Society, 45(1):61–75, 2008.

[27] Robert W Ghrist. Elementary applied topology, volume 1. Createspace Seattle, 2014.

[28] Marian Gidea and Yuri Katz. Topological data analysis of financial time series: Landscapes of crashes. Physica A: Statistical Mechanics and its Applications, 491:820–834, 2018.

[29] David Günther, Jan Reininghaus, Hubert Wagner, and Ingrid Hotz. Efficient computation of 3d morse–smale complexes and persistent homology using discrete morse theory. The Visual Computer, 28(10):959–969, 2012.

[30] Allen Hatcher. Algebraic topology. 2005.
[31] Takashi Ichinomiya, Ippei Obayashi, and Yasuaki Hiraoka. Persistent homology analysis of craze formation. Physical Review E, 95(1):012504, 2017.

[32] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065):20150202, 2016.

[33] Daniel M Kan. Abstract homotopy. Proceedings of the National Academy of Sciences of the United States of America, 41(12):1092, 1955.

[34] Peter M Kasson, Afra Zomorodian, Sanghyun Park, Nina Singhal, Leonidas J Guibas, and Vijay S Pande. Persistent voids: a new structural metric for membrane fusion. Bioinformatics, 23(14):1753–1759, 2007.

[35] Firas A Khasawneh, Elizabeth Munch, and Jose A Perea. Chatter classification in turning using machine learning and topological data analysis. IFAC-PapersOnLine, 51(14):195–200, 2018.

[36] Miroslav Kramár, Rachel Levanger, Jeffrey Tithof, Balachandra Suri, Mu Xu, Mark Paul, Michael F Schatz, and Konstantin Mischaikow. Analysis of kolmogorov flow and rayleigh–bénard convection using persistent homology. Physica D: Nonlinear Phenomena, 334:82–98, 2016.

[37] Hyekyoung Lee, Moo K Chung, Hyejin Kang, Bung-Nyun Kim, and Dong Soo Lee. Discriminative persistent homology of brain networks. In 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pages 841–844. IEEE, 2011.

[38] Hyekyoung Lee, Hyejin Kang, Moo K Chung, Bung-Nyun Kim, and Dong Soo Lee. Persistent brain network homology from the perspective of dendrogram. IEEE transactions on medical imaging, 31(12):2267–2277, 2012.

[39] Clément Maria, Jean-Daniel Boissonnat, Marc Glisse, and Mariette Yvinec. The gudhi library: Simplicial complexes and persistent homology. In International Congress on Mathematical Software, pages 167–174. Springer, 2014.

[40] Justin Matejka and George Fitzmaurice. Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pages 1290–1294, 2017.

[41] John Willard Milnor, Michael Spivak, and Robert Wells. Morse theory, volume 1. Princeton university press Princeton, 1969.

[42] Konstantin Mischaikow and Vidit Nanda. Morse theory for filtrations and efficient computation of persistent homology. Discrete & Computational Geometry, 50(2):330–353, 2013.

[43] James R Munkres. Elements of algebraic topology. CRC Press, 2018.

[44] Takenobu Nakamura, Yasuaki Hiraoka, Akihiko Hirata, Emerson G Escolar, and Yasumasa Nishiura. Persistent homology and many-body atomic structure for medium-range order in the glass. Nanotechnology, 26(30):304001, 2015.
[45] Marc Niethammer, Andrew N Stein, William D Kalies, Pawel Pilarczyk, Konstantin Mischaikow, and Allen Tannenbaum. Analysis of blood vessel topology by cubical homology. In Proceedings. International Conference on Image Processing, volume 2, pages II–II. IEEE, 2002.

[46] Ippei Obayashi. Volume-optimal cycle: Tightest representative cycle of a generator in persistent homology. SIAM Journal on Applied Algebra and Geometry, 2(4):508–534, 2018.

[47] Ippei Obayashi. Homcloud, 2020.

[48] Travis E Oliphant. Python for scientific computing. Computing in Science & Engineering, 9(3):10–20, 2007.

[49] Jose A Perea. Topological time series analysis. Notices of the American Mathematical Society, 66(5), 2019.

[50] Jose A Perea, Anastasia Deckard, Steve B Haase, and John Harer. Sw1pers: Sliding windows 1-persistence scoring; discovering periodicity in gene expression time series data. BMC bioinformatics, 16(1):257, 2015.

[51] Jose A Perea and John Harer. Sliding windows and persistence: An application of topological methods to signal analysis. Foundations of Computational Mathematics, 15(3):799–838, 2015.

[52] Henri Poincaré. Analysis situs. Gauthier-Villars, 1895.

[53] Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4741–4748, 2015.

[54] Lee M Seversky, Shelby Davis, and Matthew Berger. On time-series topological data analysis: New data and opportunities. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 59–67, 2016.

[55] Rahul R Shah and Nicholas L Abbott. Principles for measurement of chemical exposure based on recognition-driven anchoring transitions in liquid crystals. Science, 293(5533):1296–1299, 2001.

[56] Simon J Sheather. Density estimation. Statistical science, pages 588–597, 2004.

[57] Alexander Smith, Nicholas L Abbott, and Victor M Zavala. Convolutional network analysis of optical micrographs. 2020.

[58] J Spidlen, K Breuer, C Rosenberg, N Kotecha, and RR Brinkman. A resource of annotated flow cytometry datasets associated with peer-reviewed publications. Cytometry Part A, 81(9):727–731, 2012.

[59] Bernadette J Stolz, Heather A Harrington, and Mason A Porter. Persistent homology of time-dependent functional networks constructed from coupled time series. Chaos: An Interdisciplinary Journal of Nonlinear Science, 27(4):047410, 2017.

[60] Yuhei Umeda. Time series classification via topological data analysis. Information and Media Technologies, 12:228–239, 2017.
[61] Dennis Van Hoof, Woodrow Lomas, Mary Beth Hanley, and Emily Park. Simultaneous flow cytometric analysis of ifn-γ and cd4 mrna and protein expression kinetics in human peripheral blood mononuclear cells during activation. *Cytometry Part A*, 85(10):894–900, 2014.

[62] Theodore W Walker, Alex K Chew, Huixiang Li, Benginur Demir, Z Conrad Zhang, George W Huber, Reid C Van Lehn, and James A Dumesic. Universal kinetic solvent effects in acid-catalyzed reactions of biomass-derived oxygenates. *Energy & Environmental Science*, 11(3):617–628, 2018.

[63] Eric Walter and Luc Pronzato. Identification of parametric models. *Communications and control engineering*, 8, 1997.

[64] Yuan Wang, Hernando Ombao, and Moo K Chung. Topological data analysis of single-trial electroencephalographic signals. *The annals of applied statistics*, 12(3):1506, 2018.

[65] Kelin Xia, Xin Feng, Yiying Tong, and Guo Wei Wei. Persistent homology for the quantitative prediction of fullerene stability. *Journal of computational chemistry*, 36(6):408–422, 2015.

[66] Kelin Xia and Guo-Wei Wei. Persistent homology analysis of protein structure, flexibility, and folding. *International journal for numerical methods in biomedical engineering*, 30(8):814–844, 2014.

[67] Afra Zomorodian. Topological data analysis. *Advances in applied and computational topology*, 70:1–39, 2012.

[68] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.