Topological methods for data modelling

Gunnar Carlsson

Abstract | The analysis of large and complex data sets is one of the most important problems facing the scientific community, and physics in particular. One response to this challenge has been the development of topological data analysis (TDA), which models data by graphs or networks rather than by linear algebraic (matrix) methods or cluster analysis. TDA represents the shape of the data (suitably defined) in a combinatorial fashion. Methods for measuring shape have been developed within mathematics, providing a toolkit referred to as homology. In working with data, one can use this kind of modelling to obtain an understanding of the overall structure of the data set. There is a suite of methods for constructing vector representations of various kinds of unstructured data. In this Review, we sketch the basics of TDA and provide examples where this kind of analysis has been carried out.

Features
In any data set, the features are the various numerical quantities attached to data points. In a data matrix, they are the columns of the matrix, and the rows are the data points.

How to analyse large and complex data sets has become one of the most difficult problems facing the scientific community. There are many reasons this is so, but two reasons stand out. First, if we compare present-day research with the scientific work done by Galileo Galilei and Johannes Kepler, we can see that the data they used were small and simple, in that they consisted of a small number of observations. Furthermore, each observation consisted of a small set of numbers (that is, features), such as the 3D coordinates of an object. As a result, one could hope to analyse the data 'by hand'. By contrast, the size of the data scientists currently attempt to study has grown tremendously, both in terms of the number of observations and the number of features — consider, for example, X-ray imaging data[1] or particle physics data from synchrotrons[2] — and the techniques required to understand it go far beyond what was available to Galileo and Kepler. In short, the set of rows and the set of columns of the data matrices are very large.

The size of data creates computational problems for storage as well as computation. However, the second stand-out reason that data analysis is now so challenging is a purely data analytic issue. This issue is the need for unsupervised analysis, which is made much more difficult by the presence of very large numbers of features. In the past, scientists often had particular models or families of models in mind, and the data analysis was used to verify the hypotheses. For many of the problems that are now arising, we do not have a model in mind. Indeed, the goal of the analysis is to discover models from among a very large set of variables (that is, columns of the data matrix). In summary, the size of the data matrix creates substantial problems for all types of analysis.

There are, however, important challenges beyond the size of data sets. One of these is the complexity of the data. In fact, even small data sets can present problems of interpretation. The complexity comes in at least two distinct forms.

The first type of complexity is what might be called format complexity, in that the way the data is organized or presented creates difficulties. Consider, for example, a data set consisting of complex molecules. Each molecule is represented as a collection of atoms with atomic weights, and a collection of bonds and lengths of those bonds. If we order the atoms, then such a data set might be stored as an ordered list of atomic weights, followed by a list of triples describing the bonds. The triple would include a pair of integers, namely the indices of the atoms comprising the bond, and a real number, which is the length of the bond. For example, a water molecule might be encoded as $(w_H, w_H, w_O, (1, 3, l), (2, 3, l))$, where w_H and w_O represent the atomic weights of hydrogen and oxygen, respectively, l represents the length of the oxygen–hydrogen bond in water, and the vector $(1, 3, l)$ represents the bond from the first atom (hydrogen) to the third atom (oxygen). It is possible to represent any molecule in this way, but a collection of such lists is difficult to analyse because there are many representations of the same molecule, due to the ordering of the list. This multiplicity of representations makes it difficult to construct appropriate features or coordinates for the data. Similar problems arise in the study of text data, in which the data in its native form is a list of documents, each of which is a list of words. Words are not typically of the same length, and thus feature generation can also be a challenge.

The second kind of complexity is structural complexity, and applies even to data sets consisting of data matrices with numerical entries. Consider, for example, a data set of major league baseball players, in which numerous features concerning hitting are tracked, such as batting average, runs batted in, home runs and walks.

Department of Mathematics, Stanford University, Stanford, CA, USA.

e-mail: carlsson@stanford.edu

https://doi.org/10.1038/s42254-020-00249-3

Key points

- The analysis of large and complex data sets is crucial to all areas of science and industry, and is needed to support artificial intelligence. Existing methods for data analysis are often inadequate to deal with data that exhibit a great deal of complexity, because they are unable to express complicated 'data shapes'.
- Topology (the mathematical study of shape) has been extended to topological data analysis to give systematic graph representations of data sets, which are informative in many different ways. Graphs can be thought of as encoding shape.
- Graph representations of data permit systematic unsupervised analysis of data, with a variety of methods for the interrogation of the data. They constitute a compression of the data that nevertheless preserves salient features.
- Because of the flexibility of graph representations, methods for measuring the corresponding shape are required. Homology is a family of such methods. It is useful both for overall understanding of data sets and for generation of numerical features for many kinds of unstructured data.
- Topological data analysis has been applied in many different complex data situations.

In looking at this data, there are certain key properties that are evident, for example, that there are different types of player with different specialties — pitchers tend not to be good hitters. There are also different styles of play, which makes the analysis of the data set of types of player difficult. For instance, some batters are singles hitters with high averages and others are power hitters with low averages. These observations lead to the decomposition of the set of players into groups. Some of these groups might overlap, meaning that the data should not be thought of as a disjoint union; the decomposition is a so-called soft clustering decomposition. Data exhibiting this kind of complexity are often not conveniently modelled by algebraic methods, such as principal component analysis or linear regression. Because of the fact that the groups may overlap, it is also often not well modelled by clustering methods such as k-means clustering or hierarchical clustering[3]. Such problems suggest the need for a different organizing principle for large and complex data, and particularly for a method that extracts information from soft clustering decompositions into geometric information. Geometric information is extremely useful because it can encode both discrete information, such as a decomposition of a data set into discrete groups, and continuous variation within the data. The organizing principle we will discuss is topological modelling, which refers to the construction of graphs and simplicial complexes (defined below) to represent data sets and similarity relations on them. Topology is a highly developed subdiscipline within mathematics, which concerns itself with the study of shape.

The field of topology studies objects generically referred to as 'spaces'. For the purpose of this Review, a space is simply a subset of n-dimensional Euclidean space $\mathbb{R}^n$. Such a subset is simply a set of vectors of length n, and a large fraction of data analytic problems deal with such sets. Sets that are not given directly in this form can often be transformed into sets of vectors. The term 'space' is used to evoke the geometry coming from the natural extension of Euclidean distances in dimensions two and three to higher dimensions. The subject has two main threads — geometric topology and algebraic topology. Geometric topology studies geometric objects (possibly in high dimensions) directly, in particular dealing with theorems that assert that particular spaces can be represented usefully in combinatorial ways. Algebraic topology develops tools for 'measuring' shape, by performing sophisticated counting procedures that count the number of occurrences of geometric features of a certain kind. For example, one such feature would be the number of connected components, or the number of independent loops within a space. What has been initiated in the past 15–20 years is the extension of both these threads within topology to point clouds, that is, finite data sets. For purposes of this Review, this will just mean finite, but possibly large, subsets of $\mathbb{R}^n$. Point clouds inherit the distance function from $\mathbb{R}^n$, which can be thought of as a dissimilarity measure. We will occasionally allow ourselves more general notions of distance functions and spaces, called metrics and metric spaces[4]. These notions provide an abstraction of the notion of distance from Euclidean spaces, which will be useful in studying unstructured data such as molecules. Thus, all spaces are metric spaces and point clouds are finite metric spaces.

This Review aims to give the reader a high-level view of both the geometric and algebraic aspects of topology, particularly as adapted to the study of point clouds, and to do so in a way that minimizes the mathematical prerequisites required. A number of explicit applications are shown, to give an indication of the possibilities for the methods. A brief guide to available software for topological data analysis (TDA) is also included.

The main serious mathematical prerequisite is matrix algebra, with the wrinkle that the algebra uses modular arithmetic with modulus 2 (REF.[5]). That is, the entries of the matrices are in the set {0, 1}, where addition is 'exclusive or' and multiplication is 'and'. The properties of matrix algebra we use are all analogues of properties that hold for ordinary matrix algebra with real numbers. It will also useful to know the definitions of graphs, in the computer science or combinatorics sense.

To go beyond the present discussion, please see the more detailed resources in REFS[6,7].

Topological modelling

To give an idea of what is meant by the shape of a data set, suppose that we have a collection of points in the plane such as that in FIG. 1. It is a finite set of points, but to our eyes it has a shape, namely that of a loop. (It is not a perfectly round loop, but the set of points nevertheless exhibits loopy behaviour.) As a space, it is a discrete set of points, but the geometric distribution of the points produces the loopy structure. This kind of behaviour can occur in a number of situations, one being data that is time dependent and that exhibits periodic or recurrent behaviour. Another situation is data coming from the statistics of natural images[8]. In this case, the circular structure comes from a parametrization of edge patches by an angular coordinate. We would like to obtain models of the data set that reflect this loopy behaviour, and to be able to detect this aspect of shape in an automatic fashion.

Shape is a somewhat nebulous concept, which appears to be difficult to formalize and measure. By contrast,

Clustering decomposition
Any method that decomposes a data set into disjoint groups, called clusters.

Space
A set equipped with a notion of nearness. For any positive integer, subsets of $\mathbb{R}^n$ are examples, and so are metric spaces.

Connected components
The decomposition of a space into disjoint pieces that are separated from each other, and which cannot be so decomposed further.

Metric spaces
An abstraction of the notion of distance in the plane. A metric space consists of a set X and a non-negative valued distance function d on pairs of points in X, satisfying certain conditions, such as symmetry and the triangle inequality $d(x, z) \leq d(x, y) + d(y, z)$.

graphs are simple combinatorial objects, given as a pair (V, E), where V is a set of points or vertices and E is a set of edges, that is, a set of unordered pairs of vertices. Graphs may be embedded in the plane for visualization (FIG. 2), but as a mathematical object they are simply the list of vertices and edges. FIGURE 2a depicts a graph in the shape of a triangle. Note that the graph is only the boundary of the triangle and not the interior. The graph is given by a list of three vertices and three edges, namely a,b,c,{a,b},{a,c},{b,c}. Graphs are by their nature 1D objects, in the sense that they consist only of vertices and edges, where edges are simplices with only two vertices. One can extend the notion of graphs to that of simplicial complexes, where one permits simplices that are higher-dimensional edges, or faces, which correspond to unordered n-tuples of vertices for $n > 2$. Adjoining higher-dimensional faces is critical when the space is more than 1D, such as a sphere. A spherical data set could be given by the positions of a system of sensors distributed around the Earth. See REF.[4] or REF.[9] for a thorough discussion of simplicial complexes and related constructions. FIGURE 2b depicts such a simplicial complex. It is encoded by a list of vertices, edges and faces: e,f,g,h,{e,f},{e,g},{e,h}, {f,h}, {g,h},{e,g,h}. The inclusion of the set {e,g,h} corresponds to the presence of the interior of the triangle with vertices e, g and h. This shape is 2D. The inclusion of sets of vertices of cardinality four allows one to include tetrahedra, and higher cardinalities correspond to higher-dimensional versions of these shapes referred to as simplices. We formalize the notion of a simplicial complex as being given by a pair (V, Σ), where V is a finite set of vertices and Σ is a collection of subsets of V, so that if a subset $S \subseteq V$ is an element of Σ and $T \subseteq S$, then T is also an element of Σ. This condition holds for the two lists given above, and for 2D simplices (triangles), it means that if a triangle is included in the list then so are all of its edges.

Graphs and simplicial complexes can be viewed as spaces via a process called geometric realization. For instance, it is possible to reconstruct the shape of the graph in FIG. 2a directly from the list of vertices and edges, and this reconstruction is called geometric realization. Note that although the triangle in the figure lies on a 2D plane, geometric realization is not the same as graph drawing algorithms that seek to create graph layouts in 2D. Instead, geometric realization produces a simplicial complex in $\mathbb{R}^N$, where N is the number of vertices, which is generally a very high dimension. However, by moving the vertices in $\mathbb{R}^N$, it is typically possible to obtain a set in a much lower-dimensional space. For example, if the complex has simplices of dimension at most n, that is, all simplices have n or fewer vertices, then the vertices of the realization can be moved so that the complex lies in $\mathbb{R}^{2n+1}$. This means that the points of the realization can be encoded as vectors of length $2n+1$. The list of vertices and edges does not determine the exact details of the shape, including the lengths of the edges, but the list does determine the connectivity properties. Connectivity is an informal notion that refers to closeness in the limit of very small scales, without regard for the distances at any particular scale.

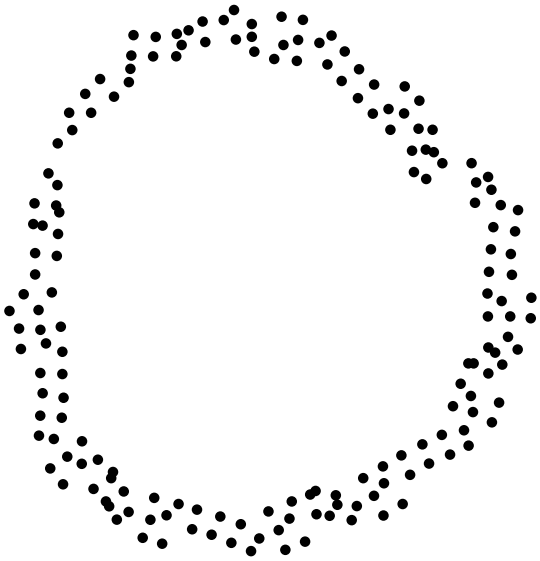

Fig. 1 | **Statistical circle.** The goal of topological modelling is to detect the loopy behaviour of the data points in an automatic and robust fashion.

For example, a triangle might be equilateral, isosceles or scalene, but it is still a triangle. Similarly, a set of two discrete points is so whether the points are a millimetre or a kilometre apart.

Geometric realization is a method for producing geometric objects from combinatorial ones. The goal of modelling is to go the other way, that is, to produce simplicial complexes from geometric objects. Topological modelling refers to any modelling procedure that produces a simplicial complex, and presumably represents the shape of the data. Should it be the case that the data are thought of as sampled (perhaps with noise) from a space X, one would like the complex constructed to be somehow comparable to X. The simplest such construction is the Vietoris–Rips complex[10,11], which takes as input a point cloud M and a threshold R, and outputs a simplicial complex $V(X; R)$, the set of vertices of which is the set of points of M, and for which the elements of $\Sigma(X; R)$ are the subsets $\{m_0, m_1, \ldots, m_k\}$ for which the distance $d(m_i, m_j) \leq R$ for all pairs (i, j) — that is, points within the distance threshold are connected to each other. In the case of the vertices of the unit square, the Vietoris–Rips complex at various scales R looks like the depiction in FIG. 3. The right-most complex ($\sqrt{2} \leq R$) is a full tetrahedron. Note that any of these complexes also include all the complexes to its left. The complexes increase as R does.

The Vietoris–Rips complex is perhaps the simplest construction that can be used to assign shapes to point clouds. Although it is simple conceptually, it is typically difficult to compute with due to the very large number of vertices and simplices, and its approximation properties are not well developed, owing to the fact that it does not arise from a nerve construction. The nerve construction is a general and useful construction for simplicial complexes and is the key tool for topological modelling. Given a set X and a collection $\mathcal{U} = \{U_0, U_1, \ldots, U_n\}$ of non-empty subsets of X, we form a simplicial complex

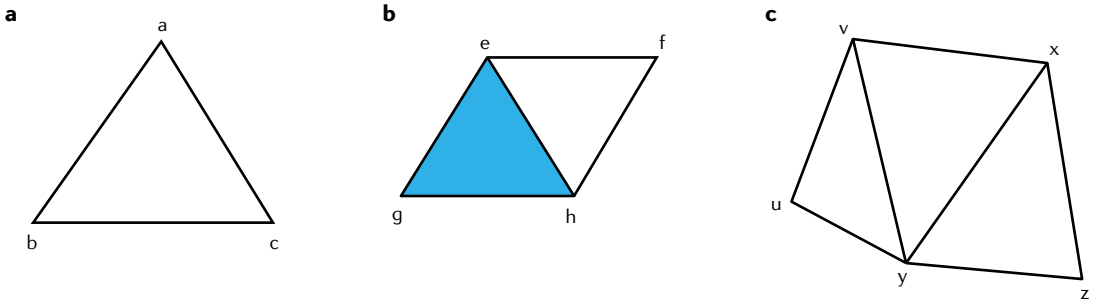

Fig. 2 | **Geometric realization. a** | A depiction of a graph consisting of vertices a, b and c, and edges {a,b}, {a,c} and {b,c}. **b** | A depiction of a simplicial complex consisting of vertices e, f, g and h, two-simplex (face) {e,g,h} (shaded) and one-simplices (edges) {e,f}, {e,g}, {e,h} and {f,h}. **c** | A graph that contains independent cycles uvy, vyx and xyz.

$N(X;\mathcal{U}) = (V_N, \Sigma_N)$ the vertices V_N of which are the set $\underline{n} = \{0, 1, \ldots, n\}$, and where a subset $\{i_0, i_1, \ldots, i_s\}$ of $\underline{n}$ is in Σ_N if and only if

$$U_{i_0} \bigcap U_{i_1} \bigcap \ldots \bigcap U_{i_s} \neq \varnothing$$

For example, consider a space that is a fattened boundary of a square, covered by four rectangular sets that intersect at the corners of the square (FIG. 4). The corresponding nerve complex has one vertex for each of the four sets, with connections between vertices drawn if and only if the corresponding sets have a non-empty intersection. Note that the intersections on the left part of FIG. 4 correspond to the edges on the right part of FIG. 4. When the set X is a space and the sets U_i are in an appropriate sense small, the nerve complex of the covering $\mathcal{U}$ often approximates the original space X well. The precise statement of an approximation theorem in this direction is called the nerve lemma[9].

There are a number of other constructions that have better computational and theoretical properties than the Vietoris–Rips complex, and which apply in different situations. Three examples, each of which is given as a nerve construction on a covering, are α-shapes[12,13] the witness complex[14] and Mapper[15]. The first two are based on analogues of the Voronoi construction[16], whereas Mapper is based on Reeb graphs[17]. Mapper is a construction based on a metric space X equipped with one or more projections from X to the real line. Each line is covered by a family of overlapping intervals, of identical length and identical degree of overlap. The inverse images of the intervals, or products of intervals, are each clustered using single linkage clustering, and these clusters create a covering of X, in which the sets can overlap because the intervals overlap. The nerve of the covering is the Mapper construction based on the chosen covering. This particular construction has been shown to be extremely useful for the direct exploratory study of complex data sets.

Covering
A covering of a set X is a collection of subsets of X whose union is all of X. The sets need not be disjoint.

Homology
An invariant that counts occurrences of geometric patterns, such as loops, in a space.

Measuring shape with homology

The notion of shape is inherently a fairly qualitative one, which appears not to be amenable to precise quantitative analysis. In fact, there is a precise method that can be used to distinguish between various different kinds of shape. It is described in detail in REFS[4,9]. The idea is to proceed the way humans do in distinguishing different shapes. Consider the shapes given by the numbers 0 and 8. Humans readily recognize the distinction between them by counting the number of loops present in the figure. The zero has a single loop and the eight has two loops. This is a very robust way of recognizing the digits, which does not change under many deformations of the figures, such as shears, rotations, warping or bending, or under a change of viewing angle. It is, however, difficult to imagine a computational way of computing the number of loops. It turns out, though, that there is a method known as homology that performs exactly this task, and we sketch it here. Consider the complex in FIG. 2a. It is apparent that the complex has a single loop, and consists of a single connected component. These are visual observations, but we claim that this information is contained in a single matrix, called the boundary matrix. The boundary matrix has its rows labelled by the vertices of the graph, and the columns are labelled by the edges. The boundary matrix for the graph in FIG. 2a is

	ab	ac	bc
a	1	1	0
b	1	0	1
c	0	1	1

The entries of the matrix are Boolean integers $\mathcal{B} = \{0, 1\}$, for which addition is given by exclusive or (that is: $1+1=0$, $1+0=1$, $0+1=1$, $0+0=0$) and multiplication is given by and. This algebraic structure is also known as modular arithmetic with modulus 2, and its properties are described in REF.[5]. Matrices with entries that are in $\mathcal{B}$ can be manipulated in ways identical to matrices with real entries. In particular, the rank of a matrix with entries in $\mathcal{B}$ makes sense in the same way as it does for matrices with real entries. The above boundary matrix above has rank equal to one, as the rows $a+b=c$. It therefore follows that the null space (which also has a meaning within modular matrix algebra) has dimension one, and that a basis is given by the vector represented by (1, 1, 1). Given the labelling of the columns, it can be formally thought of as $ab+ac+bc$. If we permit ourselves to think of addition as union, then this element corresponds to the union of all the edges, which is a loop in X. One can show that in any graph, thought of as a 1D simplicial complex, there is a corresponding boundary matrix, and the dimension of its null space is the number of 'independent cycles'. In other words, the

Simplex
A subset of $\mathbb{R}^n$ that is the convex hull of k points, where $k \leq n+1$. For $k = 2$, 3 and 4, simplices are intervals, triangles and tetrahedra, respectively.

Homotopy
For maps f and g between spaces X and Y, $f, g : X \to Y$, f and g are homotopic if there is a continuous one-parameter family of maps beginning with f and ending at g.

boundary matrix of a graph can be used to identify the number of loops in the graph. To understand this notion, consider the graph in FIG. 2c. The cycles uvy, vyx and xyz form an independent set of cycles. The cycle uvxy is thought of as dependent on the two cycles uvy and vyx because it is a union of them, with the edge vy occurring twice, and therefore treated as zero as $1 + 1 = 0$ in $\mathcal{B}$. In the null space of the boundary matrix of this graph, the null space has dimension 3, coinciding with the maximal number of independent cycles.

There is also a construction dual to the null space construction, which assigns to an $m \times n$ matrix A with entries in $\mathcal{B}$ the quotient of the $\mathcal{B}$-vector space $\mathcal{B}^m$ by the column space of A. It can be computed using column operations in the same way that the null space is computed using row operations. Moreover, the dimension of the quotient is $m - \text{rank}(A)$. The dimension of the quotient space of the column space of the boundary matrix can be interpreted as the number of connected components of the complex. In summary, two interesting geometric invariants of a complex may be identified using quantities representing algebraic properties of the boundary matrix.

It is natural to ask whether there are similar algebraic properties involving other matrices that represent higher-dimensional geometric properties of a complex X. It turns out that there are, and they are obtained as follows.

- For each $k \geq 0$, there is a boundary matrix ∂_k whose rows are identified with the k-simplices of X and whose columns are identified with the $(k+1)$-simplices of X. An entry of the matrix ∂_k is 1 if the k-simplex corresponding to its row is contained as a face in the $(k+1)$-simplex corresponding to its column, and is 0 otherwise.
- The matrix product $\partial_k \cdot \partial_{k+1}$ exists and is identically zero.
- Emmy Noether observed that one important consequence of the construction of homology is that the numerical quantities attached to shapes obtained from it are in fact equal to dimensions of certain vector spaces $H_k(X)$, defined for each $k \geq 0$. These are called the homology groups of X. The dimension of $H_k(X)$ is called the kth Betti number of X, is denoted by β_k, and in an appropriate sense counts the number of holes in X whose boundary is k-dimensional.

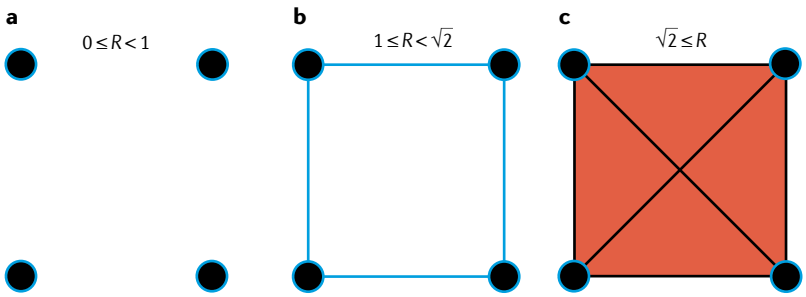

Fig. 3 | **The Vietoris–Rips complex.** The four vertices are positioned on the corners of a unit square. For a given distance threshold R, simplices are constructed that contain all vertices within a distance R. The Vietoris–Rips complex for a given R consists of the vertices and all simplices that exist for that R value.

- The Betti number β_0 is equal to the number of connected components of X.
- The Betti number β_1 is equal to the number of independent loops in X, up to homotopy.
- For a map of complexes $f : X \to Y$, there is an induced linear transformation $H_k(X) \to H_k(Y)$, satisfying the functoriality property. This property, observed by Emmy Noether, is critical to all applications and computational in the area. It means that if we have a composite $g \cdot f : X \to Y \to Z$, where $f : X \to Y$ and $g : Y \to Z$ are continuous maps, then the induced linear transformation $H_k(g \cdot f) : H_k(X) \to H_k(Z)$ is equal to the composite linear transformation $H_k(g) \cdot H_k(f)$. This means that if we equip the homology groups with bases, the transformation $H_k(g \cdot f)$ is given as the matrix product of the matrices corresponding to $H_k(f)$ and $H_k(g)$.

Persistent homology

We have seen that we can assign invariants (vector spaces) to complexes and ultimately to spaces. What we are interested in, though, is the ability to assign similar invariants in more discrete situations, specifically to point clouds. For example, given the data set shown in FIG. 1, it is desirable to be able to recognize automatically the presence of a loop in the data. We have already seen that homology provides such a recognition mechanism in the case where our space is given completely, rather than only through a sample. There is an extension of this idea to the point cloud setting, called persistent homology, which was defined in REF.[18], and developed further in REFS[19–21]. Useful surveys are given in REFS[10,22].

What we have seen above is that we are able to assign complexes to point clouds in a number of systematic ways. One of these methods is the Vietoris–Rips complex described above, and it suggests that for a point cloud X we may define the k-dimensional homology of X at scale R to be $H_k(V(X, R))$. This natural definition, however, suffers from the need to choose a scale R. In many situations, there is not a natural choice of R, and in others we may wish to understand the k-dimensional homology of X at all scales. In fact, it is possible to define a new invariant that encodes the values of H_k for all values of R in a single mathematical object. This is possible due to the functoriality property introduced above. The point is that if we consider all values of R at once, we obtain a family of vector spaces $\{V_R\}_R$, equipped with linear transformations $L(R, R') : V_R \to V_{R'}$ for every $R \leq R'$, so that $L(R', R'') \cdot L(R, R') = L(R, R'')$ whenever $R \leq R' \leq R''$. Of course, $V_R = H_k(V(X, R))$, and the linear transformations $L(R, R')$ are the linear transformations induced by the inclusions of complexes $V(X, R) \subset V(X, R')$, illustrated in FIG. 3. An object of this type, defined by a family of vector spaces parametrized by the real numbers and with transformations satisfying the above properties, is called a persistence vector space.

Just as we have the classification of vector spaces (up to isomorphism) by a non-negative integer called the dimension, so there is a classification of persistence vector spaces up to isomorphism. In the case of persistence vector spaces, the replacement for the dimension is the so-called persistence barcode. It is a finite

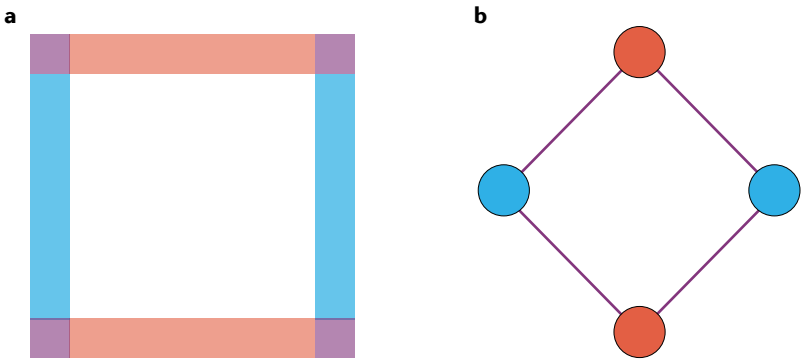

Fig. 4 | **The nerve construction of a covering. a** | The fattened boundary of a square is covered by four rectangular sets that intersect at the corners of the square. **b** | The corresponding nerve complex has one vertex for each of the four sets, with the set and the corresponding vertex having the same colour. Connections between vertices are drawn if and only if the corresponding sets have a non-empty intersection.

unordered collection of intervals, which are either finite or half-infinite of the form $[a, +\infty)$. Each interval corresponds to a feature in the k-dimensional homology of X, and indicates the scale values for which that feature exists. For instance, a point cloud A1 may appear to be a sample from a circle (FIG. 5a). If we analyse the cloud using a method such as the Vietoris–Rips complex at different scale values R, we find that there are several loops that appear for a narrow range of R values, as points become connected to each other in the simplicial complex constructed from the data. However, one loop exists for a wide range of R values, which is seen in the H_1 barcode A2 (FIG. 5b) as a long bar. This bar is our indication that the data are sampled from a space with one loop; the short-lived bars are taken to be noise. It is often useful to have another representation of barcodes, as pairs of points in the Euclidean plane, with the x value being the birth time and the y value being the death time. Points near the diagonal are features that exist for only a small range of R values; points further from the diagonal are persistent (FIG. 5c). Such a representation is referred to as a persistence diagram.

In another example, for the point cloud B1 (FIG. 5d), the H_1 barcode B2 (FIG. 5e) and persistence diagram B3 (FIG. 5f) give signals for the presence of two loops. Barcodes can be constructed for homology of any dimension. The point cloud C1 (FIG. 5g) has a H_0 barcode C2 (FIG. 5h) and a corresponding persistence diagram C3 (FIG. 5i). The H_0 barcode encodes the evolution of the connected components as R increases. The case of H_0 barcodes is a little anomalous, because it always has exactly one infinite interval (indicated by the right arrow), owing to the fact that there is always at least one connected component. For all dimensions >0, the barcodes have no infinite intervals. The infinite interval in the dimension-zero case is therefore not considered in the persistence diagram, because it would correspond to a y value of ∞. Also, the number of intervals in H_0 barcodes is equal to the number of data points in the point cloud, and is therefore very large. Displaying them for the point clouds A1 and B1 would not have been feasible.

What we have accomplished is an extension of the notion of homology and Betti numbers to the world of point clouds. The price we pay is that we no long obtain integers, but instead collections of intervals, where Betti numbers correspond to counts of 'long bars', suitably defined.

Functional persistence

It is possible to construct persistence barcodes in other ways. Supposing that we have a space X equipped with a real-valued function $f: X \to \mathbb{R}$, then we can form a persistence vector space $\{H_k(f^{-1}((-\infty, r])\}_r$ and corresponding persistence barcode. It reflects the topology of the sublevel sets as they develop over increasing values of r, regarded as a threshold value for f. To extend this idea to the case of a data set $\mathcal{D}$ (rather than a space) equipped with a real-valued function f, then we fix a choice of R and filter the Vietoris–Rips construction $V(\mathcal{D}, R)$ by the filtration that associates to a simplex $\{d_o, \ldots, d_k\}$ the filtration value $\max\{f(d_o), \ldots, f(d_k)\}$. Only simplices for which the filtration value is below some threshold are retained in the construction. This kind of construction can be quite useful in detecting geometric features that do not obviously come from connected components, loops or higher-dimensional generalizations thereof. For example, suppose our problem is to distinguish the two letters X and Y. No straightforward homology calculation is relevant, because these letters do not have loops. However, they do have ends, and, in fact, X has four ends and Y has three. If the letters are given as images, we can treat each as a collection of darkened pixels. The pixels can be given the structure of a point cloud, as they are embedded in the plane, and we can fix the distance threshold R to be $\sqrt{2}\,\delta$, where δ is the distance from any pixel to any of its four nearest neighbours. We define the centrality function C on any point cloud X (such as the set of pixels in a hand-drawn letter) by setting $C(p)$ to be the largest distance from p to any other point in X. Letting Δ denote the diameter of X, we can consider the functional persistence associated to the function $f^C(p) = \Delta - C(p)$. The first points included in the filtration are the points with highest C values, that is, those that are least central in the point cloud. The pictures of the complexes at increasing levels of f^C are given in FIG. 6.

Notice that the functional persistence barcode for the letter Y has only three bars, whereas for X there are four bars. The number of bars reflects the number of ends in the letter. One can use different kinds of functional persistence to capture other aspects of geometric behaviour, for example, corners. For many interesting kinds of data, such functions arise very naturally in the data. For example, in the case of complex molecules, quantities such as charge and electron density are of great importance, and can be used as the functional parameter.

Metrics on barcodes

The barcodes as defined above form a set without any particular structure attached. If they are to be used as features for the analysis of unstructured data, it is important to understand how they behave under small changes in the input data. For example, suppose that we have found that the persistent homology for a data set indicates the presence of some geometric features, such as loops. We might be interested in determining whether

Diameter
In any space where we have a notion of distance, the diameter is the maximum distance between any pair of points. For example, the diameter of the sphere is 2.

L_∞ distance
A notion of distance for $\mathbb{R}^n$ in which the distance between two points is the maximum of the absolute values of the differences between the coordinates of the two points.

the observed structure is meaningful, and one way to convince oneself that this is the case is to select multiple samples and see whether they generate results that are similar, but not necessarily identical. To do so, one needs to formalize the notion of similarity of barcodes. A simple and useful way to do this is by imposing a metric space structure on the set of barcodes. This can be done in numerous distinct ways. There are stability theorems that assert that small changes in the input data (suitably defined) produce small changes, in terms of the distance between barcodes, of the output. We sketch one metric between barcodes, and outline the stability properties for it that can be proven. Alternative choices of metric are defined in REFS[23–25].

The bottleneck distance is a kind of edit distance between barcodes, for which one is allowed to edit a barcode by adding an interval, deleting an interval and replacing an interval by another interval. Each edit step is assigned a penalty. For adding an interval, the penalty is the length λ of the interval being added, for deleting an interval the penalty is the length of the interval being deleted and for replacing an interval the penalty is the discrepancy $\lambda(I) + \lambda(J) - \lambda(I \cap J)$ when the interval I is replaced by the interval J. The penalty of a sequence of edit steps is the sum of the penalties of the individual steps, and the bottleneck distance is the length of an edit sequence going from one barcode to the other of minimal penalty.

There are variants of this notion of distance that use different penalty functions, and these are called the p-Wasserstein distances. The importance of these metrics derives from the existence of stability theorems. To understand the most important of these theorems, it is necessary to understand the so-called Gromov–Hausdorff distance on the collection of compact metric spaces. Two metric spaces have Gromov–Hausdorff distance equal to zero if and only if they are isometric. In the case of ordinary persistent homology, for which the input is a metric space, the stability theorem asserts that if two metric distances are a distance R apart, then each of their persistence barcodes are at most a distance R apart. When one studies functional persistence, and has two functions on the data set whose L_∞ distance is R, then if one generates the persistence barcodes of the functional persistence based on the two functions, one finds the barcodes are a distance $\leq R$ apart. There are numerous stability theorems of this type[23–25].

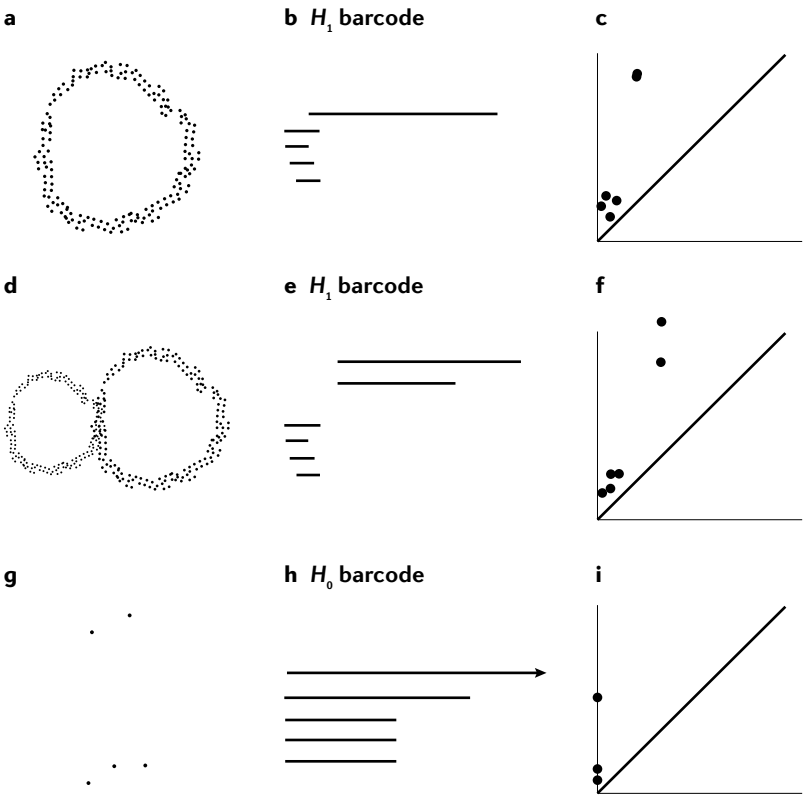

Fig. 5 | **Persistence barcodes for dimensions 0 and 1.** Data points that form a loop (panel **a**) have a 1D homology (H_1) barcode (panel **b**) that contains one long line (indicating the loop) and several shorter lines (which are noise). If the barcodes are plotted as death versus birth times (panel **c**), the loop is indicated by the single point with a large death time. Two loops (panel **d**) lead to two long bars in the H_1 barcode (panel **e**) and gives the birth-death pairs (panel **f**). For a data set that breaks clearly into components (panel **g**), the 0D (H_0) homology barcode always has one infinitely long bar (panel **h**) indicating a connected component. This bar is omitted from the persistence diagram (panel **i**).

Vectorization of persistence barcodes

In the topology of spaces where we have complete information about spaces rather than samples from a space, homology is used as an invariant that measures the shapes of individual data sets. Homology can be used to distinguish between spaces, and is often used to suggest how the space is constructed. This is also done in applied topology (that is, in the sampled situation)[8,26]. However, there is an additional family of applications to the study of data that has structure that does not fit neatly into descriptions as data matrices. Consider, for example, a situation in which a data set consists of molecules described as lists of atoms, bonds and bond lengths. On the one hand, one can formulate this data as a data matrix, but the description of molecules as vectors is not unique. For example, the ordering of the atoms and the bonds will drastically affect the vectors, so the Euclidean metric on this set of vectors will not be meaningful. On the other hand, the molecules may be treated as point clouds in their own right; the distance between two atoms can be given using the edge path distance, in which one computes the minimum of the sum of the lengths along any path in the graph given by the bonds of the molecule. This approach allows us to compute persistence barcodes for each point of a database of molecules. We can treat these barcodes as features of the data, and they can be used to understand the data set, perhaps using machine learning. One problem, though, is that the barcodes themselves are expressed as unordered sets of intervals. Such an expression is not directly suitable for applications of machine learning, which typically operate on data matrices with numerical entries. A solution to this problem is vectorization, that is, the development of methods that assign vectors to persistence barcodes, and therefore allow molecules (and other 'unstructured data') to be treated as structured data. There have been a number of methods for vectorization developed.

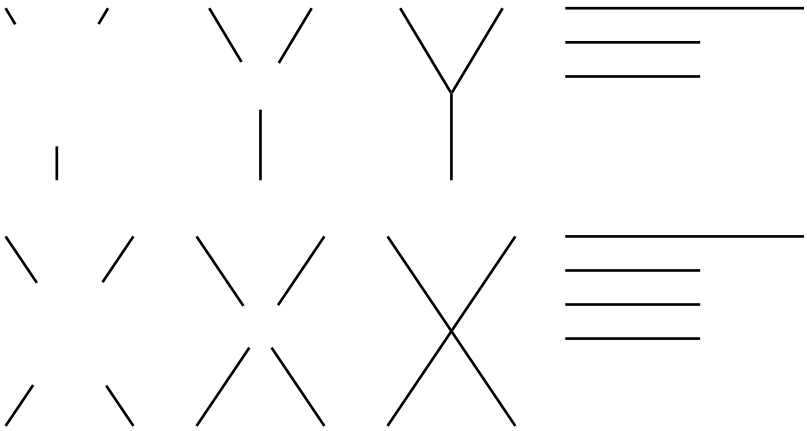

Fig. 6 | **Functional persistence barcodes.** Images of the letters Y and X can be converted into point clouds of dark pixels. One can define a function that quantifies how far from the centre of the cloud each point is, and construct simplicial complexes that are filtered by the value of this function (left). The barcodes of this filtration (right) contain three bars for a shape with three ends (such as Y) and four bars for a shape with four ends (such as X).

Persistence landscapes. This is a method that assigns a sequence $\{\varphi_k\}$ of real-valued functions to a persistence barcode $\{(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)\}$. It does so by first assigning to each interval $[a, b]$ the function $f_{a,b}$ which is zero outside $[a, b]$, which is equal to $x-a$ for $a \le x \le \frac{a+b}{2}$, and which is equal to $b-x$ for $\frac{a+b}{2} \le x \le b$. The function value $\varphi_k(x)$ is defined to be the kth-largest value among the values $f_{(a_i,b_i)}(x)$. If $k > n$, we assign the value zero to $\varphi_k(x)$. This function provides a number of ways of obtaining finite-dimensional vectors, for instance, by evaluating at finite collections of values of k and x. This representation is shown to be stable for the bottleneck distance on the set of barcodes. See REF.[27] for a complete discussion.

Persistence images. This is a method that proceeds by regarding barcodes through the persistence diagram representation, and viewing them as finite discrete sets of points in the plane. The idea is then to 'blur' the set by assigning a probability density function of fixed type to each point. Typically these will be 2D normal distributions centred at the points in question and with a fixed choice of variance. After a suitable recoordinatization of the (x,y) plane, and the multiplication by a chosen weighting function that vanishes on the x axis, the function can be regarded as lying in the first quadrant, and can therefore be regarded as an image, with the functions encoding greyscale values. This image is stable under the 1-Wasserstein distance. Details are given in REF.[28].

Polynomial functions. It is also possible to write down an algebra of functions generated by the functions f_{ij} defined by $f_{ij}(\{(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)\}) = \sum_s (a_s + b_s)^i (a_s - b_s)^j)$, where $j \ge 1$. These functions have the property (shared by persistence landscapes and persistence images) that if all these functions agree for a pair of barcodes, then the two barcodes are identical. Moreover, they are stable for the p-Wasserstein distances. They also have a tropical version, in which one considers analogues of polynomials in which the role of addition is played by the max function, and the role of multiplication is played by addition. These tropical functions enjoy the stability property for bottleneck distance. See REFS[29,30] for details of these constructions.

Tropical
A tropical algebra is a version of algebra with addition and multiplication replaced by max or min and multiplication, respectively.

Applications of topological modelling

We will discuss two classes of application. The first uses topological models (primarily Mapper)[15] directly, via the use of layout algorithms. The second class of applications applies persistent homology and uses the barcode output to obtain useful features and gain global understanding of the data set.

Direct applications. In the discussion above, topological modelling is described as the construction of simplicial complexes or graphs that capture useful information about the data. It is useful to make this definition more restrictive, and we will define topological modelling as the construction of coverings of a data set and the analysis of the corresponding nerve complexes. Of the complex constructors we discussed above, α-shapes, witness complexes and Mapper are all of this form. Mapper and α-shapes are equipped with explicit bounds on the dimension of the corresponding complexes. Supposing that we have constructed a covering $\mathcal{U}$ of a data set $\mathcal{D}$, and that we have constructed the corresponding simplicial complex X, there is a great deal of functionality that is available within the model. These include:

- Using standard graph layout algorithms to view the complex, thus providing a visualization tool for point clouds. This visualization is most effective when the complex is reasonably low dimensional. Functions on the vertex set of X are usefully represented by colourings of the vertices.
- Selecting subgroups on the laid-out complex based on the geometric structures of X. These groups can be treated as data sets in their own right, and analysed to obtain a more local understanding of the data set. This often gives more fine-grained information about $\mathcal{D}$. One can select families of subgroups to obtain segmentations of the data set.
- Given a function f on $\mathcal{U}$, one can construct a corresponding function $f^{\mathcal{U}}$ on the set of nodes of X using an averaging procedure, as follows. Each vertex v of X corresponds to a collection $\mathcal{D}_v \subseteq \mathcal{D}$, by the definition of the nerve complex, and $f^{\mathcal{U}}(v)$ can be defined to be the average of the function values $f(v)$ over the set $\mathcal{D}_v$. This procedure allows one to obtain an understanding of the behaviour of the function on $\mathcal{D}$, and in particular allows for the identification of local 'hotspots' of f.
- Given a subset $\mathcal{D}_0 \subseteq \mathcal{D}$, one can define a function on the set of vertices v of X to be the fraction of elements of $\mathcal{D}_v$ that lie in $\mathcal{D}_0$. In this way, one can understand where a chosen groups is localized in $\mathcal{D}$.
- One can obtain 'explanations' characterizing subgroups that have been selected or defined. This is done by considering all the variables, and for each computing a Kolmogorov–Smirnov score that compares the distributions of the variable on the subgroup to that on the entire set. The variables can then be ordered by this score, so that one obtains the

variables with highest Kolmogorov–Smirnov score first in the list.

- If one has two distinct coverings $\mathcal{U}$ and $\mathcal{U}'$ of $\mathcal{D}$, it is often useful to produce layouts of the corresponding nerve complexes X and X' simultaneously. This permits the selection of a group of vertices in X, and then for each vertex v of X' computing the percentage of points of $\mathcal{D}_v$ that lie in the subcollection of $\mathcal{D}$ given by the union of the collections corresponding to the selected group in X. This procedure gives a very good way to compare models.
- If we suppose that our data are given as the rows of a rectangular matrix, it is often useful to build a topological model not on the data points (rows) of the matrix, but rather on the columns (features) of the matrix. One can regard each data point of the original matrix as a function on the set of features, and construct the corresponding function on the vertices of the topological model of the set of features. This process gives a useful way to understand data sets with many features directly. There can be situations in which the number of rows is so small that building a useful topological model of them is not possible, but for which the number of features is sufficiently large to support such a model, and this method allows one to obtain useful insight about the small but high-dimensional data set. By a further averaging procedure, one can obtain functions based on subsets of the data set as well.

The capabilities described above allow one to use topological models to interrogate data sets effectively, and provide a good way to search for useful subgroups and cohorts within a data set. We now present a few examples of the application of the modelling to scientific problems.

The study of dynamics of folding of biomolecules is a classical problem in biophysics. The folding process is determined by an underlying free-energy landscape, which may contain local minima. Understanding these minima is important for understanding the folding process. Computer simulations are vital to this area of research, because experiments often cannot achieve sufficient resolution. In REF.[31], the mapping methods described above are used in the simulation of a relatively small molecule called an RNA hairpin to discover such local minima. FIGURE 7a shows the Mapper model of the space of conformations, with the corresponding conformations diagrammed below. The presence of a somewhat irregular area in the middle of the Mapper model was the clue to the presence of the local minima. The idea is that the space of conformations of the molecules is large, and needs to be represented in a compact way that is nevertheless capable of capturing the local information around the minima.

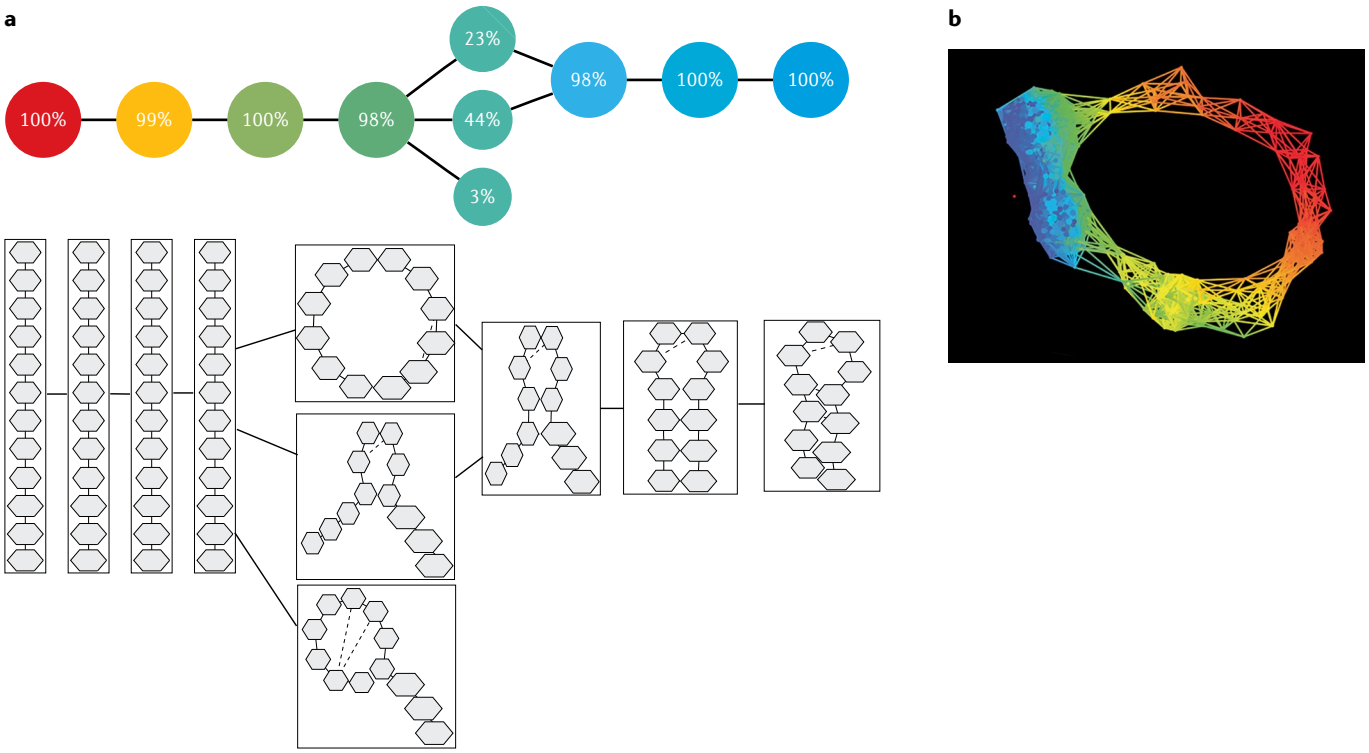

Fig. 7 | **Applications of topological data analysis. a** | Analysis of a simulation of RNA hairpin folding. The lower part shows contact maps of states of the molecule as it folds (from left to right). Mapper analysis of the simulation data (upper part) reveals two dominant pathways from the unfolded state to the folded. Colours indicate the value of a conditional density filter and the percentage labels indicate the fraction of configurations of the same level (based on the values of the conditional density filter) first included in the node. **b** | Infectious disease Mapper model. The underlying data points are 78 data points from a longitudinal study of three mice infected with malaria. The data points are converted into a network using Mapper. The network reveals a loop structure in which the transition from healthy to disease state is distinct from the recovery transition from disease to healthy state. The model is coloured by a quantitative measure of the presence of granulocytes, a particular type of white blood cell. Part **a** adapted with permission from REF.[31], AIP Publishing. Part **b** reprinted from REF.[34], CC BY 4.0.

Spectroscopy is an important tool for many applications throughout the physical sciences and engineering. The data sets often have large numbers of features, and exhibit a great deal of complexity. References[32,33] report on the application of topological methods to various problems in this area, and performing comparative evaluations of TDA with more conventional methods such as principal component analysis and hierarchical clustering, and find that TDA obtains additional resolution beyond both of these methods.

Reference[34] reports on using topological methods to study the progression of infectious diseases, such as malaria and influenza (FIG. 7b). The idea is to use various physiological and genomic variables to produce a model for a space of states coming from the study of progression of infectious disease. The topological model for this situation should be a loop, where one imagines that one begins with the healthy state then moves along an arc as the disease develops, and where one traverses a much different arc in the return to the healthy state.

The topological model is constructed without the use of the time variable, and so gives a time-invariant notion of the state within the disease process. It is important to have such a time-invariant model, because one might not have information about the time of infection and the speed with which the states are traversed depends on resilience of the subjects. References[34,35] go further in relating the various types of microarray data to the traversal of the model, and also describes a method for predicting whether or not a given subject will in fact recover.

There are many other applications of this kind of modelling. In REF.[36], a domain-specific method for constructing graphical models in the context of molecular chemistry is introduced, and REF.[37] provides another example of work in this direction. In REF.[38], another domain-specific graph construction for the study of salt solutions is given. Some highlights in the biomedical domain are REFS[39–48].

Applications of persistent homology. There are numerous applications of persistent homology. A very interesting one occurs in materials science. There is a large body of theory that is powerful in the analysis of crystalline structures. Conversely, amorphous solids are much more difficult to characterize and analyse. One can initially study the short-range order, that is, the statistics of pairs of nearest neighbours in the structure. Doing so is important, but is insufficient in many situations, which motivates the study of longer-scale interactions, the medium-range order. In addition, it has been observed that such materials often exhibit hierarchical structures, with differing phenomena occurring at various scales. One approach is to study the distributions of bond angle and dihedral angle, which gives additional information, but which still only gives information of the atomic configurations involving the second- and third-nearest neighbours. To study longer-scale interactions, one may consider the substance as a network, with atoms as nodes and bonds as edges, and perform counts of rings (closed edge paths in the network) of various lengths and types, as in REF.[49]. Doing so gives yet more information, but the method is only applicable to crystalline materials and continuous random networks, cannot take distances (bond lengths) into account and furthermore cannot account for hierarchical structures. Persistent homology provides a method that enables the systematic study of geometric features, such as rings, in a way that accounts for lengths and hierarchical structure. The material is studied by treating it as a point cloud in its own right, in which points are the atoms and the distances are given by 3D Euclidean distance. The persistence diagrams of such point clouds are used in REF.[50] to quantify and make precise geometric properties of several substances, namely silica glass, the Lennard–Jones system and Cu–Zr metallic glass. In the case of silica glass, for example, the persistence diagram classified rings in both the short- and medium-range orders, found the hierarchical relationship between them, and found that the first sharp diffraction peak was computable from the persistence diagram. In addition, it was able to make predictions regarding elastic response. Another example of persistent homology applied to materials science is given in REF.[51].

Another application of persistent homology is in the study of the effects of forces acting on dense granular media, which consist of collections of granular particles. External forces acting on a granular material produce complicated interparticle forces within the material. An isotropic external force produces an internal force field, the magnitude of which in a particular direction can be viewed as a scalar function on the set of granular particles. The granular particles can be viewed as a point cloud, and the force field gives a function on this set, which permits the application of functional persistence. This approach gives a method to produce qualitative understanding of this force field, including the discovery of local maxima and minima. Two interesting examples of this kind of work are REFS[52,53].

A different direction is in the study of complex molecules, including drug discovery. Each molecule is treated as a finite metric space, with the atoms constituting it being the points and the distances computed using the bonds in the molecule. Analysis based on the persistence diagrams associated to these metric space structures has been successfully applied to problems in drug design[54,55]. In addition, specifically designed functional persistence barcodes provide features that effectively distinguish between molecules.

Other interesting examples includes the study of the distribution of matter in the Universe[56,57] and the study of non-covalent bonds for systems containing heavy elements[37]. These examples use persistent homology as a method for feature generation for data sets in which the elements themselves are equipped with a geometry.

Persistent homology has also been used to study the structure of an entire data set. Two examples of this notion are given in REFS[8,26] for applications in the study of statistics of natural images and in the study of viral evolution, respectively.

Computational aspects

A systematic survey of algorithms and implementations for computing the persistent homology of filtered complexes was presented in REF.[58], to which we direct a reader interested in implementing persistent homology or using

existing software.The years since that survey was published have seen continued algorithmic developments in a variety of directions, including parallelization, efficient collapse methods for large complexes and the use of more complicated diagrams of complexes[59–62], as well as implementations intended for massively parallel and graphics processing unit architectures[63–65]. A toolkit for general data science applications is described in REF.[66]. In REFS[36,37], domain-specific toolkits are developed for the graphical analysis of molecular and condensed systems. Many of the innovations in computing persistent homology since the mid-2000s have focused on improving the performance of the reduction algorithm introduced in REF.[21], but persistent homology as presented here is only the first version of homology. There are numerous extensions that required more sophisticated algorithms, and the ability to support them was only recently fully implemented in REF.[67].

Outlook

TDA is still in the early stages of development. There are number of important research directions, which we summarize here. There are excellent applications to solid-state physics and materials science, in which the persistence barcodes provide invariants of large sets of points distributed in the plane or in space, as in REF.[50]. The points can be used to construct density functions, and the sublevel sets of these functions are also of a great deal of interest. The same kind methodology is also applied to the study of the cosmic web[56,57]. This kind of application is an active area of research, with the general idea being that one is able to attach invariants in a systematic way to large ensembles of objects. In this level of generality, there are also many applications in biology. Dynamical systems is another active area of study, in which topological invariants such as Conley indices[68] are useful in understanding the structure of the dynamics[69–71]. Within the subject, there is a great deal of work being done in understanding the statistical and probabilistic nature of the persistence barcodes. This is of fundamental importance for many problems in data science, as one would like to do inference concerning shapes from the barcodes, and that cannot be done without a detailed understanding of distributions on spaces of barcodes. Another important direction is work towards understanding directly the stability properties of the Mapper construction, as well as other complex constructions, rather than their barcodes. This work is exemplified by REF.[72]. Persistent homology as we have defined it is actually the first example of the application of diagrams of vector spaces to study data. There are a number of extensions, including zig-zag persistent homology[73] (generally giving more refined invariants than ordinary persistence) and multidimensional persistence[74] (which permits vectors spaces that depend on more than one parameter). Computational problems involving these constructions are currently being studied. Two examples of this kind of work are REFS[67,75].

Published online: 10 November 2020

1. Berkowitz, J. Big data hits beamline. *Berkeley Lab. Comput. Sci.* https://cs.lbl.gov/news-media/news/2013/big-data-hits-the-beamline/ (2013).
2. Gaillard, M. CERN Data Centre passes the 200-petabyte milestone. *CERN* https://home.cern/news/news/computing/cern-data-centre-passes-200-petabyte-milestone (2017).
3. Everitt, B., Landaum S., Leese, M. & Stahl, D. *Cluster Analysis* (John Wiley, 2011).
4. Armstrong, M. *Basic Topology* (Springer, 1983).
5. Dummit, D. & Foote, R. *Abstract Algebra* Vol. 1 (Wiley, 2004).
6. Edelsbrunner, H. & Harer, J. *Computational Topology. An Introduction* (American Mathematical Society, 2010).
7. Chazal, F. & Michel, B. An introduction to topological data analysis: fundamental and practical aspects for data scientists. Preprint at *arXiv* https://arxiv.org/abs/1710.04019 (2017).
8. Carlsson, G., Ishkhanov, T., De Silva, V. & Zomorodian, A. On the local behavior of spaces of natural images. *Int. J. Computer Vis.* **76**, 1–12 (2008).
9. Hatcher, A. *Algebraic Topology* (Cambridge Univ. Press, 2002).
10. Carlsson, G. Topological pattern recognition for point cloud data. *Acta Numer.* **23**, 289–368 (2014).
11. Vietoris, L. Über den höheren Zusammenhang kompakter Räume un eine klasse von zusammenhangstreuen Abbildungen. *Math. Ann.* **97**, 454–472 (1927).
12. Edelsbrunner, H., Kirkpatrick, D. & Seidel, R. On the shape of a set of points in the plane. *IEEE Trans. Inf. Theory* **29**, 551–559 (1983).
13. Akkiraju, N. et al. Alpha shapes: definition and software. *Geometry Center* http://www.geom.uiuc.edu/software/cglist/GeomDir/shapes95def/index.html (1995).
14. de Silva, V. & Carlsson, G. Topological estimation using witness complexes. *Eurographics* https://doi.org/10.2312/SPBG/SPBG04/157-166 (2004).
15. Singh, G., Memoli, F. & Carlsson, G. Topological method for the analysis of high dimensional data sets and 3D object recognition. *Eurographics* https://doi.org/10.2312/SPBG/SPBG07/091-100 (2007).
16. Aurenhammer, F., Klein, R. & Lee, D. *Voronoi Diagrams and Delaunay Triangulations* (World Scientific, 2013).
17. Reeb, G. Sur les points singuliers d'une fome de Pfaff completement integrable ou d'une fonction numerique. *C. R. Seances Acad. Sci.* **222**, 847–849 (1946).
18. Robins, V. Towards computing homology from finite approximations. *Topol. Proc.* **24**, 503–532 (1999).
19. Frosini, P. & Landi, C. Size theory as a topological tool for computer vision. *Pattern Recognit. Image Anal.* **9**, 596–603 (1999).
20. Edelsbrunner, H., Letscher, D. & Zomorodian, A. Topological persistence and simplification. *Discrete Comput. Geom.* **28**, 511–533 (2002).
21. Zomorodian, A. & Carlsson, G. Computing persistent homology. *Discrete Comput. Geom.* **33**, 249–274 (2005).
22. Edelsbrunner, H. & Harer, J. Persistent homology — a survey. *Contemp. Math.* **453**, 257–282 (2008).
23. Chazal, F., Cohen-Steiner, D., Guibas, L., Memoli, F. & Oudot, S. Gromov–Hausdorff stable signatures for shapes using persistence. *Comput. Graph. Forum* **28**, 1393–1403 (2009).
24. Cohen-Steiner, D., Edelsbrunner, H. & Harer, J. Stability of persistence diagrams. *Discrete Comput. Geom.* **37**, 103–120 (2007).
25. Steiner, D. C., Edelsbrunner, H., Harer, J. & Mileyko, Y. Lipschitz functions have L_p-stable persistence. *Found. Computat. Math.* **10**, 127–139 (2010).
26. Chan, J., Carlsson, G. & Rabadan, R. Topology of viral evolution. *Proc. Natl Acad. Sci. USA* **110**, 18566–18571 (2013).
27. Bubenik, P. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.* **16**, 77–102 (2015).
28. Adams, H. et al. Persistence images: a stable vector representation of persistent homology. *J. Mach. Learn. Res.* **18**, 1–35 (2017).
29. Adcock, A., Carlsson, E. & Carlsson, G. The ring of algebraic functions on persistence barcodees. *Homol. Homotopy Appl.* **18**, 381–402 (2016).
30. Kalisnik, S. Tropical coordinates on the space of persistence barcodes. *Found. Comput. Math.* **19**, 101–129 (2019).
31. Yao, Y. et al. Topological methods for exploring low-density states in biomolecular folding pathways. *J. Chem. Phys.* **130**, 144115 (2009).
32. Duponchel, L. Exploring hyperspectral imaging data sets with topological data analysis. *Anal. Chim. Acta* **1000**, 123–131 (2018).
33. Offroy, M. & Duponchel, L. Topological data analysis: a promising big data exploration tool in biology, analytical chemistry, and physical chemistry. *Anal. Chim. Acta* **910**, 1–11 (2016).
34. Torres, B. et al. Tracking resilience to infections by mapping disease space. *PLoS Biol.* **14**, e1002494 (2016).
35. Louie, A., Song, K. H., Hotson, A., Thomas Tate, A. & Schneider, D. S. How many parameters does it take to describe disease tolerance? *PLoS Biol.* **14**, e1002485 (2016).
36. Bhatia, H., Gyulassy, A., V. Lordi, P. J., Pascucci, V. & Bremer, P. TopoMS: comprehensive topological exploration for molecular and condensed-matter systems. *J. Comput. Chem.* **39**, 936–952 (2018).
37. Olejniczak, M., Gomes, A. & Tierny, J. A topological data analysis perspective on non-covalent interactions in relativistic calculations. *Int. J. Quantum Chem.* **120**, e26133 (2019).
38. Lukasczyk, J. et al. Viscous fingering: a topological visual analytic approach. *Appl. Mech. Mater.* **869**, 9–19 (2017).
39. Lee, J. et al. Spatiotemporal genomic architecture informs precision oncology in glioblastoma. *Nat. Genet.* **49**, 594e599 (2017).
40. Camara, P., Levine, A. & Rabadan, R. Inference of ancestral recombination graphs through topological data analysis. *PLoS Comput. Biol.* **12**, e1005071 (2016).
41. Camara, P. Topological methods for genomics: present and future directions. *Curr. Opin. Syst. Biol.* **1**, 95–101 (2017).
42. Nicolau, M., Levine, A. & Carlsson, G. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proc. Natl Acad. Sci. USA* **108**, 7265–7270 (2011).
43. Romano, D. et al. Topological methods reveal high and low functioning neuro-phenotypes within fragile X syndrome. *Hum. Brain Mapp.* **35**, 4904–4915 (2014).

44. Nielson, J. et al. Topological data analysis for discovery in preclinical spinal cord injury and traumatic brain injury. *Nat. Commun.* **6**, 8581 (2015).
45. Saggar, M. et al. Towards a new approach to reveal dynamical organizaton of the brain using topologial data analysis. *Nat. Commun.* **9**, 1399 (2018).
46. Li, L. et al. Identification of type 2 diabetes subgroups through topological analysis of patient similarity. *Sci. Transl. Med.* **7**, 311ra174 (2015).
47. Hinks, T. et al. Multidimensional endotyping in patitents with severe asthma reveals inflammatory heterogeneity in matrix metalloproteinases and chitinase 3-like protein 1. *J. Allergy Clin. Immunol.* **138**, 61–75 (2016).
48. Hinks, T. et al. Innate and adaptive T-cells in asthmatics patients: relationship to severity and disease mechanisms. *J. Allergy Clin. Immunol.* **136**, 323–333 (2015).
49. Leroux, S. & Jund, P. Ring statistics analysis of topological networks: new approach and application to amorphous GeS_2 and SiO_2 systems. *Comput. Mater. Sci.* **49**, 70–83 (2010).
50. Hiraoka, Y. et al. Hierarchical structures of amorphous solids characterized by persistent homology. *Proc. Natl Acad. Sci. USA* **113**, 7035–7040 (2016).
51. MacPherson, R. & Schweinhart, B. Measuring shape with topology. *J. Math. Phys*. **53**, 073516 (2012).
52. Kramar, M., Goullet, A., Kondic, L. & Mischaikow, K. Persistence of force networks in compressed granular media. *Phys. Rev. E* **87**, 042207 (2013).
53. Mueth, D., Jaeger, H. & Nagel, S. Force distribution in a granular medium. *Phys. Rev. E* **57**, 3164–3169 (1998).
54. Cang, Z. & Wei, G. TopologyNet: topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLoS Comput. Biol.* **13**, e100569 (2017).
55. Nguyen, D. et al. Mathematical deep learning for pose and binding affinity prediction and ranking in D3R Grand Challenges. *J. Comput. Aided Mol. Des.* **33**, 71–82 (2019).
56. Sousbie, T. The persistent cosmic web and its filamentary structure — I. Theory and implementation. *Mon. Not. R. Astron. Soc.* **414**, 350–383 (2011).
57. Sousbie, T., Pichon, C. & Kawahara, H. The persistent cosmic web and its filamentary structure — II. Illustrations. *Mon. Not. R. Astron. Soc.* **414**, 384–403 (2011).
58. Otter, N., Porter, M., Tillmann, U., Grindrod, P. & Harrington, H. A roadmap for the computation of persistent homology. *EPJ Data Sci.* **6**, 17 (2017).
59. Henselman, G. & Ghrist, R. Matroid filtrations and compputational persistent homology. Preprint at *arXiv* https://arxiv.org/abs/1606.00199 (2016).
60. Yoon, H. *Cellular Sheaves and Cosheaves for Distributed Topological Data Analysis.* Thesis, Univ. Pennsylvania (2018).
61. Boissonnat, J.-B., Pritam, S. & Pareek, D. Strong collpase for persistencey. Preprint at *arXiv* https://arxiv.org/abs/1809.10945 (2018).
62. Kerber, M. & Schreiber, H. Barcodes of towers and a streaming algorithm for persistent homology. *Discrete Comput. Geom.* **61**, 852–879 (2018).
63. Zhang, S., Xiao, M. & Wang, H. GPU-accelerated computation of Vietoris–Rips persistence barcodes. Preprint at *arXiv* https://arxiv.org/abs/2003.07989 (2020).
64. Zhang, S. et al. HYPHA: a framework based on separation of parallelisms to accelerate persistent homology matrix reduction (ACM, 2019).
65. Morozov, D. & Nigmetov, A. Towards lockfree persistent homology (ACM, 2020).
66. Tierny, J., Favelier, G., Levine, J., Gueunet, C. & Michaux, M. The topology toolkit. *IEEE Trans. Vis. Comput. Graph.* **24**, 832–842 (2017).
67. Carlsson, G., Dwaraknath, A. & Nelson, B. J. Persistent and zigzag homology: a matrix factorization viewpoint. Preprint at *arXiv* https://arxiv.org/abs/1911.10693 (2019).
68. Batko, B., Mischaikow, K., Mrozek, M. & Przybylski, M. Conley index approach to sampled dynamics. *SIAM J. Appl. Dyn. Syst.* **19**, 665–704 (2020).
69. Mischaikow, K., Mrozek, M., Reiss, J. & Szymczak, A. Construction of symbolic dynamics from experimental time series. *Phys. Rev. Lett.* **82**, 1144 (1999).
70. Zgliczynski, P. & Mischaikow, K. Rigorous numerics for partial differential equations: the Kuramoto–Sivashinsky equation. *Found. Comput. Math.* **1**, 255–288 (2013).
71. Chen, G., Mischaikow, K., Laramee, R., Pilarczyk, P. & Zhang, E. Vector field editing and periodic orbit extraction using Morse decomposition. *IEEE Trans. Vis. Comput. Graph.* **13**, 769–785 (2007).
72. de Silva, V., Munch, E. & Patel, A. Categorified Reeb graphs. *Discrete Comput. Geom.* **55**, 854–906 (2016).
73. Carlsson, G. & de Silva, V. Zigzag persistence. *Found. Comput. Math.* **10**, 367–405 (2010).
74. Carlsson, G. & Zomorodian, A. The theory of multidimensional persistence. *Discrete Comput. Geom.* **42**, 71–93 (2009).
75. Lesnick, M. & Wright, M. Interactive visualization of 2-D persistence modules. Preprint at *arXiv* https://arxiv.org/abs/1512.00180 (2015).

Acknowledgements
This article has benefited greatly from discussions with J. Carlsson, P. Lum, S. Locklin and B. Mann.

Competing interests
The author declares no competing interests.

Peer review information
Nature Reviews Physics thanks Vanessa Robins and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.