Mapper–type algorithms for complex data and relations

Paweł Dłotko
Dioscuri Centre in Topological Data Analysis
Mathematical Institute PAN, Warsaw, PL

Davide Gurnari
Dioscuri Centre in Topological Data Analysis
Mathematical Institute PAN, Warsaw, PL

Radmila Sazdanovic
Department of Mathematics
North Carolina State University, Raleigh, NC

July 28, 2024

Abstract

Mapper and Ball Mapper are Topological Data Analysis tools used for exploring high dimensional point clouds and visualizing scalar–valued functions on those point clouds. Inspired by open questions in knot theory, new features are added to Ball Mapper that enable encoding of the structure, internal relations and symmetries of the point cloud. Moreover, the strengths of Mapper and Ball Mapper constructions are combined to create a tool for comparing high dimensional data descriptors of a single dataset. This new hybrid algorithm, Mapper on Ball Mapper, is applicable to high dimensional lens functions. As a proof of concept we include applications to knot and game theory.

Keywords: Visualization, Exploratory Data Analysis, Dimension Reduction, Topological Data Analysis

1 Introduction

Mapper (Singh et al., 2007) and Ball Mapper (Dłotko, 2019) algorithms are Topological Data Analysis (TDA) tools for exploring and visualizing data. The Mapper algorithm was introduced in 2007 and since then it has been used in a number of different applications, most notably medical data analysis (Nicolau et al., 2011; Li et al., 2015), and material science (Lee et al., 2017). Ball Mapper (Dłotko, 2019) was introduced in 2019 as an easy-to-use effective alternative to the Mapper algorithm, see Qiu et al. (2020) for applications to data in economics.

Both techniques take a point cloud X, possibly with a scalar–valued function $f : X \to \mathbb{R}$, as an input, and return an abstract graph $G = (V, E)$, with a induced function $\hat{f} : V \to \mathbb{R}$. The "layout" of G resembles the multidimensional layout of the input set X. In addition, visualising $\hat{f}$ over the vertices of G, typically using an appropriate color scale, provides insights into properties of the input function f. One of the first and most famous examples is the work of Nicolau et al. (2011), where the input point cloud consists of gene expression data for breast cancer patients, and the function f determines the survival rate. Mapper associated rare cancer to subtypes with 100% survival rate and allowed the authors to characterize its genetic profile.

This research introduces novel techniques that broaden and improve the scope and applicability of mapper-type algorithms to utilize additional structure of data and visualize the maps between datasets. Main contributions include:

1. *Equivariant Ball Mapper EqBM*: the most natural choice of mapper-type algorithms when the input data admits an action of a group of isometries or distance preserving bijection. The structure of the resulting Ball Mapper graph reflects the action of the input isometries.

2. *Mapper on Ball Mapper MoBM*: Extension of the Mapper algorithm that allows the use of high dimensional lens functions $f : X \to \mathbb{R}^n$, with $n \gg 1$. In the proposed technique, a Ball Mapper graph of the range of f is built to obtain an *adaptive cover* which is then used to construct the final Mapper graph.

3. *MappingMappers*: Mapper and Ball Mapper–based representations of high dimensional datasets X and Y can be used to visualize a given relation $f : X \to Y$. For example, if X and Y are values of different descriptors of the object of interest, such as different knots, MappingMappers can be used to compare descriptors' relevance and discover potential correlations or

dependencies.

The main running and motivating example in this paper is an analysis of data coming from knot theory which has recently opened up to big data techniques such as machine learning (Hughes, 2020; Jejjala et al., 2019; Levitt et al., 2019; Davies et al., 2021; Gukov et al., 2021, 2023). Knot theory point clouds created from polynomial knot invariants such as the Alexander, Jones and HOMFLYPT, are perfect for showcasing the strengths of the Equivariant, MappingMappers and Mapper on Ball Mapper algorithms. An additional example related to game theory is discussed to highlight the presented algorithms.

Techniques developed in this work are accompanied with sample public-domain implementations and have wide applicability in different areas of science. The software implementing the discussed techniques and the interactive visualizations of all the plots in this paper are available at the webpage `https://dioscuri-tda.org/BallMapperKnots.html` and in Dłotko et al. (2023).

The paper is organized as follows. Section 2 provides the necessary background on the Mapper and Ball Mapper algorithms. Section 3 focuses on adaptations and new developments of Mapper algorithms that are applicable to any point cloud. In Section 3.1 we develop a version of Ball Mapper that takes into account symmetries (global or partial) of the data. Section 3.2 describes a way to construct Mapper graphs for lens functions with high dimensional domains and codomains. In Section 3.3 we discuss how to combine the strengths of the Mapper and Ball Mapper algorithms to analyze data, relations between high dimensional point clouds, and visualize maps between different datasets. Section 4.1 provides informal minimal background information about relevant knot invariants while Section 4.2 describes a way to turn knot invariants into point clouds so they can be analyzed by TDA techniques. Section 4.3 present the results of application of the the presented Mapper–type techniques to the space of knots and their invariants, while Sections 4.4 and 4.5 focuses on the comparison of knot invariants using techniques from Section 3.3 and shows how the presented technique can benefit theoretical mathematics. Finally, Section 5 provides additional examples of the use of the proposed techniques in game theory.

2 Background

In this section we present the Mapper algorithm (Singh et al., 2007), a standard tool of Topological Data Analysis, as well as the recently developed Ball Mapper (Dłotko, 2019). Both tools serve

to represent a finite point cloud as an abstract graph that encapsulates the essential features of the point cloud. This graph is obtained from a given *overlapping cover* of the point cloud, a finite collection of sets such that each point belongs to at least one set. Each set (element of the cover) represents a vertex of this graph, and two vertices are connected by an undirected edge whenever the corresponding sets have a non-empty intersection. In the language of modern topology this construction amounts to taking the 1-skeleton of the nerve of the cover. The difference between the Mapper and Ball Mapper algorithms lies in the way such an overlapping cover is obtained.

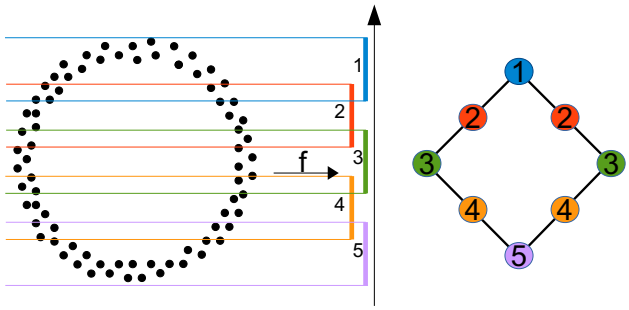


Figure 1: The Mapper construction illustration. The input is the 2-dimensional point cloud X in a shape of a circle shown on the left. The function $f : X \to \mathbb{R}^1$ is a projection of a point to its second coordinate. We cover the range of f with five intervals, enumerated from 1 to 5, then compute the inverse image of each interval. In this example, the inverse images of intervals 1 and 5 contain one cluster, while inverse images of intervals $2, 3$ and 4 contain two clusters each. Moreover there are obvious connections between the clusters in the inverse image of intervals i and $i + 1$ for $i \in 1, 2, 3, 4$. They give rise to edges in the Mapper graph presented on the right. Note that the (non-unique) enumeration of vertices comes from the enumeration of elements of the cover $\mathcal{C}$.

2.1 Mapper

The *Mapper algorithm*, introduced in 2007 by Singh et al. (2007), is one of the most recognizable tools of Topological Data Analysis. It can be considered a discrete approximation of a *Reeb graph* (Reeb, 1946; Munch and Wang, 2016). An input for Mapper is a collection of points X, often embedded in some high dimensional space, and a function $f : X \to \mathbb{R}^n$, referred to as the

lens function. Typically, the range of the lens function is 1-dimensional. Next, construct the cover of the range $f(X) \subset \mathbb{R}^n$ by a *collection of overlapping cubes* $\mathcal{C}$. For $n = 1$, $\mathcal{C}$ consists of k intervals covering $f(X)$ such that two constitutive intervals overlap on p percent of their length. The number of intervals k and the overlapping percentage p, often refereed to as *resolution* and *gain*, are additional input parameters of the Mapper algorithm.

For each element $I \in \mathcal{C}$, consider $f^{-1}(I) \subset X$ and search for the clusters therein. The clustering algorithm used for that purpose is yet another parameter of the Mapper construction. For $I \in \mathcal{C}$ let C_I indicate the collection of clusters found in $f^{-1}(I)$. Each cluster in $\bigcup_{I \in \mathcal{C}} C_I$ corresponds to a vertex of an abstract graph $G = (V, E)$. Given a cluster $A \in C_I$, the vertex corresponding to A is denoted by $v(A)$. An undirected graph G is constructed using the following rule: for any two vertices $v(A)$ and $v(B)$ corresponding to clusters $A \in C_I$ and $B \in C_J$, an undirected edge is placed between $v(A)$ and $v(B)$ if and only if $A \cap B \neq \emptyset$. The resulting graph $G = (V, E)$ is called a *Mapper graph*. An illustration of the Mapper construction on a input point cloud embedded in $\mathbb{R}^2$ is shown in Figure 1.

While Mapper is a well established tool for data analysis, its stability with respect to perturbation of input parameters is still open for explorations. Main results in this direction focus on convergence of Mapper graphs to the Reeb graph (Reeb, 1946) of the manifold from which the points are sampled from, when the number of sampled points goes to infinity (Brown et al., 2021; Carrière et al., 2018). However, the practical scope of those results is limited, especially when dealing with finite noisy samples.

2.2 Ball Mapper

The *Ball Mapper Algorithm* introduced in (Dłotko, 2019; Qiu et al., 2020), provides a conceptually different and simpler way to obtain a cover of the input cloud X than the original Mapper. Starting from a point cloud X, and a constant $\epsilon > 0$, a subset $L \subset X$ is selected having the property that for every $x \in X$, there exists $l \in L$ such that $d(x, l) \leq \epsilon$. Such a subset L is called an ϵ*–net* of X and its points are referred to as "landmarks". Algorithm 1 is an example of a greedy algorithm to compute an ϵ–net. Other algorithms can also be used to compute ϵ–nets (Haussler and Welzl,

1987).

Data: Point cloud X, $\epsilon > 0$

Result: $L \subset X$, an ϵ-net

$L = \emptyset$;

for $x \in X$ **do**

  if *x is farther than ϵ from every point in L* **then**

    $L = L \cup \{x\}$;

  end

end

return L

Algorithm 1: Greedy ϵ–net (Dłotko, 2019).

Note that $X \subset \bigcup_{l \in L} B(l, \epsilon)$. The *Ball Mapper graph* consists of a vertex $v(l)$ for each ball $B(l, \epsilon)$ (hence one vertex for each landmark point in the ϵ-net), and an edge is placed between any two vertices when their corresponding balls jointly cover points from X. An illustration of the Ball Mapper algorithm, for the dataset from Figure 1, is shown in Figure 2.

2.3 Mapper and Ball Mapper: plotting scalar-valued functions

Let G be a Mapper or Ball Mapper graph. Its vertices cover collections of points of the input point cloud X. Therefore, given a function $g : X \to \mathbb{R}$ we define an *induced function* $\hat{g} : G \to \mathbb{R}$, on G. The value of the induced function $\hat{g}$ on a vertex $v(a) \in G$ covering $A \subset X$ is an average value of g over all points in A: $\hat{g}(v(a)) = \frac{\sum_{x \in A} g(x)}{|A|}$. Since the structure of Mapper or Ball Mapper graph G reflects the shape of the input data, visualizing the induced function on G using a color scale provides an insight into the variation of a scalar–valued function g on X. This procedure can be seen as a generalization to high dimensional samples of standards techniques, such as heatmaps, which allow for visualization of a scalar function on a compact subset of $\mathbb{R}^2$ by means of a color scale. To confirm that the induced function $\hat{g}$ is a good approximation of the original function g one should check that the standard deviation of the values of g on each element of the cover is small enough compared to the value of the induced function. If this is the case, $\hat{g}$ can be used for further data analysis.

Mapper and Ball Mapper are data visualization tools that offer the initial understanding of the data and provide a way to state various hypotheses (see examples in Section 4.5). Due to

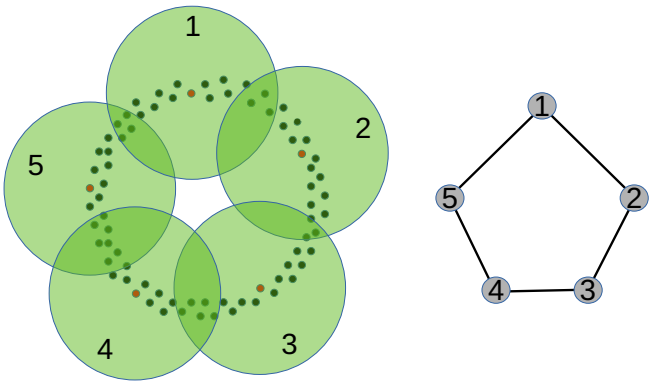


Figure 2: Ball Mapper construction illustration. Using the point cloud X in the Figure 1 we first select an ϵ–net (red points in the left figure). Construct balls of radius ϵ centered in all the points of the net (green). The union of those balls provides a cover of our data space. Ball Mapper gives a one dimensional nerve of the obtained cover. In more detail, to each ball $B(A, \epsilon)$ we assign a vertex $v(a)$ in the abstract graph $G = (V, E)$. Two balls $B(N, \epsilon)$ and $B(K, \epsilon)$ that jointly cover some points from X give rise to an edge between $v(N)$ and $v(K)$. The Ball Mapper graph of this cover is shown on the right with the corresponding balls and vertices labeled with the same number.

the stability issues, the Mapper and Ball Mapper graphs should be computed for a range of parameters and permutation of the input data, for example see Figure 7 and 8 and compared using MappingMappers or Mapper on Ball Mapper construction.

3 New developments of Mapper algorithms

3.1 Equivariant Ball Mapper

In this section we adapt the Ball Mapper algorithm to take into account the action of a finite automorphism group acting on our data (X, d) that lives in some metric space. Ideally, the sample contains the whole orbit of any of its points i.e. for every $h \in H$, $h(x) \in X$. If that is not the case, the data should be augmented to ensure that the sample reflects all the symmetries imposed by the group action.

Since H is an automorphism group, for every $x, y \in X$ and every $h \in H$, $d(x, y) = d(h(x), h(y))$.

For every point $x \in X$, and isometry $g \in H$ the *orbit* $\Omega_g(x)$ of x contains the sequence of points $x, h(x), h^2(x), \ldots, h^n(x) \in X$, where $h^{n+1}(x) = x$ for some $n \in \mathbb{N}$. The relation (Eqn. 1) for Jones polynomial data provides a dataset with a simple automorphism group generated by a permutation of the coordinates given by the exchange matrix.

Given a point cloud X and an automorphism group H acting on it, we modify the Ball Mapper algorithm in such a way that there is an induced action of H on the Ball Mapper graph G. The induced action is described in the following way: given a vertex $v(l) \in G$ consider $B(l, \epsilon) \cap X$, i.e. all the points in X covered by a ball corresponding to $v(l) \in G$. For every isometry $h \in H$ we require the "covering" condition:

- all the points in $h(B(l, \epsilon) \cap X)$ are covered by the ball $B(h(l), \epsilon)$ and
- there are no other points in this ball.

The vertex in G corresponding to the ball $B(h(l), \epsilon)$ is therefore denoted by $\hat{h}(v(l))$. Note that such $\hat{h}$, induced by h, is acting on an abstract graph, and therefore certain properties of h will not be reflected in $\hat{h}$. An example of this construction is given in Figure 3.

To ensure the "covering" condition described above is satisfied, the procedure of selection of ϵ-net $L \subset X$ is adjusted. In the Ball Mapper implementations L is chosen in the greedy way presented in Algorithm 1. The main idea is to add the whole orbit $\Omega(x) = \{h(x)\}_{h \in H}$ to the constructed set of landmark points together with the any added point x. This idea is formalized in the Algorithm 2, which adjusts Algorithm 1 so that the obtained ϵ-net L is invariant under the action of H. We refer to it as an H-equivariant ϵ-net.

Data: Point cloud X, $\epsilon > 0$, group H acting on X
Result: $L \subset X$, a H-equivariant ϵ-net
$L = \emptyset$
for $x \in X$ **do**
  if *x is farther than ϵ from every point in L* **then**
    $L = L \cup \Omega(x)$;
  end
end
return L

Algorithm 2: Equivariant greedy ϵ-net.

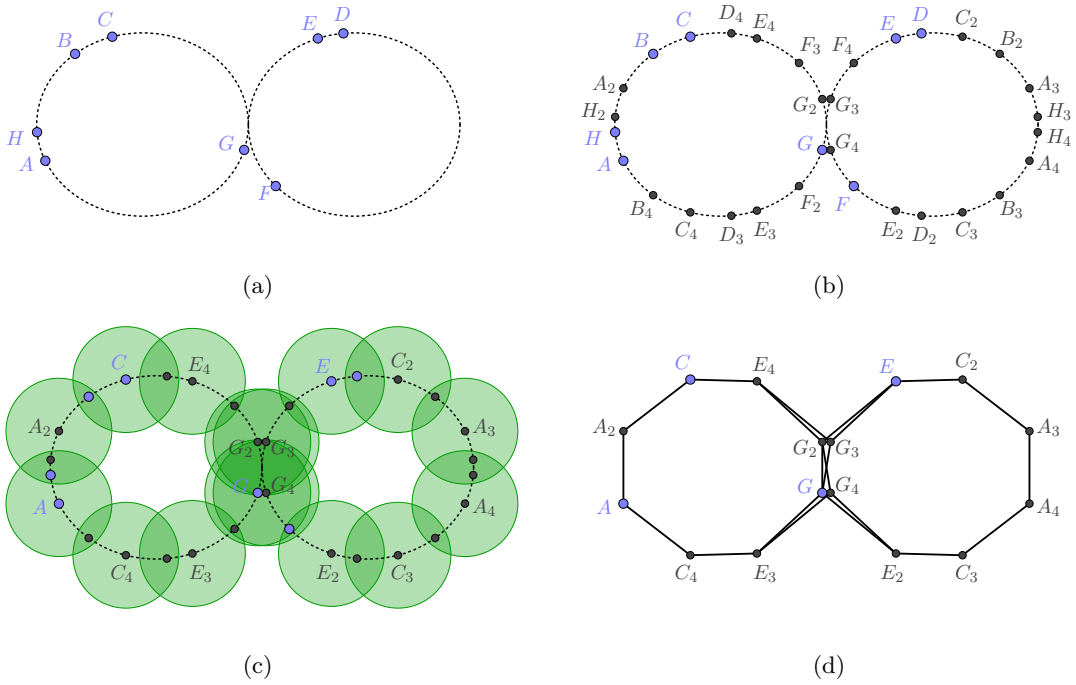


Figure 3: Example of the Equivariant Ball Mapper construction on a point cloud X sampled from a wedge of two circles (a) with a symmetry group determined by the reflections on the horizontal and vertical axis. The point cloud enriched by each point's orbit is shown in (b). Selected landmarks A, C, E and G, and all the points in their orbits that are selected as well are shown in (c) with a covering consisting of 16 balls. The resulting equivariant Ball Mapper graph is depicted in (d).

Since each $h \in H$ is an isometry, for every $l \in L$, if $y \in B(l, \epsilon) \cap X$, then $h(y) \in B(h(l), \epsilon) \cap X$. As each $B(l, \epsilon) \cap X$ corresponds to a vertex in the Ball Mapper graph, the action of H on X is induced to the vertices of the Ball Mapper graph as described above.

In Algorithm 2, the *for* loop iterates once through the points in X. The time needed to locate all points at a distance less than or equal to ϵ to a given point is bounded by $|X|$, hence the overall complexity of the presented procedure is bounded by $|X|^2$.

This equivariant construction is Ball Mapper specific and it is not clear that the analogous construction for Mapper is possible. For example, the equivariant Mapper construction requires the lens function to be either invariant with respect to the group action (i.e. points in the same orbit should obtain the same value of the function), or to map points from the input point cloud to different cover elements in such a way that there is an induced group acting on the cover. Adjusting the lens function to satisfy either of these requirements is non-trivial and also requires the clustering algorithm to be performed in such a way that there is an appropriate group action induced on the obtained clusters. These obstacles to obtaining equivariant Mapper emphasize the advantage of Ball Mapper with respect to equivariance and justify our choice.

3.2 Mapper on Ball Mapper

This section provides a new construction and practical generalization of the Mapper graph, as described in Section 2.1. Recall that the Mapper construction is based on a interval cover of the range of the lens function $f : X \to \mathbb{R}^n$. Typically $n = 1$ or another small positive integer, as the range of f needs to be covered with a collection of overlapping cubes, defined as a product of intervals. There is tension between wanting large n when the lens function is more likely to preserve essential information about the point cloud, and the fact that having k intervals in each of n directions requires k^n cover elements which is not computationally feasible for large values of n.

To overcome this obstacle we propose the following *Mapper on Ball Mapper (MoBM)* construction to leverage the overlapping, adaptive cover of the point cloud obtained from Ball Mapper. This algorithm applies to the more general setting with two point clouds X, Y and a relation $p \subset X \times Y$. MoBM is formalized by the following pseudocode:

Data: Point clouds X, Y, a relation $p \subset X \times Y$, $\epsilon > 0$, a clustering algorithm C

Result: MoBM graph for X, Y, p

Compute Ball Mapper $G(Y, \epsilon)$;

$\mathcal{C}_{\mathcal{X}} = \emptyset$;

for *each vertex* $v(l)$ *in* $G(Y, \epsilon)$ **do**

$X_v = \{x \in X \,|\, y \in B(l, \epsilon) \,\text{AND}\, (x, y) \in p\}$;

$\mathcal{C}_{\mathcal{X}} = \mathcal{C}_{\mathcal{X}} \cup C(X_v)$;

end

MoBM = 1-dimensional nerve of such $\mathcal{C}_{\mathcal{X}}$;

return MoBM

Algorithm 3: Mapper on Ball Mapper.

This construction, illustrated in Figure 4, requires only two parameters: the radius ϵ and the choice of the clustering algorithm. There is no need to define the number of intervals or the overlapping percentage as in the conventional Mapper algorithm; the covering of Y, being the range of the lens, is completely determined by the Ball Mapper graph. However, varying the selection of landmark points can lead to potentially different Ball Mapper graphs. Selection of different landmarks can be obtained by permuting the points of Y and re-running the algorithm. The time complexity of MoBM is bounded by the time required to run both Mapper and Ball Mapper.

3.3 MappingMappers

The standard construction of Mapper or Ball Mapper for a point cloud X with a metric $d : X \times X \to \mathbb{R}_{\geq 0}$ provides a model of the point cloud X. If the point cloud X is equipped with a function $f : X \to \mathbb{R}$, an induced function $\hat{f}$ can be defined on the Mapper or Ball Mapper graph G as explained in Sections 2.1 and 2.2. This way, Mapper and Ball Mapper graphs can be used to visualize functions $f : \mathbb{R}^n \to \mathbb{R}$, where f is defined on $X \subset \mathbb{R}^n$.

Consider a more general question of using Mapper and Ball Mapper to visualize functions $f : \mathbb{R}^n \to \mathbb{R}^m$ for larger values of m and n. Assume that we are given $X \subset \mathbb{R}^n$ and $Y \subset \mathbb{R}^m$ and a relation $f \subset X \times Y$ (note that a function is a particular case of such a relation). In this instance we focus on the Ball Mapper–based construction. A construction for Mapper is analogous. In the first step, let us build Ball Mapper graphs $G(X)$ and $G(Y)$ corresponding to point clouds X and Y and denote with $V(X)$ and $V(Y)$ the corresponding vertex sets. Given a relation $f \subset X \times Y$

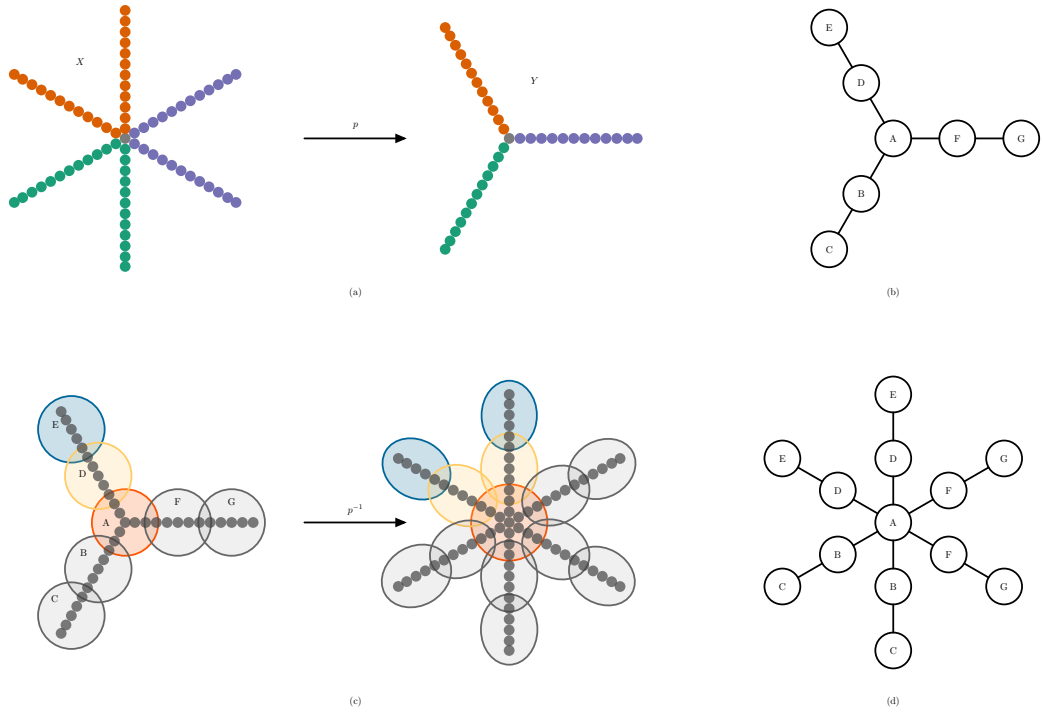
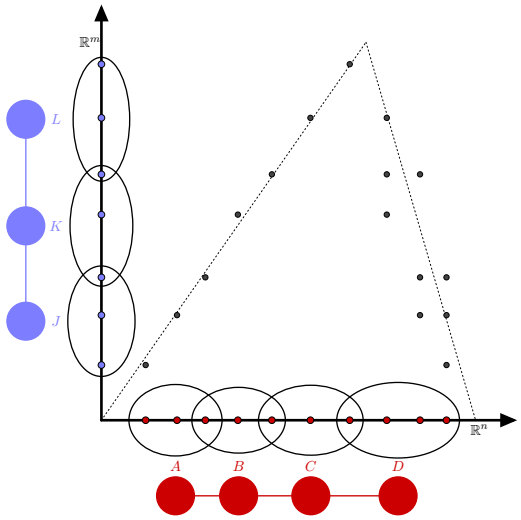


Figure 4: The Mapper on Ball Mapper construction: an illustration. The point cloud X is mapped via p to the point cloud Y (a) where the map is indicated by corresponding colors. The Ball Mapper graph for Y (b) is used to obtain a cover of Y (C, left). Such cover is then pulled-back to obtain a covering of X (C, right). The resulting Mapper on Ball Mapper graph for X is shown in (d), where, similarly to Figure 1, node labels indicate the originating covering ball in Y.

assigning points from X to the points from Y, define a map $\tilde{f} : V(X) \to [0,1]^{V(Y)}$ in the following way. For every v in $V(X)$, corresponding to a ball $B(l_X, \epsilon_X)$ compute $f(B(l_X, \epsilon_X) \cap X) \subset Y$. For every vertex w in $V(Y)$ corresponding to a ball $B(l_Y, \epsilon_Y)$, compute $\frac{|(B(l_Y,\epsilon_Y)\cap Y)\cap f(B(l_X,\epsilon_X)\cap X)|}{|B(l_Y,\epsilon_Y)\cap Y|}$. This fraction indicate the percentage of points in $B(l_Y, \epsilon_Y) \cap Y$ that are in the image of the points covered by the vertex v in $G(X)$. When computed for every vertex in $G(X)$, this fraction gives us $|V(X)|$ different coloring functions on $G(Y)$ indicating where the image of each vertex v is mapped.

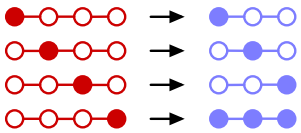
This construction works analogously for arbitrarily large unions of vertices of the graph $G(X)$. The procedure described above is automatized and a reliable interface can be found in (Dłotko et al., 2023). It provides a way to relate regions of $G(Y)$ to a chosen region of $G(X)$. By doing so, a visualization of the map $f : X \to Y$ is obtainticed. A simplified example of the procedure is given in Figure 5.



(a)

	J	K	L
A	1	0	0
B	0	1	0
C	0	0	1
D	1	1	1

(b)



(c)

Figure 5: Consider a relation f between $X \subset \mathbb{R}^n$ located on the x-axis and $Y \subset \mathbb{R}^m$ on the y-axis, represented by the black dots in the plane, see (a). Next, create Ball Mapper graphs on both domain X and co-domain Y. Set X is covered by four balls corresponding to the red vertices in the Ball Mapper graph, while Y, with the three balls corresponding to the blue vertices in the Ball Mapper graph. Based on the relation, the points in X covered by A are mapped into points covered by J in Y. Similarly, B into K, C into L. Lastly, the points in D in X are mapped to points in J, K and L. In addition, when we track the proportions of points in each of the ball J, K and L that are reached by points in A, B, C or D. Such proportions are indicated in the matrix (b). Each row in this matrix provides a coloring function on the image Ball Mapper graph (c). This idea generalizes and provides a way to visualize maps between point clouds in high dimensional spaces.

4 Applications: knot theory

Analyzing and visualizing data from knot theory is the main motivation and inspiration for the development of algorithms presented in Sections 3.1, 3.3 and 3.2. This section discusses the basics of knot theory and the data used in the our analysis, followed by the results obtained using Equivariant Ball Mapper, Mapper on Ball Mapper, and MappingMappers.

4.1 Knot theory: a brief introduction

A knot is a class of embeddings of S^1 into $\mathbb{R}^3$ up to ambient isotopy (Livingston, 1993; Lickorish, 2012; Jablan and Sazdanovic, 2007). Knots are hard to distinguish and their classification and tabulation (Hoste et al., 1998; Hoste, 2005; Bar-Natan and Morrison, 2011–2019; Livingston and Moore, 2022; Burton, 2018) solicits techniques from a range of mathematical disciplines. Consequently, a number of *knot invariants* have been introduced in an attempt to compare and classify knots. A knot invariant should be thought of as a quantity assigned to each knot such that if two knots are the same (isotopic), the values assigned to them are the same. The most common knot invariants are integers, one- or two-variable polynomials, groups, etc. In this paper we focus on the following knot invariants and their relations:

- Polynomial: Alexander $\Delta(K)(t)$ (Alexander, 1928), Jones $J(K)(q)$ (Jones, 1985), HOMFLYPT $P(K)(a,z)$) (Freyd et al., 1985).
- Numerical: minimal crossing number, signature $\sigma(K)$ (Kauffman and Taylor, 1976) defined as a signature of a matrix obtained using a Seifert surface, determinant of a knot $det(K) = |\Delta(K)(-1)| = |J(K)(-1)|$.

Since knot invariants often rely on advanced algebraic, geometric, and combinatorial topology, in lieu of definitions we provide references (Livingston, 1993; Rolfsen, 2003; Lickorish, 2012; Jablan and Sazdanovic, 2007), and key insights sufficient for analyzing these knot invariants and their relations. Note that Alexander and Jones polynomials are 1-variable polynomials, while HOMFLYPT polynomial is of two variables. Moreover, HOMFLYPT is more general than both the Alexander and Jones polynomial. HOMFLYPT specializes to the Jones polynomial (Jones, 1985) by substituting $a = t^{-1}$ and $z = \sqrt{t} - \frac{1}{\sqrt{t}}$, and to the Alexander polynomial (Alexander, 1928) by substituting $a = 1$ and $z = \sqrt{t} - \frac{1}{\sqrt{t}}$.

Note that it is common for knot tables to contain only one knot from each mirror pair $(K, mir(K))$, where the mirror of a knot K is the knot $mir(K)$ whose diagram is obtained by changing all the crossings in a diagram of K. A knot K is achiral if it coincides with its mirror $mir(K)$, and many invariants either do not distinguish between mirrors or have a straightforward relation between the two values. For example, the Alexander polynomial does not distinguish then while the Jones

Invariant	Unknot	Trefoil	Data vector
Alexander	1	$t^{-1} - 1 + t$	(0,1,-1,1,0)
Jones	1	$q + q^3 - q^4$	(0,0,0,0,0,1,0,1,-1)
HOMFLYPT	1	$-a^4 + 2a^2 + a^2z^2$	(0,0,0,0,0,0,2,0,-1,**0,0,0,0,0,0,1,0,0**)

Table 1: Values of several knot polynomials for the unknot and the trefoil, and the corresponding data vector. Note that for the 2-variable HOMFLYPT polynomial the coefficient matrix is flattened into a vector: the variable z corresponding to rows, a to columns.

polynomial of a knot and its mirror (Lickorish, 2012; Jones, 1985) satisfies the following relation:

$$J(mir(K))(q) = J(K)(q^{-1}). \tag{1}$$

The signature of a knot and its mirror have opposite signs (Lickorish, 2012),

$$\sigma(mir(K)) = -\sigma(K). \tag{2}$$

and for the HOMFLYPT polynomials of mirror knots the following relation holds:

$$P(mir(K))(a, z) = P(K)(a^{-1}, z) \tag{3}$$

4.2 Knot invariants as point clouds

The use of big data techniques (Hughes, 2020; Jejjala et al., 2019; Levitt et al., 2019; Gukov et al., 2021, 2023) is warranted by the result of Ernst and Sumners (1987) showing that the number of knots with a given number of crossings grows exponentially. The point clouds are obtained in the way introduced and described in Levitt et al. (2019), where each knot is represented by a vector of coefficients of a knot polynomial such as the Alexander, Jones or HOMFLYPT polynomial. The datasets we consider were created by Levitt et al. (2019), and preprocessed by D. Gurnari. The data is freely available at Dłotko et al. (2023) and includes Alexander and Jones polynomials, together with numerical knot invariants like minimal crossing number and signature for all 9755329 knots up to 17 crossings. HOMFLYPT polynomials are provided for all 313231 knots up to 15 crossings.

Following Levitt et al. (2019), given a finite collection of knots $\mathcal{K}$, we construct a point cloud $\mathcal{I}(\mathcal{K})$ corresponding to the coefficients of the one-variable polynomial invariant $\mathcal{I}$, in the following way:

Step 1 Given a knot $K \in \mathcal{K}$ and its single variable polynomial $I(K)$, extract a vector of the coefficients.

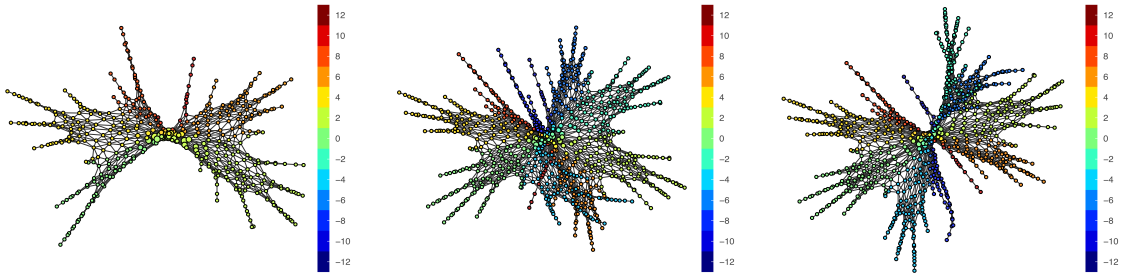
Step 2 Compute the minimal and maximal powers min_t, max_t of the variable denoted by t among all knots in $\mathcal{K}$. Then the maximal length of all considered vectors is $d = max_t - min_t + 1$.

Step 3 Add zeros on both sides of each vector of coefficients to obtain a vector $I(K)_v \in \mathbb{R}^d$ to ensure a correct alignment of corresponding powers.

Note that in this way all vectors are of the same length determined by the overall minimum and maximum exponent, and the coefficients of a given power are in the same position in the vector for all the considered knots. In case of a two-variable polynomial, such as the HOMFLYPT polynomial, we apply Steps 1-3, as described above in the case of one-variable polynomial, to both variables. In this way we first obtain a matrix padded with zeros, and then create a corresponding vector by linearizing this matrix (concatenating its rows). HOMFLYPT data belongs to $\mathbb{R}^d$ where $d = (max_a - min_a + 1)(max_z - min_z + 1)$ where a and z stand for the two variables in the corresponding polynomial. Examples of the coefficient vectors obtained from both the unknot and the trefoil are presented in Table 1.

Unlike some databases, we choose to consider knots and their mirrors although that increases the dimension of the coefficient point cloud or the size of the coefficient vectors. The Jones coefficient vector of its mirror is obtained by reversing the original vector, see relation (1), and therefore the point cloud admits a symmetry given by the exchange matrix. The HOMFLYPT coefficient matrix of the mirror knot can be obtained by reversing the columns of the original matrix (3). Hence, in the case of the Jones polynomial $min_t(mir(K)) = -max_t(K)$ and $max_t(mir(K)) = -min_t(mir(K))$ as a consequence of relation (1) and the point cloud belongs to $\mathbb{R}^d$ where $d = 2max(|max_t|, |min_t|)+1$.

The size of the obtained tables of polynomial coefficients are the following: 9755329 rows $\times$ 17 columns for Alexander, 19510658 rows $\times$ 51 columns for Jones and 626462 rows $\times$ 152 columns for HOMFLYPT. Note that the number of rows in Jones and HOMFLYPT data is double the number of prime knots since mirrors are also included.

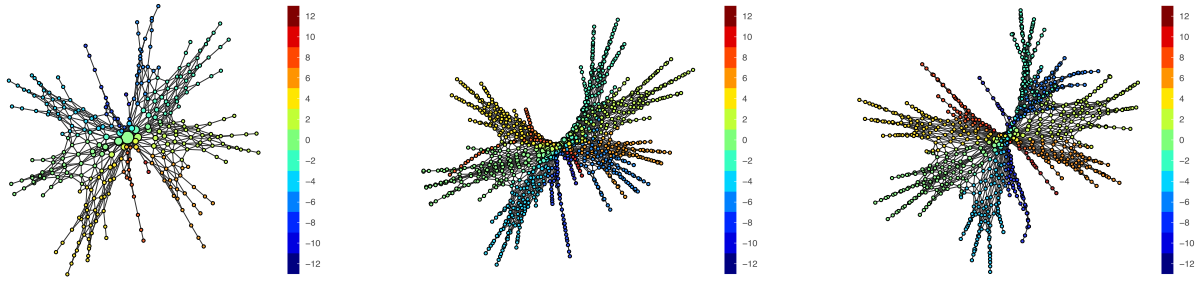


(a) (b) (c)

Figure 6: Ball Mapper applied to Jones polynomial data of knots up to 17 crossings with (a) just one choice of a mirror, (b) knots and their mirrors with standard Ball Mapper, and (c) Equivariant Ball Mapper. Color reflects the average signature of knots in each cluster. Note that the graph in (c) is symmetric, although this fact is not accurately represented in this image due to the chosen graph plotting subroutine.

4.3 Ball Mapper: structure of knot polynomial data

In this section we apply standard and Equivariant Ball Mapper to data obtained from Jones, Alexander and HOMFLYPT polynomials for all knots up to 17 crossings. The choice to use Ball Mapper is natural, as there is no obvious lens function for the Mapper construction.



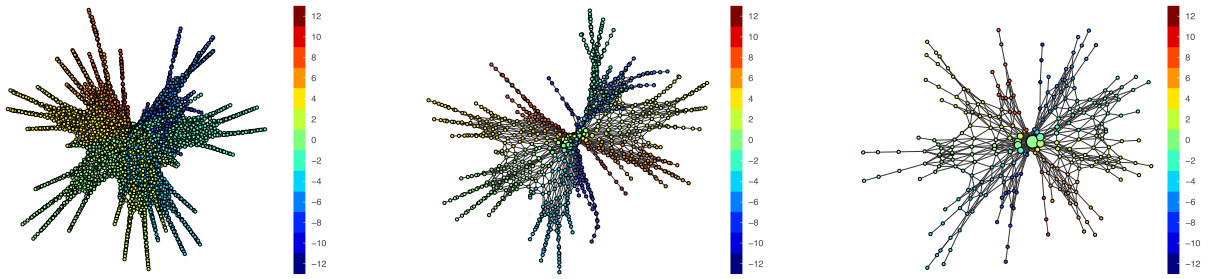
(a) (b) (c)

Figure 7: Stability of Equivariant Ball Mapper for Jones polynomial data with respect to the crossing number filtration: (a) EqBM graphs of knots up to 15 crossings $\epsilon = 30$ with 826 nodes, (b) 16 with $\epsilon = 50$ with 1008 nodes, and (c) 17 crossings with $\epsilon = 100$ with 890 nodes.

Since our data contains knots and their mirror images, the Jones polynomial data cloud admits a symmetry generated by the permutation of the coordinates for those knots which are not identical to their mirrors (see Eqn.(1)). Figure 6 shows Ball Mapper graphs with just one of the mirrors included

6a, knots and their mirrors with the standard Ball Mapper 6b and the equivariant Ball Mapper construction from Section 3.1 in 6c. The symmetry of the data is preserved by the equivariant Ball Mapper: for each flare there is an identical one with opposite signature, as a consequence of relation (2).

The structure of the BM graph is stable with respect to the filtration by the number of crossings, as illustrated in Figure 7, in addition to stability across the choice of parameter/radius shown as shown in Figure 8. This observation is important since sampling knot data is a hard problem and it is known that knots with lower crossing number do not provide a sample representative of the space of knots (Levitt et al., 2019).



(a) (b) (c)

Figure 8: Stability of Equivariant Ball Mapper with respect to the choice of parameter/radius ϵ: (a) EqBM graphs of knots up to 17 crossings $\epsilon = 50$ with 3840 nodes, (b) $\epsilon = 100$ with 890 nodes, and (c) $\epsilon = 200$ with 254 nodes. Color is determined by the average signature of knots in that node/cluser.

The Ball Mapper graph for the Alexander polynomial data has linear structure, see Figure 9b. The two flares contain clusters of knots whose signature modulo 4 is equal to zero or two, respectively. Similarly, the Ball Mapper graphs for HOMFLYPT data exhibit a star-like structure whose flares contain knots with the same signature, Figure 9c. As in the case of the Jones polynomial data, all Ball Mapper graphs are stable with respect to the crossing number filtration and the choice of Ball Mapper parameter ϵ.

4.4 MappingMappers and Mapper on Ball Mapper: comparing knot invariants

In this section, we further investigate data corresponding to the Jones, Alexander, and HOMFLYPT polynomials whose Ball Mapper graphs are provided in Section 4.3. The main goal is to compare

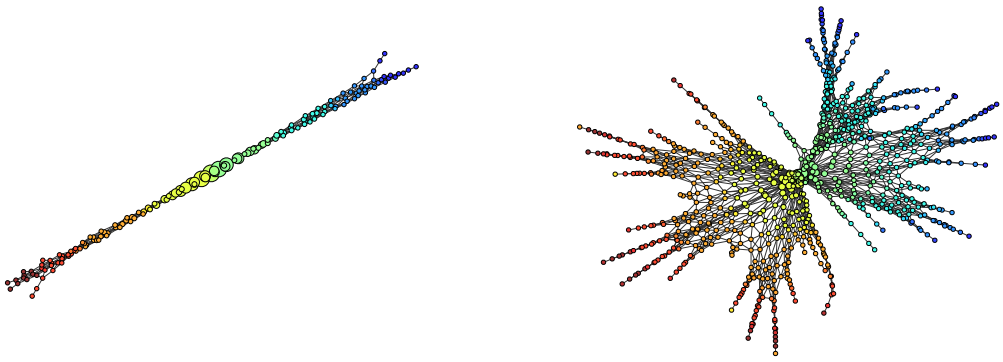
(a) (b) (c)

Figure 9: Equivariant Ball Mapper graphs for the (a) Jones, (b) Alexander, and (c) HOMFLYPT polynomial data of knots up to 17 crossings colored by the average signature of knots in each cluster (a), (c) or signature mod 4 in (b).

these spaces using Ball Mapper based tools developed in Sections 3.3 and 3.2, in order to, implicitly, compare the invariants. First, spaces of two invariants can be compared by constructing their Ball Mapper graphs and visualizing the maps between them as described in Section 3.3. Next, Mapper on Ball Mapper construction from Section 3.2 is used to emphasize relative strengths of two invariants with respect to distinguishing knots. To be more specific, given two invariants A and B, in general data descriptors, of a dataset $\mathcal{K}$ we can think of them as maps $A : \mathcal{K} \to M_A$ and $B : \mathcal{K} \to M_B$, where M_A and M_B are metric spaces. Most commonly, M_A and M_B are finite point clouds in Euclidean spaces of different dimensions. Inspired by comparison of knot invariants, that invariant A is considered to be *stronger* than invariant B if the elements covered by a single vertex or several closely-connected vertices in the Mapper on Ball Mapper graph of M_B are spread across different regions of the Mapper on Ball Mapper graph of M_A. MappingMappers and Mapper on Ball Mapper can be used for this type of analysis.

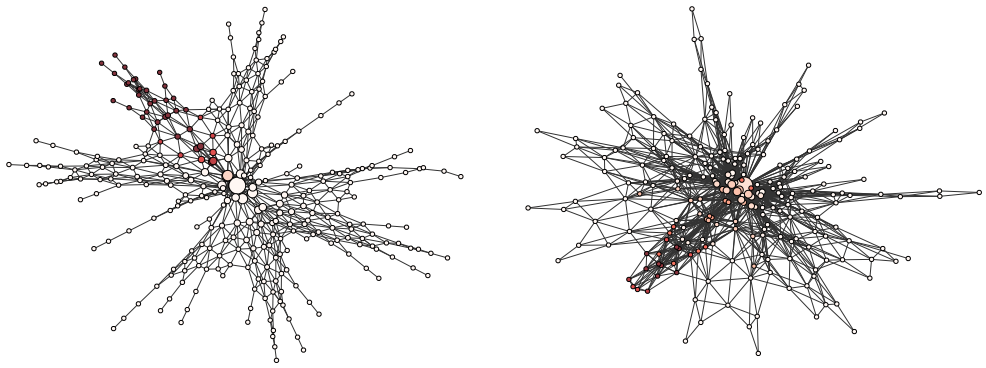
MappingMappers, defined in Section 3.3, uses two point clouds $X \subset \mathbb{R}^n$ and $Y \subset \mathbb{R}^m$ and a relation $f \subset X \times Y$ as inputs. To illustrate this technique we use the collection $\mathcal{K}$ of knots up to 17 crossings along with the Jones $J(\mathcal{K}) \subset \mathbb{R}^{51}$ and the Alexander $A(\mathcal{K}) \subset \mathbb{R}^{17}$ point clouds obtained in Section 4.2.

Using the set of knots $\mathcal{K}$ as the common indexing set, we define the relation $f \subset A(\mathcal{K}) \times J(\mathcal{K})$ in the following way: for a given knot $K \in \mathcal{K}$ its Alexander polynomial $A(K)$ in $A(\mathcal{K})$ is related to its Jones polynomial $J(K)$ in $J(\mathcal{K})$. Note that this is a relation rather than a function, since some knots from $\mathcal{K}$ may have the same Alexander polynomial, but different Jones polynomials,



(a) (b)

Figure 10: MappingMappers: Representation of a map from the space of the Alexander (a) and the space of the Jones polynomials of knots up to 17 crossings (b). Rainbow coloring of consecutive clusters of the linear embedding of the Equivariant Ball Mapper graph for the Alexander data and used to color the corresponding regions of the Equivariant Ball Mapper graph for the Jones data.



(a) (b)

Figure 11: MappingMappers: Visualizing the map from the space of Jones polynomials to the space of HOMFLYPT polynomials of knots up to 15 crossings using the Ball Mapper graph (containing 326 nodes) of Jones polynomial with $\epsilon = 50$ (a) and the Ball Mapper graph (containing 258 nodes) of HOMFLYPT polynomial of with $\epsilon = 50$ shown in (b). All clusters containing knots with signature equal to zero in both Ball Mapper graphs are shown as a shade of red.

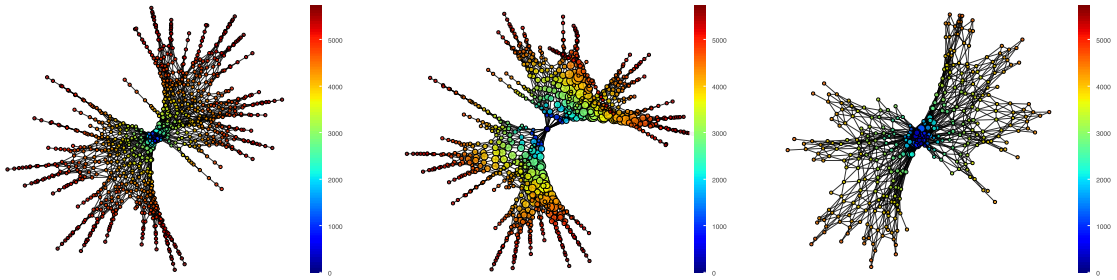
and vice versa. In this case, a single point from $A(\mathcal{K})$ can be related to multiple points of $J(\mathcal{K})$ or the other way around. Figure 10 illustrates this relation: colors indicate matching regions in the Alexander and Jones Ball Mapper graphs. Roughly speaking, the linear structure of the Alexander

Ball Mapper induces the linear structure among the flares in the star-like Ball Mapper graph of the Jones data. In the opposite direction, flares of the Jones Ball Mapper merge according to the signature modulo 4 which is consistent with the fact that one end of the Alexander Ball Mapper contains knots whose signature mod 4 is zero and the other one is two. Figure 11 illustrates the analogous relation between the Jones and HOMFLYPT Ball Mapper graphs, respectively. The non-linear nature of the Jones Ball Mapper graph prohibits using a gradient-like coloring; instead, the cluster color in this graph reflects the percentage of knots with signature equal to zero and the corresponding clusters in the HOMFLYPT Ball Mapper graph.

4.5 New insights into relations between knot invariants

The following section features applications of methods introduced in this paper to knot theory: exploratory analysis and hypothesis formulation.

4.5.1 Jones polynomial, determinant and signature of a knot



(a) (b) (c)

Figure 12: Equivariant Ball Mapper graphs for the Jones polynomial data of (a) all, (b) alternating and (c) non-alternating knots up to 17 crossings colored by the average determinant of knots in each cluster.

The star-like embedding of the Ball Mapper graph of the Jones polynomial data in Figure 8 suggests that the flares tend to be monochromatic when colored by values of knot signature. On the other hand, Figure 12 indicates that the values of the determinant (as well as the norm of the Jones polynomial) increases along the flares as one moves away from the center. Consequently, we consider the hypothesis that for knots with sufficiently large determinant (or the norm of its

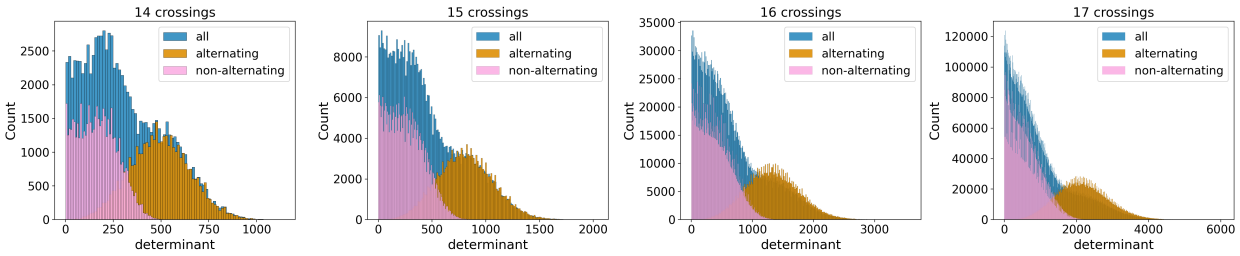


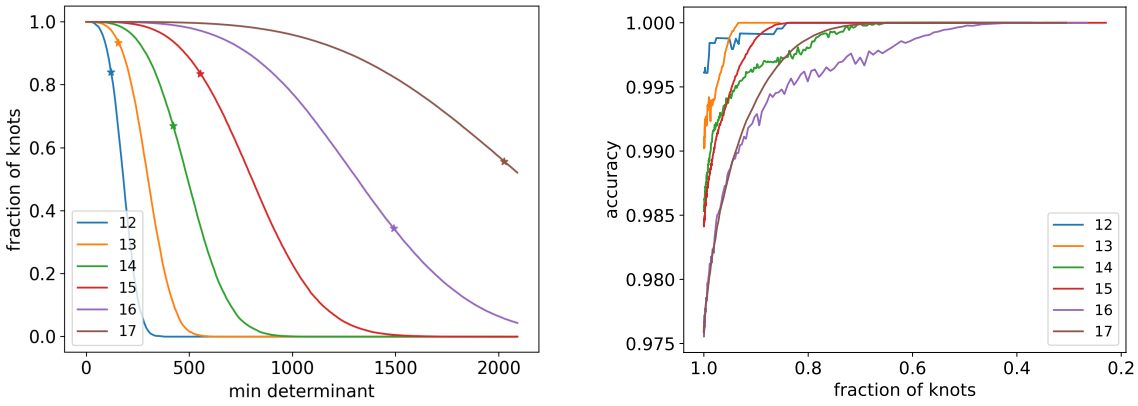
Figure 13: Distribution of the absolute value of the determinant knots with 14, 15, 16 and 17 crossings for all (blue), alternating (orange), and non-alternating (pink) knots.

Jones polynomial), the coefficients of the Jones polynomial determine the knot's signature. This assumption is necessary, since for 385822 out of 19510658 (1.97%) knots up to 17 crossings there exists at least one other knot with identical Jones polynomial but different signature.

Since there exist non-trivial non-alternating knots whose whose determinant equals 1, the determinant of the unknot, but which are not unknots e.g. torus knots T(p,q) with prime p and q, we restrict this hypothesis to alternating knots. Restricting the hypothesis to alternating knots is further supported by the distribution of determinants and norms of the Jones polynomial, as shown in Figure 13. For alternating knots it resembles a normal distribution while the one for non-alternating is concentrated at smaller values and then tapers off Levitt et al. (2019).

To confirm the hypothesis for alternating knots with minimal crossing number $n \in \{12, 13, \ldots, 17\}$ we are looking for $r_n > 0$, $n \in \{12, 13, \ldots, 17\}$ such that after removal of all the n-crossing knots of determinant smaller than r_n, linear Support Vector Machine (Bishop and Nasrabadi, 2006) (a technique known to be resistant to overfitting) is capable to predict the knot's signature from the coefficients of its Jones polynomial with 100% accuracy. Our results together with the information about the distribution of determinants of n-crossing alternating knots are summarized in Figure 14. The stars indicate the minimal value of the knot determinant above which 100% accuracy in predicting the signature from the coefficients of Jones polynomial is achieved.

To provide a further evidence for our hypothesis we normalize the coefficients of the Jones polynomials of the n–crossing knots with determinant larger than r_n. Next, we apply the principal component analysis to the normalized data and project it to the two principal directions, see Figure 15b. The obtained projection suggests existence of conical regions in the ambient space consisting solely of knots with equal signature which correspond to monochromatic arcs in the



(a) (b)

Figure 14: (a) Fraction of alternating n–crossing knots for $n \in \{12, \ldots, 17\}$ with determinant above a certain threshold. The stars indicate the minimal value of the determinant for which the knot's signature can be determined from its Jones polynomial with 100% accuracy using a linear SVM classifier. (b) Accuracy of the linear SVM classifier in predicting the knot's signature from its Jones polynomial as a function of the fraction of considered knots. The minimal determinant for each fraction can be extracted from (a).

PCA projection.

4.5.2 Knot signature mod 4

In this section we illustrate the potential of the tools introduced in this paper and big data analysis approach in theoretical mathematics in general and formulating or (re)discovering theorems. Let us reconsider the Ball Mapper graph in Figure 9b. The distribution of signature values suggests that signature mod 4 can be determined as a function of the coefficients of the Alexander polynomial. This hypothesis was tested by training a linear Support Vector Machine classifier (Bishop and Nasrabadi, 2006). According to it, the perfect separation between two classes of knots, those whose signature mod 4 is zero and those with it equal to 2, is achieved with an "anti-diagonal" hyperplane with normal vector $[1, -1, 1, -1, \ldots]$. This observation indicates that the sign of the alternating sum of the coefficients of the Alexander polynomial determines the signature mod 4 , which is the well-known Theorem 6.4.7 in Murasugi (2008). This Theorem also applies to the Jones polynomial but the SVM fails to converge to an optimal solution on the Jones data. However, combining the

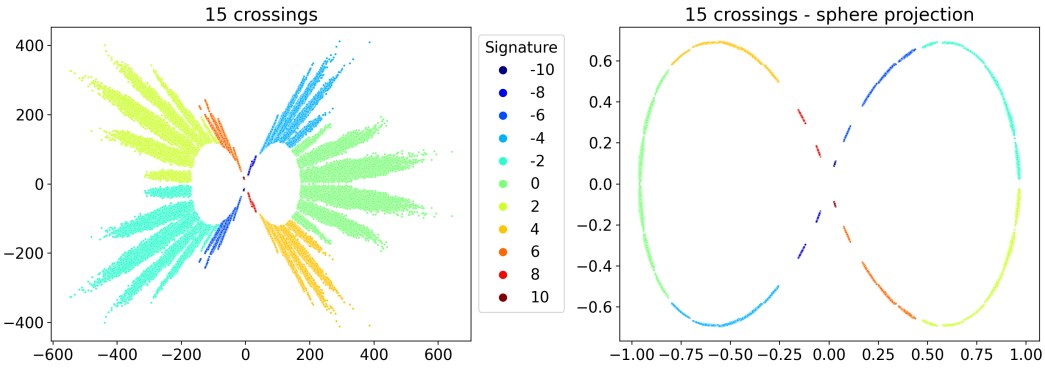


Figure 15: (left) 2-dimensional PCA projection of the coefficients of Jones polynomials for alternating knots with 15 crossing with determinant greater than 553. (right) 2-dimensional PCA projection for the normalized Jones coefficients of the same set of knots.

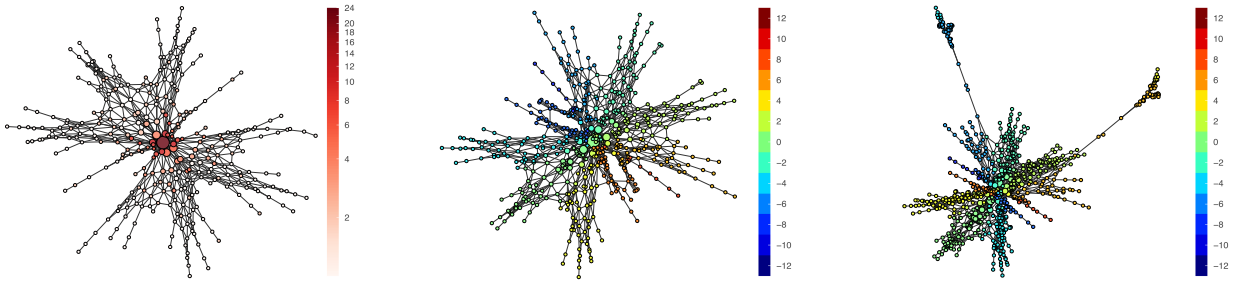
SVM on Alexander and the MappingMappers correspondence between Alexander and Jones data (see Figure 10) recovers the Theorem for the Jones polynomial. Our approach provides a new way of obtaining such a Theorem and paves the way for using mapper-type algorithms to aid discovery in knot theory in particular, and theoretical mathematics and sciences in general.

4.5.3 Relation between HOMFLYPT and Jones polynomials

In this section we apply MoBM to the algebraic relation between the HOMFLYPT polynomial and both the Alexander and Jones polynomials, see Section 4.1. These specializations tie in with the framework introduced in Section 3.2 as they can be used as lenses for Mapper algorithm when using the Mapper on Ball Mapper, or MoBM, construction. As a clustering method in the MoBM construction, we use the DBSCAN algorithm (Ester et al., 1996). DBSCAN requires a new parameter ϵ_{DB} in addition to the ϵ denoting the radius of balls used in the Ball Mapper construction.

The MoBM construction using knots data is illustrated in Figure 16: Ball Mapper on Jones data Figure 16a is used as the input covering to the MoBM graph 16b the HOMFLYPT data whose Ball Mapper is in Figure 11b. The coloring in Figure 16a represents the number of clusters into which the points in each node split when they are pulled back from the space of Jones to HOMFLYPT. This pullback is not trivial (clusters split into more than one cluster in the pre-image) both in the center region and in between the flares. The obtained MoBM graph thus achieves better separation

in those regions. Unlike Ball Mapper graph for HOMFLYPT data, this MoBM graph has two flares that consist of knots with the same s-invariant and different signature. This observation touches on the open question when s-invariant and signature differ as we know that they coincide for alternating knots but not much more is known.



(a) (b) (c)

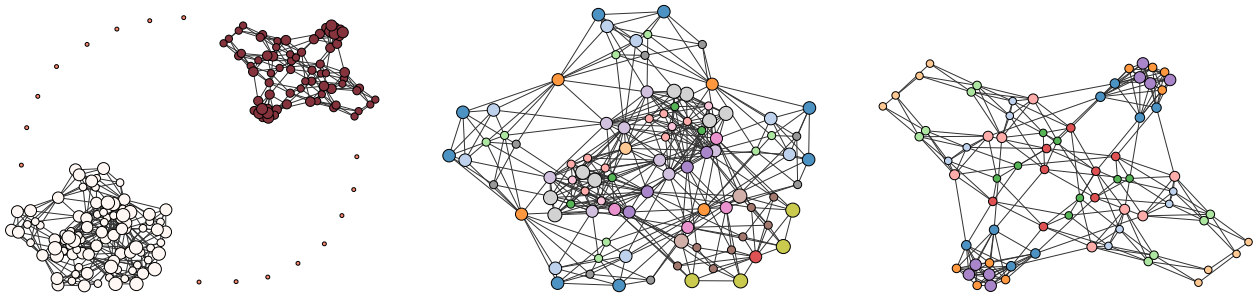
Figure 16: The Mapper on Ball Mapper graph on Jones-HOMFLYPT data pair. Image (a) shows the Ball Mapper graph for Jones data for knots up to 15 crossings at $\epsilon_{BM} = 50$ with a total of 326 nodes colored by the number of clusters found in each node. The Mapper on Ball Mapper construction in (b) is obtained from the Jones Ball Mapper (a) and the HOMFLYPT data for the DBSCAN clustering algorithm parameter equal $\epsilon_{DB} = 40$ with 644 nodes, and colored by signature. Analogous MoBM construction for $\epsilon_{DB} = 30$ reveals a different structure on HOMFLYPT data (c): the two long flares emerging in (c) consist knots with the same s-invariant (Rasmussen, 2010), but different values of signature.

5 Equivariant Ball Mapper on Tic-Tac-Toe data

Any dataset with a nontrivial isometric group action can be visualized with preservation of the action using Equivariant Ball Mapper. As an example we analyze the Tic-Tac-Toe endgame dataset (Dua and Graff, 2017) that consists of all possible board configurations at the end of Tic-Tac-Toe games. These configurations, 958 in total, are represented by 3 by 3 matrices (interpreted as vectors in $\mathbb{R}^9$) since the game is played between two players on a 3 by 3 grid where the first player places crosses and the second places noughts. The winner is the first player who places three noughts or crosses in a vertical, horizontal or a diagonal line.

The input for our analysis consists of 3 by 3 grids, interpreted as vectors with 9 entries, with

values -1 (corresponding to a nought), 0 (corresponding to an empty slot) or 1 (corresponding to a cross). Moreover, we use the l_1 norm as distance function which sets the distance between an empty cell and a filled in cell to 1, and the distance between a cross and a nought in the same cell to 2. This is consistent with the rules of the game, as symbols can only be placed on empty cells and replacing the symbol in an already filled-in cell is not allowed. The symmetries of the 3×3 configurations are given by a dihedral group consisting of four rotations and four reflections. It is straightforward to see that all configurations in one orbit are all wins, loses or ties, since rotation and reflection of the board does not change the outcome of the game. These symmetries induce relations between vectors in $\mathbb{R}^9$ and the Euclidean distance between any two configurations and their images via one of the actions will be the same. Hence, Equivariant Ball Mapper is the natural choice for this data; see the resulting Ball Mapper graph in the Figure 17a.



(a) (b) (c)

Figure 17: The Equivariant Ball Mapper graph for the Tic-Tac-Toe dataset. Figure (a) shows the game outcome: the wins clusters are colored white, the losses clusters red.

The sixteen isolated orange clusters correspond to all possible ties. The wins and losses clusters are shown in (b) and (c) respectively, with color denoting orbits. Note that even if the same color palette is used, there is no relation between the nodes in (b) and (c) as the dihedral symmetry does not change the outcome of a game.

While our purpose is to showcase the technique rather than to draw a conclusion about the output, we include several observations. The Equivariant Ball Mapper graph for radius $\epsilon = 2.5$ (using l_1 distance) provides a perfect separation between the configurations in which the first player wins the game (white), loses the game (red) or when there is a tie (orange), see Figure 17a. We attribute the clear separation of win-loss-tie clusters, as well as the symmetries of the win clusters (Figure 17b) and loss clusters (Figure 17c), to the combinatorial properties of the game. In panels

(b) and (c) nodes belonging to the same orbit are colored with the same color. Note that different orbits might have different lengths. A configuration with no rotation or reflection symmetry will lead to a length 8 orbit. On the other hand, a configuration that already has some symmetries will have a shorter orbit. The maximally symmetric configuration has an orbit of length 1. This corresponds to the only red node in the bottom right of Figure 17b. Intuition on the perfect win-loss-tie separation can be built by considering the smallest l_1 distance between configurations. It is not difficult to show that the distance between winning and losing configurations is at least 3. Similarly, the minimum distance between two ties is 4 and the minimum distance between winning (resp. loosing) configuration and a tie is 4 (resp. 3). On the other hand, any pair of winning (resp. losing) configurations can be connected by a sequence of winning (resp. loosing) configurations spaced by at most 2. In lieu of the proof, consider a Ball Mapper graph with radius $2 < \epsilon < 3$, like the one depicted in Figure 17a, where all the winning (resp. losing) configurations are in the same connected component evidencing the existence of such a path.

6 Discussion

The main contributions of this paper consist of new Mapper-inspired algorithms that expands applicability and utility of standard TDA tools for exploring high-dimensional datasets. Equivariant Ball Mapper allows to encode the symmetries of the input point cloud, thus returning a more faithful graph representation. Mapper on Ball Mapper provides a new, flexible way of obtaining an overlapping cover of the lens function's domain, overcoming the conventional Mapper's limitation that excludes lens functions with high-dimensional ranges. Finally, MappingMappers is a new visualization tool that enables immediate and effective comparisons between Mapper graphs. In depth applications to knot theory underscore the potential of combining the introduced methods with quantitative statistical and machine learning methods to gain insights on complex datasets. Results are accompanied by Python implementations of all new algorithms, designed to ensure easy utilization by researchers and data science practitioners.

7 Supplementary Materials

Dataset and Python code Jupyter notebooks containing code to perform the experiments described in the article. The same files can be retrieved from the following Zenodo record Dłotko et al. (2023). The package also contains all datasets used as examples in the article. (knotsBM.zip, zipped file).

8 Acknowledgements

PD and DG acknowledge support by Dioscuri program initiated by the Max Planck Society, jointly managed with the National Science Centre (Poland), and mutually funded by the Polish Ministry of Science and Higher Education and the German Federal Ministry of Education and Research. DG is also with the University of Warsaw within the Doctoral School of Exact and Natural Sciences. RS is partially supported by NSF grant DMS-1854705. The authors declare no competing interests.

References

Alexander, J. W. (1928), "Topological invariants of knots and links," *Transactions of the American Mathematical Society*, 30, 275–306.

Bar-Natan, D. and Morrison, S. (2011–2019), "The Knot Atlas," URL `http://katlas.org`. `http://katlas.org`.

Bishop, C. M. and Nasrabadi, N. M. (2006), *Pattern recognition and machine learning*, volume 4, Springer.

Brown, A., Bobrowski, O., Munch, E., and Wang, B. (2021), "Probabilistic convergence and stability of random mapper graphs," *J. Appl. Comput. Topol.*, 5, 99–140.

Burton, B. A. (2018), "The next 350 million knots," `http://regina-normal.github.io/data.html`. `http://regina-normal.github.io/data.html`.

Carrière, M., Michel, B., and Oudot, S. (2018), "Statistical Analysis and Parameter Selection for Mapper," *Journal of Machine Learning Research*, 19, 1–39.

Davies, A., Veličković, P., Buesing, L., Blackwell, S., Zheng, D., Tomašev, N., Tanburn, R., Battaglia, P., Blundell, C., Juhász, A., et al. (2021), "Advancing mathematics by guiding human intuition with AI," *Nature*, 600, 70–74.

Dua, D. and Graff, C. (2017), "UCI Machine Learning Repository," URL `http://archive.ics.uci.edu/ml`.

Dłotko, P. (2019), "Ball mapper: a shape summary for topological data analysis," *arXiv:1901.07410*.

Dłotko, P., Gurnari, D., and Sazdanovic, R. (2023), "Mapper-type algorithms for complex data and relations - datasets and code," URL `https://doi.org/10.5281/zenodo.7669610`. `https://doi.org/10.5281/zenodo.7669610`.

Ernst, C. and Sumners, D. W. (1987), "The growth of the number of prime knots," *Mathematical Proceedings of the Cambridge Philosophical Society*, 102, 303–315, URL `https://www.cambridge.org/core/journals/mathematical-proceedings-of-the-cambridge-philosophical-society/article/growth-of-the-number-of-prime-knots/B03EBE9A337FA60207A40DC68D490B09`.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996), "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, Portland, Oregon: AAAI Press.

Freyd, P., Yetter, D., Hoste, J., Lickorish, W. R., Millett, K., and Ocneanu, A. (1985), "A new polynomial invariant of knots and links," *Bulletin (new series) of the American mathematical society*, 12, 239–246.

Gukov, S., Halverson, J., Manolescu, C., and Ruehle, F. (2023), "Searching for ribbons with machine learning," URL `http://arxiv.org/abs/2304.09304`. ArXiv:2304.09304 [cs, math].

Gukov, S., Halverson, J., Ruehle, F., and Sułkowski, P. (2021), "Learning to unknot," *Machine Learning: Science and Technology*, 2, 025035, URL `https://dx.doi.org/10.1088/2632-2153/abe91f`.

Haussler, D. and Welzl, E. (1987), "ϵ-nets and simplex range queries," *Discrete & Computational Geometry*, 2, 127–151, URL https://doi.org/10.1007/BF02187876.

Hoste, J. (2005), "The enumeration and classification of knots and links," in *Handbook of knot theory*, Elsevier, 209–232.

Hoste, J., Thistlethwaite, M., and Weeks, J. (1998), "The first 1,701,936 knots," *The Mathematical Intelligencer*, 20, 33–48.

Hughes, M. C. (2020), "A neural network approach to predicting and computing knot invariants," *Journal of Knot Theory and Its Ramifications*, 29, 2050005.

Jablan, S. V. and Sazdanovic, R. (2007), *LinKnot: knot theory by computer*, volume 21, World Scientific.

Jejjala, V., Kar, A., and Parrikar, O. (2019), "Deep learning the hyperbolic volume of a knot," *arXiv preprint arXiv:1902.05547*.

Jones, V. F. R. (1985), "A polynomial invariant for knots via von Neumann algebras," *Bull. Amer. Math. Soc. (N.S.)*, 12, 103–111.

Kauffman, L. H. and Taylor, L. R. (1976), "Signature of links," *Transactions of the American Mathematical Society*, 216, 351–365.

Lee, Y., Barthel, S. D., Dłotko, P., Moosavi, S. M., Hess, K., and Smit, B. (2017), "Quantifying similarity of pore-geometry in nanoporous materials," *Nature Communications*, 8, 15396. Number: 1 Publisher: Nature Publishing Group.

Levitt, J. S., Hajij, M., and Sazdanovic, R. (2019), "Big data approaches to knot theory: Understanding the structure of the Jones polynomial," *arXiv preprint arXiv:1912.10086*.

Li, L., Cheng, W.-Y., Glicksberg, B. S., Gottesman, O., Tamler, R., Chen, R., Bottinger, E. P., and Dudley, J. T. (2015), "Identification of type 2 diabetes subgroups through topological analysis of patient similarity," *Science Translational Medicine*, 7, 311ra174 – 311ra174.

Lickorish, W. R. (2012), *An introduction to knot theory*, volume 175, Springer Science & Business Media.

Livingston, C. (1993), *Knot theory*, volume 24, Cambridge University Press.

Livingston, C. and Moore, A. H. (2022), "KnotInfo: Table of Knot Invariants," URL: `knotinfo.math.indiana.edu`.

Munch, E. and Wang, B. (2016), "Convergence between Categorical Representations of Reeb Space and Mapper," *SOCG*.

Murasugi, K. (2008), *Knot Theory & Its Applications*, Boston, MA: Birkhäuser, URL `http://link.springer.com/10.1007/978-0-8176-4719-3`.

Nicolau, M., Levine, A. J., and Carlsson, G. (2011), "Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival," *Proceedings of the National Academy of Sciences*, 108, 7265–7270.

Qiu, W., Rudkin, S., and Dłotko, P. (2020), "Refining understanding of corporate failure through a topological data analysis mapping of Altman's Z-score model," *Expert Systems with Applications*, 156, 113475.

Rasmussen, J. (2010), "Khovanov homology and the slice genus," *Inventiones mathematicae*, 182, 419–447.

Reeb, G. (1946), "Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique," *C. R. Acad. Sci.*, 222, 847–849.

Rolfsen, D. (2003), *Knots and links*, volume 346, American Mathematical Soc.

Singh, G., Mémoli, F., and Carlsson, G. E. (2007), "Topological methods for the analysis of high dimensional data sets and 3d object recognition." *PBG@ Eurographics*, 2.