Austrian Journal of Statistics
2026, Volume 55, Issue 3, 91–112.
http://www.ajs.or.at/

Clustering Persistence Barcodes of Multidimensional Scaling Representations from Random Initial Configurations

Melinda A. Kleczynski
National Institute of
Standards and Technology
Gaithersburg, MD, USA

Anthony J. Kearsley
National Institute of
Standards and Technology
Gaithersburg, MD, USA

Abstract

Multidimensional scaling (MDS) is a popular technique for exploring complex datasets. A common application is examination of collections of objects equipped with pairwise dissimilarities. Iterative optimization produces points in Euclidean space whose pairwise distances approximate the pairwise dissimilarities between the original data objects. For a single dataset, starting at different initial configurations during optimization may produce MDS representations with substantially different structural features. This presents both challenges and opportunities for those who use these methods. We use persistence barcodes, a descriptor from topological data analysis (TDA), to reveal clusters of multidimensional scaling representations. By applying our methods to datasets consisting of photographs of rotating objects, we uncover robust and interpretable but relatively uncommon MDS configurations which have higher stress than many representations obtained through random initialization. These interpretable configurations may be easily missed in standard analysis. We are optimistic that the results described in this manuscript will encourage further consideration of MDS in cases when preliminary configurations returned for a dataset do not depict known structural features.

Keywords: MDS, initial configuration, configurational similarity, topological data analysis.

1. Introduction

Exploratory analysis is difficult for many modern datasets. The data format may be high-dimensional, and the dissimilarities between objects may be non-Euclidean. A common strategy is to obtain a reasonable representation of the data in a low-dimensional Euclidean space. Representations in 2-dimensional space are especially useful for visualization.

Multidimensional scaling (MDS) is a widely used technique which attempts to represent data objects as points in a (typically low-dimensional) Euclidean space (Borg and Groenen 2005; Borg, Groenen, and Mair 2018). We consider datasets equipped with pairwise dissimilarities between objects. The n data objects can be described by the $n \times n$ dissimilarity matrix

$\Delta = (\delta_{ij})$ satisfying

$$\begin{aligned}\delta_{ii} &= 0\\ \delta_{ij} &\geq 0\\ \delta_{ij} &= \delta_{ji}\end{aligned}$$

for $i, j = 1, \ldots, n$. The goal is to obtain a configuration of points whose pairwise distances approximate the given pairwise dissimilarities.

Performing MDS involves identifying coordinates X minimizing some quantity, frequently stress (Kruskal 1964). If δ_{ij} is the original dissimilarity between data objects i and j, and $d_{ij}(X)$ is the Euclidean distance between the corresponding points in the target low-dimensional space, then we seek to minimize the stress,

$$\sigma(X) = \sum_{i<j} (\delta_{ij} - d_{ij}(X))^2. \tag{1}$$

Since pairwise dissimilarities differ in magnitude for different applications, it may also be of interest to consider the related quantity

$$\sigma_n(X) = \frac{\sum_{i<j} (\delta_{ij} - d_{ij}(X))^2}{\sum_{i<j} \delta_{ij}^2}, \tag{2}$$

which may be referred to as normalized stress (Borg and Groenen 2005, p. 248) or explicitly normalized stress (Rusch, Mair, and Hornik 2021). It is also possible to assign different weights to different terms in the summation, but we don't consider this generalization here.

In practice, it is standard to obtain configurations with low stress through iterative optimization. The goal is to return configurations which are near local minimizers, a process which may be sensitive to the starting point used for optimization (Kearsley, Tapia, and Trosset 1998). There may be several local minimizers with similar stress values and qualitatively different MDS representations; this makes consideration of multiple initial configurations valuable for visualization (Borg and Mair 2017). There is interest in considering MDS configurations which may actually have higher stress, but better represent the underlying structure of the dataset (Borg 2020; Mair, Borg, and Rusch 2016).

In this work, we analyze datasets whose points are approximately parameterized by a single periodic variable. In many cases, standard MDS pipelines effectively reveal the circular structure of a dataset. Examples of previous such applications are MDS of color similarity data (Borg and Groenen 2005; Borg and Mair 2017), simulated fMRI data (Ellis, Lesnick, Henselman-Petrusek, Keller, and Cohen 2019), and rotating synthetic data (Li, Storm, Li, Needham, and Wang 2023). For datasets such as those considered in the current work, MDS representations may not always capture known properties, such as dependence on a rotation angle.

Structures such as loops and tunnels are readily detectable using topological data analysis (TDA) (Carlsson 2009), raising the possibility that we may be able to use techniques from applied topology to characterize MDS results. MDS has frequently been used in combination with TDA, but often with performing MDS as one of the initial steps in TDA pipelines. This has produced successful results for analyzing data from single-cell RNA sequencing (Rizvi, Camara, Kandror, Roberts, Schieren, Maniatis, and Rabadan 2017), cardiac resynchronization therapy patients (Veres, Schwertner, Tokodi, Szijártó, Kovács, Merkel, Behon, Kuthi, Masszi, Gellér, Zima, Molnár, Osztheimer, Becker, Kosztin, and Merkely 2024), and the human gut microbiome (Lymberopoulos, Gentili, Alomari, and Sharma 2021). MDS has also been applied to visualize completed topological summaries to study tumor microenvironments (Stolz, Dhesi, Bull, Harrington, Byrne, and Yoon 2024), chaotic versus periodic dynamics (Myers,

Chumley, Khasawneh, and Munch 2023), and antibody conformations (Kleczynski, Bergonzo, and Kearsley 2025).

When comparing MDS results, we typically want to consider two configurations as being essentially the same if they differ only by actions such as translation, rotation, and reflection. One approach is to enforce constraints which fix one or more of these aspects of the MDS configuration (Groenen 1993; Kearsley *et al.* 1998). Another option is to perform MDS as usual, but process the final configurations using Procrustean transformations to mitigate differences between MDS configurations which are due to these effects (Borg and Leutner 1985; Borg and Mair 2017, 2022).

A final approach is to use features which can be obtained from the matrix of pairwise Euclidean distances between MDS points (Borg and Leutner 1985). This matrix is unaffected by the actions of translation, rotation, and reflection (Lele and Richtsmeier 2001). We employ the final option. This work proposes topological data analysis as an extension of this family of techniques for MDS characterization. We demonstrate the success of this approach through an application to two commonly analyzed image datasets.

In this work, we use topological data analysis to identify interpretable MDS representations which have substantially different structure than the minimum stress MDS representation identified. Our approach is to obtain MDS representations from many random initial configurations. We quantify the topological structure of each MDS representation using two barcode coordinates. We cluster these barcode coordinates, and view the minimum stress MDS representation from each cluster, rather than just the minimum stress result across all the initial configurations. This approach yields interesting MDS representations. By either selecting a single most interpretable MDS representation, or viewing the set of MDS representations as an ensemble, one can gain a fuller understanding of the structure of a dataset. We focus on datasets which have a strong signature of a single persistent loop, but we mention alternative choices for topological quantification that make the overall approach suitable for a wider range of datasets.

2. Topological data analysis (TDA) background

We begin with a brief overview of the relevant techniques from TDA. A thorough discussion of TDA is beyond the scope of this work, but can be found in many other texts (Chazal and Michel 2021; Dey and Wang 2022; Ghrist 2014). We restrict our attention to analysis of finite sets of data points equipped with pairwise distances between points. As with MDS, the input data can be described by a dissimilarity matrix. In the current work, we only consider Euclidean distances. For other analysis pipelines, more general distances or dissimilarities can also be used for topological analysis.

Given such a dataset and a fixed distance value ϵ, we form a simplicial complex by connecting sets of data points with pairwise distances at most ϵ. This is a particular type of simplicial complex called a Vietoris–Rips complex (Ghrist 2008); as this is the only type of simplicial complex used in the current work, we can use the more general term without ambiguity. Examples of simplicial complexes are shown in the top row of Figure 1.

We can think of a Vietoris–Rips complex as a collection of vertices and a collection of simplices. Each vertex corresponds to an object in the dataset of interest. In the simplicial complexes in Figure 1, vertices correspond to points in the MDS representation. For the simplicial complexes that generate the persistence barcodes in Figure 2, vertices correspond to photographs at various object rotations. We can think of simplices as connections. In a Vietoris–Rips complex, a set of vertices is connected by a simplex when the corresponding data objects have pairwise distances at most ϵ. In this work, we only compute dimension 1 persistent homology, so the key components of the Vietoris–Rips complex are 1-simplices and 2-simplices. 1-simplices connect pairs of vertices; in Figure 1, these are drawn as black edges. 2-simplices connect trios of vertices; in Figure 1, these are drawn as gold triangles.

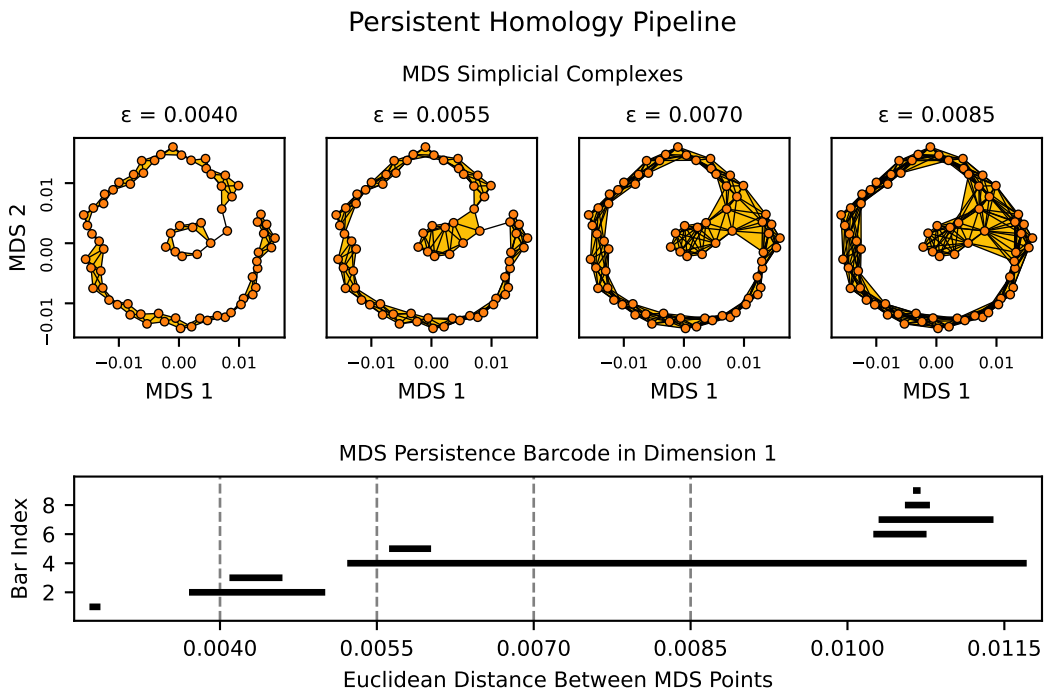

Figure 1: In a common TDA pipeline, we form simplicial complexes by connecting sets of points whose pairwise distances are at most ϵ. For data points which are sampled from a closed loop, many choices of ϵ will generate a simplicial complex which encircles an empty center region. Barcodes track topological features across values of ϵ. The long bar in this barcode signals that the corresponding MDS representation has a persistent closed loop structure.

The simplicial complexes in Figure 1 capture structural features of the MDS representation. In this case, each simplicial complex encircles a single empty region, reflecting the fact that the points in the MDS representation are arranged in a loop. In the first simplicial complex, for $\epsilon = 0.0040$, there is a small loop at the center of the MDS representation, representing a local feature involving a smaller number of points in the MDS representation. The remaining simplicial complexes form a larger loop corresponding to global structure of the MDS representation.

A key strategy in topological data analysis is to consider the persistence of features with respect to ϵ. This is accomplished through the use of persistent homology (Otter, Porter, Tillmann, Grindrod, and Harrington 2017; Zomorodian and Carlsson 2005). Informally, there is a wide interval of values of ϵ for which the simplicial complex formed from the data in Figure 1 will enclose a large empty center region. The exact intervals of ϵ corresponding to loop structures at different distance scales are recorded in a topological summary called a dimension 1 persistence barcode (Ghrist 2008). Simplicial complexes also generate persistence barcodes for other dimensions. For example, persistence barcodes in dimension 0 track connected components for varying values of ϵ, analogous to hierarchical clustering. Persistence barcodes in dimension 2 track enclosed volumes. In this work, we only consider persistence barcodes in dimension 1. Thus, we may refer to them simply as persistence barcodes or just barcodes for convenience.

In Figure 1, we show the barcode for the MDS representation of the tomato dataset. It is also possible to generate barcodes for the original image datasets; the barcode of the scaled Euclidean distances between images for the tomato dataset is shown in the top plot of Figure 2. Both barcodes contain a long interval, or bar. This is an indication of the overall loop structure of the MDS representation and the dataset from which the MDS representation was generated.

In order to obtain persistence barcodes using standard software, the user provides a dissimilar-

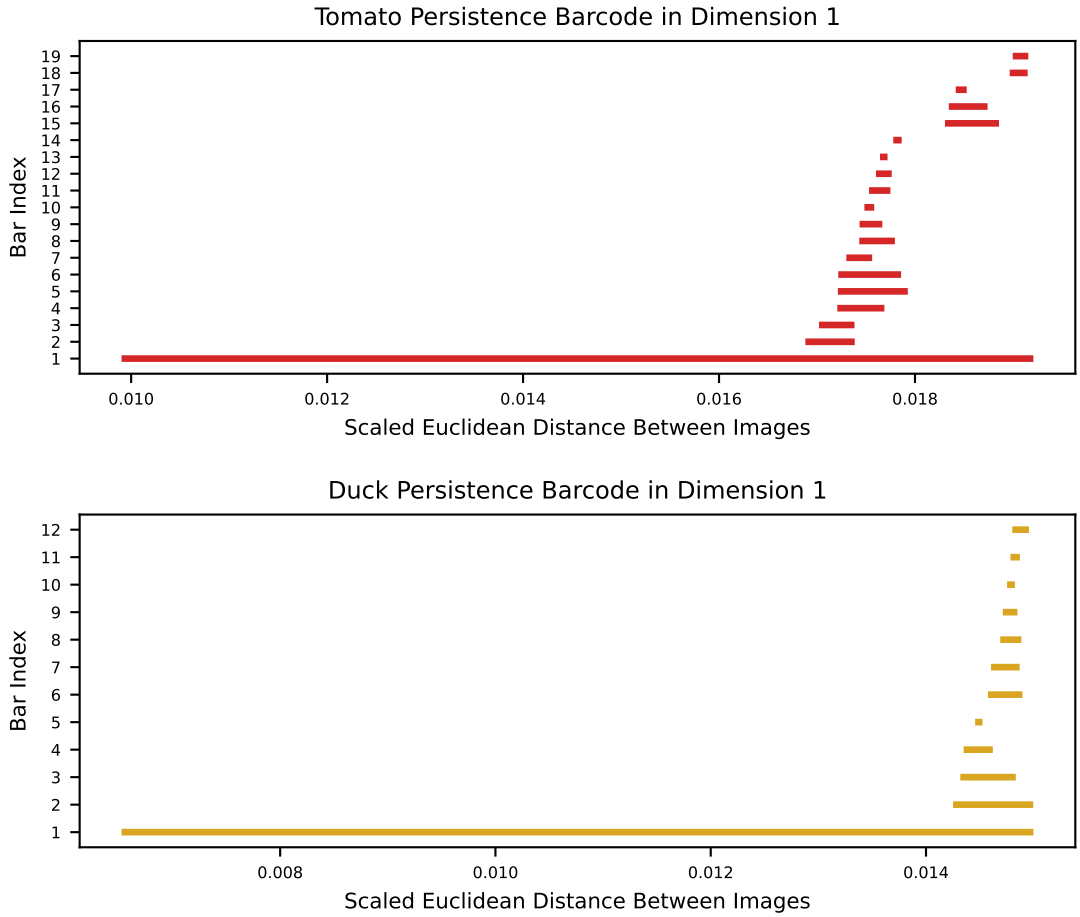

Figure 2: We obtain persistence barcodes of the same distance matrices which are used for performing MDS

ity matrix or collection of point coordinates. Algorithms for computing persistent homology track the relevant simplicial complexes, and output persistence barcodes. Figure 3 shows four example datasets and the corresponding persistence barcodes in dimension 1. The first dataset, consisting of points arranged in a circle, generates a single long interval, or bar, in the persistence barcode. The second dataset, which has an additional, smaller circle adjacent to the original circle, has an additional, smaller bar in the barcode. Persistent homology is ideally suited for detecting multiscale features such as these. The third dataset contains nested circles. The smaller circle takes up part of the interior of the larger circle, shortening the bar corresponding to the larger circle. The fourth dataset consists of points arranged in a circle, but with added noise. There is still a long bar indicating the overall closed loop structure of the datset, but there are additional, shorter bars generated by smaller loops among nearby points. In this case, we consider the smaller bars to be noise.

A persistence barcode is a multiset of intervals, or bars. Each bar gives a range of values of ϵ for which a particular feature is present. Barcodes of similar datasets can have different numbers of bars. For example, the circle dataset in Figure 3 generates a persistence barcode with one bar. However, the noisy circle dataset has a persistence barcode with four bars. The bars can be assigned convenient indices for plotting; for example, the bars can be ordered based on the appearance of each feature (the left endpoint of the bar).

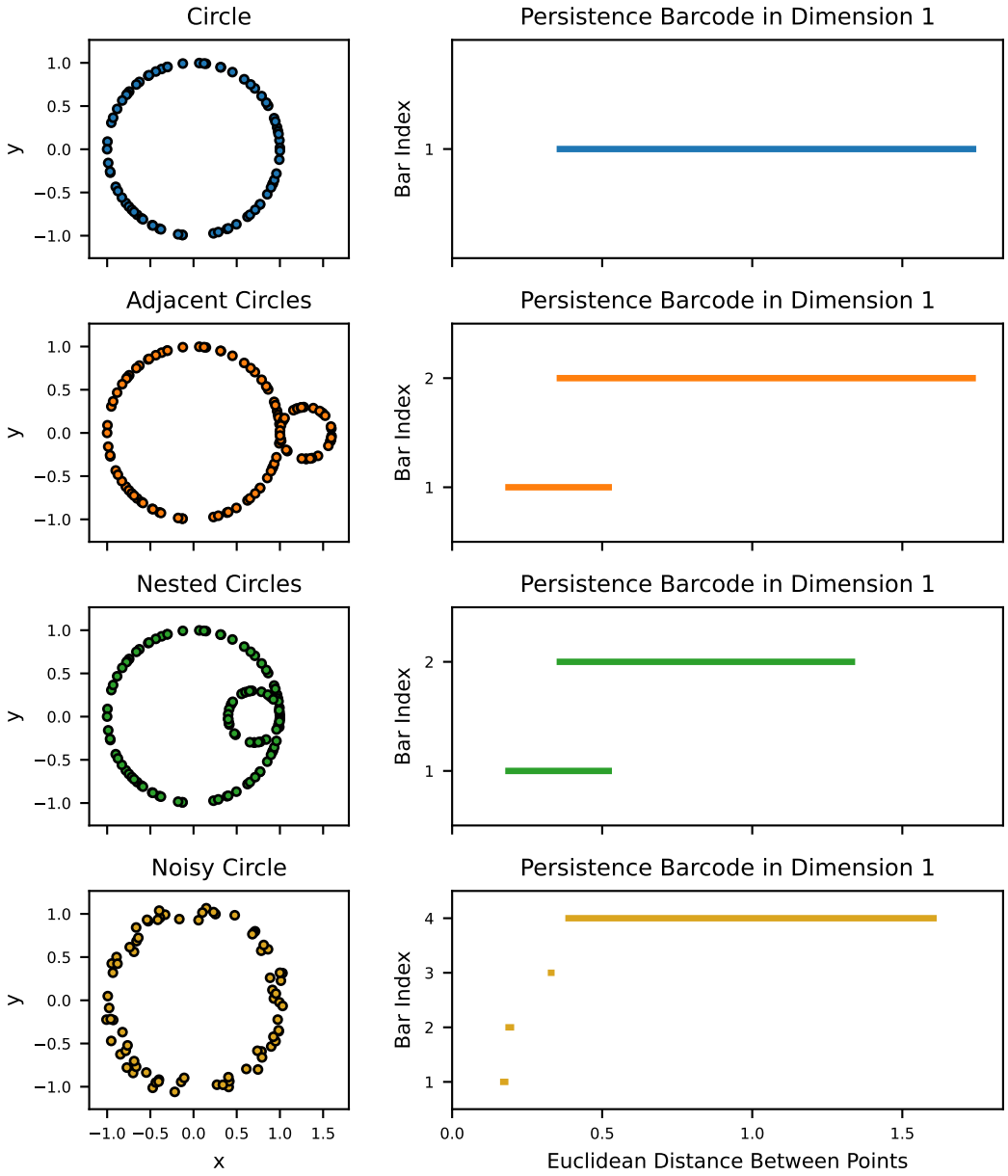

Figure 3: Four example datasets and their persistence barcodes in dimension 1

3. Methods

In this section, we discuss the software and data used in the current work. We provide the specifications we use for producing random initializations and performing MDS. Finally, we outline the additional analysis involved in characterizing and clustering the MDS representations.

The following is an overview of the steps involved. We will describe each step in more detail in this section.

1. Process the dataset to obtain a pairwise dissimilarity matrix.
2. Repeat the following for many random states.
   (a) Generate a random initial configuration.
   (b) Perform iterative optimization to obtain the corresponding MDS representation.
   (c) Record the stress of the MDS representation.
   (d) Obtain the persistence barcode of the MDS representation.
   (e) Compute barcode coordinates from the persistence barcode.
3. Use the barcode coordinates to cluster the MDS representations.
4. View the minimum stress MDS representation from each cluster.
5. Select the final MDS representation(s) for dataset visualization or additional analysis.

3.1. Software

All analysis is performed in Python 3.13.2 (Python Software Foundation 2025).[1] The packages used are summarized in Table 1. We use the scikit-learn 1.7.0 (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot, and Duchesnay 2011) implementations of MDS, DBSCAN, and Gaussian Mixture Models (GMMs). For topological data analysis, we use Open Applied Topology (OAT) (Henselman-Petrusek, Hang, Ziegelmeier, and Giusti 2024c) with backend oat_rust (Henselman-Petrusek, Hang, Ziegelmeier, and Giusti 2024b) accessed through oat_python 0.1.1 (Henselman-Petrusek, Hang, Ziegelmeier, and Giusti 2024a). We also use the packages Matplotlib 3.10.1 (Hunter 2007) for plotting, NumPy 2.2.4 (Harris, Millman, van der Walt, Gommers, Virtanen, Cournapeau, Wieser, Taylor, Berg, Smith, Kern, Picus, Hoyer, van Kerkwijk, Brett, Haldane, del Río, Wiebe, Peterson, Gérard-Marchant, Sheppard, Reddy, Weckesser, Abbasi, Gohlke, and Oliphant 2020) for handling arrays and numerical computations, and SciPy 1.15.2 (Virtanen, Gommers, Oliphant, Haberland, Reddy, Cournapeau, Burovski, Peterson, Weckesser, Bright, van der Walt, Brett, Wilson, Millman, Mayorov, Nelson, Jones, Kern, Larson, Carey, Polat, Feng, Moore, VanderPlas, Laxalde, Perktold, Cimrman, Henriksen, Quintero, Harris, Archibald, Ribeiro, Pedregosa, van Mulbregt, and SciPy 1.0 Contributors 2020) for computing pairwise distances and performing least-squares optimization.

Table 1: Python packages used in our implementation

Package	Version	Role in analysis
Matplotlib	3.10.1	plotting
NumPy	2.2.4	arrays and numerical computations
oat_python	0.1.1	persistence barcodes
scikit-learn	1.7.0	MDS, clustering
SciPy	1.15.2	pairwise distances, least-squares optimization

[1]Certain equipment, instruments, software, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement of any product or service by NIST, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

3.2. Data

We analyze two Columbia Object Image Library (COIL) datasets (Nene, Nayar, and Murase 1996a,b). The COIL datasets have been used for demonstrating many TDA techniques, such as Mapper (Carrière, Michel, and Oudot 2018), circular coordinate recovery (Perea 2020; Schonsheck and Schonsheck 2024), and optimization using topological losses (Carrière, Theveneau, and Lacombe 2024; Clémot, Digne, and Tierny 2025; Nelson and Luo 2022). Each dataset consists of 72 images of an everyday object, with the object rotated at a different angle in each image. This can generate a closed loop structure due to the proximity of images with similar rotation angles. The COIL-20 (processed) library (Nene *et al.* 1996b) consists of 128 pixel by 128 pixel grayscale PNG images which we read as 128×128 numerical arrays. The COIL-100 library (Nene *et al.* 1996a) consists of 128 pixel by 128 pixel RGB color PNG images which we read as $128 \times 128 \times 3$ numerical arrays.

Overviews of the datasets included in our analysis are given in Table 2. Throughout the text, we refer to them simply by their descriptions. We compute Euclidean distances between images. More sophisticated dissimilarities may produce better results for image handling, but the simple distances we consider are sufficient for the analysis in this work. Throughout the manuscript, we frequently use the term "dissimilarities" to refer to distances between images, to distinguish these from the distances between MDS points, even though we compute Euclidean distances in both cases.

Table 2: The two rotating image datasets included in the analysis

Dataset	Library	Object ID
tomato	COIL-100 Nene *et al.* (1996a)	4
duck	COIL-20 Nene *et al.* (1996b)	1

3.3. Preprocessing

For each image, we divide by the maximum value in the corresponding array. The resulting image arrays have values between 0 and 1, inclusive. Numerical values in this work are dimensionless quantities obtained from these normalized image arrays.

The original indices of the images are directly related to the image rotation, so as a matter of principle we shuffle the order of the images prior to performing MDS. In many applications, the ground truth ordering of data objects within a closed loop structure may not be known. We only use the original image order when we plot selected MDS representations, to draw connections between MDS points representing adjacent rotations. In these plots, each point in the MDS representation corresponds to an image. We draw edges between points if the corresponding images depict adjacent angles of rotation of the subject (the tomato or the duck).

Normalized stress (Equation (2)) is obtained from stress (Equation (1)) by dividing by $\sum_{i<j} \delta_{ij}^2$, where δ_{ij} is the dissimilarity between images i and j. In our case, the dissimilarities between images do not have informative units. We immediately multiply the unprocessed matrix of pairwise image dissimilarities by a scalar to obtain a dissimilarity matrix $\Delta = (\delta_{ij})$ such that

$$\sum_{i<j} \delta_{ij}^2 = 1. \tag{3}$$

These are the dissimilarities with which we perform TDA and MDS. Consequently, stress and normalized stress are equal. For simplicity of exposition, we will simply use the term stress throughout.

3.4. Random initialization

We use the technique outlined by Kearsley *et al.* (1998), which we review here. Our input is an $n \times n$ matrix $\Delta = (\delta_{ij})$ of dissimilarities (in our analysis, $n = 72$ and δ_{ij} is the dissimilarity between images i and j). We aim to find an MDS representation in Euclidean space of dimension p (in our analysis, $p = 2$ or $p = 3$). We set

$$\begin{aligned} m &= n(n-1)/2, \\ S &= \sum_{i<j} \delta_{ij}^2, \\ \sigma &= \sqrt{S/(2pm)}. \end{aligned}$$

Note that due to Equation (3), this simplifies to

$$\begin{aligned} m &= n(n-1)/2, \\ S &= 1, \\ \sigma &= \sqrt{1/(2pm)}. \end{aligned}$$

For $i = 1, ..., n$, pseudorandom values $\tilde{x}_i$ and $\tilde{y}_i$ (and $\tilde{z}_i$ when applicable) are each drawn from the univariate standard normal distribution, respectively. Then the coordinates of the points in the initial configuration are $x_i = \sigma\tilde{x}_i$ and $y_i = \sigma\tilde{y}_i$ (and $z_i = \sigma\tilde{z}_i$ when applicable), $i = 1, ..., n$. The advantage of generating initial configuration points using this method is that the expected squared distance between points is equal to the average squared dissimilarity of the input data.

3.5. MDS

For all examples, we input precomputed pairwise dissimilarities and output low-dimensional MDS representations. For the main analysis, we produce 2-dimensional configurations. For the duck dataset, we also show a 3-dimensional MDS representation, which contextualizes the 2-dimensional results.

For both datasets, we perform MDS using scikit-learn (Pedregosa *et al.* 2011) which utilizes SMACOF (Scaling by Majorizing a Complicated Function) (Borg and Groenen 2005; de Leeuw and Mair 2009). We set the maximum number of iterations equal to 5,000 (sufficiently high so that this limit is not encountered). We set $\epsilon_{stress} = 10^{-8}$, where ϵ_{stress} determines the threshold for the change in stress values when deciding whether to continue iterative optimization.

Our approach does not require a particular MDS implementation. To demonstrate, for the duck dataset, we also show results from performing MDS by minimizing stress via a nonlinear least-squares problem. We utilize the SciPy (Virtanen *et al.* 2020) least-squares implementation with the Levenberg-Marquardt algorithm (Moré 1978).

3.6. Barcode coordinates

A persistence barcode is a multiset of intervals, or bars. A barcode is not a vector; the number of intervals is not predetermined, and finding an optimal "matching" between bars of two persistence barcodes can be a nontrivial task. Nevertheless, there are many techniques for quantifying barcodes.

A stable, established method for quantifying persistence barcodes is through the use of tropical coordinates (Kališnik 2019), among the simplest of which are the quantities

$$\max_t \lambda_t$$

and

$$\sum_t \lambda_t,$$

where λ_t is the length of bar t, and t ranges over all the bars in the barcode. These quantities are the maximum bar length and the sum of the bar lengths, respectively. For datasets characterized by a single highly persistent topological feature, these quantities may be highly correlated. If this is the case, one can replace these tropical coordinates with the linear combination

$$c_1 = \max_t \lambda_t, \tag{4}$$

$$c_2 = \left(\sum_t \lambda_t\right) - \left(\max_t \lambda_t\right). \tag{5}$$

The coordinate c_1 is the maximum bar length, and c_2 is the sum of the remaining bar lengths. Throughout this work, we refer to c_1 and c_2 as barcode coordinates, following the terminology of Adcock, Carlsson, and Carlsson (2016). These coordinates are the basis for clustering the MDS representations.

In our setting, namely dimension 1 Vietoris–Rips filtrations of finite collections of data points equipped with Euclidean distances, persistent homology produces barcodes with finitely many bars (Chevyrev, Nanda, and Oberhauser 2020; Zomorodian and Carlsson 2005), and all bars have finite length (Blumberg, Gal, Mandell, and Pancia 2014). Consequently, the quantities in Equations (4) and (5) are well-defined and independent of the ordering of the bars. It is technically possible that a persistence barcode in dimension 1 could be empty, in which case we would set $c_1 = 0$ and $c_2 = 0$.

3.7. Clustering

Clustering of MDS representations from different random initial configurations is performed using the barcode coordinates generated by each MDS representation. These coordinates are defined in Equations (4) and (5). Both barcode coordinates quantify bar length(s), and we do not find it necessary to normalize either feature prior to clustering. Clustering only two coordinates facilitates visualization of the clusters for guiding selection of the clustering method and hyperparameters.

We recommend clustering methods which can handle clusters of different sizes. We use either Gaussian Mixture Models (GMMs) for more elliptical clusters (tomato dataset) or DBSCAN (Ester, Kriegel, Sander, and Xu 1996) for more irregular clusters (duck dataset). For both, we use the implementations from scikit-learn (Pedregosa *et al.* 2011).

For the tomato dataset, we use GMM clustering with two components. The number of components is selected based on the visual appearance of the plot of the barcode coordinates in Figure 4. The tolerance is set to 10^{-8} (clustering is not the most time-intensive part of the pipeline, so choosing a strict tolerance is reasonable). We require an estimated probability of at least 0.8 for assignment to a cluster.

For the duck dataset, we use the DBSCAN (Ester *et al.* 1996) clustering algorithm with hyperparameter values Eps = 0.001 and MinPts = 50. Eps can be selected based on the cluster separation observed in the plot of the barcode coordinates in Figure 4. The value of MinPts can be adjusted based on the number of MDS representations obtained (in our analysis, we obtain 10,000 MDS representations for each combination of dataset and MDS method).

For all datasets, we choose the names of the clusters so that the largest cluster (in terms of number of members) is the first cluster, and the sizes progressively decrease. This convention, rather than relying on default cluster names, enables easy comparison of the SMACOF versus least-squares results for the duck dataset. MDS representations which are not assigned to a cluster are labeled as noise.

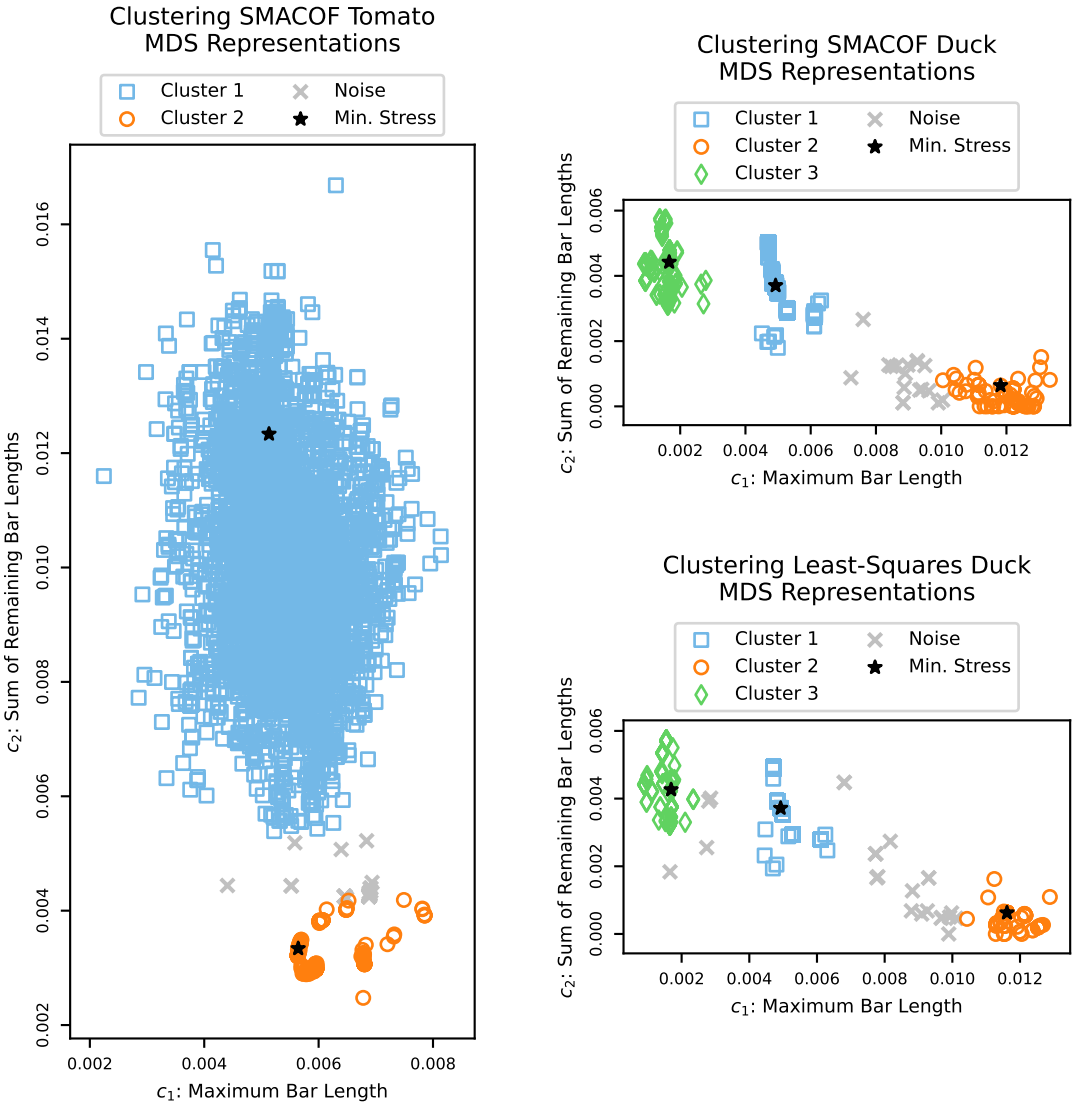

Figure 4: MDS representations from different random initial configurations are clustered based on barcode coordinates c_1 and c_2. The label "Noise" indicates MDS representations not assigned to a cluster. The stars correspond to the MDS representations with the lowest stress in each cluster. These are shown in Figures 5, 6 and 7.

4. Results

The persistence barcodes in dimension 1 of the scaled Euclidean distances between images for the tomato and duck datasets are shown in Figure 2. Both persistence barcodes contain a single long bar, or interval, and many smaller bars. The persistence barcode of the tomato dataset contains 19 bars, or intervals, while the barcode of the duck dataset contains 12 bars.

An overview of the number of SMACOF iterations executed for the various initial configurations is given in Table 3. Both datasets showed a significant range in the number of iterations required depending on the random initial configuration used. In general, the tomato dataset seemed to require more iterations than the duck dataset.

Table 3: The number of SMACOF iterations performed to obtain the MDS representations

Dataset	Minimum	Median	Maximum
tomato	147	472	2,369
duck	73	202	1,351

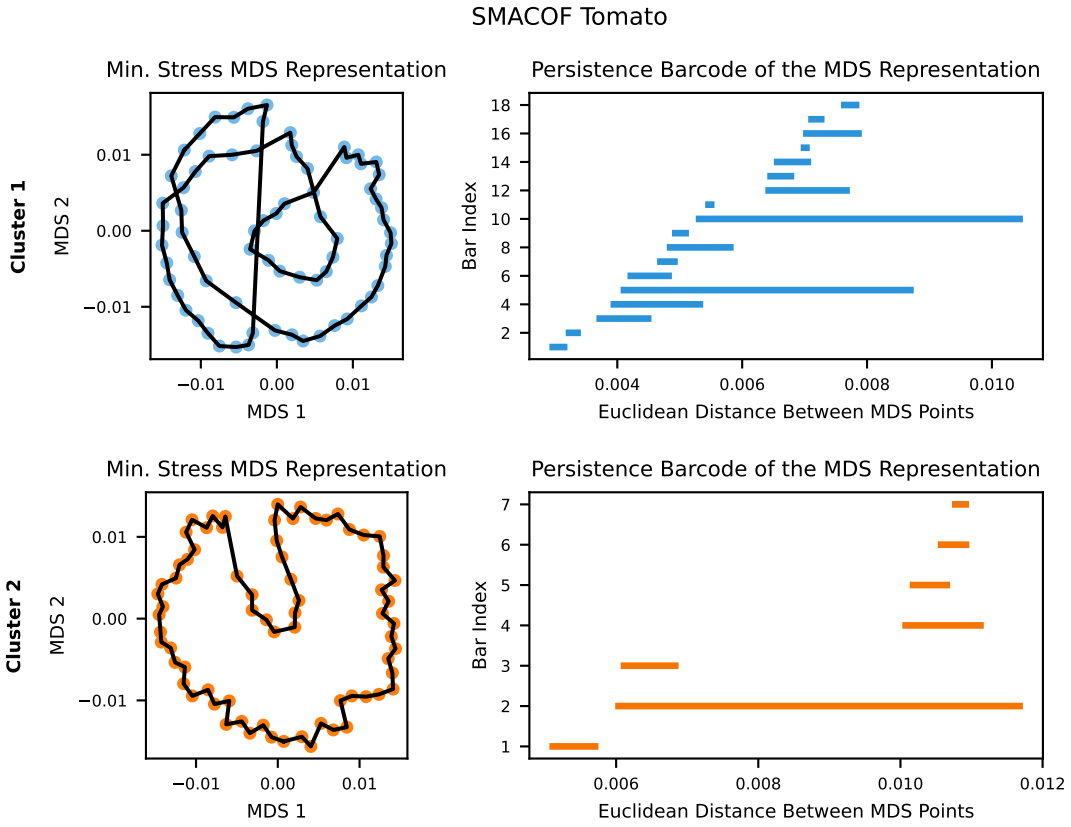

Figure 5: The minimum stress SMACOF MDS representation from each cluster of the tomato dataset. Black line segments represent adjacent images in terms of object rotation (ground truth information not used in performing MDS). The persistence barcodes of these MDS representations are also shown; compare to the persistence barcode of the tomato image dataset before performing MDS, shown in Figure 2.

The barcode coordinates of MDS representations of the tomato dataset from different random initial configurations are shown in the left panel of Figure 4. Clustering with Gaussian Mixture Models resulted in two clusters. As shown in Table 4, more than 97% of the MDS representations were assigned to Cluster 1. About two percent of the MDS representations belong to Cluster 2. The remainder were unassigned, and labeled as noise. As shown in Table 5, the MDS representation with the lowest stress is in Cluster 1.

Table 4: The number of MDS representations in each cluster

Dataset	Algorithm	Cluster 1	Cluster 2	Cluster 3	Noise	Total
tomato	SMACOF	9,727	229	N/A	44	10,000
duck	SMACOF	9,679	183	120	18	10,000
duck	least-squares	9,738	125	113	24	10,000

Table 5: The minimum stress of the MDS representations in each cluster

Dataset	Algorithm	Cluster 1	Cluster 2	Cluster 3	Noise
tomato	SMACOF	9.57×10^{-2}	9.90×10^{-2}	N/A	9.88×10^{-2}
duck	SMACOF	4.17×10^{-2}	5.36×10^{-2}	5.73×10^{-2}	5.37×10^{-2}
duck	least-squares	4.17×10^{-2}	5.36×10^{-2}	5.73×10^{-2}	5.75×10^{-2}

From each cluster of MDS representations of the tomato dataset, the MDS representation with the lowest stress in the cluster is selected for further consideration. The barcode coordinates of these representations are marked by stars in Figure 4 and the representations themselves are plotted in the left column of Figure 5. Each point in the MDS representation corresponds to a photograph of the tomato. In the plots, the points are connected with line segments if the corresponding images were generated by adjacent object rotations.

Out of the two MDS representations shown for the tomato dataset, the one from Cluster 2 has the clearest closed loop structure and the most consistent closeness of pairs of points from adjacent object rotation angles (designated in the plots by black line segments). The persistence barcodes of the two minimum stress MDS representations are shown in the right column of Figure 5. Both persistence barcodes have a bar with high persistence, but the Cluster 1 barcode has more bars and a higher second highest persistence.

The barcode coordinates of MDS representations of the duck dataset are shown in the right panel of Figure 4. The top plot was obtained using SMACOF MDS, while the bottom plot was obtained using least-squares MDS. The SMACOF and least-squares results are very similar and will be discussed together.

For the duck dataset, clustering with DBSCAN resulted in three clusters. As shown in Table 4, more than 96% of the MDS representations were assigned to Cluster 1 (slightly higher, more than 97%, for least-squares optimization). Less than two percent of the MDS representations are in Cluster 2. Slightly more than one percent of the MDS representations are in Cluster 3. A small number of the remaining MDS representations are unassigned, and labeled as noise. As shown in Table 5, the MDS representation with the lowest stress is in Cluster 1.

The minimum stress MDS representation from each cluster for the duck dataset is selected for further visualization. For SMACOF MDS, the three minimum stress MDS representations and their persistence barcodes are shown in Figure 6. For least-squares MDS, the minimum stress representations are shown in Figure 7. The SMACOF and least-squares MDS representations in each cluster are structurally similar. The MDS representation of Cluster 2 has the clearest closed loop structure, and the corresponding persistence barcode has the strongest signature of a single persistent feature. We do not perform full analysis of the 3-dimensional MDS representations of the duck dataset, but we show a sample configuration in Figure 8.

For both datasets, the cluster we found to have the most interpretable configuration is Cluster 2. In all cases, Cluster 2 contains significantly fewer MDS representations than Cluster 1. In addition, Cluster 1 is always the cluster with the MDS representation with the lowest stress.

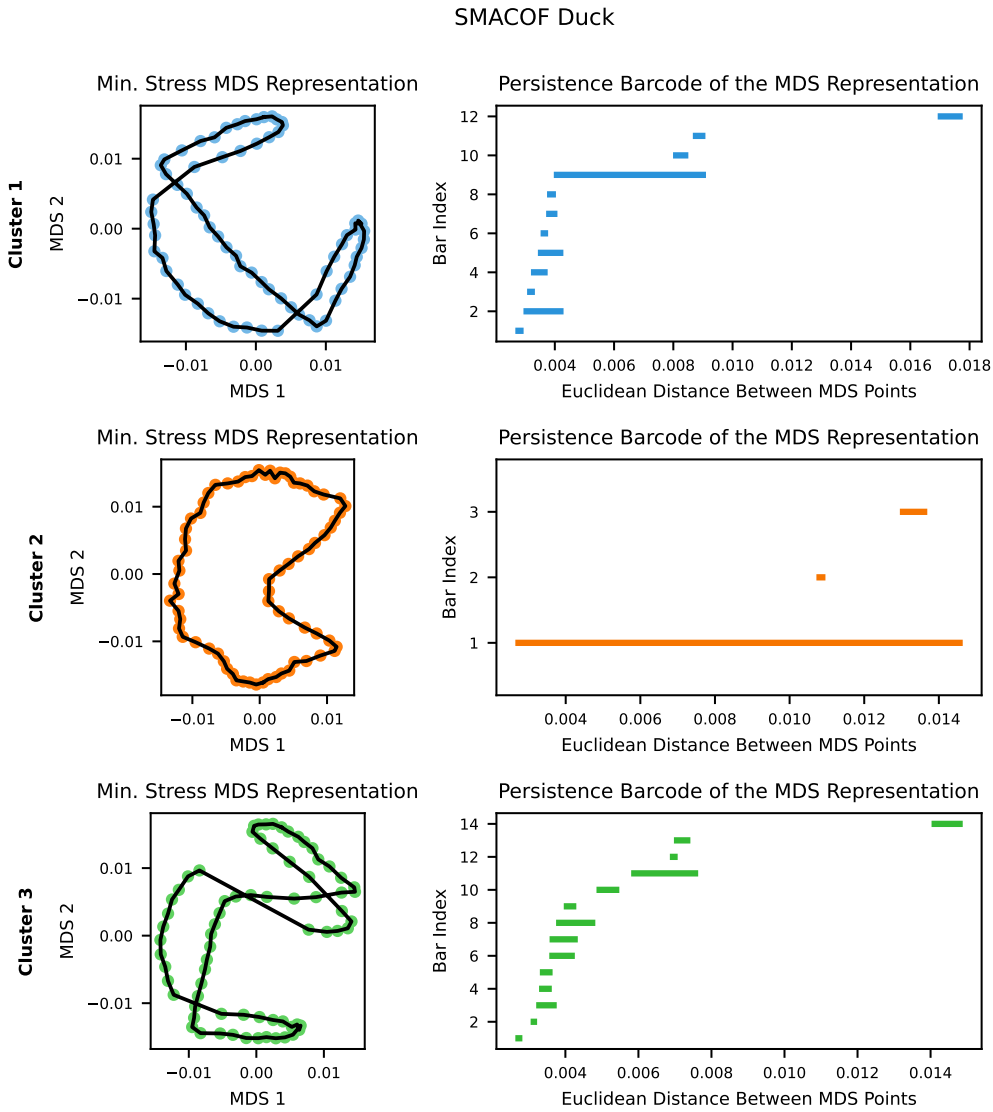

Figure 6: The minimum stress SMACOF MDS representation from each cluster of the duck dataset. Black line segments represent adjacent images in terms of object rotation (ground truth information not used in performing MDS). The persistence barcodes of these MDS representations are also shown; compare to the persistence barcode of the duck image dataset before performing MDS, shown in Figure 2.

5. Discussion

In our analysis, we encounter many MDS configurations which have lower stress than the configurations we find to be more interpretable. For the tomato dataset, the percentage increase in stress for the interpretable configuration versus the minimum stress configuration is well within what has been considered acceptable in applications of confirmatory MDS (Borg and Groenen 2005; Borg *et al.* 2018). For the duck example, the 2-dimensional configurations we identified that have a clearer closed loop structure have stresses that are relatively high (almost 30% higher than the minimum stress of any 2-dimensional representation). Viewing a 3-dimensional MDS configuration provides some additional context. The 3-dimensional representation we show in Figure 8 has a stress value close to the minimum stress of all the 3-dimensional representations we obtained. There is in fact a strong signature of a closed loop structure, but the loop does not lie close to a flat plane. Rather, it folds toward itself in 3-dimensional space. Cluster 1 and Cluster 2 of the 2-dimensional representations reveal complementary features of the dataset.

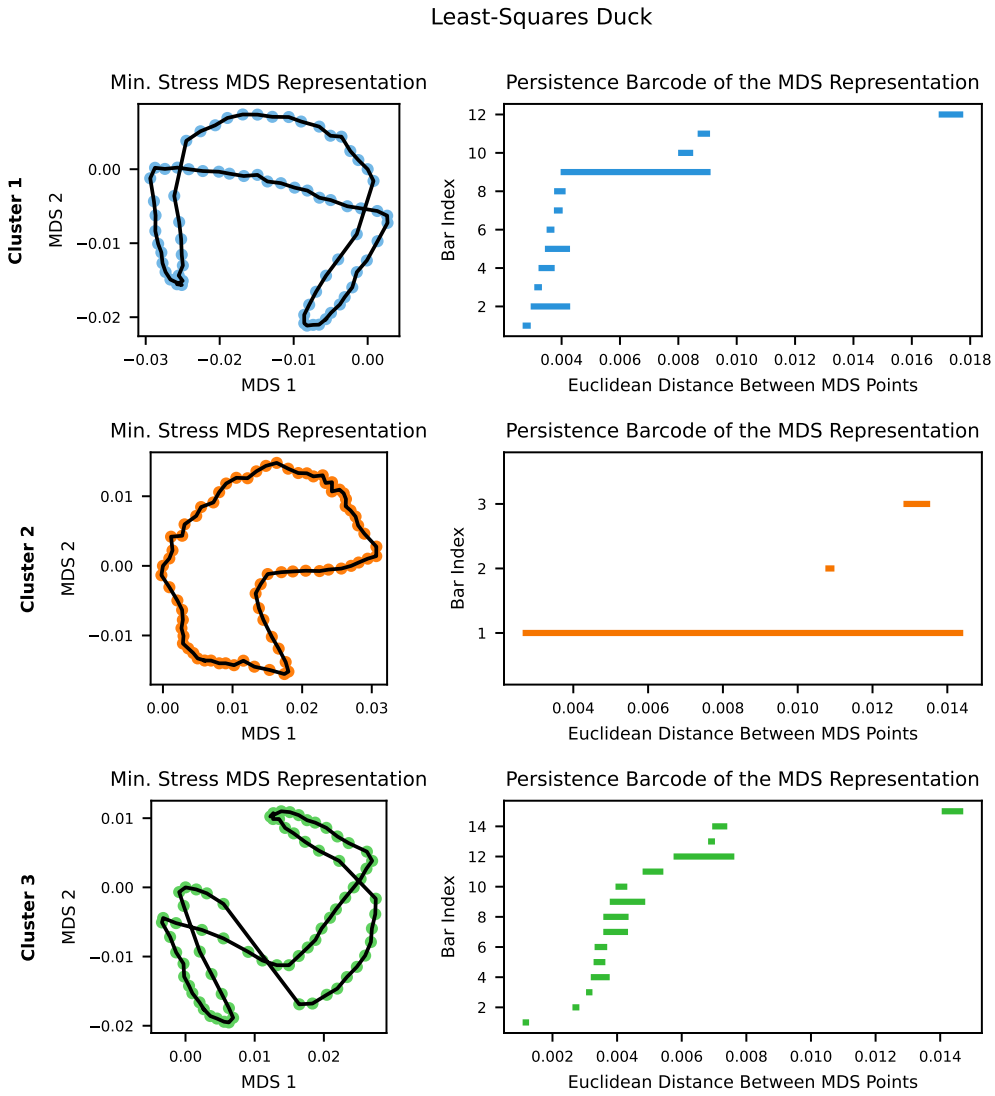

Figure 7: The minimum stress least-squares MDS representation from each cluster of the duck dataset. Black line segments represent adjacent images in terms of object rotation (ground truth information not used in performing MDS). The persistence barcodes of these MDS representations are also shown; compare to the persistence barcode of the duck image dataset before performing MDS, shown in Figure 2.

By viewing the minimum stress MDS representation of each cluster, rather than just the minimum stress MDS representation of the entire dataset, we can obtain a more complete picture of the dataset structure. Viewing such an ensemble of MDS representations reveals the variation in topological structure of representations from different random initial configurations. However, each individual MDS representation is still obtained using standard MDS techniques, which may improve interpretability as opposed to incorporating topological terms in the optimization itself.

In each example, Cluster 2 contains the MDS representation which we observed as being the most interpretable. That is, without knowing the ground truth angles associated with the points, we can nevertheless see that they lie in a closed loop, generated by their dependence on an underlying angle. Cluster 2 is a small cluster in all examples, but the interpretable configurations are repeatable. Each cluster contains more than 100 MDS representations. In short, the desirable configurations may be uncommon, but they are robust in the sense that obtaining a configuration with comparable barcode coordinates is not an isolated incident.

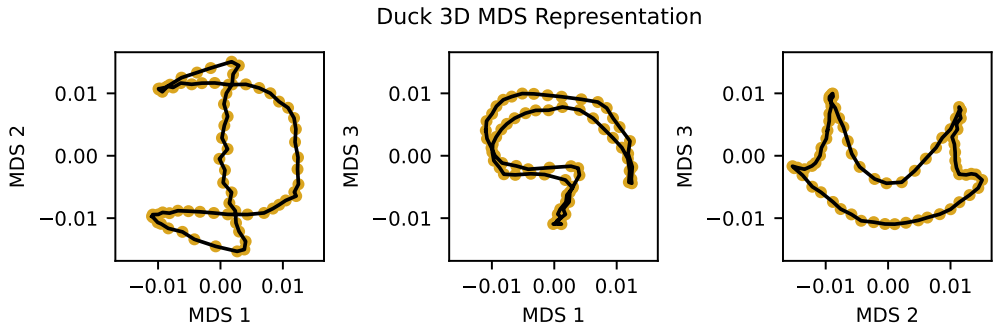

Figure 8: Viewing a low-stress 3-dimensional MDS representation of the duck dataset shows a strong signature of a closed loop structure whose visibility is dependent on the angle from which the configuration is viewed. Black line segments represent adjacent images in terms of object rotation (ground truth information not used in performing MDS).

Based on our observations, the greatest risk in terms of failing to identify an interpretable MDS representation is examining an insufficient number of initial configurations. It may not be immediately apparent that one has not trialed sufficiently many initial configurations. One option is to determine structural properties of the original dataset, and (within reason) increase the number of initial configurations if few or none of the final configurations reflect the known structure. In our case, we obtained the persistence barcodes of the pairwise dissimilarity data of the image datasets. If few or none of the persistence barcodes of the final MDS representations are similar to the barcode of the image dissimilarity data, then one could consider expanding the search by increasing the number of random initial configurations. Another option is to use known structural properties in combination with MDS. In previous work, we obtained interpretable MDS representations by using initial configurations constructed using topological data analysis (Kleczynski and Kearsley 2025). Using structural information to generate a strategic initial configuration falls in the category of weak confirmatory MDS analysis (Borg *et al.* 2018).

Early termination is another potential risk when using SMACOF methods for MDS (Kearsley *et al.* 1998). When performing MDS using the SMACOF implementation in scikit-learn (Pedregosa *et al.* 2011), we set the maximum allowable number of iterations equal to 5,000 and ϵ_{stress} equal to 10^{-8}. The default values are 300 for the maximum allowable number of iterations and either 10^{-3} or 10^{-6} for ϵ_{stress} (depending on the package version). As shown in Table 3, the median number of iterations for the tomato dataset is greater than the default maximum allowable number of iterations, and for both datasets there are some random initial configurations which require substantially more than 300 iterations to satisfy the convergence criterion. Ensuring that all the MDS runs are allowed to converge sufficiently can improve the clustering process by reducing the prevalence of MDS representations with atypical topology due to not being near a local stress minimum. Therefore, we recommend increasing the maximum allowable number of iterations and decreasing ϵ_{stress} if clear clusters are not initially apparent.

In this work, the clusters were apparent on visual inspection, and the cluster labels were readily obtained using standard clustering methods. If this is not the case for other datasets, one could establish an acceptable stress threshold to be applied prior to clustering. Removing MDS representations with excessively high stress could make the clusters more distinct. The threshold would need to be selected so that desirable MDS representations are not inadvertently excluded. Another approach would be to explore more sophisticated clustering techniques. In this work, we used Gaussian Mixture Models and DBSCAN. These performed well for the datasets we considered, and were able to handle the substantial variation in cluster sizes we observed. For other applications, additional techniques could be considered such as Ordering Points To Identify the Clustering Structure (OPTICS) (Ankerst, Breunig, Kriegel, and Sander 1999) and associated statistics (Rusch, Hornik, and Mair 2018).

One clustering consideration which is of heightened importance for our analysis is avoiding placing points in the "wrong" cluster. Consider the MDS representations of the tomato dataset, whose barcode coordinates are plotted in Figure 4. Due to requiring a sufficiently high estimated probability in order to be eligible for cluster assignment, some of the MDS representations are not assigned to a cluster. In particular, MDS representations whose barcode coordinates place them at the interface of the two clusters are left out. This is beneficial, because it prevents low stress MDS representations from Cluster 1 from inadvertently being included in Cluster 2. If this occurred, it may overshadow the interesting but higher stress representations in Cluster 2. If the minimum stress MDS representation in a cluster has barcode coordinates which place it near the edge of the cluster, it warrants further investigation to verify that the qualitative structure of this MDS representation is in fact typical of that cluster.

For our analysis we used Vietoris–Rips complexes. One advantage of this type of complex is that the same pipeline can be used to compute persistent homology of both the original dataset and the MDS representation in the event that comparisons are of interest. For our application, the number of objects in each dataset is relatively small, and this approach works well. For larger datasets, the computational cost of computing Vietoris–Rips persistent homology of thousands of MDS representations may be prohibitive. Fortunately, there are many options for reducing the time needed to obtain topological summaries (especially when approximations are acceptable). These include using other types of complexes such as alpha or witness complexes (Otter *et al.* 2017).

In this work, two barcode coordinates were sufficient. These coordinates are based on two tropical coordinates (Kališnik 2019). For collections of persistence barcodes requiring further description, additional tropical coordinates are available (Kališnik 2019), for example

$$\max_{s<t} (\lambda_s + \lambda_t) .$$

These additional coordinates may be particularly useful for persistence barcodes with several longer bars. If needed, there are also additional barcode coordinate systems, such as Adcock-Carlsson coordinates (Adcock *et al.* 2016).

Although not necessary in this work, there are many techniques for generating vectors from persistence barcodes (Ali, Asaad, Jimenez, Nanda, Paluzo-Hidalgo, and Soriano-Trigueros 2023), which allows for more flexibility in characterizing barcodes of more general datasets. We recommend starting with simpler persistence barcode descriptors, as these can be very competitive (Ali *et al.* 2023), and using more complicated vectorizations only when necessary. It is also possible to compute distances between topological summaries without first converting them to vectors; see Kerber, Morozov, and Nigmetov (2017) for a discussion of the computational complexity of this approach.

We could also explore using alternate families of initial configurations. We obtain coordinates from a normal distribution, as described in Section 3.4. Other approaches include generating coordinates from a uniform distribution or adding noise to a strategically chosen initial configuration (Borg and Mair 2017). The default behavior of the implementation in scikit-learn (Pedregosa *et al.* 2011) is to use a uniform distribution for initial coordinates.

We considered the topological features of MDS configurations from varying initial conditions. It would also be interesting to use these descriptors for characterizing other choices in the representation pipeline. For example, Structure Optimized Proximity Scaling (STOPS) in Rusch, Mair, and Hornik (2023) supports hyperparameter tuning for an expanded family of methods encompassing common MDS techniques; STOPS is a general procedure that would admit topological approaches for structure quantification.

6. Conclusion

We used tools from topological data analysis to characterize and cluster MDS representations, a pipeline which identified highly interpretable MDS configurations. These configurations may otherwise be easily missed, due to being less common and having somewhat higher stress values. In our analysis, we found persistence barcodes to be effective, interpretable, and dependent on few parameter choices. Consequently, we suspect they will be of use in characterizing MDS representations for a range of datasets, perhaps in combination with classical descriptors. For example, MDS representations of biological datasets can contain closed loop structures due to periodic phenomena such as the cell cycle (Liu, Lin, Yardımcı, and Noble 2018), and the pipeline outlined in this work may be of use if this structure is not immediately apparent. The growing availability of TDA software in a range of programming languages makes these techniques increasingly accessible for exploratory data analysis.

Although we utilize topological data analysis, the final chosen representations are still the outputs of a standard multidimensional scaling pipeline. We use a standard loss function (stress), and the final configurations are still sets of points whose pairwise distances approximate the pairwise dissimilarities between images. We use TDA for selecting the final MDS representations, but not for producing them. Once we have identified the random seeds which lead to interpretable representations, the final results can be reproduced using standard techniques. In conclusion, our method is effective and the outputs accessible for visualizing complex datasets.

Code availability

Code to reproduce the main results in this manuscript is available at https://github.com/usnistgov/cluster-MDS-persistence-barcodes.

Acknowledgments

This research was performed while MAK held a National Institute of Standards and Technology (NIST) National Research Council (NRC) Research Postdoctoral Associateship Award at the NIST Applied and Computational Mathematics Division under the supervision of AJK.

The authors thank the anonymous reviewers for their thoughtful evaluation and helpful suggestions.

References

Adcock A, Carlsson E, Carlsson G (2016). "The Ring of Algebraic Functions on Persistence Bar Codes." *Homology, Homotopy and Applications*, **18**(1), 381–402. doi:10.4310/HHA.2016.v18.n1.a21.

Ali D, Asaad A, Jimenez MJ, Nanda V, Paluzo-Hidalgo E, Soriano-Trigueros M (2023). "A Survey of Vectorization Methods in Topological Data Analysis." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **45**(12), 14069–14080. doi:10.1109/TPAMI.2023.3308391.

Ankerst M, Breunig MM, Kriegel HP, Sander J (1999). "OPTICS: Ordering Points To Identify the Clustering Structure." *SIGMOD Record*, **28**(2), 49–60. doi:10.1145/304181.304187.

Blumberg AJ, Gal I, Mandell MA, Pancia M (2014). "Robust Statistics, Hypothesis Testing, and Confidence Intervals for Persistent Homology on Metric Measure Spaces." *Foundations of Computational Mathematics*, **14**, 745–789. doi:10.1007/s10208-014-9201-4.

Borg I (2020). "Data Fit (Stress) vs. Model Fit (Recovery) in Multidimensional Scaling." *Austrian Journal of Statistics*, **49**(2), 43–52. doi:10.17713/ajs.v49i2.918.

Borg I, Groenen PJF (2005). *Modern Multidimensional Scaling: Theory and Applications*. 2 edition. Springer, New York, NY. ISBN 978-0-387-25150-9. doi:10.1007/0-387-28981-X.

Borg I, Groenen PJF, Mair P (2018). *Applied Multidimensional Scaling and Unfolding*. 2nd edition. Springer, Cham, Switzerland. ISBN 3319734709. doi:10.1007/978-3-319-73471-2.

Borg I, Leutner D (1985). "Measuring the Similarity of MDS Configurations." *Multivariate Behavioral Research*, **20**(3), 325–334. doi:10.1207/s15327906mbr2003_6.

Borg I, Mair P (2017). "The Choice of Initial Configurations in Multidimensional Scaling: Local Minima, Fit, and Interpretability." *Austrian Journal of Statistics*, **46**(2), 19–32. doi:10.17713/ajs.v46i2.561.

Borg I, Mair P (2022). "A Note on Procrustean Fittings of Noisy Configurations." *Austrian Journal of Statistics*, **51**(4), 1–9. doi:10.17713/ajs.v51i4.1423.

Carlsson G (2009). "Topology and Data." *Bulletin of the American Mathematical Society*, **46**(2), 255–308. doi:10.1090/S0273-0979-09-01249-X.

Carrière M, Michel B, Oudot S (2018). "Statistical Analysis and Parameter Selection for Mapper." *Journal of Machine Learning Research*, **19**(12), 1–39. URL http://jmlr.org/papers/v19/17-291.html.

Carrière M, Theveneau M, Lacombe T (2024). "Diffeomorphic Interpolation for Efficient Persistence-Based Topological Optimization." In A Globerson, L Mackey, D Belgrave, A Fan, U Paquet, J Tomczak, C Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 27274–27294. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2feff80094b297bcfb42dbb01f34b875-Abstract-Conference.html.

Chazal F, Michel B (2021). "An Introduction to Topological Data Analysis: Fundamental and Practical Aspects for Data Scientists." *Frontiers in Artificial Intelligence*, **4**. doi:10.3389/frai.2021.667963.

Chevyrev I, Nanda V, Oberhauser H (2020). "Persistence Paths and Signature Features in Topological Data Analysis." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **42**(1), 192–202. doi:10.1109/TPAMI.2018.2885516.

Clémot M, Digne J, Tierny J (2025). "Topological Autoencoders++: Fast and Accurate Cycle-Aware Dimensionality Reduction." *arXiv preprint arXiv:2502.20215*. URL https://arxiv.org/abs/2502.20215.

de Leeuw J, Mair P (2009). "Multidimensional Scaling Using Majorization: SMACOF in R." *Journal of Statistical Software*, **31**(3), 1–30. doi:10.18637/jss.v031.i03.

Dey TK, Wang Y (2022). *Computational Topology for Data Analysis*. Cambridge University Press. doi:10.1017/9781009099950.

Ellis CT, Lesnick M, Henselman-Petrusek G, Keller B, Cohen JD (2019). "Feasibility of Topological Data Analysis for Event-Related fMRI." *Network Neuroscience*, **3**(3), 695–706. doi:10.1162/netn_a_00095.

Ester M, Kriegel HP, Sander J, 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, p. 226–231. AAAI Press. URL https://dl.acm.org/doi/10.5555/3001460.3001507.

Ghrist R (2008). "Barcodes: The Persistent Topology of Data." *American Mathematical Society. Bulletin. New Series*, **45**(1), 61–75. doi:10.1090/S0273-0979-07-01191-3.

Ghrist R (2014). *Elementary Applied Topology*. 1.0 edition. Createspace.

Groenen PJF (1993). *The Majorization Approach to Multidimensional Scaling: Some Problems and Extensions*. DSWO Press, Leiden University, The Netherlands.

Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, del Río JF, Wiebe M, Peterson P, Gérard-Marchant P, Sheppard K, Reddy T, Weckesser W, Abbasi H, Gohlke C, Oliphant TE (2020). "Array Programming with NumPy." *Nature*, **585**(7825), 357–362. doi:10.1038/s41586-020-2649-2.

Henselman-Petrusek G, Hang H, Ziegelmeier L, Giusti C (2024a). "oat_python: User-Friendly Tools for Applied Topology in Python." https://crates.io/crates/oat_python.

Henselman-Petrusek G, Hang H, Ziegelmeier L, Giusti C (2024b). "oat_rust: User-Friendly Tools for Applied Topology." https://crates.io/crates/oat_rust.

Henselman-Petrusek G, Hang H, Ziegelmeier L, Giusti C (2024c). "Open Applied Topology." https://github.com/openappliedtopology.

Hunter JD (2007). "Matplotlib: A 2D Graphics Environment." *Computing in Science & Engineering*, **9**(3), 90–95. doi:10.1109/MCSE.2007.55.

Kališnik S (2019). "Tropical Coordinates on the Space of Persistence Barcodes." *Foundations of Computational Mathematics*, **19**, 101–129. doi:10.1007/s10208-018-9379-y.

Kearsley AJ, Tapia RA, Trosset MW (1998). "The Solution of the Metric STRESS and SSTRESS Problems in Multidimensional Scaling Using Newton's Method." *Computational Statistics*, **13**(3), 369–396.

Kerber M, Morozov D, Nigmetov A (2017). "Geometry Helps to Compare Persistence Diagrams." *ACM Journal of Experimental Algorithmics*, **22**. doi:10.1145/3064175.

Kleczynski M, Bergonzo C, Kearsley AJ (2025). "Spatial and Sequential Topological Analysis of Molecular Dynamics Simulations of IgG1 Fc Domains." *Journal of Chemical Theory and Computation*. doi:10.1021/acs.jctc.5c00161.

Kleczynski MA, Kearsley AJ (2025). *Topological Initialization for Multidimensional Scaling*. U.S. Dept. of Commerce, National Institute of Standards and Technology, Gaithersburg, MD. doi:10.6028/NIST.TN.2349.

Kruskal JB (1964). "Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis." *Psychometrika*, **29**, 1–27. doi:10.1007/BF02289565.

Lele SR, Richtsmeier JT (2001). *An Invariant Approach to Statistical Analysis of Shapes*. Chapman & Hall/CRC, Boca Raton, FL.

Li M, Storm C, Li AY, Needham T, Wang B (2023). "Comparing Morse Complexes Using Optimal Transport: An Experimental Study." In *2023 IEEE Visualization and Visual Analytics (VIS)*, pp. 41–45. doi:10.1109/VIS54172.2023.00017.

Liu J, Lin D, Yardımcı GG, Noble WS (2018). "Unsupervised Embedding of Single-Cell Hi-C Data." *Bioinformatics*, **34**(13), i96–i104. doi:10.1093/bioinformatics/bty285.

Lymberopoulos E, Gentili GI, Alomari M, Sharma N (2021). "Topological Data Analysis Highlights Novel Geographical Signatures of the Human Gut Microbiome." *Frontiers in Artificial Intelligence*, **4**. doi:10.3389/frai.2021.680564.

Mair P, Borg I, Rusch T (2016). "Goodness-of-Fit Assessment in Multidimensional Scaling and Unfolding." *Multivariate Behavioral Research*, **51**(6), 772–789. doi:10.1080/00273171.2016.1235966.

Moré JJ (1978). "The Levenberg-Marquardt Algorithm: Implementation and Theory." In GA Watson (ed.), *Numerical Analysis*, pp. 105–116. Springer, Berlin, Heidelberg. ISBN 978-3-540-35972-2. doi:10.1007/BFb0067700.

Myers AD, Chumley MM, Khasawneh FA, Munch E (2023). "Persistent Homology of Coarse-Grained State-Space Networks." *Physical Review E*, **107**, 034303. doi:10.1103/PhysRevE.107.034303.

Nelson BJ, Luo Y (2022). "Topology-Preserving Dimensionality Reduction via Interleaving Optimization." *arXiv preprint arXiv:2201.13012*. URL https://arxiv.org/abs/2201.13012.

Nene SA, Nayar SK, Murase H (1996a). "Columbia Object Image Library (COIL-100)." In *Technical Report, Department of Computer Science, Columbia University CUCS-006-96*. URL https://cave.cs.columbia.edu/repository/COIL-100.

Nene SA, Nayar SK, Murase H (1996b). "Columbia Object Image Library (COIL-20)." In *Technical Report, Department of Computer Science, Columbia University CUCS-005-96*. URL https://cave.cs.columbia.edu/repository/COIL-20.

Otter N, Porter MA, Tillmann U, Grindrod P, Harrington HA (2017). "A Roadmap for the Computation of Persistent Homology." *EPJ Data Science*, **6**, 17. doi:10.1140/epjds/s13688-017-0109-5.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011). "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, **12**(85), 2825–2830. URL http://jmlr.org/papers/v12/pedregosa11a.html.

Perea JA (2020). "Sparse Circular Coordinates via Principal $\mathbb{Z}$-Bundles." In NA Baas, GE Carlsson, G Quick, M Szymik, M Thaule (eds.), *Topological Data Analysis*, pp. 435–458. Springer, Cham, Switzerland. ISBN 978-3-030-43408-3. doi:10.1007/978-3-030-43408-3_17.

Python Software Foundation (2025). "Python." Available at https://www.python.org/ (25 June 2025).

Rizvi AH, Camara PG, Kandror EK, Roberts TJ, Schieren I, Maniatis T, Rabadan R (2017). "Single-Cell Topological RNA-Seq Analysis Reveals Insights into Cellular Differentiation and Development." *Nature Biotechnology*, **35**, 551–560. doi:10.1038/nbt.3854.

Rusch T, Hornik K, Mair P (2018). "Assessing and Quantifying Clusteredness: The OPTICS Cordillera." *Journal of Computational and Graphical Statistics*, **27**(1), 220–233. doi:10.1080/10618600.2017.1349664.

Rusch T, Mair P, Hornik K (2021). "Cluster Optimized Proximity Scaling." *Journal of Computational and Graphical Statistics*, **30**(4), 1156–1167. doi:10.1080/10618600.2020.1869027.

Rusch T, Mair P, Hornik K (2023). "Structure-Based Hyperparameter Selection with Bayesian Optimization in Multidimensional Scaling." *Statistics and Computing*, **33**(28). doi:10.1007/s11222-022-10197-w.

Schonsheck NC, Schonsheck SC (2024). "Spherical Coordinates from Persistent Cohomology." *Journal of Applied and Computational Topology*, **8**, 149–173. doi:10.1007/s41468-023-00141-w.

Stolz BJ, Dhesi J, Bull JA, Harrington HA, Byrne HM, Yoon IHR (2024). "Relational Persistent Homology for Multispecies Data with Application to the Tumor Microenvironment." *Bulletin of Mathematical Biology*, **86**, 128. doi:10.1007/s11538-024-01353-6.

Veres B, Schwertner WR, Tokodi M, Szijártó Á, Kovács A, Merkel ED, Behon A, Kuthi L, Masszi R, Gellér L, Zima E, Molnár L, Osztheimer I, Becker D, Kosztin A, Merkely B (2024). "Topological Data Analysis to Identify Cardiac Resynchronization Therapy Patients Exhibiting Benefit from an Implantable Cardioverter-Defibrillator." *Clinical Research in Cardiology*, **113**, 1430–1442. doi:10.1007/s00392-023-02281-6.

Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat İ, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, SciPy 10 Contributors (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." *Nature Methods*, **17**, 261–272. doi:10.1038/s41592-019-0686-2.

Zomorodian A, Carlsson G (2005). "Computing Persistent Homology." *Discrete & Computational Geometry*, **33**, 249–274. doi:10.1007/s00454-004-1146-y.

Affiliation:

Anthony J. Kearsley
Applied and Computational Mathematics Division
National Institute of Standards and Technology
100 Bureau Drive, Mailstop 8910, Gaithersburg MD 20899 USA
E-mail: anthony.kearsley@nist.gov
URL: https://www.nist.gov/people/anthony-j-kearsley