TDA - filament network classification

November 2019

Abstract

The actin cytoskeleton plays a critical role in plant cells. The filamentous structure of actin proteins can be viewed as a network endowed with a topology. We propose a novel, automated classifier, combining topological data analysis (TDA) with a machine learning framework in order to investigate and leverage the topology of actin networks. Our classifier is non-distance-based, instead using a persistence vectorization. We attain additional power, at a relatively low computational cost, through resampling. We benchmark our classifier against several distance and non-distance based classifiers, using a synthetic dataset to measure accuracy and sensitivity. We succeed in classifying the simulated networks with very high accuracy. Finally, we demonstrate an application with real data from confocal microscopy, classifying myosin-mutant and wildtype *Arabidopsis* root cells.

1 Introduction

The actin cytoskeleton is a complex network of proteins that is present in all eukaryotic cells. In addition to its function as cellular scaffolding, the actin cytoskeleton enables several basic cellular functions including the control of cellular shape and direction of movement [20]. These basic functions are critical to many higher order physiological processes such as cell division, expansion, mobility and motility[8].

In the actin cytoskeleton, actin filament organization is thought to be governed partially by the interaction of filaments and partially by myosin motor proteins. Actin filaments are polar structures, ploymerized by globular actin proteins. Many actin-binding proteins have potential to bind to actin filaments at various sites along the filament. These binding proteins allow actin filaments to assemble and disassemble spatiotemporally. The binding proteins give rise to a dense cross-linking where filaments develop into networks at with many filaments and binding sites. To understand certain behaviors of cells, it is of tremendous importance to understand the processes that govern actin filament network organization. A key driver of these processes may be the relationship between actin-binding proteins, individual filaments and emergent networks.

Our goal in this work is to develop a framework for the classification of actin newtworks. The images of actin networks that we are interested in are captured via confocal mcroscopy. These images are very high resolution (**TODO**: add typical resolution), but still suffer from several types of noise such as: filaments moving through the focal plain, rounding of the cell at the edges, neighboring cells poluting the image, changes in microscopic conditions/settings, and many more. Therefor, confocal microsocopy has the advantage of providing a lot of high quality data at the expense of also including many noisy data. This means that in order to automatically study these images, without interjections by researchers (eliminating potential for an introduction of unforseen bias), a tool is called for which is highly robust to these types of noise.

In the fast developing field of machine learning, topological data analysis (TDA) has become increasingly popular as a tool for noisy network and signal classification. To date, researchers have used TDA to solve many real-world problems including signal identification [13], materials classification [10, 16], shape recognition [3, 12], histologic image analysis [2, 17, 18], ecology of human mobility [5, 6], and cosmology [19, 21]. A review of TDA and its applications is provided in [22]. A sub-method of TDA, persistence homology, is a

popular method used to measure differences in topological features, due to its robustness in the face of purturbation of data. Persistence homology records when homological features (connections and voids) appear and vanish within data. These patterns vary between data. All of the appearances and disappearances of homological features are summarised in persistence barcodes and/or diagrams. In this work, we encode the geometric features of filaments networks into persistence diagrams and show a method of classification on the vectorization of the persistence space (a persistence space is not itself a vector space, so it has no mean for instance **TODO:cite**). We compare this approach to a traditional, distance-based classififfications which attempts to summarize the similarities of the actin network topologies in the persistence space.

We are aided in our investigation of classification methods by a high quality dataset of simulated actin networks, which we use to benchmark each method. We are provided the outputs of simulations which combine theoretical physical properties with experimental stochastic simulation in order to emualte actin network dyanmics. With this methodology one can control the known factors, which will affect the structure of networks. Varying these initial conditions enables researchers to compare the conditional difference in outcomes of the simulated networks. This experimental strategy can provide an opportunity to independently examine the role each factor plays in the process. These factors could include the cross-linker density(number of cross-linkers per certain area), cross-linker stiffness, maximum angle that can exist between two filament segments to be crosslinked, and so on[8, 9]. This control mechanism also allows us to test our methods on a highly controlled and clean dataset, in order to test sensitivity and to compare between methods.

After testing several methods of actin network classification on the simulated data, we choose the top performing method and addapt it to the microscopy images. We perform a classification between myosin-mutant and wildtype *Arabidopsis* root cells.

Move to discussion?: **In this work, we propose a machine learning approach to classify filament networks generated with varied cross-linker density. Our method leverages the topology of the actin networks through Topological Data Analysis(TDA). Our exploratory work is the first time filament networks have been studied by direct topological classification. This work could serve as a pilot for future research in actin cytoskeleton organization. In the future, this work should be useful in the course of research on cytoplasmic streaming to be able to classify real cells based on images of their actin networks. This could provide biologists a method of disentangling the interaction of myosin motor proteins, the actin network, and streaming, i.e. by imaging the actin structure and clustering cells based on their actin network topology, the researcher may be able to fix a network structure while varying parameters specific to myosin.**

The structure of this work is as follows: In section 2, we describe the data and introduce the background of persistence homology. Section 3 demonstrates two algorithms for classifying filament networks. Section 4 exhibits the numerical results. Section 5 will give conclusions and a discussion of future directions.

2 Persistence Homology and Filament Networks

To classify filament networks, we need to build a structure which reveals geometric features hidden in the data. We construct this structure with simplicial complexes in the typical way of persistence homology by using the 2-dimensional coordinates of actin beads along the filaments as the initial nodes. Simplcial complexes provide a bridge between the data space and a topological space in which computation of distances between sets of data points can be realized. A simplicial complex is a finite collection of simplices of different dimensions such that faces of simplices are also simplices, and intersections of the simplices are either empty or a face of both [7]. In particular, higher dimensional simplices are constructed from lower dimensional simplices. Vertices are 0-dim simplices. A 1-dim simplex is called an edge and is created by its two vertices as faces (note that a higher dimensional edge is constructed from lower dimensional points). A 2-dim simplex

or a triangle has three edges as faces. Further more, a 3-dim simplex or a tetrahedron has four triangles as faces, another nesting of several lower dimensional features to build one of higher dimension.

2.1 Data

Microscopy data: Microscopy data were provided in the form of one grayscale image per cell. The actin filaments fluoresce and so the intensity of each pixel of an image can be thought of as likely indicating the presense of a filament in that region of the image. In order to study the homology of an actin network, we must perform a topological tranformation on the data. Since the images contain hundreds of thousands of pixels, we chose to sample from the images a set of points, where the probability of choosing a pixel is propotionate to the pixel's intensity. We can make a choice of a number of points that we think sufficiently summarizes a network. We will perform our topological transformation on these new point clouds.

Simulated data: Our synthetic data come from simulations with varried numbers of crosslinking protiens. As discussed, actin filaments are thought to be organized by cross-linking on actin-binding proteins. Filaments and inter-filament structure can then be simulated by a physical model [8, 9]. The change of initial conditions in a eukaryotic cell will cause variation in later measurement of filament networks. Our network data is simulated by three different cross-linker densities. Higher cross-linker density means more opportunities for filaments to be cross-linked, i.e. the binding and unbinding processes can be more active. As shown in Fig. 1(a), three kinds of filaments networks were simulated with different numbers of cross-linkers: 825, 1650 and 3300. All simulated cells were bound by a $20\,\mu m \times 20\,\mu m$ square. Therefore, the cross-linker density of each network is 2.06, 4.13 and 8.25 per μm^2, respectively. In each network, there are a total of 100 filaments with average length $10\,\mu m$, where filaments are model as polar warm-like chain in red and blue dots represent barbed ends of these filaments. We also record the locations of the actin beads that make up the filaments, which are shown as small black circles in Fig. 1(b). Each actin bead is of radius $0.5\,\mu m$. Our interest is in developing an automated method to accurately classify cross-linker density of filament networks from simulated network data. The actin beads of the simulated networks will act as our point clouds in the topological transormations of these synthetic data.

2.2 Persistence Homology

In order to build simplicial complexes, we adopt the procedure of forming Vietoris-Rips complexes on each dataset (actin network) by introducing a sequence of ϵ-balls with increasing radius ϵ centered at each data point (a sampled pixel for image data or an actin bead in the synthetic data). Simplicial complexes are constructed based on intersections of these ϵ-balls and each value of ϵ corresponds to an unordered group of homological features, which is called a homology group. Considering values of ϵ as a timeline, we only record when a homological feature appears and disappears. These indexes are called the birth times and death times of a particular homological feature. Moreover, the lifespan (death minus birth) of a homological feature is referred to as the feature's persistence. A set of homological features gives rise to a set of persistence measures. At the end of this procedure, when radius ϵ is sufficient larger so that the homology group remains unchanged even by further increase the radius, information of a filament network's persistence homology (the set of persistpent homology measurements) is summarized in a persistence diagram.

Figure 3 depicts the process of discovering and summarizing the persistence homology of a simplified filament network. When $\epsilon = 0$, the sampled points of the filament network are each their own connected component. As the ϵ-spheres grow, connecetd components begin to merge. In the first column of te figure, when $\epsilon = 0.3$, several of the sampled points have already connected. We denote the points which have died (having merged into a larger connected component), by ending their bar in the presistence barcode below, and plotting a point at 0 on the x-axis and at their precise time of death on the y-axis in the persistence diagram below that. When $\epsilon \approx 1$, two holes form. The holes are evident in the top row, second column, of the figure, where one can see one larger and one smaller hole. We begin plotting bars for the holes in the corresponding

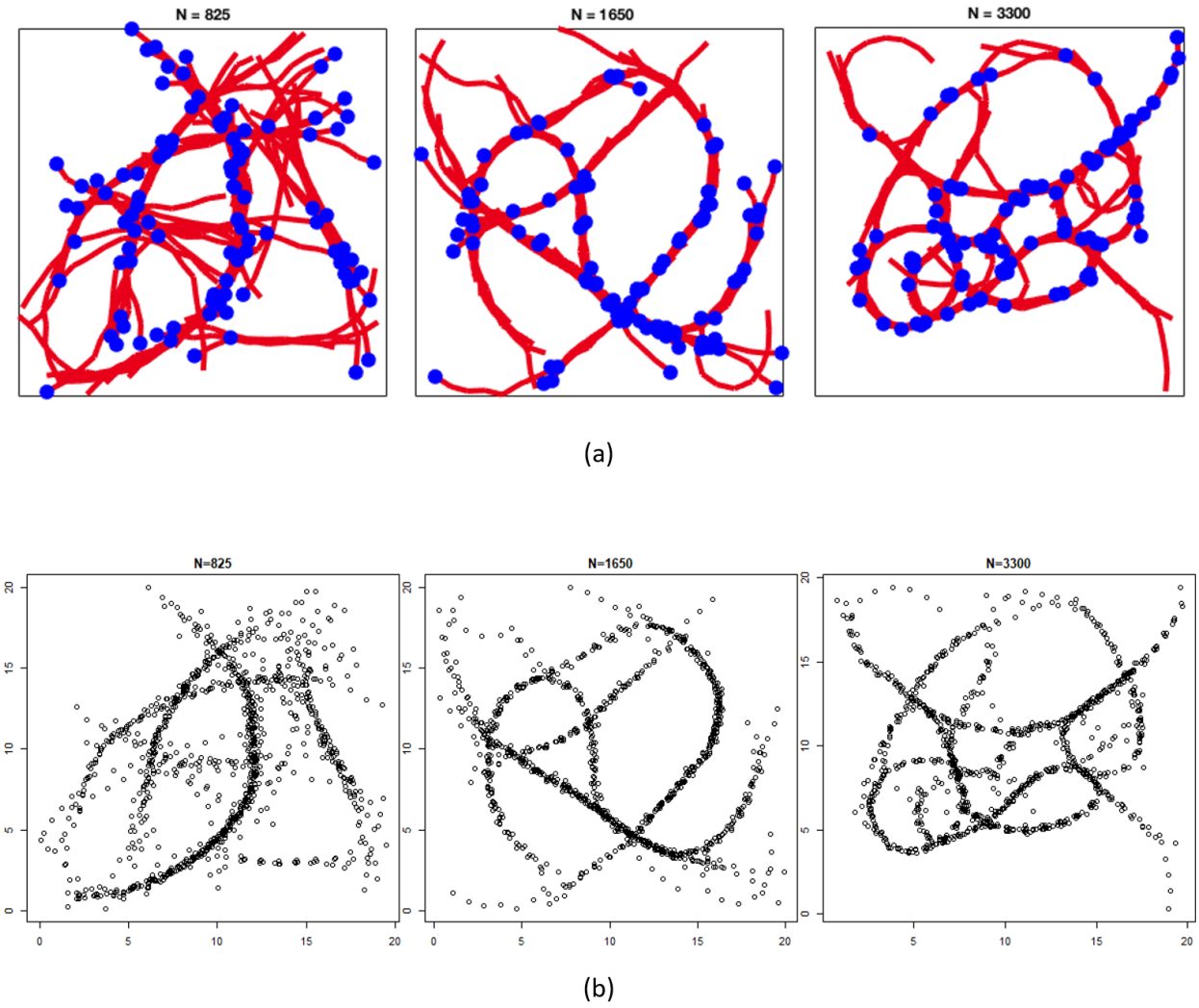

Figure 1: Filament networks. Panel (a) shows three filament networks generated by 825, 1650 and 3300 cross-linkers, respectively, in a 20 µm × 20 µm area. Each network contains 100 filaments which are represented as red lines. The blue dots are the barbed ends of these filaments. Panel (b) shows the locations of the actin beads that make up the filaments exhibited in Panel (a).

persistence barcode below. Note that there are not yet reccords of the holes in the persistence diagram, because we require the death time of the holes in order to plot them in the 2-dimensional diagram.

In the third column of Figure 3, when $\epsilon = 1.1$, the smaller hole has closed, and become part of a larger connected component. Now a single point is plotted in the persistence diagram below and the bar is terminated.

As the algorithm progresses, the larger hole eventually dies and its corresponding bar in the persistence barcode is terminated. A point is added at the corresponding birth and death in the persistence diagram (a seconf red triangle appears in the final column). The algorithm could continue to $\epsilon = \inf$, but it is evident in this example that no more homologic information will be discovered as the spheres continue to grow. We choose to arbitrarily terminate the algorithm at $\epsilon = 2$, terminate the final bar in the final barcode, and plot at (0,2) in the persistence diagram to denote final death time.

Overall, persistence homology indirectly summarizes the hidden shape of the data and transcribes this shape to the persistence diagram. With the persistent homology of each point cloud, a classifier can be generated either from the distance [14] between persistence diagrams or by alternative vectorizations of the diagrams [1, 4, 15].

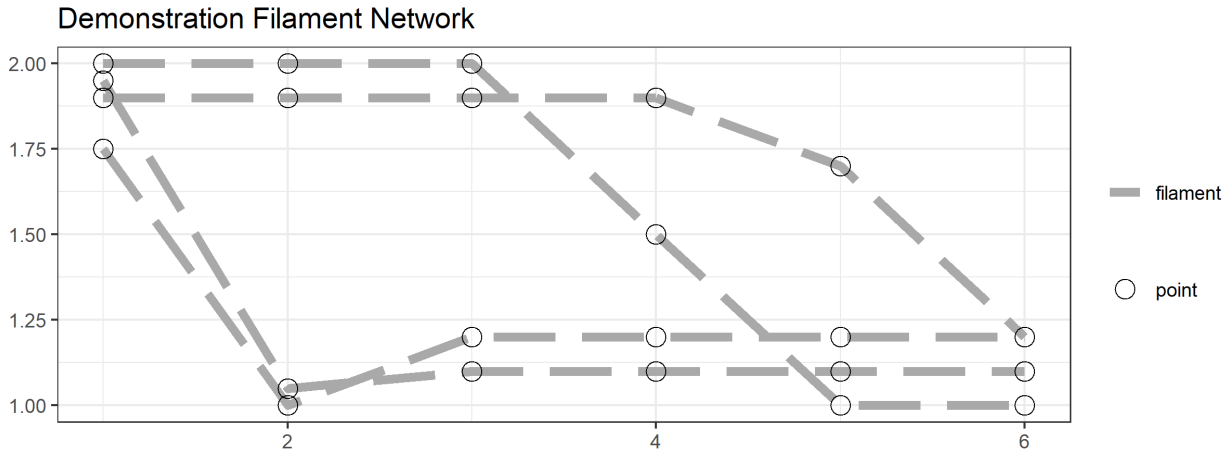

Figure 2: Demonstration fillament network. This network contains 3 filaments. Points are sampled along the filaments, in order to produce a point cloud from which persistence homology can be studied.

3 Filament network classifier

Once we have persistence diagrams corresponding to the point clouds of actin networks, we are ready to classify these networks. In this work, we propose two methodologies as candidates for a filament network classifier, one is distance-based method while the other one is based on a vectorization.

3.1 Distance-based network classifier

Given any two persistence diagrams, we need a way to quantify the difference between them in the space of persistence diagrams. In TDA, two methods are commonly used to calculate the distance between persistence diagrams: the Bottleneck and Wasserstein distances [1, 4, 7, 11, 22]. These distances calculate the optimal (minimal) cost in matching the points between two persistence diagrams. They assume infinitely many points of infinite multiplicity on the diagonal (where birth equals death), so that off-carnality points are

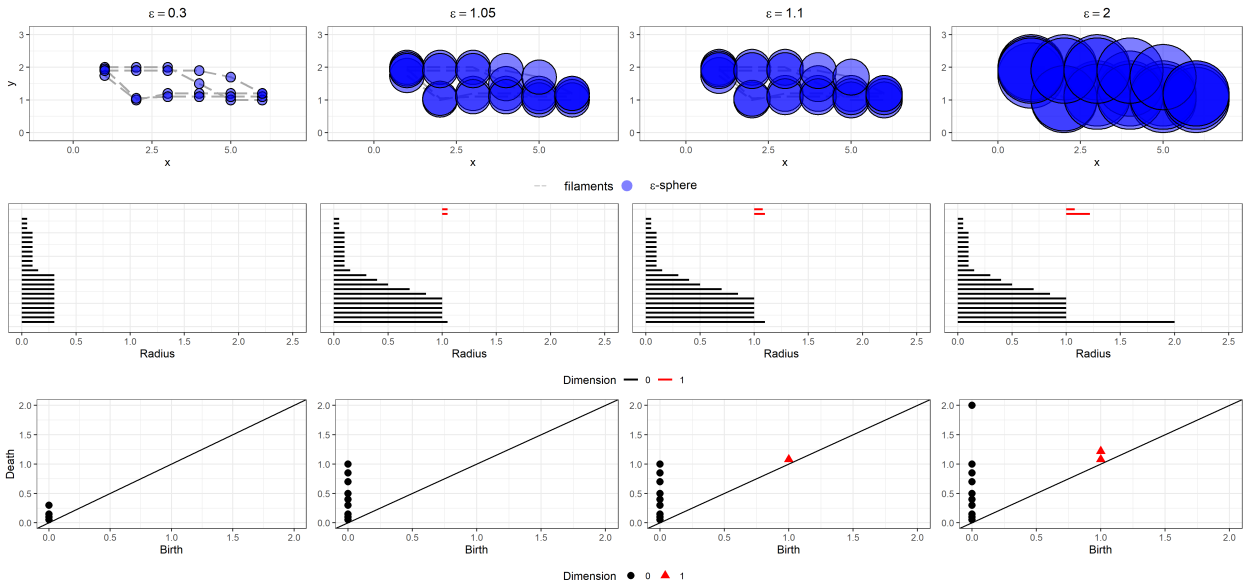

Figure 3: Investigation of the persistence homology of the demonstration filament network in Figure 2. The first row of figures, showes the growing ϵ-spheres about the sampled points of the fillaments. The second row shows the corresponding persistence barcode. The third row shows the corresponding persistence diagram. The columns progress with the algorithm, right-to-left.

matched to a point in this artificial set. In addition to the Wasserstein and Bottleneck distances, in this work we adopt a new distance, called d_p^c distance, which is proposed in [14] and has been proved to be stable in [15]. The cardinality of a persistence diagram may carry important information in applications, especially for those homological features which die very quickly and may be considered as insignificant in Wasserstain distance. However, the d_p^c distance accounts uneven cardinalities between persistence diagrams by assigning a regularization term with the parameter c rather than connecting extra points to the diagonal as is done in the Wasserstein distance. This method allows one to adjust the weight assigned to data that might be considered "topological noise" to fit the particular use case.

Definition 1. Let D_X and D_Y be two persistence diagrams with cardinalities n and m respectively such that $n \leq m$ and denote $D_x = \{x_1, ..., x_n\}$, $D_y = \{y_1, ..., y_m\}$. Let $c > 0$ and $1 \leq p < \infty$ be fixed parameters. The d_p^c distance between two persistence diagrams D_x and D_y is

$$d_p^c(D_x, D_y) = \left(\frac{1}{m} \left(\min_{\pi \in \Pi_m} \sum_{l=1}^{n} min(c, ||x_l - y_{\pi(l)}||_\infty)^p + c^P |m - n| \right) \right)^{\frac{1}{p}}, \tag{1}$$

where Π_m is the set of permutations of $(1, ..., m)$. If $m < n$, define $d_p^c(D_x, D_y) := d_p^c(D_y, D_x)$.

The d_p^c distance not only calculates the distance of points in two persistence diagrams without the simulated points on the diagonal, it also adds a penalty term on the difference in cardinalities of the two sets of points. The parameter c in eq. (1) is a constant that controls the weight of penalization to be added in the d_p^c distance. Larger values of c will yield a larger penalization. The parameter p is typically chosen as 2 since this corresponds to the Euclidean distance. We tend to evaluate c between 0 and 1 as these have been empirically found to be appropriate options in real-world applications [15].

Since persistence diagrams can summarize the homological features of multiple dimensions in one diagram, such as in the last panel of Fig. 3, the persistence diagram contains both 0-dim features(connected compo-

nents) with cardinality 5 and 1-dim features(holes) with cardinality 1. We can further define the d_p^c distance of a certain dimensional feature between a persistence diagram and a group of persistence diagrams.

Definition 2. Denote $\mathcal{D}$ as a collection of persistence diagrams in the same class. For a specific β-dim homological feature, $\beta = 0, 1, 2, ...$, the d_p^c distance between a persistence diagram D_x and the set of persistence diagrams $\mathcal{D}$ is

$$d_\beta(D_x, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{D \in \mathcal{D}} d_p^c(D_x, D), \tag{2}$$

where $|\mathcal{D}|$ represents the size of class $\mathcal{D}$.

With all preparations above complete, we can build the d_p^c-based network classifier. For K classes of filament networks, every network in a class is generated under a unique set of constraints. Therefore, we have K sets of persistence diagrams, where each set corresponds to a class of network generated under unique constraints. Given a new filament network with its persistence diagram D', our goal is to classify under which constraints the network was most likely generated, i.e. to which class k it most likely belongs. We can estimate this membership by calculating the distance between D' and each class of persistence diagrams. We then assign the new network to the class with the smallest distance. Additionally, we parameterize the relative weights for different dimensions of homological features in the calculation of the distance and force the weights' sum to 1. The classifier is summarized in Algorithm 1.

Algorithm 1 d_p^c-based network classifier

Let B is highest dimension of homological features under consideration.
1. Take the training set $T_1, T_2, ..., T_K$ from each class of diagrams $\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_K$,
2. For a new network with its corresponding persistence diagram D', compute

$$d(D', T_k) = \sum_{\beta=0}^{B} w_\beta d_\beta(D', T_k), \tag{3}$$

where $\sum_{\beta=0}^{B} w_\beta = 1$, and w_β determine how much β-dim homological feature is considered,
3. Assign D' a class label c' such that,

$$c' = \arg\min_{1 \le k \le K} d(D', T_k), \tag{4}$$

3.2 Non-distance-based network classifier

3.2.1 Sub-sampling approach

In a second analytical branch, rather than work on the coordinates of the actin beads directly, we took several pre-processing steps in order to extract more information from the provided network samples. The actin bead data included both spatial coordinates as well as an index indicating the filament sampled. That is, for our networks $\{X_1, X_2...X_{150}\}$, we had 100 filaments in each network $\{F_1, F_2...F_{100}\}$ and therefor our sampled points have features x, y, and f, where x and y are the coordinates in the plane and f is the index in the identity map of F.

Using Rs spatial package, each filament was constructed by connecting the (x,y) bead-coordinates in each filament. A single filament network was then constructed as a multiline object from the corresponding collection of lines formed from the bead-coordinates. These multiline objects were then projected onto square grids of 200×200 cells. Then, for each network we sampled the multiline object into a grid with the

identity function (i.e. the grid cells have value 0 unless a line and grid cell intersect then the value of the grid cell goes to 1). We now had a grid of 4000 cell values taking $\{0, 1\}$. We filtered these 4000 values to only those of value 1 and used the corresponding planar coordinates to build our new simplicial complexes and persistence diagrams. All re-sampling steps are summarized in Algorithm 2.

Algorithm 2 Re-sampling Algorithm

Let $X_i \in \{X_1 ... X_{150}\}$ the set of simulated actin networks.
Let $a \in \{A_{i,(x,y,f)} ... A_{i,(x,y,f)}\}$ the set of points in X_i. a is a pair, $(i, (x, y, f))$, where i is an index to X_i, x and y are planar coordinates and f is an index to a filament of X_i, $F_{i,f} \in \{F_{i,1} ... F_{i,100}\}$, along which a lies. See Panel A of Fig. 4 for a graphical depiction of one X_i. For simplicity, we will describe how our algorithm is applied to X_1:

1. Reconstruct the paths of each $F_{1,f}$ by connecting $A_{1,(x,y,f)}$. See Panel B, Fig. 4.
2. We project F_1 onto a 200x200 square lattice with the same bounds as the given by the data. The cells, $c_{x,y}$ of the lattice take values $\{0, 1\}$, where $c_{x,y} = 1$ if it is intersected by one of F_1 and 0 otherwise. See Panel C of Fig. 4.
3. Retain only $c_{x,y}$ where $c_{x,y} = 1$.
4. Draw 1000 random $c_{x,y}$ without replacement and call these the elements of $R_{1,r}$. Repeat this twice so that r indexes three new samples drawn from X_1. The three colors in Panel D of Fig. 4 denote the index r.
5. Generate three separate persistence diagrams, $D_{1,r}$, for the new samples $R_{1,r}$. Shown in Panel E of Fig. 4.

Due to the exponential growth of the computation of the Vietoris-Rips complex, rather than computing a Vietoris-Rips complex per network, 1000 points were randomly sampled from each network 3 times without replacement. This gives us 3 sets of 1000 points per network. This increased the sample size from 150 to 450 networks. We computed persistence diagrams for these 450 networks and could used these to construct a classifier using the same procedure as outlined in Algorithm 1.

3.2.2 Persistence vectorization/classification

In addition to distance based classification, a classification was performed on vectorized features of the persistence diagrams. A matrix was generated from the persistence diagrams which had row entries for each diagram and column features with the mean and standard error of persistence diagrams considering only the 0^{th} or 1^{th} persistence features alone and all topological features. That is, for each filament network, there is one persistence diagram which corresponds to one row of the vectorized matrix with mean and standard error of birth or death times for 0^{th}, 1^{th}, or all features. This gives us 12 columnar features. However, the mean and standard error of births for 0^{th} features are constant 0 and are not considered. Therefor, we are left with 10 entries per vector. The vectorization procedure is shown in Algorithm 3.

4 Classification result

4.1 d_p^c-based classifier

In our data set, we are provided three classes of filament networks. Each class of filament network is generated with a different number of cross-linking proteins. Each class contains 50 samples. Therefore, there are total of 150 individual provided filament networks.

In order to compare classifiers, we employed 10-fold cross validation to estimate overall classification accuracy. All of the networks are randomly partitioned into 10 mutually exclusive sets. 9 partitions are selected as a

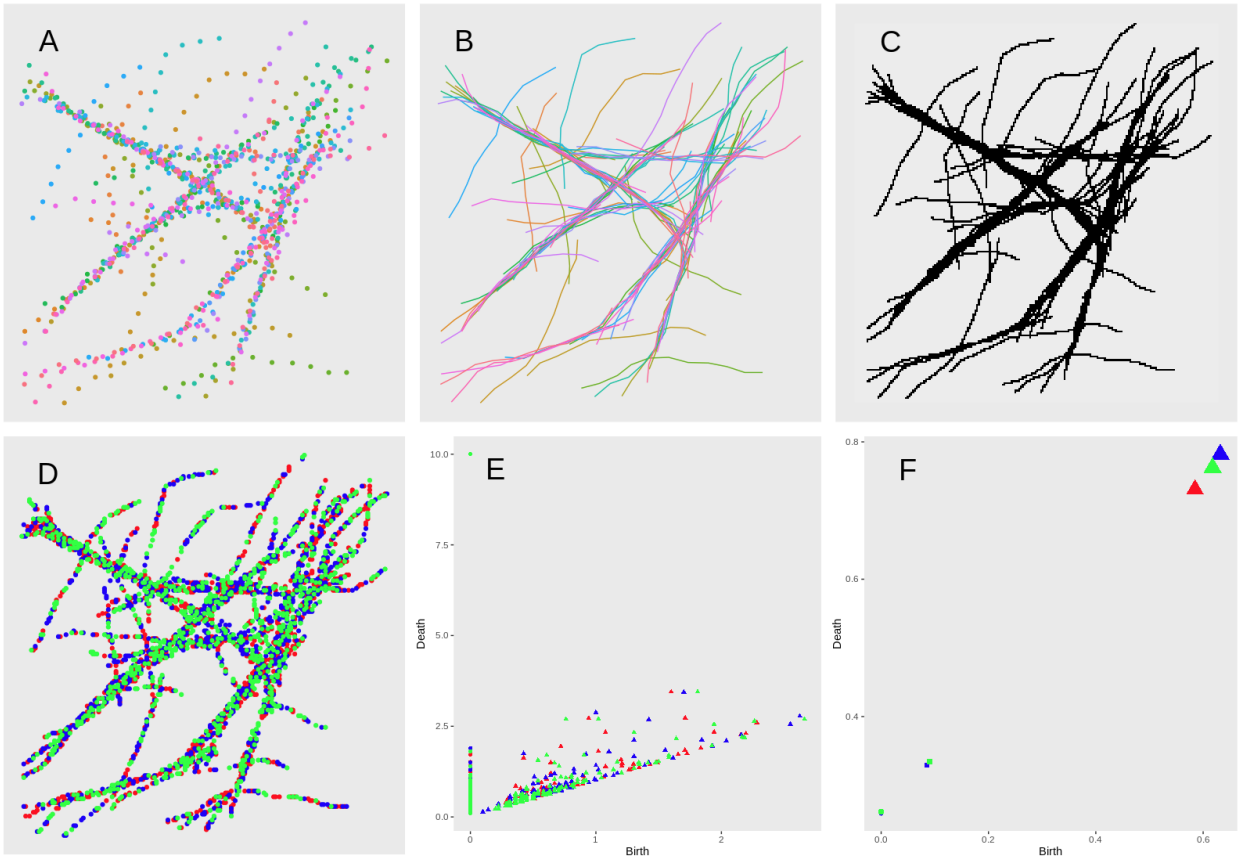

Figure 4: Example of one filament construction (B) from the supplied point cloud (A), rasterization (C), re-sampling (D), construction of persistence diagram (E) and vectorization of persistence diagram (F). The point cloud in (A) was supplied with an index that allowed for the constrution of the filaments in (B); note the coloring. This was projected onto a 200x200 raster (C). The raster was sampled into 3 sets (colored) of 1000 points (D) without replacement and the x,y coordinates are used to generate a VR complex. The persistence diagrams and vectorizations in (E) and (F) are colored the same as (D), to show the variability that arises in the vectorized persistence diagram features when this approach is taken.

Algorithm 3 Vectorization of persistence diagrams

We vectorize $D_{1,r}$ from Algorithm 2 by taking the mean *birth* and mean *death* of connected components and holes. These data are plotted in Panel F of Figure 3. For $D_{1,r}$ we have 3 row-vectors in the form:

$$v_{1,r} = \begin{bmatrix} \bar{x}(\text{Death of Connected components}) \\ s(\text{Death of Connected components}) \\ \bar{x}(\text{Birth of holes}) \\ s(\text{Birth of holes}) \\ \bar{x}(\text{Death of holes}) \\ s(\text{Death of holes}) \\ \bar{x}(\text{Birth of all features}) \\ s(\text{Birth of all features}) \\ \bar{x}(\text{Death of all features}) \\ s(\text{Death of all features}) \end{bmatrix}$$

Note: the mean $\bar{x}$ and standard error s of the birth of connected components would be 0 for all vectors, so these are not found in $v_{1,r}$. Repeat for all X_i. Our complete matrix of data has a final dimension 450×10.

training set, while the remaining 1 partition is used for testing. We repeat the classification 10 times such that every partition acts as a testing set exactly once. We consider the overall classification accuracy rate as the mean accuracy across all partitions.

After generating the persistence diagram for each filament network from the locations of actin beads, we tested our classifier on this data set using the d_p^c-based classifier in Algorithm 1. We consider only 0-dim and 1-dim topological homological features. We chose $p = 2$ to imitate Euclidean distance. The choices of parameter w_0, w_1, c were found to be optimal based on cross validation. When $w_0 = 0.5, w_1 = 0.5$, connected components and holes are weighted equally, and $c = 0.2$ gives a relatively small contribution from the cardinality difference in the d_p^c distances, the best classification accuracy rate is 89%.

4.2 non-distence based

A support vector machine (SVM) was used to develop a network classifier based on the 3-times re-sampled and vectorized persistence diagrams. 10-fold cross-validation was again used to assess the accuracy of the SVM classifier. Accuracy was measured only on one of the three sub-sampled networks $R_{i,1}$ per filament network X_i so as to fairly compare to the methods without re-sampling. The mean accuracy of the classifier was 96.3%. This was the highest accuracy attained on this set of simulated networks.

4.3 Comparison to other classifiers

We also list the accuracy rate by using other classifier in Table 1.

Our findings show that re-sampling of data can produce a more reliable filament network classifier even when data are mapped to a topological space and summarized in persistence diagrams. The d_p^c-based classifier performed much better than Wasserstein-based classifier. Our re-sampled and vectorized persistence diagram based classifier and d_p^c-based classifier are superior than any other common classifiers. These findings suggest that future with with real data microscopy data may be classified with similar methods.

classifier	accuracy
SVM, re-sampled	96%
d_p^c-based	89%
Wasserstein-based, re-sampled	87%
Wasserstein-based	83%
Bayes Factor [16]	83%
SVM PI [1]	75%
Neural Net PI	71%
SVM Raster	65%
Random Forest Raster	55%

Table 1: Accuracy rate of classifiers

4.4 Application to classifying myosin mutants

We are given labeled images of cells belonging to one of two classes: myosin mutant (MM) or wild type (WT). The MM cells have a genetic knockout such that they do not a produce one of their myosin proteins. The WT cells are the control with no knockout. Myosin motor proteins are known, via empirical observation, to influence the connectivity and shape of the actin filament network. This experiment differs from our synthetic experiment in that our synthetic experiment involved the direct manipulation of the actin network through changes in cross-linker abundance. Whereas cross-linking is fundamental to the structure of the actin network, the linkage betweeen mysoin motors and actin networks is more complex and likely modulated by several interacting factors.

Figure 5: Example of a raw WT cell image.

We perform classification of MMs and WTs using our top classifier, the re-sampled SVM. The data are provided as tifs from 22 MM and 20 WT cells. Each tif is a a very high resolution (roughly 2500x500 pixels with 0.043 μm^2 pixels) grayscale image, wherein the pixel values are an 8-bit inensity. We perform preprocessing on the images simply by performing a gaussian blur and then thresholding to the top $\frac{1}{3}$ of values. The thresholded image is then a grid of values taking only 0 or 1, where 1 indicates the presense of a filament. We then take the processed images and proceed in classification from step (C) in Figure 4, resampling, computing PDs, vecorizing the PDs, and finally classifying with an SVM. We find a cross-validated (k=5) accuracy of roughly 73%.

Interpretation of these results is complicated by our uncertainty of the true, cumulative impact of the moyin knockout on the actin network. While the accuracy in this experiment is significantly lower than found in the analysis of the synthetic data, this could easily be explained by the single motor protein knockout having a relatively minor impact on actin organization when compared with up or down regualtion of cross-linkers. Out of curiosity, we conducted a small survey (n=6), wherein we asked respondents who were familiar with the experiment to sepparate the cells into two classes. We found that respondents had an average accuracy of 63%, with a minimum accuracy of random chance and a maximum accuracy of 75%. Since our algorithm is (on average) more effective than manual classification by an expert, we conclude that there is likely a weak relationship between this mysoin knockout and actin network structure, and our classification algorithm is likely running into the ceiling of discernible difference in classes.

5 Conclusion

In this work, we successfully classified simulated actin filament networks generated under different initial conditions with very high accuracy. We combined a machine learning framework with TDA and proposed two classifiers. One is a distance-based classifier based on a new advanced distance on the persistence diagram space, i.e., the d_p^c distance; Another one is a non-distance-based classifier with re-sampling and persistence vectorization. Both of these methods were built on the foundation of topological homology by encoding geometric features hidden in the data into the topological features and summarizing those into persistence diagrams.

From these results, we are able to confidently classify actin filament networks based on unique inital conditions. Accordingly, these methods could reveal the key factors in the filament network generating process and provide biologists with the opportunities to uncover the interaction of motor proteins, actin networks and streaming. In addition, we show that it is possible to classify cells by images of their actin networks, since our data reflects a 2 dimensional projection of the network such as would come from microscopy. Overall, our work is the first time that actin filament networks have been studied by topological classification, researchers could further advance their understanding of cell physiology through this work and future research.

References

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

[2] Francisco Belchi, Mariam Pirashvili, Joy Conway, Michael Bennett, Ratko Djukanovic, and Jacek Brodzki. Lung topology characteristics in patients with chronic obstructive pulmonary disease. *Scientific reports*, 8(1):5341, 2018.

[3] Thomas Bonis, Maks Ovsjanikov, Steve Oudot, and Frédéric Chazal. Persistence-based pooling for shape pose recognition. In *International Workshop on Computational Topology in Image Context*, pages 19–29. Springer, 2016.

[4] Peter Bubenik. Statistical topological data analysis using persistence landscapes. *The Journal of Machine Learning Research*, 16(1):77–102, 2015.

[5] Yen-Chi Chen and Adrian Dobra. Measuring human activity spaces from gps data with density ranking and summary curves. *arXiv preprint arXiv:1708.05017*, 2017.

[6] Yen-Chi Chen et al. Generalized cluster trees and singular measures. *The Annals of Statistics*, 47(4):2174–2203, 2019.

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

[8] Simon L Freedman, Shiladitya Banerjee, Glen M Hocky, and Aaron R Dinner. A versatile framework for simulating the dynamic mechanical structure of cytoskeletal networks. *Biophysical journal*, 113(2):448–460, 2017.

[9] Simon L Freedman, Glen M Hocky, Shiladitya Banerjee, and Aaron R Dinner. Nonequilibrium phase diagrams for actomyosin networks. *Soft matter*, 14(37):7740–7747, 2018.

[10] Yasuaki Hiraoka, Takenobu Nakamura, Akihiko Hirata, Emerson G Escolar, Kaname Matsue, and Yasumasa Nishiura. Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences*, 113(26):7035–7040, 2016.

[11] Michael Kerber, Dmitriy Morozov, and Arnur Nigmetov. Geometry helps to compare persistence diagrams. *Journal of Experimental Algorithmics (JEA)*, 22:1–4, 2017.

[12] Chunyuan Li, Maks Ovsjanikov, and Frederic Chazal. Persistence-based structural recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1995–2002, 2014.

[13] Andrew Marchese and Vasileios Maroulas. Topological learning for acoustic signal identification. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1377–1381. IEEE, 2016.

[14] Andrew Marchese and Vasileios Maroulas. Signal classification with a point process distance on the space of persistence diagrams. *Advances in Data Analysis and Classification*, 12(3):657–682, 2018.

[15] Vasileios Maroulas, Cassie Putman Micucci, and Adam Spannaus. A stable cardinality distance for topological classification. *arXiv preprint arXiv:1812.01664*, 2018.

[16] Vasileios Maroulas, Farzana Nasrin, and Christopher Oballe. Bayesian inference for persistent homology. *arXiv preprint arXiv:1901.02034*, 2019.

[17] Monica Nicolau, Arnold J Levine, and Gunnar Carlsson. 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(17):7265–7270, 2011.

[18] Nikhil Singh, Heather D Couture, JS Marron, Charles Perou, and Marc Niethammer. Topological descriptors of histology images. In *International Workshop on Machine Learning in Medical Imaging*, pages 231–239. Springer, 2014.

[19] Thierry Sousbie, Christophe Pichon, and Hajime Kawahara. The persistent cosmic web and its filamentary structure–ii. illustrations. *Monthly Notices of the Royal Astronomical Society*, 414(1):384–403, 2011.

[20] Clément Thomas, Stéphane Tholl, Danièle Moes, Monika Dieterle, Jessica Papuga, Flora Moreau, and André Steinmetz. Actin bundling in plants. *Cell motility and the cytoskeleton*, 66(11):940–957, 2009.

[21] Rien van de Weygaert, Erwin Platen, Gert Vegter, Bob Eldering, and Nico Kruithof. Alpha shape topology of the cosmic web. In *2010 International Symposium on Voronoi Diagrams in Science and Engineering*, pages 224–234. IEEE, 2010.

[22] Larry Wasserman. Topological data analysis. *Annual Review of Statistics and Its Application*, 5:501–532, 2018.