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 classify myosin-mutant from wildtype *Arabidopsis* root cells by their actin network topologies, captured in confocal microscopy images. We benchmark our classifier against several distance and non-distance based classifiers, using a simulated dataset to compare sensitivities. We succeed in classifying the simulated networks with very high accuracy and carry over the highest accuracy method to classify the *Arabidopsis* root cell images. Our top performing method is an 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 low computational cost, through sub-sampling.

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 [24]. These basic functions are critical to many higher order physiological processes such as cell division, expansion, mobility and motility[8].

Actin filament organization is thought to be largely governed by the interaction of the filaments themselves and 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 spatiotemporally assemble and disassemble. The binding proteins give rise to a dense cross-linking where filaments develop into networks consisting of many filaments and very many binding sites. Therefore, one key driver of actin network dynamics is the relationship between actin-binding proteins, individual filaments, and emergent networks [8, 9]. Another key driver is the activitity of myosin motor proteins [25, 13, 21, 20], which perform several critical functions including pulling intracellular bodies about the cell. To understand certain behaviors of cells, it is of tremendous importance to understand the processes that govern actin filament network organization including the binding of actin filaments and the role of myosin motor activity.

Our goal is to develop a general framework for the classification of actin network topologies. The images of actin networks we examined are captured via confocal microscopy. These images are very high resolution (typically 1500×400 to 2500×500 pixels, where a pixel is approximately $0.04\,\mu m^2$), but 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, and changes in microscopic conditions/settings. Consequently, confocal microsocopy data have the advantage of providing many high quality data at the expense of also including many noisy data. In order to automatically study these images, without myriad interjections by researchers (and thus eliminating a potential introduction of bias), an automated tool is called for which is highly robust to these types of noise. This work seeks to develop a classifier, robust to the noise of confocal microscopic images of actin networks, and requiring minimal intervention from researchers.

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 [14], materials classification [10, 17], shape recognition [3, 12], histologic image analysis [2, 19, 22], ecology of human mobility [5, 6], and cosmology [23, 26]. A review of TDA and its applications is provided in [27]. A sub-method of TDA, persistence homology, is a popular mechanism used to measure differences in topological features, due to its robustness in the face of purturbated data. Persistence homology records when homological features (connections and voids) appear and vanish in 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, and, for instance, may not have a unique mean [18]). We compare this approach to traditional, distance-based classifications which attempt to summarize the similarities of the actin network topologies in the persistence space.

We are aided in our investigation of classification methods of actin networks by a dataset of simulated networks, which we use to benchmark our candidate methods. We are provided the outputs of simulations which combine theoretical physical properties with experimental stochastic simulations in order to emulate actin network dyanmics. These simulations allow the researcher to control the instantiating factors which will drive the emergent structure of the networks. Varying the initial conditions enables us to compare the conditional difference in outcomes of the simulated networks. This experimental strategy can provide an opportunity to independently examine the role individual factors play in the process of network formation. 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 gives us the opportunity to test our classification methods on an analagous, highly controlled and clean dataset.

After testing several methods of actin network classification on the simulated data, we take our top performing method and adapt it to the classification of microscopy images. We test our classifier, by measuring it's accuracy in a task of supervised learning wherein we label myosin-mutant (MM) and wildtype (WT) *Arabidopsis* root cells. The MM cells have a genetic knockout on one myosin motor protein and the WT cells are a control with no knockout. Our top performing method provides very high accuracy in classifying the simulated data, but less so in the microscopy data. We address the likely reasons for this descrepency as well as future improvements in our final discussion.

The structure of this work is as follows: In Section 2, we describe the data and introduce persistence homology. Section 3 describes our methods including several detailed algorithms for classifying simulated and imaged filament networks. Section 4 compares the results from the simulated networks and exhibits teh final results on the classfication of microscopy images. Finally, in Section 5, we discuss the implications, limitations and conclusions of our work.

2 Filament Networks and Persistence Homology

To quantify the differences in filament networks, we need to tranform our data in a manner that reveals its hidden geometric features. We perform this transformation using simplicial complexes in a manner typical of persistence homology. We use the 2-dimensional coordinates of sampled points along the filaments as 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

We are provided data in two spaces. The focus of this work is to develop a method for the classification of actin networks in confocal microscopic imagery. These data are given to us in the form of one 2-dimensional image per cell, where the data can be seen as a rectangular lattice of pixel instensities. We are also provided benchmarking data that come from simulations. The outputs we are provided from these simulations are point-clouds, with coordinates in 2-dimensions.

Microscopy data: The microscopy data were provided in the form of one grayscale image per cell. An example image is shown in Figure 1. The actin filaments fluoresce in the images and so the intensity of each pixel of an image can be thought of as indicating the likely presense of a filament in that region of the cell. 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 make a choice of a number of pixels that we think is likely to sufficiently summarize a network. We sample the pixels coordinates, weighted by pixel intensity, to get a set of points in 2 dimensions. We can then perform our topological transformation on these new point clouds.

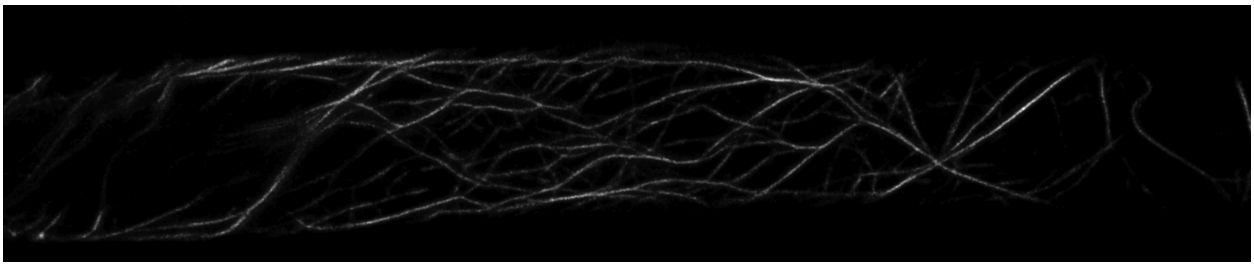

Figure 1: Example of an original, grayscale WT cell image.

We are given labeled images of cells belonging to one of two classes: MM or 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 [25, 13, 21, 20]. Therefore, we expect to the MM and WT cells to differ in their topologies.

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 are simulated with 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 Figure 2(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 $10\,\mu m$. In figure 2, the filaments are modeled as polar worm-like chains in red and blue dots represent barbed ends of these filaments. The locations of the actin beads that make up the filaments, which are shown as small black circles in Fig. 2(b) are the outputs that we are provided. Each actin bead is of radius $0.5\,\mu m$. These actin beads will act as our point clouds in the topological transormations of these simulated data.

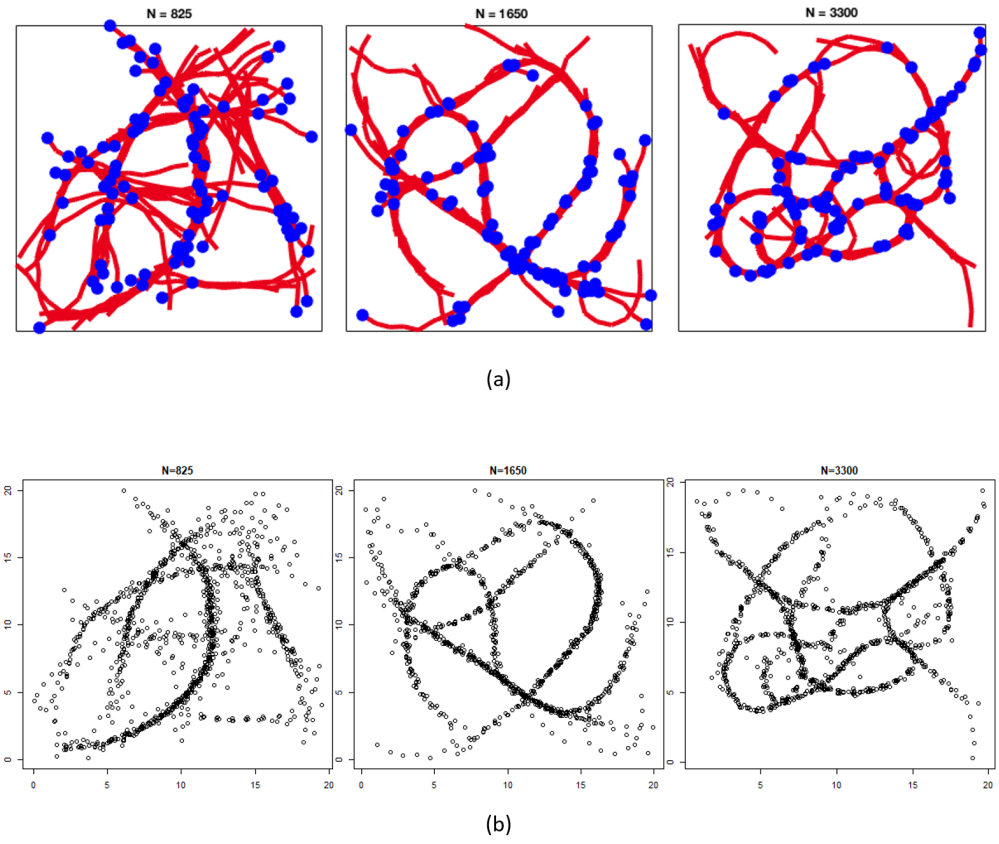

Figure 2: Filament networks. Panel (a) shows three filament networks generated by 825, 1650 and 3300 cross-linkers, respectively, in a 20 µm × 20 µm area. Each network contains 100 filaments which are represented as red lines. The blue dots are the barbed ends of these filaments. Panel (b) shows the locations of the actin beads that make up the filaments exhibited in Panel (a).

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 the image data or an actin bead for the simulated 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 the homological features. 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 the procedure, when radius ϵ is sufficiently large so that the homology group remains unchanged by any further increase to the ϵ, information about a filament network's persistence homology (the set of persistence homology measurements) is summarized in a persistence barcode and/or diagram.

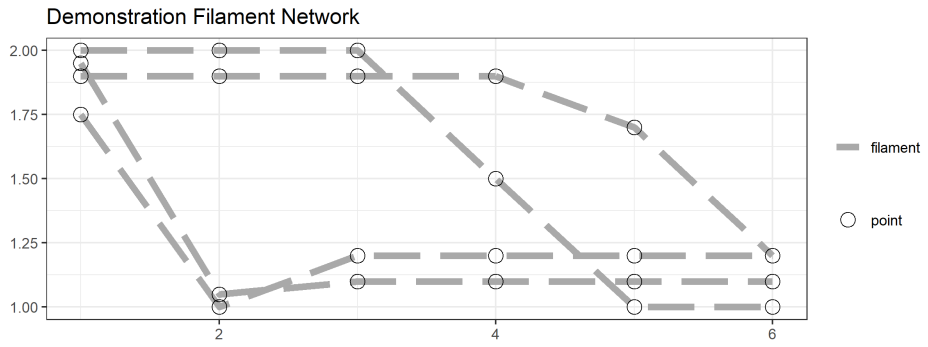

Figure 3: Demonstration fillament network. This network contains 3 filaments. Points are sampled along the filaments simil, in order to produce a point cloud from which persistence homology can be studied.

For clarity, we demonstrate the formation of the persistence on a filament network. We use a simplified filament network which is shown in detail in Figure 3. Figure 4 depicts the process of discovering and summarizing the persistence homology of the simple network shown in Figure 3. These illustrations should make clear the connection between filament networks, the Vietoris-Rips Complex and the encoding of the persistence information.

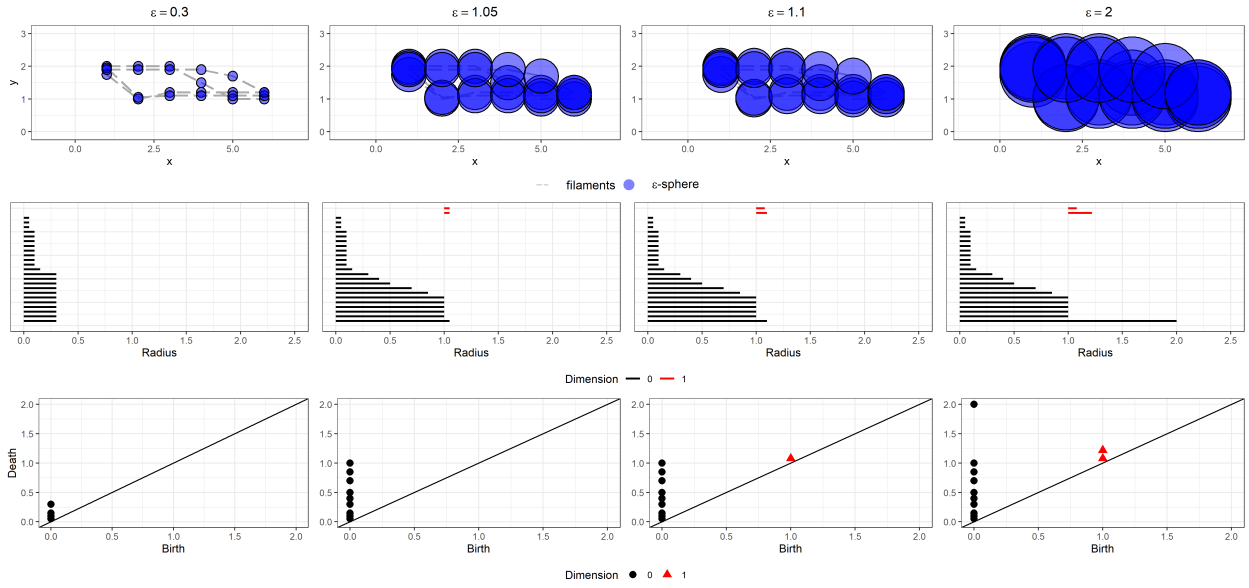

Figure 4: Investigation of the persistence homology of the demonstration filament network in Figure 3. The first row of figures, showes the growing ϵ-spheres about the sampled points of the fillaments. The second row shows the corresponding persistence barcode. The third row shows the corresponding persistence diagram. The columns progress with the algorithm, right-to-left.

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 Figure 4, when $\epsilon = 0.3$, several of the sampled points have already connected. We denote the points which have died (having connected into a larger connected component), by ending their bar in the persistence barcode below. In the third row (of persistence diagrams), we plot a point at 0 (their birth) on the x-axis and at their precise time of death on the

y-axis. When $\epsilon \approx 1$, two holes form. The holes are evident in the top row, second column, of Figure 4, where one can see a larger and a smaller hole. We begin plotting bars for the holes in the corresponding persistence barcode below. Note that there are not yet reccords (points) for the holes in the persistence diagram. This is because we must know the death time of the holes in order to plot them in the 2-dimensional persistence diagram.

In the third column of Figure 4, when $\epsilon = 1.1$, the smaller hole has closed, and become part of a larger connected component. Now a single point (red triangle) is plotted in the persistence diagram below and the corresponding bar in the persistence barcode is terminated.

As the algorithm progresses, the larger hole eventually dies and its corresponding bar in the final persistence barcode is terminated. A point is also added at the corresponding birth and death in the persistence diagram (a second red triangle appears in the final column, bottom row of Figure 4). The algorithm could continue to $\epsilon = \inf$, but it is evident in this example that no more homological information will be discovered as the spheres continue to grow. We choose to terminate the algorithm at $\epsilon = 2$, terminate the final bar in the persistence barcode, and plot a point at (0,2) in the persistence diagram to denote final death time of the connected component which includes all the merged spheres.

Persistence homology indirectly summarizes the hidden shape of the data. In this work, we transcribe the persistence information of each cell to a persistence diagram. Having reccorded the persistence information of the networks, a classifier can be generated either from the distance [15] between persistence diagrams or by alternative vectorizations of the diagrams [1, 4, 16]. Equipped with persistence diagrams, we will now detail several methods of classification built on the diagrams' homological information.

3 Methods

We divide our methods into two subsections. In Subsection 3.1, we describe our methods of sub-sampling the simulated and real data. Since the microscopy data and the simulated data live in two distinct spaces, the sub-sampling procedure varies between the two data spaces. Next, in Subsection 3.2, we describe the classifiers used in this work. These vary primarily by whether a vectorization of the persistence diagram is performed for an ML approach, or a distance is calculated between diagrams for a traditional TDA approach. The same classifiers can be applied to both the simulated and the real data.

3.1 Sub-sampling

3.1.1 Sub-sampling microscopy data

The provided microscopy images are extremely large for the computation of persistence homology. Several images are over $1.25e^6$ pixels (data points). At this time, it is computationaly infeasable to compute persistence diagrams for such images using even a small fraction of the pixels. However, we can efficiently compute the persistence homology of several independent, small sub-samples. Since the intensity of each pixel should be correlated to the probability of the presense of an actin filament in that region of the cell, we can use the pixel intensities as weights on a uniform, random sampler. We take 3 samples, without replacement, of 1000 pixels, where the probability of sampling a pixel is propotionate to its intensity. The coordinates of the sampled pixels then form 3 sub-sampled 1000-point point clouds for which we may calculate a persistence diagram per point cloud.

3.1.2 Sub-sampling of simulated data

The simulated data, representing sets of actin beads, included spatial coordinates as well as an index indicating the filament that the beads belonged to. 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 each network's sampled points are triples (x, y, f), where x and y are the coordinates in the plane and f is the index in the identity map of F.

Using R's 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 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 or 1 for each simulated cell. We filtered these 4000 values to only those of value 1 and used the corresponding planar coordinates to build our point clouds, simplicial complexes and persistence diagrams. All sub-sampling steps are summarized in Algorithm 3.1.2.

Algorithm 1 Sub-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. 5 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. 5.
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. 5.
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. 5 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 5.

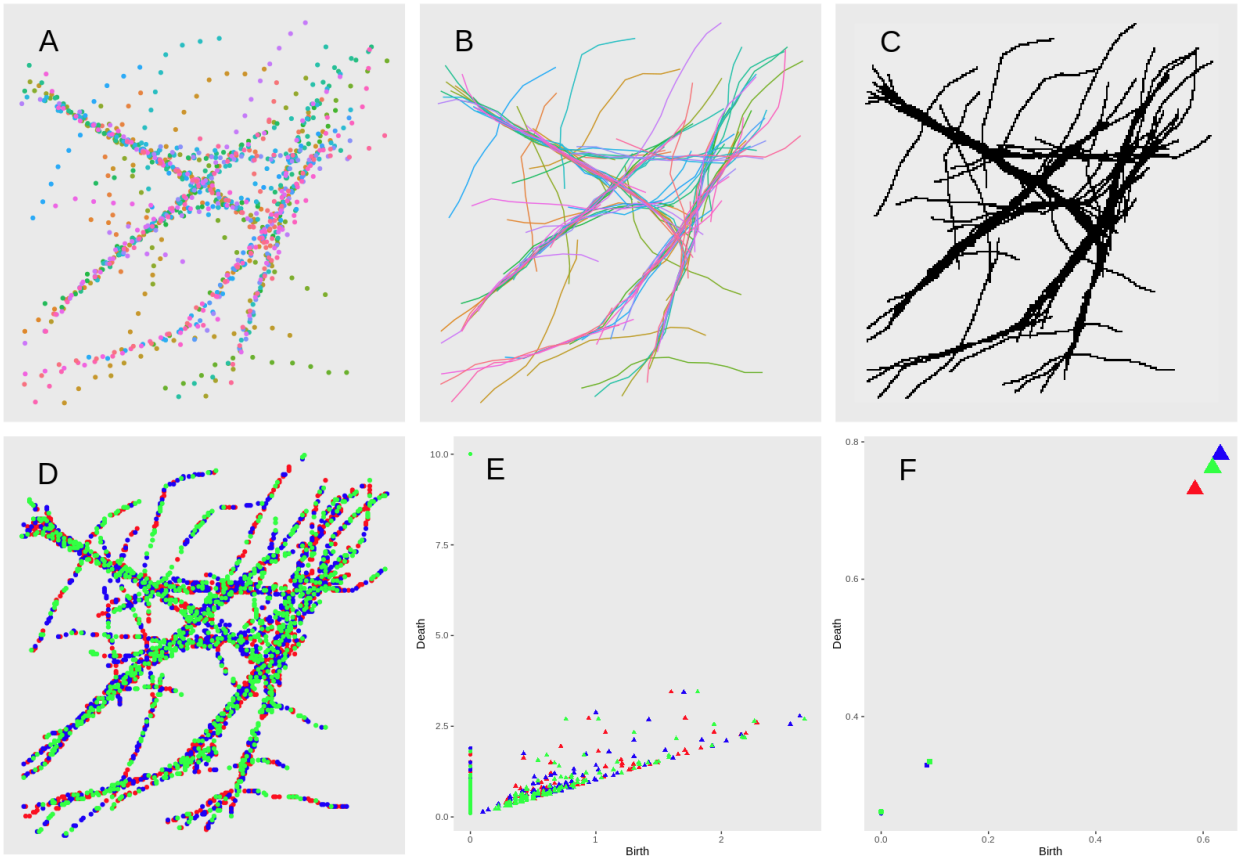

Figure 5: Example of one filament construction (B) from the supplied point cloud (A), rasterization (C), sub-sampling (D), construction of persistence diagram (E) and vectorization of persistence diagram (F). The point cloud in (A) was supplied with an index that allowed for the constrution of the filaments 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.

As in the microscopy sub-sampling procedure, rather than computing a single 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 simulated networks. We computed persistence diagrams for these 450 networks and used these to construct a classifiers.

3.2 Classification

3.2.1 Persistence vectorization (non-distance based)

Our key method performs a classification on vectorized features of the persistence diagram. A matrix was generated from the persistence diagrams which had row entries for each diagram and column features with the mean and standard deviation of the persistence diagrams considering only the 0^{th} or 1^{th} persistence features alone. That is, for each filament network, there is one persistence diagram which corresponds to one row of the vectorized matrix with mean and standard deviation of birth or death times for 0^{th}, 1^{th}, or all features. This gives us 12 columnar features. However, the mean and standard deviation of births for

0^{th} features are constant 0 and are not considered. Therefore, we are left with 10 entries per vector. This process is identical between the simulated and real data. The vectorization procedure is shown in Algorithm 2.

Algorithm 2 Vectorization of persistence diagrams

We vectorize $D_{1,r}$ from Algorithm 3 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 deviation 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.

With the vectorized data we have the freedom to use any number of commmon classification methods. We present results from implementations of neural networks, random forests, and support vector machines (SVMs).

3.2.2 Distance-based classifications

Classifying diagrams in the persistence space, we need a way to quantify the difference between two diagrams. There are two methods commonly used to calculate the distance between persistence diagrams: the Bottleneck and Wasserstein distances [1, 4, 7, 11, 27] directly from the persistence diagram space. 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 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 [15] and has been proved to be stable in [16]. 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 [16].

Since persistence diagrams can summarize the homological features of multiple dimensions in one diagram, such as in the last panel of Fig. 4, the persistence diagram contains both 0-dim features(connected components) 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 3.

Algorithm 3 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,

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

3.3 Prediction

The sub-sampling procedure introduces one idiosyncrasy, which is an ambiguity of how to utilize multiple predictions per network. For classifiers using sub-sampled data, predictions were made for each of the test-fold sub-sampled networks $R_{i,\chi}$. These predictions were considered as votes and the majority label was taken as the final prediction for X_χ. This returns us to a state of one prediction per test-fold network.

4 Results

The cross-validated ($k = 10$) classification of the simulated networks was performed with many common, competing methods. The top performer was a support vector machine (SVM), trained on the sub-sampled and topologically transformed data. The next best performance was achieved by the d_p^c-based classifier. We also provide the accuracy rates of several other classifiers tested in Table 1.

classifier	**mean accuracy**
SVM, re-sampled	94%
d_p^c-based	89%
Wasserstein-based, re-sampled	87%
Wasserstein-based	83%
Bayes Factor [17]	83%
SVM PI [1]	75%
Neural Net PI	71%
SVM Raster	65%
Random Forest Raster	55%

Table 1: Cross-validated accuracy rate of classifiers on simulated actin networks.

Our findings show that sub-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. Since, our re-sampled and vectorized persistence diagram far outperformed all other classifiers, we proceeded to test this method on the microscopy data.

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 ($n = 42$). We find a cross-validated ($k = 5$) accuracy of approximatley 83%. We used fewer folds in testing accuracy on the microscopy data, only because of the relatively samll sample size.

5 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 a topological transformation. Our exploratory work is the first time filament networks have been studied by homological classification. This work should be useful in the course of research on cytoplasmic streaming. This tool provides 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 should be able to fix a network structure while varying parameters specific to myosin.

The microscopy experiment differs from the simulated experiment in that our simulated experiment involved the direct manipulation of the actin network through changes in cross-linker abundance. The groups in the microscopy experimetn differ in by a far less direct manipulation. Whereas cross-linking is fundamental to the structure of the actin network, the linkage betweeen a single type of mysoin motor and resulting actin networks is more complex and likely modulated by several interacting factors. The simulated actin networks were critical to the development of our new tool, both in their clarity and in that they allowed us to benchmark algorithms without overfitting to our small sample of microscopy data.

Interpretation of these results is complicated by our uncertainty of the true, cumulative impact of a myosin knockout on the actin network. While the accuracy of our method on real data is significantly lower than found in the analysis of the simulated 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 more effective than manual classification by an expert, we are left to 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.

5.1 Conclusion

In sum, we successfully classified simulated actin filament networks generated under three different initial conditions with very high accuracy. We combined a machine learning framework with TDA for our classification. We compared this method with several distance-based classifiers. Our most successful method uses a sub-sampling technique, a vectorization of the persistence diagram, an SVM as the classifier and a voting mechanism for the final predictions. We then showed an application of our method to real confocal microscopic data. Our 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.

In future work, we hope that this method will reveal key factors in the filament network organizing process and provide biologists with opportunities to study the interaction of motor proteins, actin networks and cytoplasmic streaming. We demonstrate that it is simple to bridge our method between simulations and real images, laying the groundwork for additional work comparing outcomes of simulations to real experiments.

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] Stephanie L Madison, Matthew L Buchanan, Jeremiah D Glass, Tarah F McClain, Eunsook Park, and Andreas Nebenführ. Class xi myosins move specific organelles in pollen tubes and are required for normal fertility and pollen tube growth in arabidopsis. *Plant Physiology*, 169(3):1946–1960, 2015.

[14] 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.

[15] 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.

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

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

[18] Yuriy Mileyko, Sayan Mukherjee, and John Harer. Probability measures on the space of persistence diagrams. *Inverse Problems*, 27(12):124007, 2011.

[19] 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.

[20] Eunsook Park and Andreas Nebenführ. Myosin xik of arabidopsis thaliana accumulates at the root hair tip and is required for fast root hair growth. *PloS one*, 8(10):e76745, 2013.

[21] Valera V Peremyslov, Alexey I Prokhnevsky, and Valerian V Dolja. Class xi myosins are required for development, cell expansion, and f-actin organization in arabidopsis. *The Plant Cell*, 22(6):1883–1897, 2010.

[22] 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.

[23] 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.

[24] 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.

[25] Haruko Ueda, Etsuo Yokota, Natsumaro Kutsuna, Tomoo Shimada, Kentaro Tamura, Teruo Shimmen, Seiichiro Hasezawa, Valerian V Dolja, and Ikuko Hara-Nishimura. Myosin-dependent endoplasmic reticulum motility and f-actin organization in plant cells. *Proceedings of the National Academy of Sciences*, 107(15):6894–6899, 2010.

[26] 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.

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