Bridging global actin network patterns to local molecular dynamics: a combined modeling and machine learning framework

Minghao W. Rostami[1], Brittany E. Bannish, Kelsey Gasior, Rebecca L. Pinals, Calina Copos*, Adriana T. Dawes*

Abstract

Cells rely on their cytoskeleton for key processes including division and directed motility. Actin filaments are a primary constituent of the cytoskeleton. They can create a variety of network structures linked to cell functions, although the microscale interactions that give rise to these macroscale structures are not well understood. In this work, we investigate the microscale mechanisms that produce specific actin network architectures. Identifying these mechanisms can be viewed as a classification problem where the inputs are the measurable network density and geometry, and the output is the combination of molecular mechanisms underlying that network configuration. We develop a two-pronged approach. The first prong is a simple yet comprehensive, agent-based model that simulates the growth of actin networks with precise control over the microscale dynamics. The second prong is applying machine learning techniques to classify actin networks. Extensive computational experiments reveal that the most accurate method uses a combination of supervised learning based on network density and unsupervised learning based on network symmetry. This framework can potentially serve as a powerful tool to discover the molecular interactions that produce the wide variety of actin network configurations associated with normal development as well as pathogenic conditions such as cancer.

Keywords: Actin, branched actin network, lamellipodium, machine learning, agent-based model, cell cytoskeleton.

1. Introduction

Actin, the most abundant protein in eukaryotic cells, is involved in a wide range of key cellular functions including cell motility, cell differentiation, muscle contraction, and cytokinesis (Michelot and Drubin, 2011, Schwarz and Gardel, 2012, Svitkina, 2018). Actin monomers form rod-like actin filaments, which are constantly assembled, disassembled, and remodeled by accessory proteins and molecular motors into networks. As a consequence of the noise-dominated complex dynamics, actin networks have a variety of architectures with distinguishable biochemical and mechanical properties. Notably, the cell cytoskeleton, which provides

*Corresponding authors
Email addresses: c.copos@northeastern.edu (Calina Copos), dawes.33@math.ohio-state.edu (Adriana T. Dawes)
[1]First author

the mechanical support for the cell and is the main determinant of cellular rheology, is composed of actin filaments, microtubules, intermediate filaments and over 100 associated proteins (Moeendarbary et al., 2013). Not surprisingly, defects in actin networks are often correlated with disease states (Clarkson et al., 2004). Measurements using atomic force microscopy show that cancerous human bladder cells are softer than non-malignant bladder cells, implying that the integrity of actin structures is altered (Lekka et al., 1999). Studies of actin-related primary immunodeficiencies have also pointed to the mounting evidence of the pivotal role of actin structures in a successful immune response (Moulding et al., 2013). Despite the implications that actin structures have on cellular functions, a direct and comprehensive link from local molecular dynamics to network organization to cellular behavior has yet to emerge.

A means to probe this connection is to measure intracellular forces with high spatiotemporal resolution at the same time as visualizing filamentous actin structures by fluorescence microscopy (Svitkina and Borisy, 1999). Simultaneously, the *in vitro* reconstitution of purified proteins has also been a powerful tool for identifying the molecular conditions that are necessary and sufficient determinants of actin structure organization (Liu et al., 2008, Ierushalmi et al., 2020, Reymann et al., 2012). To complement these existing experimental approaches, we propose a computational technique that combines stochastic agent-based simulations based on known molecular interactions, machine learning algorithms, and image processing tools. When combined with experiments, this approach allows us to identify targetable processes at the molecular level that lead to a network organizational change which in turn affects the cellular function.

In this work, the focus is on branched actin structures observed in protrusions, such as the lamellipodium, for cell spreading and locomotion (Theriot and Mitchison, 1991). Branched (or protrusive) networks emerge as a result of the complex interplay between various known molecular processes listed next (Svitkina, 2018). Actin filaments can grow or shrink through the addition or loss of actin monomers from their pointed and barbed ends. Filament branching occurs when an Arp2/3 protein complex binds to a filament and creates a nucleation site for a new filament to extend from the existing one at a 70° angle (Svitkina and Borisy, 1999, Young et al., 2004). Capping proteins can prevent the additional growth of an actin filament by binding to the barbed end. Depending on the intracellular environment that actin networks grow in, including the availability and distribution of resources (actin monomers) and regulators (capping proteins and Arp2/3 complexes), branched actin structures can exhibit vastly different shapes and internal organizations.

We present a novel computational pipeline to link local molecular processes that take place during the local formation of a branched structure and the resulting global network architecture. The pipeline is a two-pronged approach based on synthetic networks with prescribed local dynamics and statistical learning tools. The first prong is an agent-based stochastic model that simulates the construction of a protrusive actin structure in two dimensions. Building on our previous published work, we create a stochastic model that can generate actin structures by tuning various interactions and parameters (Copos et al., 2021). A primary benefit of this *in silico* approach over traditional *in vitro* or *in vivo* approaches is that networks can be generated quickly and inexpensively while molecular processes and parameters are controlled in a way

that is challenging in an experimental setting.

The computer-generated actin networks are then used in the second part of our approach, which is to apply machine learning techniques to identify the primary dynamics that are prescribed to give rise to a particular architecture. We formulate this process as a classification problem in which we classify images of networks to the prescribed underlying local molecular interactions. We leverage a variety of machine learning algorithms that have been successful at image classification including convolutional neural networks (CNNs), which specialize in processing data on a uniform grid, like our actin density data. We perform both fine-grained classification and coarse-grained classification; the former aims to "pinpoint" the exact combination of molecular processes that give rise to a specific actin structure, whereas the latter uncovers the distinct dominating mechanisms. The coarse-grained classification is developed based on the observation that different mechanisms can lead to very similar network architectures, making fine-grained classification unreliable. It is a notable approach since it combines supervised and unsupervised learning methods, specifically, k-means clustering and convolutional neural network, and utilizes both raw and symmetry transformed data. Using simulated data, we find that the fine-grained classifier has an accuracy ranging between 66% and 90% depending on the network growth condition, while the coarse-grained classifier boasts an accuracy of 87% – 99%. The unsupervised re-grouping of the growth conditions into coarse-grained classes informs us that limiting capping proteins and branching complexes is a primary driver of network architecture. These results indicate that our method could serve as an unbiased tool to analyze experimentally-obtained intracellular actin networks in healthy and diseased cells, and more broadly, to uncover the connections between local rules of interactions and global network morphology for other biological systems.

2. Methods

2.1. Generation of branching actin networks

In our previous work, we built a tractable agent-based computational framework to capture the local micro-structure of branching actin networks under various intracellular conditions (Copos et al., 2021) (top of Figure 1). The framework uses a molecular-level representation of dynamically assembling and disassembling polymers (actin filaments) in two dimensions. In this work, in order to capture more of the biological complexity of branching networks, the framework is extended to include capping and limited molecular resources in addition to polymerization, depolymerization, and branching of polar actin filaments.

We briefly summarize the dynamics used to generate synthetic actin networks (see Figure 2A for an illustration). A branching actin network is formed from a single nucleation site in an unbounded two dimensional domain. By excluding the third spatial dimension in our modeling framework, the branching actin network is assumed to be relatively flat as observed in lamellipodia (Laurent et al., 2005, Urban et al., 2010). Filaments are represented as rigid rods with a spatially fixed base and a barbed tip capable of growing or shrinking. Changes in both actin filament length and overall network structure are due to the addition

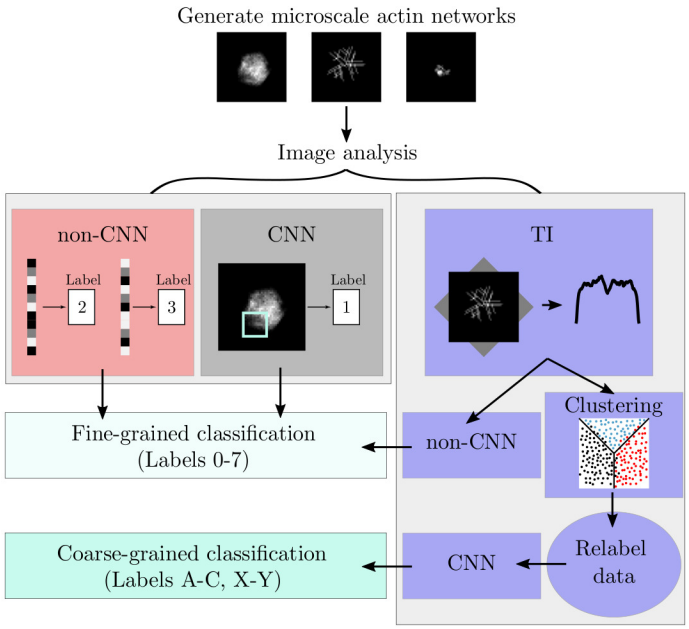

Figure 1: Outline of the analysis pipeline for identifying the governing molecular-level dynamics of observed architectures of actin networks. A detailed description of the analysis, which combines agent-based modeling, machine learning methods and image processing, is in Section 2.

or removal of single actin monomers and branching via branching proteins (Arp2/3 nucleation complexes) at the free end. Filament growth and shrinkage occur in discrete increments corresponding to the length of an actin monomer, 0.0027 μm (Lodish et al., 2000). Based on the known effect of the Arp2/3 complex, the newly branched filament is assigned to grow in a direction that is 70° on either side of the preexisting filament tip (Goley and Welch, 2006). The simulation framework also allows for capping proteins to regulate actin filament length by binding to the barbed end of a filament, which blocks the addition or loss of monomers from that filament (Edwards et al., 2014). The concentration of free monomers and branching proteins can be either unlimited or limited; both cases are considered in our work. Additional implementation details are available in Appendix A.1 along with model parameters and computational constants found in Table A1 and explained in Appendix A.2. The simulation and analysis of actin networks in this work are implemented in a custom MATLAB code, which will be made available upon request.

The spatial domain of each simulation is a 10 μm × 10 μm square, which is comparable to the size of eukaryotic cells. To extrapolate a corresponding density from a discrete actin network architecture, the computational domain is uniformly subdivided into 49×49 grid of size 0.2 μm × 0.2 μm. The choice of grid resolution represents a balance between computational cost and underlying filament and monomer sizes. At the end of a given simulation, we calculate the total length of actin filaments contained in each discretized grid box. Since this length is a scalar multiple of the number of actin monomers contained in a grid box, we take the length divided by the area of the grid box, 0.04 μm^2, to be the actin density at the center of that

Label	Capping	Branching	Monomers
0 ●	×	Unlimited	Unlimited
1 ●	✓	Unlimited	Unlimited
2 ●	×	Limited	Unlimited
3 ●	×	Unlimited	Limited
4 ●	✓	Limited	Unlimited
5 ●	✓	Unlimited	Limited
6 ●	×	Limited	Limited
7 ●	✓	Limited	Limited

Table 1: Eight possible fine-grained growth conditions for actin networks

box. The result is a 49×49 grayscale image of actin densities, where the brighter a pixel is, the higher the actin density is at that location.

The data used to train and validate the classification algorithms consists of density images of actin networks generated under eight possible growth conditions labeled 0, 1, 2,..., 7, that is, all combinations of capping or no capping, limited or unlimited Arp2/3 branching complexes, and limited or unlimited actin monomers (see Table 1). For each growth condition, 300 independent simulations are run, totaling 2400 simulated actin networks to be analyzed and classified for their dominating dynamics. The networks are divided into training, validation, and test sets as follows. A group of 75 networks for each growth condition (600 networks total) are randomly selected for the training data. From the remaining networks, we randomly select a group of 25 networks per condition (200 networks total) can be used for validation. The still remaining networks (200 per condition, 1600 networks total) comprise the test data. We emphasize that the training, validation, and test sets are disjoint and produced by independent runs of the microscale model.

2.2. Fine-grained actin network classification

To assign a class or label to a given actin network as resulting from one of the eight growth conditions, we build classifiers using supervised machine learning techniques based on the images of actin densities and compare their performance. Foreshadowing the findings in the next section, we refer to this type of classification as "fine-grained."

2.2.1. Classification based on density

The supervised machine learning algorithms considered are: convolutional neural network (CNN) (LeCun, 1989), support-vector machine (SVM) (Cristianini and Shawe-Taylor, 2000), k-nearest neighbors (k-NN) (Altman, 1992), ensemble algorithms (Breiman, 1996, 2001, Ho, 1998, Seiffert et al., 2008), and neural network (NN) (McCulloch and Pitts, 1943). We use "non-CNN" to mean supervised machine learning methods other than CNN. All classifiers are built for the two-dimensional actin density plots described in Section 2.1. The CNN classifier uses the density data in its natural dimension, a 49×49 matrix (image), whereas the non-CNN classifiers use the same data arranged into a $49^2 \times 1$ vector. We proceed as follows:

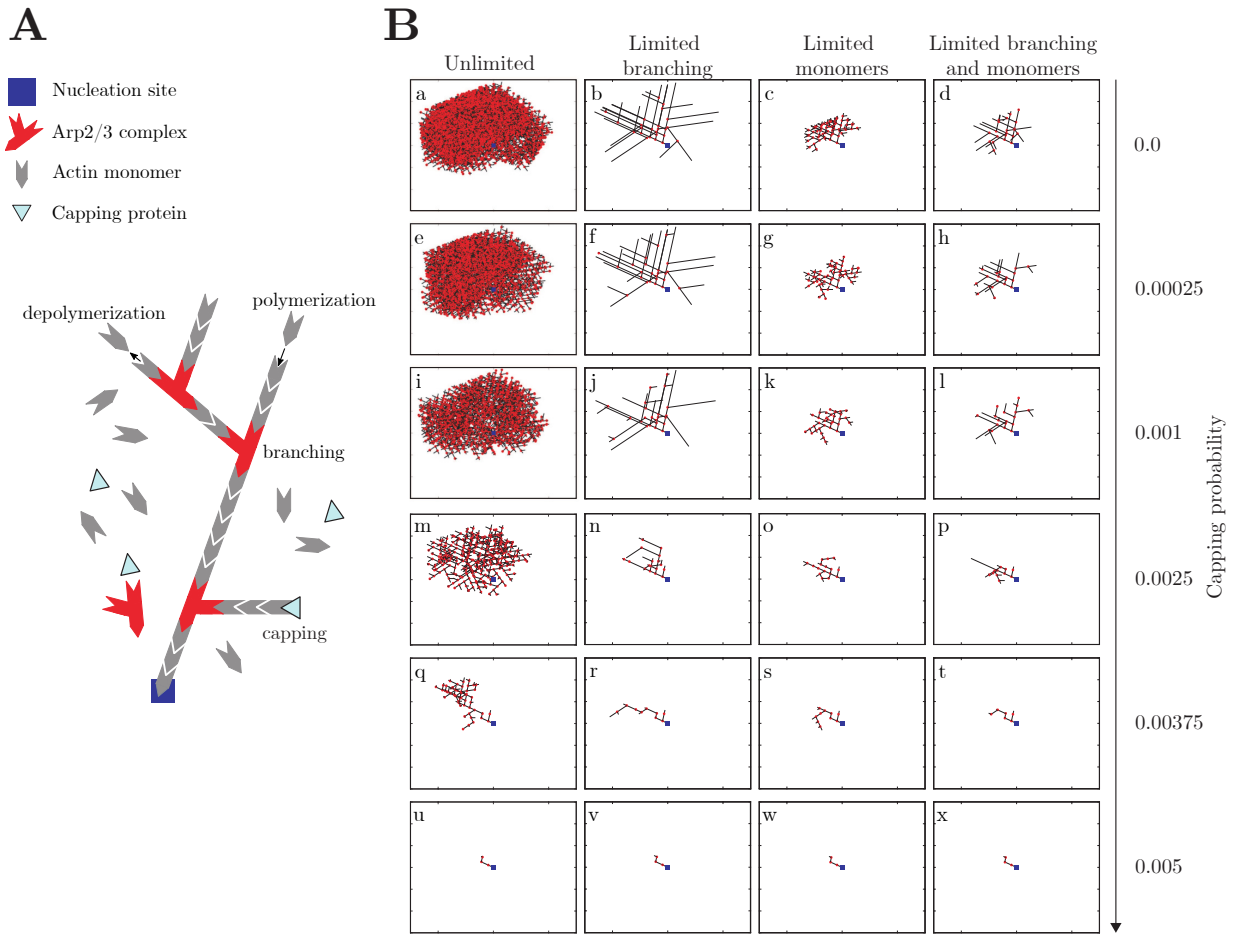

Figure 2: (A) Schematic of actin kinetics incorporated in our simulation framework. Actin filament growth begins at the initial nucleation site (blue) through polymerization. Monomers have a polymerization probability of p_{poly} and a depolymerization probability of p_{depoly}. Attachment of Arp2/3 complex (red) to the filament facilitates Y-branching. Association or disassociation of monomers to/from a filament can be halted by the binding of capping proteins (light blue) with probability p_{cap}. (B) Sample actin network architectures arising from various growth conditions. All networks are shown after 10 s with a simulation domain of a 10 μm × 10 μm square. For ease of comparison, the initial angle of growth from the nucleation site was kept constant between all simulations.

Step 1 Generate training data: The training data is composed of density plots of 600 actin networks simulated by the same number of independent runs of the microscale model (75 networks per growth condition) and labeled by their classes $0, 1, 2, \ldots, 7$ according to their respective growth conditions.

Step 2 Train classifiers: We consider CNN and 22 non-CNN classifiers:

- six SVM variants: linear, quadratic, cubic, fine Gaussian, medium Gaussian, and coarse Gaussian;
- six k-NN variants: fine, medium, coarse, cosine, cubic, and weighted;
- five ensemble algorithms: boosted trees, bagged trees, subspace discriminant, subspace k-NN, and random undersampling boosted trees;

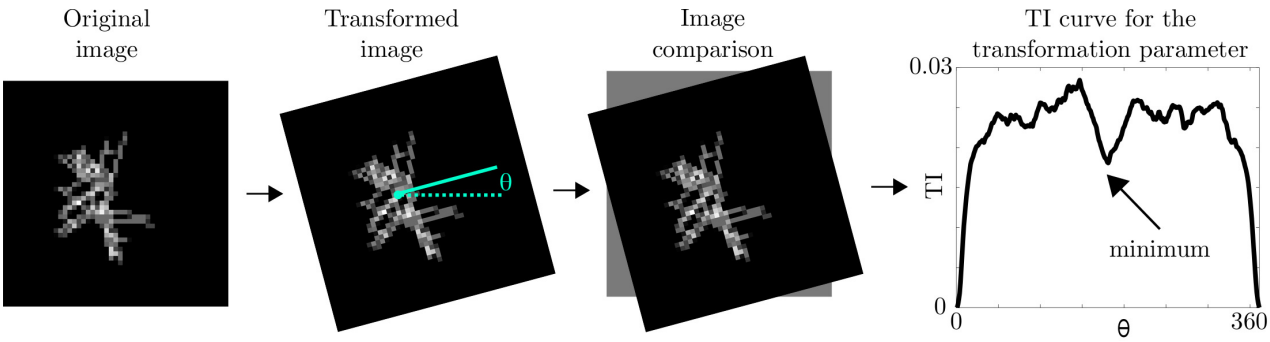

Figure 3: Schematic of the image processing method for quantification of approximate symmetries in the actin density images using transformation information. The image is modified using an affine transformation (in this case, rotation by angle θ). The original and transformed images are compared to determine the amount of information lost by using the newly transformed image to approximate the original. Minima of the TI curve indicate approximate symmetries in the actin network.

- five NN variants: narrow, medium, wide, bilayered, and trilayered.

MATLAB's Classification Learner App[2] is used for all non-CNN classifiers and the CNN classifier is implemented using a custom code using built-in functions from MATLAB's Deep Learning Toolbox [3] and Statistics and Machine Learning Toolbox[4]. All 23 classifiers are trained on networks in the training data and the outcome is a predicted label for each network, that is, an integer between 0 and 7. During training, the parameters of each machine learning classifier are optimized so that the predicted label agrees with the true label for as many actin density plots in the training data as possible.

Step 3 Evaluate performance: We calculate and compare the accuracy of all classifiers on data not used in the training process. A classifier is successful at labeling an actin density plot if its predicted label agrees with the true label and unsuccessful otherwise. The percentage accuracy of a classifier is defined as

$$\frac{\text{number of actin density plots correctly labeled by the classifier}}{\text{total number of actin density plots}} \cdot 100\%. \tag{1}$$

First, we calculate the percentage accuracy of the 22 non-CNN classifiers on the validation data composed of 200 images of actin density (25 per growth condition). This allows us to identify the best non-CNN classifier as the one with the highest percentage accuracy. Then, to compare the best performing non-CNN classifier and the CNN classifier, we compute their respective percentage accuracy on the test data, which consists of 1600 actin density plots (200 per condition).

[2] `https://www.mathworks.com/help/stats/classificationlearner-app.html`
[3] `https://www.mathworks.com/help/deeplearning/index.html`
[4] `https://www.mathworks.com/help/stats/index.html`

Capping probability	Fine-grained non-CNN (best)	Fine-grained CNN	Fine-grained TI-based non-CNN (best)	Coarse-grained CNN
0.00025	69% (quadratic SVM)	82%	53% (trilayered NN)	99%
0.00050	84% (linear SVM)	90%	64% (linear SVM)	98%
0.00100	81% (quadratic SVM)	89%	66% (quadratic SVM)	96%
0.00150	77% (medium Gaussian SVM)	84%	63% (median NN)	87%
0.00250	67% (narrow NN)	74%	61% (narrow NN)	87%
0.00375	65% (medium Gaussian SVM)	66%	52% (quadratic SVM)	98%

Table 2: Accuracy of machine learning classification of actin network architectures

2.2.2. Quantification of shape symmetry

To identify geometric properties of simulated actin networks, we calculate the transformation information (TI) associated with each network density plot as outlined in Gandhi et al. (2021). The TI function quantifies the amount of information lost by approximating the original image by the transformed image. More specifically, the value of the TI function measures how approximate the symmetry is, with minima of the TI function indicating the transformation that results in the least change from the original image. Here, an affine transformation corresponding to rotation is used to quantify the symmetry properties in the actin network. The function $\mu(\mathbf{x})$ is the actin density at location $\mathbf{x}$ in each image. The steps for calculating the TI curve are outlined below and shown graphically in Figure 3:

Step 1 Obtain TI measurements by:

$$TI = \frac{1}{|\widetilde{D}|} \int_{\widetilde{D}} \mu(\mathbf{x}) \ln\left(\frac{\mu(\mathbf{x})}{T_\theta \mu(\mathbf{x})}\right) dA, \tag{2}$$

where T_θ is the rotation of the image by angle θ, $\widetilde{D}$ is the domain intersection of the original and transformed images, $|\widetilde{D}| = \int_{\widetilde{D}} dA$, and dA is the area element. We repeat the calculation and compute the TI for each increment of the transformation parameter, angle θ, to obtain TI as a function of angle θ.

Step 2 Identify approximate symmetries: Find and rank minima of the TI curve in order to identify approximate symmetries of the actin density plots. For example, a TI value of 0 indicates perfect symmetry.

Further details of the TI calculation and MATLAB code can be found in Gandhi et al. (2021).

2.2.3. Classification based on shape symmetry

For each plot of the actin network density, the TI calculation produces a 361×1 vector (or a curve); the ith entry of the vector is the TI measurement for the transformation parameter $\theta_i = (i-1)$ degrees for $i = 1, 2, \ldots, 361$. The non-CNN supervised classifiers can also be applied to the TI curves extracted from

the actin density plots. The process is similar to the one described in Section 2.2 and the TI curves carry the same fine-grained labels $0, 1, \ldots, 7$ associated with the original actin density plots and their respective growth conditions.

2.3. Coarse-grained actin network classification

Different growth conditions may result in similar network architectures. Therefore, fewer, more identifiable classes are sometimes desired. In order to reveal governing motifs of emergent network architectures, we propose a methodology known as "coarse-grained" classification which results in fewer than eight classes. Our approach combines supervised and unsupervised machine learning techniques; first, an unsupervised algorithm is applied to identify the new, fewer than eight, groupings in the dimensionally-reduced TI curves extracted from actin density plots, and then a supervised learning algorithm is used to build a classifier that determines which new grouping (or label) to assign to a particular actin density plot. The procedure is outlined below.

Step 1 Identify new classes: To abstract away fine-level detail, we use principle component analysis (PCA) to transform each TI curve (a 361×1 vector) to a 2×1 vector referred to as the "reduced" TI curve. The reduced TI curve consists of the two leading coefficients given by PCA. Next, we apply a widely used unsupervised clustering algorithm, called k-means, to identify the number of groupings (clusters) of the reduced TI curves (MacQueen, 1967, Hastie et al., 2004). We use MATLAB's built-in function for k-means to perform unsupervised clustering[5]. The algorithm is repeated over a number of clusters from $k = 1$ to $k = 12$. A value for number of clusters k, smaller than eight, is chosen based on the minimum of the sum of magnitudes between the each data point and its group centroid. This process ensures that we choose a relatively small k that is still large enough for the k clusters to be identifiable. We then proceed to divide the original eight growth conditions into $k < 8$ groups. Growth condition i is placed in the jth group if the majority of the reduced TI curves originally labeled with i are placed in the jth group by k-means.

Step 2 We relabel each actin density plot according to the correspondence between the original eight growth conditions and the new k groups identified in Step 1. As an example, consider the case where $k = 3$ and the three groups are labeled A, B, C. If the original label 4 is deemed a member of the group A in Step 1, then all the density plots previously labeled with 4 will be relabeled with A.

Step 3 Rebuild and re-evaluate the CNN classifier: Repeat Steps 2 to 3 in Section 2.2.1 using the relabeled data.

[5] `https://www.mathworks.com/help/stats/index.html`

3. Results

We present a robust computational pipeline to associate an actin network observed microscopically with its underlying molecular kinetics. In the pipeline, an actin network is fed through the fine-grained classifier to identify molecular growth conditions (Sections 2.2.1 and 2.2.3) or through the coarse-grained classifier to identify primary dynamics (Section 2.3). However, training the supervised statistical methods for accurate classification may require a large set of networks labeled according to their underlying molecular kinetics. To rapidly generate thousands of actin networks with prescribed dynamics, we employ our molecular kinetics-based modeling framework outlined in Section 2.1.

3.1. Stochastic simulations produce actin networks with a variety of architectures, particularly in the presence of limited resources

In our microscale model, we can easily adjust parameter values to generate diverse actin networks, some of which are shown in Figure 2B. For the networks presented here, the polymerization and depolymerization probabilities are held constant across simulations, while the capping probability is varied (see Appendix A.2 for details).

As expected, the addition of capping proteins and limited resources to the actin kinetics modeling framework, individually or in combination, changes the resulting network architecture (Figure 2B). With unlimited resources and no capping, a dense and highly branched network forms (Figure 2B.a). The network remains dense and branched for low capping probabilities (p_{cap} = 0.00025 and p_{cap} = 0.001), but as the capping probability increases further, polymerization and depolymerization are halted, resulting in much smaller, sparser networks (Figure 2B.m,q,u). In cases with low or no capping, the addition of limited Arp2/3 branching complexes (Figure 2B.b,f,j) results in longer filaments once all the Arp2/3 complexes are used; with no additional Arp2/3 complexes, no further branching can occur, so filaments only polymerize and depolymerize. These longer filaments are not seen at higher capping probabilities (Figure 2B.n,r,v) because filaments are capped before they can grow too long. Regardless of the capping probability, simulations with limited monomers result in significantly smaller networks (Figure 2B.c,g,k,o,s,w); this is because once all the actin monomers are used, no further bulk growth can occur and only slight structural modifications to the network continue, as individual monomers depolymerize and re-polymerize elsewhere in the network. Finally, in cases with both limited Arp2/3 complexes and limited monomers (Figure 2B.d,h,l,p,t,x), networks are spatially reduced, with emergence of some longer filaments at low capping probabilities (Figure 2B.d,h,l,p) that are not present at higher capping probabilities (Figure 2B.t,x).

3.2. Extreme capping rates obscure the role of other dynamics when classifying network structures using standard machine learning techniques

Our first attempt at constructing a classifier relies on standard supervised machine learning algorithms: support-vector machines, k-nearest neighbors, ensemble algorithms, and neural networks. These methods

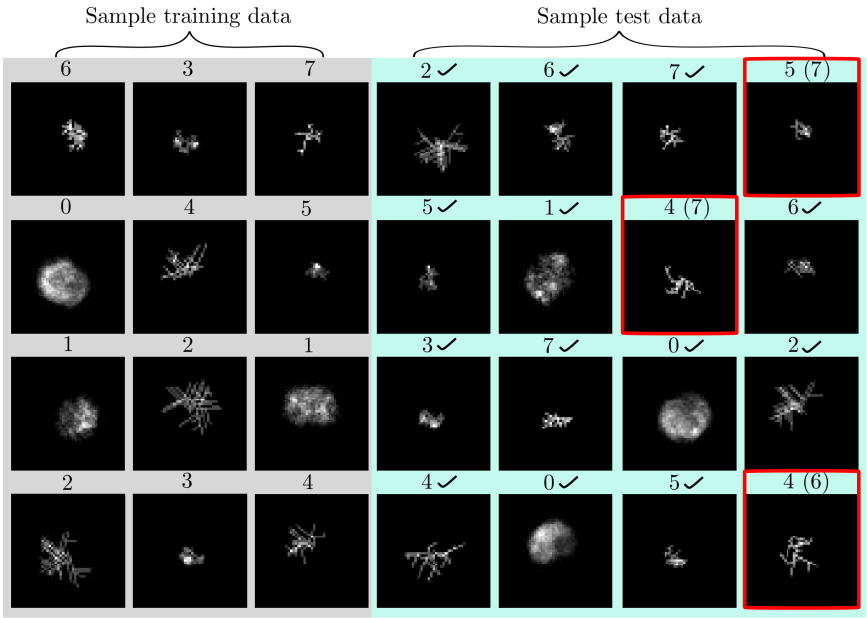

Figure 4: Randomly selected actin density plots and their corresponding fine-grained labels. A brighter pixel indicates a higher actin density. Across all images, the capping probability is held fixed at an intermediate value ($p_{\text{cap}} = 0.001$). The 12 plots (left, gray) are part of the training set used for tuning the classifiers. The 16 images (right, blue) are from the test set used for evaluating and comparing the classifiers. A check mark indicates that the CNN classifier accurately predicts the true label. The density plots wrongly classified by CNN are outlined in red. For any incorrect CNN predictions, the true labels are listed in parentheses.

rely on statistical analysis and numerical optimization to predict labels (in our case, growth conditions) for given data points. For each capping probability listed in Table 2, we generate 300 actin density plots for each of the eight growth conditions with labels $j = 0, 1, 2, \ldots, 7$. The generated data is partitioned into training, validation, and test sets as described in Section 2.1. The parameters of the classifiers are fine-tuned on the training set. The leftmost three columns in Figure 4 show 12 actin density plots randomly selected from the training set and their labels when the capping probability is 0.001. Once the training step is completed, the accuracy of each algorithm is evaluated on the validation set to determine which classifier has the highest accuracy. The accuracy of this classifier for the test data is reported in Table 2 for each of the six capping probabilities considered. We find that the accuracy of the best non-CNN classifier ranges between 65% and 90% depending on the capping probability. Interestingly, the highest accuracy is obtained with mid-range capping probabilities ($0.0005 - 0.0015$). That is, at extreme capping rates, the underlying dynamics are not fully uncovered by any of the non-CNN classifiers.

3.3. Actin networks grown under widely varying conditions are distinguishable when both actin density and network geometry are evaluated

To improve the overall accuracy of the classification pipeline in the previous section, we turn our attention to convolutional neural networks (CNNs). This class of statistical learning methods have been highly successful at detection of visual imagery (LeCun, 1989, Ciresan et al., 2012, Krizhevsky et al., 2017). Motivated by these findings, we train and evaluate the performance of a CNN classifier as implemented in Section 2.2.1. The validation data is used to inform the choice of CNN hyper-parameters such as the number of network layers and network nodes. Lastly, the accuracy of the CNN classifier on the test set is summarized in Table 2. The accuracy ranges between 66% and 90%, which is an improvement over the best non-CNN classifier for every capping probability. Like the non-CNN classifiers, the CNN classifier is also more accurate when the capping probability is neither too high or too low. In the rightmost four columns in Figure 4, for a fixed capping probability of 0.001, we show 16 actin density plots randomly selected from the test set, along with their true labels and the labels predicted by the CNN classifier if the two are different.

Since the CNN classifier uses image data (rather than vectorized data) and hence preserves spatial information of the simulated actin networks, the better performance of the CNN classifier over the non-CNN classifiers suggests that the geometric properties of the network may be important for classification. For higher capping probabilities such as 0.00375, none of the classifiers, CNN or non-CNN, are able to uncover the growth condition with high accuracy. We speculate that this is because when the capping probability is high, actin networks generated under different growth conditions can approach a similar morphology.

3.4. Symmetric properties of actin networks alone are insufficient to distinguish underlying generating mechanisms

To test the hypothesis that only geometric properties of the actin networks may be important for accurate classification, we use a novel and robust measure of symmetry that is grounded in concepts of entropy and information theory called transformation information (Gandhi et al., 2021, Vstovsky, 1997). We apply this method to quantify asymmetries of actin density plots under image transformations parameterized by the rotation angle θ as outlined in Section 2.2.3 (Figure 3). The outcome is a TI curve associated with each network as a function of angular rotation of the network.

Next, we apply the same standard non-CNN machine learning techniques from Section 3.2 to the TI curves extracted from the same data. That is, we train and evaluate the performance of 22 non-CNN classifiers on TI curves of actin networks grown under the six capping probabilities considered previously. For each capping probability, we identify the best non-CNN classifier as in Section 3.2 and report its accuracy on the test set in Table 2. The accuracy ranges between 52% and 66%. Similar to the density-based classifiers, it performs worse for the extremes of the capping probability range. Surprisingly, the best symmetry-based non-CNN classifier is always less accurate than the best density-based non-CNN classifier, indicating that

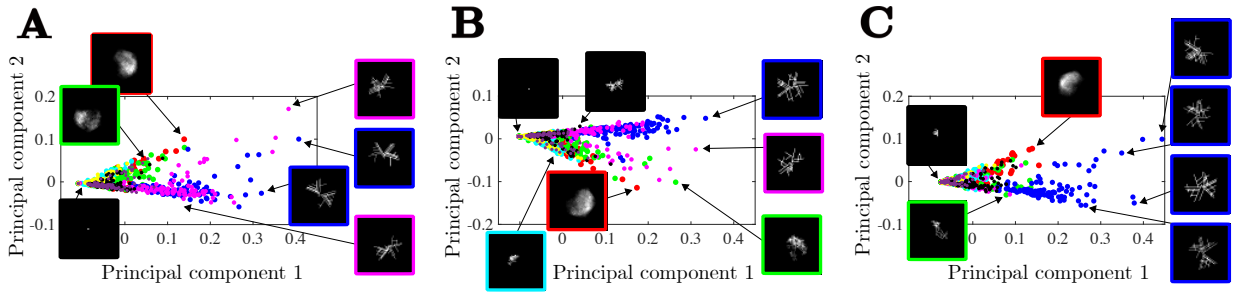

Figure 5: Visualization of the reduced TI curves extracted from actin density plots in the training set. Principal components capture key shapes corresponding to "round" versus "spiked-edge" networks. Each dot represents an individual image, and representative images are indicated. Label 0: red; label 1: green; label 2: blue; label 3: cyan; label 4: magenta; label 5: yellow; label 6: black; label 7: purple. (A) Low capping probability, $p_{\text{cap}} = 0.00025$. (B) Intermediate capping probability, $p_{\text{cap}} = 0.001$. (C) High capping probability, $p_{\text{cap}} = 0.00375$.

symmetry of the network alone is insufficient to reveal the mechanisms underlying the growth conditions of an actin network.

To investigate the potential role of network symmetry, we would like to visualize the TI curves but cannot easily do so due to the large dimension of the data. A dimension reduction by linear principal component analysis (PCA) is performed on the TI curves. The first two principal components of the linear PCA of the TI curves are calculated, and they account for at least 93% of the variance in the data. The visualization of the symmetry information in a reduced two-dimensional space in Figure 5 leads us to hypothesize that the symmetry information can be used to identify coarser level categorization of the data. For example, for the case of low capping probability in Figure 5A, networks with labels 2 (blue) and 5 (magenta) appear to have a similar "vascular" topology. Likewise, networks with labels 0 (red) and 1 (green) share a similar "round" topology, while networks with label 6 (black) are relatively small. Similar conclusions can be drawn for cases with intermediate and high capping probabilities in Figure 5B and 5C respectively.

3.5. Symmetry-informed unsupervised clustering reveals three dominant network architecture groupings

Based on the visualization of the dimensionally reduced TI curves of actin density plots, we postulate that there exists a smaller set of dominant dynamics contributing to a given network architecture. We apply k-means clustering algorithm to the reduced TI vectors extracted from the training set to identify a partitioning of the data into k groups, where k is an integer between 1 and 12. In Figure 6A, for three representative capping probabilities 0.00025, 0.001, and 0.00375 respectively, we show the sum of distances between each reduced TI vector and the centroid of the group that it belongs to as the number of groups increases from $k = 1$ to $k = 12$. These plots suggest that $k = 3$ achieves a good balance between the tightness of the groups and the number of groups. The subplots in Figure 6B show the three groups of reduced TI vectors partitioned by k-means for each capping probability. Interestingly, we find that the grouping appears to be predominantly along the first principal component. For capping probabilities of 0.00025 and 0.001,

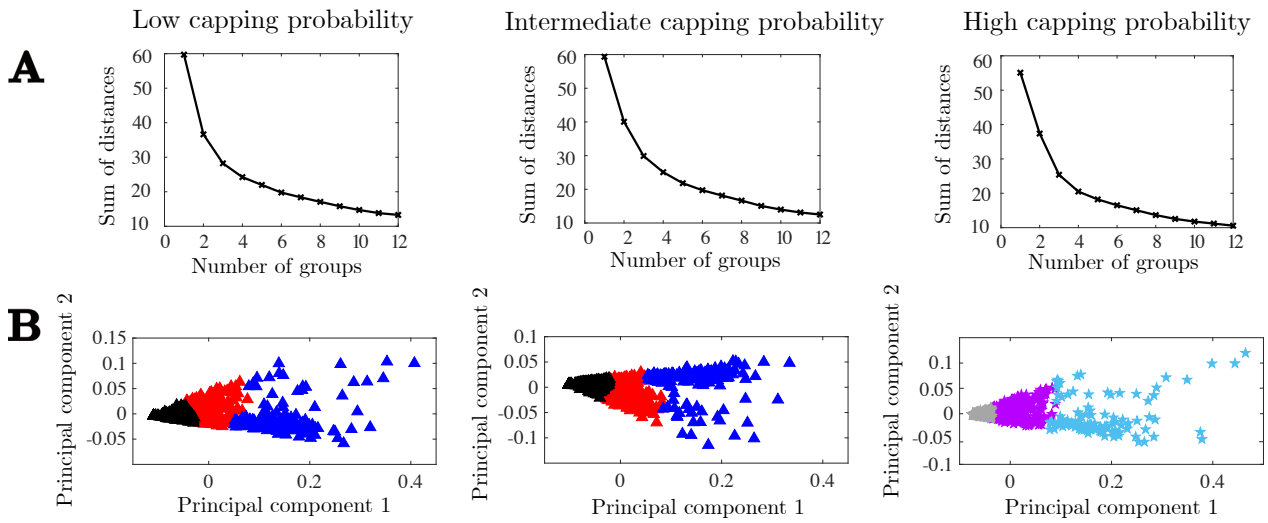

Figure 6: Identification of coarse-grained labels through unsupervised clustering of the reduced TI curves extracted from the training set. (A) The sum of the distances between the reduced TI vectors and their respective group centroid, as a function of the number of groups used in the k-means algorithm. (B) Partition of the reduced TI vectors into three groups by k-means. Each triangle or star represents the reduced TI vector extracted from an actin density plot. Label A: red triangle; label B: blue triangle; label C: black triangle; label X: purple star; label Y: gray star; label Z: light blue star. Low, intermediate, and high capping probabilities correspond to p_{cap} = 0.00025, 0.001, and 0.00375, respectively.

we label the three clusters as "A", "B", and "C" and the correspondence between the eight original labels and the three new labels is in Table 3. For low and intermediate capping probabilities, whether and which resources are limited are the main factors in determining the network architecture. In particular, growth conditions with unlimited resources are labeled "A", growth conditions with limited Arp2/3 complexes and unlimited monomers are labeled "B", and growth conditions with limited actin monomers are labeled "C".

In the case of high capping probability, we label the three clusters as "X", "Y", and "Z" and Table 4 gives the correspondence between the eight fine-grained labels and these three new labels. Whether capping is activated and whether branching is limited are the main contributors to the network architecture. Growth conditions that do not include capping are labeled "X" or "Z, while growth conditions that do include capping are all labeled "Y". As only one growth condition without capping (the condition which has limited Arp2/3 complexes and unlimited actin monomers) is labeled by "Z", we can conclude that while capping is the biggest determinant in labeling the data, limited branching is also a feature in distinguishing network architecture.

3.6. Actin networks with defined microscale dynamics can be grouped into three classes with high accuracy even in the presence of extreme capping rates

A CNN classifier for coarse-grained classification is constructed, which labels actin networks according to their dominant mechanisms rather than their original growth conditions. It still takes images of actin

New label	Old label	Capping	Branching	Monomers
A ▲	0	×	Unlimited	Unlimited
A ▲	1	✓	Unlimited	Unlimited
B ▲	2	×	Limited	Unlimited
B ▲	4	✓	Limited	Unlimited
C ▲	3	×	Unlimited	Limited
C ▲	5	✓	Unlimited	Limited
C ▲	6	×	Limited	Limited
C ▲	7	✓	Limited	Limited

Table 3: Three coarse-grained groups for actin networks with low and intermediate capping probability

New label	Old label	Capping	Branching	Monomers
X ★	0	×	Unlimited	Unlimited
X ★	3	×	Unlimited	Limited
X ★	6	×	Limited	Limited
Y ★	1	✓	Unlimited	Unlimited
Y ★	4	✓	Limited	Unlimited
Y ★	5	✓	Unlimited	Limited
Y ★	7	✓	Limited	Limited
Z ★	2	×	Limited	Unlimited

Table 4: Three coarse-grained groups for actin networks with high capping probability

density and produces predicted labels; the difference is that the labels now correspond to the clusters found by k-means for $k = 3$. The same training and testing data sets are used as previously. In Table 2, the accuracy of the new TI-informed CNN classifier on the test set is reported for the six capping probabilities. We note that the dominant mechanisms can vary with the capping probability, as shown in Section 3.5. For every capping probability considered, the TI-informed CNN classifier is more accurate at the coarse-grained classification than the original CNN classifier is at the fine-grained classification. Its accuracy ranges between 87% and 99%, and interestingly, it performs exceedingly well for extreme capping probabilities, a regime where the original CNN classifier has lower accuracy for fine-grained classification. Lastly, in Figure 7, for intermediate capping probability of 0.001, we re-plot the 12 actin networks from the training set and 16 actin networks from the test set shown previously in Figure 4 but now with their coarse-grained labels. The labels predicted by the TI-informed CNN classifier are also included if they are different from the correct coarse-grained labels.

4. Discussion

The actin cytoskeleton, where actin filaments are organized into a complex interconnected network, is a critical structural element in cells. The organization of the actin filaments in this network can have profound consequences for cell functions, and network disorganization can be a hallmark of disease processes such as cancer (Dugina et al., 2019). Despite the importance of actin network organization, a precise method for classifying and identifying actin network organizations according to their underlying dynamics has not been developed. Towards this end, we develop a two-pronged approach: the first part is a stochastic model that allows us to "grow" actin networks in a wide variety of settings, and the second part is classification using machine learning techniques to identify the dominant mechanisms leading to a given actin network. The model is able to effectively and efficiently produce a large number of actin networks with known, diverse growth conditions, which are necessary to train our classifiers. The machine learning algorithms in turn identify a small number of meaningful categories hidden in the vast amount of training data and allow us to make educated guesses about the microscale dynamics responsible for a given actin network whose growth conditions are unknown. We use relatively standard algorithms for classification such as CNN, and do not rely on state-of-the-art algorithms for image detection (Pham et al., 2021, He et al., 2016). While it is conceivable that the performance of our methodology could be improved with more sophisticated algorithms, we achieve high accuracy results with CNN.

We consider a variety of machine learning algorithms based on network density and/or network symmetry and find that the most successful approach is the following: first use k-means clustering (an unsupervised machine learning algorithm) based on symmetry data to identify the dominant mechanisms and then use CNN (a supervised machine learning algorithm) based on density data to build a classifier. CNN specializes in processing data on a grid, for example, images. Since the actin density is sampled on a two-dimensional grid, CNN is the natural method to use. Indeed, we observe that CNN is more accurate at network classification than other non-CNN methods considered including the SVMs. This implies that the geometry of actin networks, not just the density, is important in identifying their growth conditions. Since multiple growth conditions that we initially considered can lead to similar actin networks, the accuracy of classifiers built based on density alone and symmetry alone both leave room for improvement. The importance of network geometry and the need to identify more prominent, well-defined classes motivate our approach of coupling k-means based on symmetry data and CNN based on density data.

Our work showcases how machine learning can help "demystify" the formation of intricate biological networks, which are ubiquitous. We see many potential applications of the classification tools that we have developed. An experimentalist imaging *in vitro* actin network formation may observe very different network architectures, and may wish to identify the biological growth dynamics that led to the given architecture. Using the machine learning tools described here, specifically k-means coupled with CNN, we can predict a label for the given network. That label can provide insight into the growth condition of the network, for

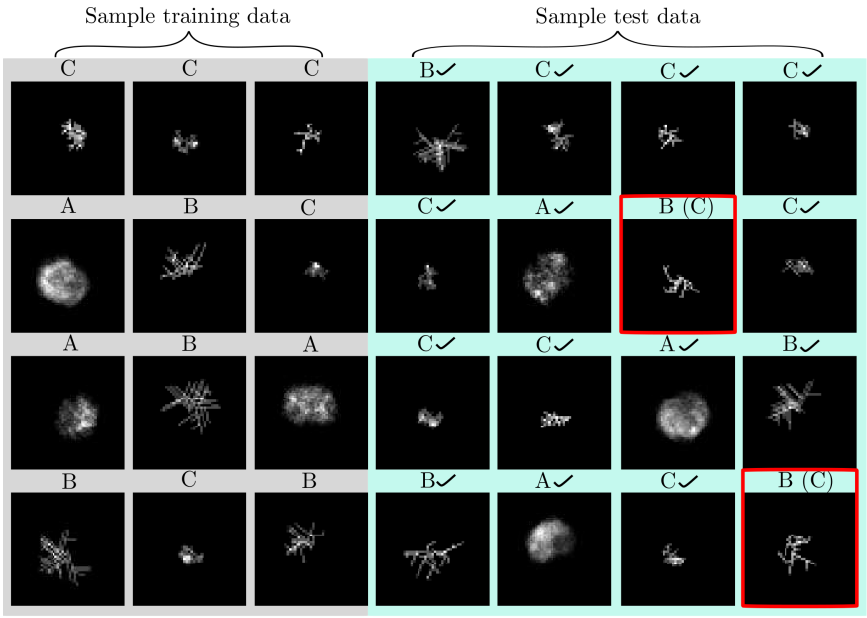

Figure 7: The same 12 randomly selected actin density plots from the training set (three leftmost columns) and the same 16 randomly selected actin density plots from the test set (four rightmost columns) as in Figure 4, with their corresponding coarse-grained labels. The images from the test set misclassified by the TI-informed CNN classifier are outlined in red. Their labels predicted by the CNN are listed in parentheses. A check mark indicates that the CNN-predicted label matches the true label. The plots shown are for an intermediate capping probability, $p_{\text{cap}} = 0.001$.

example, whether or not there are limited Arp2/3 branching complexes. We illustrate the workflow of such an analysis in Figure 8. This prediction can be used in conjunction with other sources of data to better understand the dynamics of the system. In some disease states like cancer (Dugina et al., 2019) actin network growth is disrupted, and the machine learning method described here would be a way to use images of the network architecture to diagnose possible causes of the disrupted growth. More broadly, the framework of analysis developed in this work can be applied to understand analogous network-forming biological systems, including extensions of fungal hyphae, blood vessels arising from endothelial cell self-assembly, and more.

Acknowledgments

M.W.R. was supported, in part, by the National Science Foundation (NSF) under Grant No. DMS-1818833 and Grant No. DMS-2146191 (CAREER). K.G. was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant No. RGPIN-2022-03924. R.L.P. acknowledges support from the Schmidt Science Fellows program, in partnership with the Rhodes Trust. C.C. was supported in part by the NSF under Grant No. DMS-2209494. A.T.D. was supported by the NSF under Grant No. DMS-1554896 (CAREER) and the National Institutes of Health (NIH) under Grant No. R01GM132651.

We would also like to thank the Banff International Research Station (BIRS) for hosting us in May 2022.

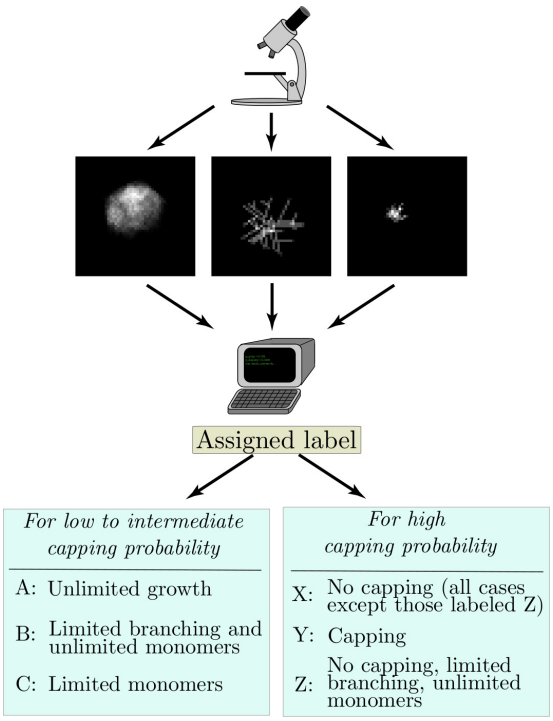

Figure 8: *In vitro* experiments can generate images of actin networks, which can be computationally analyzed using the methods described in this paper, and then classified with respect to the underlying dynamics. Tables 3 and 4 list the biological growth conditions for the different labels.

This BIRS Focused Research Group (FRG) meeting was crucial to the development of this work.

Author Credits

M.W.R.: Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Resources, Data Curation, Writing - Original Draft, Writing - Review and Editing, Visualization, Funding acquisition; B.B: Conceptualization, Methodology, Software, Visualization, Writing - Original Draft; K.G.: Conceptualization, Visualization, Writing - Review and Editing; R.P.L.: Conceptualization, Methodology, Investigation, Writing - Review and Editing; C.C.: Conceptualization, Methodology, Software, Formal analysis, Writing - Original Draft, Project administration; A.T.D.: Conceptualization, Methodology, Software, Formal analysis, Writing - Original Draft, Project administration.

References

O. Akin and R. D. Mullins. Capping protein increases the rate of actin-based motility by promoting filament nucleation by the arp2/3 complex. *Cell*, 133(5):841–851, 2008. ISSN 0092-8674. doi: 10.1016/j.cell.2008.04.011. URL https://www.sciencedirect.com/science/article/pii/S0092867408004984.

N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992. doi: 10.1080/00031305.1992.10475879. URL https://www.jstor.org/stable/2685209.

K. Amann and T. Pollard. Direct real-time observation of actin filament branching mediated by Arp2/3 complex using total internal reflection fluorescence microscopy. *Proceedings of the National Academy of Sciences*, 98:15009–15013, 2001. doi: 10.1073/pnas.211556398. URL https://www.jstor.org/stable/3057401.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. doi: 10.1007/BF00058655. URL https://link.springer.com/content/pdf/10.1007/BF00058655.pdf.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324. URL https://link.springer.com/content/pdf/10.1023/A:1010933404324.pdf.

D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, 2012. doi: 10.1109/CVPR.2012.6248110.

E. Clarkson, C. Costa, and L. Machesky. Congenital myopathies: diseases of the actin cytoskeleton. *J. Pathol.*, 204:407–417, 2004. doi: doi.org/10.1002/path.1648.

C. Copos, B. Bannish, K. Gasior, R. Pinals, M. Rostami, and A. Dawes. Connecting actin polymer dynamics across multiple scales. In R. Segal, B. Shtylla, and S. Sindi, editors, *Using Mathematics to Understand Biological Complexity*, volume 22, pages 7–33. Springer, Cham, 2021. doi: 10.1007/978-3-030-57129-0_2. URL https://link.springer.com/content/pdf/10.1007/978-3-030-57129-0.pdf.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000. doi: doi.org/10.1017/CBO9780511801389.

V. B. Dugina, G. S. Shagieva, and P. B. Kopnin. Biological role of actin isoforms in mammalian cells. *Biochemistry (Moscow)*, 84:583–592, 6 2019. ISSN 16083040. doi: 10.1134/S0006297919060014. URL https://link.springer.com/article/10.1134/S0006297919060014.

M. Edwards, A. Zwolak, D. Schafer, D. Sept, R. Domniguez, and J. Cooper. Capping protein regulators fine-tune actin assembly dynamics. *Nat. Rev. Mol. Cell Biol.*, 15:677–689, 2014. doi: 10.1038/nrm3869.

P. Gandhi, M.-V. Ciocanel, K. Niklas, and A. T. Dawes. Identification of approximate symmetries in biological development. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2213), dec 2021. ISSN 1364-503X. doi: 10.1098/RSTA.2020.0273. URL `https://royalsocietypublishing.org/doi/epdf/10.1098/rsta.2020.0273`.

E. Goley and M. Welch. The Arp2/3 complex: an actin nucleator comes of age. *Nat. Rev. Mol. Cell Biol.*, 7:713–726, 2006. doi: 10.1038/nrm2026.

T. Hastie, J. Friedman, and R. Tibshirani. *The elements of statistical learning.* Springer, 2004. doi: 10.1007/978-0-387-21606-5.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998. doi: 10.1109/34.709601. URL `https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=709601`.

N. Ierushalmi, M. Malik-Garbi, A. Manhart, E. Abu Shah, B. Goode, A. Mogilner, and K. Keren. Centering and symmetry breaking in confined contracting actomyosin networks. *eLife*, 9:e55368, 2020. doi: 10.7554/eLife.55368.

M. Jensen, E. Morris, R. Huang, G. Rebowski, R. Dominguez, D. Weitz, J. Moore, and C.-L. Wang. The conformational state of actin filaments regulates branching by actin-related protein 2/3 (Arp2/3) complex. *Journal of Biological Chemistry*, 287:31447–31453, 2012. doi: 10.1074/jbc.M112.350421. URL `https://www.jbc.org/article/S0021-9258(20)63089-7/fulltext`.

A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. *Comm. ACM*, 60:84–90, 2017. doi: doi:10.1145/3065386.

V. Laurent, S. Kasas, Y. A., T. Schäffer, S. Catsicas, G. Dietler, A. Verkhovsky, and J.-J. Meister. Gradient of rigidity in the lamellipodia of migrating cells revealed by atomic force microscopy. *Biophys. J.*, 89: 667–675, 2005. doi: 10.1529/biophysj.104.052316.

Y. LeCun. Generalization and network design strategies. Technical Report CRG-TR-89-4, University of Toronto, 1989.

M. Lekka, P. Laidler, D. Gil, J. Lekki, Z. Stachura, and A. Hrynkiewicz. Elasticity of normal and cancerous human bladder cells studied by scanning force microscopy. *Eur. Biophys. J.*, 28:312–316, 1999. doi: doi.org/10.1007/s002490050213.

A. Liu, D. Richmond, L. Maibaum, S. Pronk, P. Geissler, and D. Fletcher. Membrane-induced bundling of actin filaments. *Nat. Phys.*, 4:789–793, 2008. doi: 10.1038/nphys1071.

H. Lodish, A. Berk, S. Zipursky, et al. *Molecular Cell Biology, 4th ed.* W. H. Freeman, New York, USA, 2000.

J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297, Statistical Laboratory of the University of California, Berkeley, 1967.

W. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics.*, 5(4):115–133, 1943. doi: 10.1007/BF02478259. URL `https://link.springer.com/content/pdf/10.1007/BF02478259.pdf`.

A. Michelot and D. Drubin. Building distinct actin filament networks in a common cytoplasm. *Curr. Biol.*, 21:R560–R569, 2011. doi: 10.1016/j.cub.2011.06.019.

E. Moeendarbary, L. Valon, M. Fritzsche, A. Harris, D. Moulding, A. Thrasher, E. Stride, L. Mahadevan, and G. Charras. The cytoplasm of living cells behaves as a poroelastic material. *Nat. Mater.*, 12:253–261, 2013. doi: 10.1038/nmat3517.

A. Mogilner. Mathematics of cell motility: have we got its number? *Journal of Mathematical Biology*, 58: 105–134, 2009. doi: 10.1007/s00285-008-0182-2. URL `https://link.springer.com/content/pdf/10.1007/s00285-008-0182-2.pdf`.

D. Moulding, J. Record, D. Malinova, and A. Thrasher. Actin cytoskeletal defects in immunodeficiency. *Immunol. Rev.*, 256:282–299, 2013. doi: 10.1111/imr.12114.

H. Pham, Z. Dai, Q. Xie, and Q. Le. Meta pseudo labels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11557–11568, 2021. doi: 10.1109/CVPR46437.2021.01139.

T. Pollard and G. Borisy. Cellular motility driven by assembly and disassembly of actin filaments. *Cell*, 112: 453–465, 2003. doi: 10.1016/s0092-8674(03)00120-x. URL `https://www.sciencedirect.com/science/article/pii/S009286740300120X`.

A. Reymann, R. Boujemaa-Paterski, J. Martiel, C. Guérin, W. Cao, H. Chin, E. De La Cruz, M. Théry, and L. Blanchoin. Actin network architecture can determine myosin motor activity. *Science*, 336:1310–1314, 2012. doi: 10.1126/science.1221708.

U. Schwarz and M. Gardel. United we stand – integrating the actin cytoskeleton and cell–matrix adhesions in cellular mechanotransduction. *J. Cell Sci.*, 125:3051–3060, 2012. doi: 10.1242/jcs.093716.

C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano. RUSBoost: Improving classification performance when training data is skewed. In *2008 19th International Conference on Pattern Recognition*,

pages 1–4, 2008. doi: 10.1109/ICPR.2008.4761297. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4761297.

B. A. Smith, K. Daugherty-Clarke, B. L. Goode, and J. Gelles. Pathway of actin filament branch formation by arp2/3 complex revealed by single-molecule imaging. *Proceedings of the National Academy of Sciences of the United States of America*, 110:1285–1290, 1 2013. doi: 10.1073/pnas.1211164110. URL https://www.pnas.org/doi/epdf/10.1073/pnas.1211164110.

T. Svitkina. The actin cytoskeleton and actin-based motility. *Cold Spring Harb. Perspect. Biol.*, 10:a018267, 2018. doi: 10.1101/cshperspect.a018267.

T. Svitkina and G. Borisy. Arp2/3 complex and actin depolymerizing factor/cofilin in dendritic organization and treadmilling of actin filament array in lamellipodia. *Journal of Cell Biology*, 145:1009–1026, 1999. doi: 10.1083/jcb.145.5.1009. URL https://rupress.org/jcb/article-pdf/145/5/1009/1285979/9903075.pdf.

J. Theriot and T. Mitchison. Actin microfilament dynamics in locomoting cells. *Nat.*, 352:126–131, 1991. doi: 10.1038/352126a0.

E. Urban, S. Jacob, M. Nemethova, G. Resch, and J. Small. Electron tomography reveals unbranched networks of actin filaments in lamellipodia. *Nat. Cell Biol.*, 12:429–435, 2010. doi: 10.1038/ncb2044.

M. Vinzenz, M. Nemethova, F. Schur, J. Mueller, A. Narita, E. Urban, C. Winkler, C. Schmeiser, S. A. Koestler, K. Rottner, G. P. Resch, Y. Maeda, and J. V. Small. Actin branching in the initiation and maintenance of lamellipodia. *Journal of Cell Science*, 125:2775–2785, 6 2012. doi: 10.1242/jcs.107623. URL https://journals.biologists.com/jcs/article/125/11/2775/32327/Actin-branching-in-the-initiation-and-maintenance.

G. Vstovsky. Transform information: a symmetry breaking measure. *Foundations of Physics*, 27:1413–1444, 1997. doi: 10.1007/BF02551520. URL https://link.springer.com/content/pdf/10.1007/BF02551520.pdf.

M. A. Wear, A. Yamashita, K. Kim, Y. Maéda, and J. A. Cooper. How capping protein binds the barbed end of the actin filament. *Current Biology*, 13(17):1531–1537, 2003. ISSN 0960-9822. doi: 10.1016/S0960-9822(03)00559-1. URL https://www.sciencedirect.com/science/article/pii/S0960982203005591.

M. Young, J. Cooper, and P. Bridgman. Yeast actin patches are networks of branched actin filaments. *J. Cell Biol.*, 166:629–635, 2004. doi: doi.org/10.1083/jcb.200404159.

Appendix

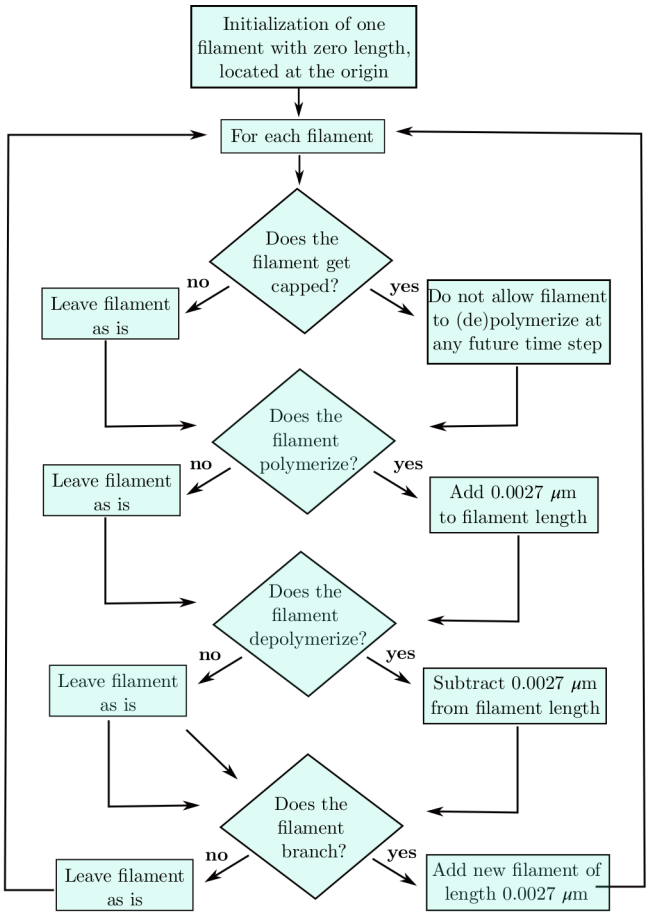

Figure A1: Microscale model flow diagram.

A.1. Detailed description of stochastic simulation of actin networks

In our previous work, we built a tractable agent-based stochastic model to capture the local microstructure of branching actin networks under various intracellular conditions Copos et al. (2021). The model includes the actin dynamics of polymerization, depolymerization, and branching of filaments initiated from a single nucleation site. The actin filaments are represented as rigid rods with a spatially fixed base and a barbed tip capable of growing or shrinking. Changes in both actin filament length and overall network structure are due to the addition or removal of actin monomers and branching via the Arp2/3 complex. In the initial model, we assume that there is an unlimited pool of free monomers and Arp2/3 complexes available to enable filament growth and branching, respectively. For simplicity, other dynamics of actin networks such as capping, sliding, bundling, etc. are not considered in the original model.

Briefly, each simulation starts with an actin filament of length zero located at the "nucleation site" placed at the origin. The filament is assigned an angle of growth which is drawn from a static uniform distribution;

the angle of growth prescribes the direction of growth for that particular filament. At each time step, there are four possibilities for filament dynamics: (1) the filament grows with probability p^0_{poly}; (2) the filament shrinks with probability p^0_{depoly}, provided that the filament has a nonzero length at the start of the time step; (3) the filament remains the same length; or (4) the filament branches with probability p^0_{branch} to create a new filament, provided that the original filament is at least some critical length L_{branch} (measured from the closest branch point). Filament growth and shrinkage occur in discrete increments corresponding to the length of an actin monomer, $\Delta x = 0.0027\ \mu$m. Based on the known biological interaction of the Arp2/3 complex with actin, the newly branched filament is assigned to grow in a direction that is $\pm 70°$ from the preexisting filament tip.

To determine which of the four outcomes happens at a time step, two independent uniformly distributed random numbers are generated for each filament in the simulation. If the first random number is less than p^0_{poly}, then polymerization occurs, and if it is greater than $1 - p^0_{\text{depoly}}$, then depolymerization occurs. If the first random number simultaneously satisfies both inequalities, then the filament length is unchanged since both polymerization and depolymerization occur in the same time step. Likewise, filament length is unchanged if neither inequality is satisfied, since this means neither polymerization nor depolymerization occur in the given time step. If the second random number is less than p^0_{branch}, and the filament is at least length L_{branch}, then the filament branches to form a new filament, capable of autonomous growth and branching. This step-wise process is repeated until the final simulation time is reached. Ultimately, this stochastic model captured molecular-level effects within the network and sensitivity analysis revealed that the biological parameters responsible for filament growth and branching kinetics impacted resulting network dynamics and morphology in a complementary manner. Simulations of branching actin networks were implemented in a custom MATLAB code, with additional implementation details available in Copos et al. (2021).

In the current study, to capture more biologically relevant branching actin networks, we include capping proteins, limited actin monomers, and limited Arp2/3 branching complexes as follows (schematic in Figure 2A; the full algorithm flow chart is shown in Appendix Figure A1). The capping protein regulates actin polymerization by binding to the barbed end of an actin filament, which blocks the addition and loss of monomers from that filament. In our simulation, capping dynamics are modeled as follows. At each time step, a third uniform random number is generated for each uncapped filament. If the capping probability (p_{cap} = rate of capping × time step) is greater than the random number, then the filament can no longer polymerize or depolymerize. Capping is irreversible; once a filament is capped, it is capped for the remainder of the simulation. In simulations that include limited resources, we start with a fixed number of actin monomers ($M_0 = 10,000$) and/or a fixed number of Arp2/3 complexes ($A_0 = 24$). A polymerization (or branching) event can only occur if there are available actin monomers (Arp2/3 complexes), and the probability of polymerization (branching) depends on how many actin monomers (Arp2/3 complexes) remain. In our initial model with unlimited resources, the polymerization rate was fixed: $p^0_{\text{poly}} = 0.324\,\Delta t/\Delta x$, where Δt

is the constant time step, Δx is the constant space step. Now, with limited resources, we assume that the polymerization rate depends on the number of actin monomers: $p_{\text{poly}} = \frac{M}{M_0}\, e^{(M-M_0)/M_0}\, p^0_{\text{poly}}$, where M is the number of available actin monomers at the given time step. Similarly, the branching probability is modified by the number of available Arp2/3 complexes: $p_{\text{branch}} = \frac{A}{A_0}\, p^0_{\text{branch}}$, where p^0_{branch} is the branching probability from the unlimited resources case (chosen from a cumulative distribution function of the standard normal distribution with mean 2 and standard deviation 1), and A is the number of available Arp2/3 complexes at the given time step. Values for parameters are listed in Appendix Table A1, and explained in detail in the next section.

A.2. Stochastic simulation parameters

Wherever possible, model parameters are based on experimental values. The probability of polymerization is calculated from Pollard and Borisy (2003), which states that actin monomers elongate the barbed ends of actin filaments at a velocity of 0.3 μm/s Pollard and Borisy (2003). Using the formula

$$\text{assembly speed} = \text{polymerization probability} \times \text{length added to filament} \times \#\ \text{time steps per second},$$

we calculate that

$$0.3\,\frac{\mu\text{m}}{\text{s}} = p_{\text{poly}} \times 0.0027\,\mu\text{m} \times \frac{1}{0.005\ \text{s}},$$

which implies that $p_{\text{poly}} = 0.56 \approx 0.6$. Similarly, Pollard and Borisy (2003) give a range of dissociation rates of actin monomers from $1.4 - 8\ \text{s}^{-1}$ (Pollard and Borisy, 2003). We choose an intermediate value of $5.0\ \text{s}^{-1}$ and then calculate

$$\text{disassembly speed} = \text{depolymerization probability} \times \text{length removed from filament} \times \#\text{time steps per second},$$

where disassembly speed is obtained via

$$5.0\frac{\text{subunit}}{\text{s}} \times 0.0027\frac{\mu\text{m}}{\text{subunit}} = 0.0135\,\frac{\mu\text{m}}{\text{s}}.$$

Therefore,

$$0.0135\,\frac{\mu\text{m}}{\text{s}} = p_{\text{depoly}} \times 0.0027\,\mu\text{m} \times \frac{1}{0.005\ \text{s}},$$

which yields $p_{\text{depoly}} = 0.025$. Literature measurements of actin filament length per branch vary from 0.02 to 5 μm (Mogilner, 2009, Svitkina and Borisy, 1999, Amann and Pollard, 2001, Jensen et al., 2012, Vinzenz et al., 2012, Smith et al., 2013), so we choose $L_{\text{branch}} = 0.2\ \mu$m, which is similar to the values from Vinzenz et al. (2012), Smith et al. (2013) and is an intermediate value between the orders-of-magnitude-different literature values. The probability of capping is calculated using the formula

$$\text{capping probability} = \text{capping rate} \times \text{time step} \times \text{concentration of capping protein},$$

which gives

$$p_{\text{cap}} = 6.3\,\frac{1}{\mu\text{M}\cdot\text{s}} \times 0.005\,\text{s} \times X\ \mu\text{M}.$$

The capping rate of 6.3 $\mu M^{-1}s^{-1}$ is from Wear et al. (2003), and X, the concentration of capping protein, can take on values up to 0.168 μM Akin and Mullins (2008), which corresponds to p_{cap} = 0.0053. Hence, we look at a range of capping probabilities from 0 to 0.00375; capping probabilities greater than this do not result in appreciable actin network growth.

Parameter	Meaning	Value
p^0_{poly}	Probability of polymerizing	0.6*
p^0_{depoly}	Probability of depolymerizing	0.025*
p_{cap}	Probability of capping	0 – 0.00375
L_{branch}	Critical length before branching can occur	0.2 μm†
p^0_{branch}	Probability of branching	normal CDF (μ = 2, σ = 1)
T_{end}	Computational run time	10 s
Δt	Computational time step	0.005 s
A_0	Initial number of branching complexes	24
M_0	Initial number of actin monomers	10,000

Table A1: Microscale model parameter values. Values flagged with one star (*) were calculated from Pollard and Borisy (2003). For the value flagged with a dagger (†): literature measurements of actin filament length per branch vary from 0.02–5 μm Mogilner (2009), Svitkina and Borisy (1999), Amann and Pollard (2001), Jensen et al. (2012), Vinzenz et al. (2012), Smith et al. (2013). We choose L_{branch} = 0.2 μm as an intermediate estimate between these orders-of-magnitude-different values from the literature, similar to the values from Vinzenz et al. (2012), Smith et al. (2013). p^0_{branch} is chosen from a Gaussian distribution with mean μ = 2 and standard deviation σ = 1. A range of p_{cap} values from 0 to 0.00375 are investigated.