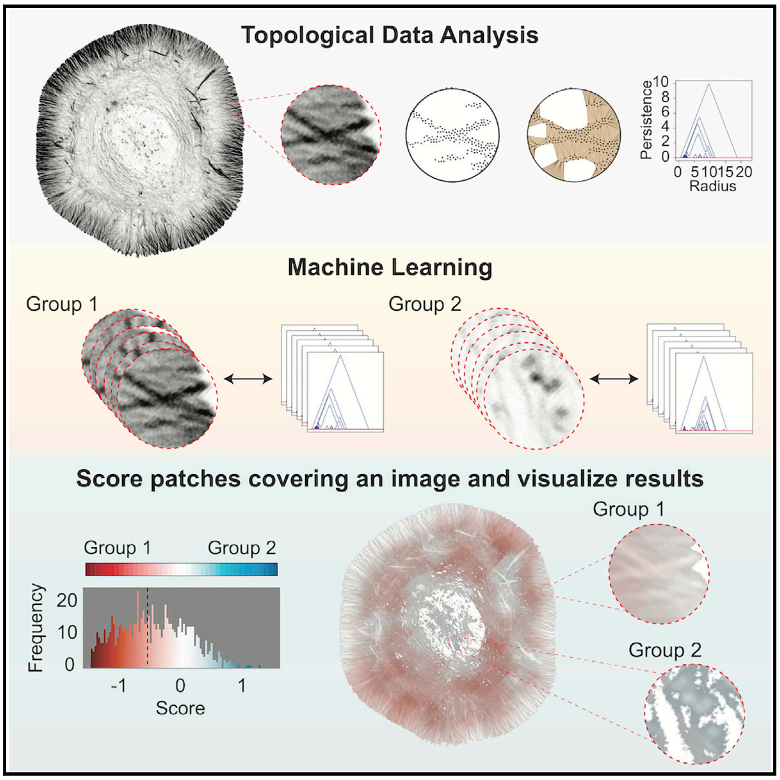
Patterns

TDAExplore: Quantitative analysis of fluorescence microscopy images through topology-based machine learning

Graphical abstract



Highlights

- TDAExplore combines topological data analysis with machine learning classification
- As few as 20–30 high-resolution images can be used to train TDAExplore models
- TDAExplore is robust to different microscopy modes, dataset size, image features
- TDAExplore quantifies where and how much each image resembles the training data

Authors

Parker Edwards, Kristen Skruber, Nikola Milićević, James B. Heidings, Tracy-Ann Read, Peter Bubenik, Eric A. Vitriol

Correspondence

peter.bubenik@ufl.edu (P.B.), evitriol@augusta.edu (E.A.V.)

In brief

Machine learning is exceptionally powerful at categorizing imaging data into different classes, yet most methods do not provide insight into how classification tasks were made. Here, Edwards et al. address this limitation with TDAExplore, an image analysis pipeline combining machine learning classification with topological data analysis. In addition to being able to accurately classify fluorescent microscopy data over a broad range of examples, TDAExplore quantifies where and how much images resemble the data that were used to train it.

Edwards et al., 2021, Patterns 2, 100367
November 12, 2021 © 2021 The Author(s).
https://doi.org/10.1016/j.patter.2021.100367

CellPress

Article

TDAExplore: Quantitative analysis of fluorescence microscopy images through topology-based machine learning

Parker Edwards,[1,7] Kristen Skruber,[2,7] Nikola Milićević,[3] James B. Heidings,[4] Tracy-Ann Read,[5] Peter Bubenik,[6,*] and Eric A. Vitriol[5,8,*]

[1]Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556, USA
[2]Department of Cellular and Molecular Pharmacology and Howard Hughes Medical Institute, University of California, San Francisco, CA 94143, USA
[3]Department of Mathematics, Pennsylvania State University, University Park, PA 16802, USA
[4]Department of Pharmacology and Therapeutics, University of Florida College of Medicine, Gainesville, FL 32601, USA
[5]Department of Neuroscience and Regenerative Medicine, Medical College of Georgia, Augusta University, Augusta, GA 30912, USA
[6]Department of Mathematics, University of Florida, Gainesville, FL 32611, USA
[7]These authors contributed equally
[8]Lead contact
*Correspondence: peter.bubenik@ufl.edu (P.B.), evitriol@augusta.edu (E.A.V.)

https://doi.org/10.1016/j.patter.2021.100367

THE BIGGER PICTURE Traditional intensity-based measurements of fluorescent microscopy data limit its potential to reveal new information about its sample. Here, we present an image analysis pipeline called TDAExplore, which is based on topological data analysis and machine learning classification. In addition to being highly accurate in assigning images to their correct group, TDAExplore quantifies how much images resemble the training data and identifies which parts are different, an improvement over other machine learning models that do not permit insight into how classification tasks were made. The next steps for TDAExplore will be to expand its capabilities into three-dimensional, multivariate, and time series datasets. This work represents progress into a future where machine learning identifies and describes nuanced image features in ways that allow researchers to answer important biological questions and generate new hypotheses for future studies.

1 2 **3** 4 5 **Development/Pre-production:** Data science output has been rolled out/validated across multiple domains/problems

SUMMARY

Recent advances in machine learning have greatly enhanced automatic methods to extract information from fluorescence microscopy data. However, current machine-learning-based models can require hundreds to thousands of images to train, and the most readily accessible models classify images without describing which parts of an image contributed to classification. Here, we introduce TDAExplore, a machine learning image analysis pipeline based on topological data analysis. It can classify different types of cellular perturbations after training with only 20–30 high-resolution images and performs robustly on images from multiple subjects and microscopy modes. Using only images and whole-image labels for training, TDAExplore provides quantitative, spatial information, characterizing which image regions contribute to classification. Computational requirements to train TDAExplore models are modest and a standard PC can perform training with minimal user input. TDAExplore is therefore an accessible, powerful option for obtaining quantitative information about imaging data in a wide variety of applications.

INTRODUCTION

Microscopy images contain an incredible amount of complex information. The development of machine learning has substantially accelerated the ability to identify and extract relevant features,[1] as well as to classify images into groups.[2,3] These methods, such as convolutional neural networks (CNNs), perform extraordinarily well in segmentation and classification.[4,5] However, identifying how machine learning pipelines interpret training data to determine which features to extract is challenging.

Here, we present a new set of features for image analysis based on two topological data analysis (TDA) methods: persistent homology[6] and persistence landscapes.[7] TDA is a mathematical method that uses algebraic topology to quantify the shape of data[8] and has been used for other machine-learning-based methods.[9–11] We then exhibit an image analysis pipeline, TDAExplore, that uses these features for image classification and image segmentation. The segmentation model is "weakly supervised" and learns to identify regions whose shape features most strongly characterize each class of image using only image labels. Using multiple datasets, we demonstrate that this method achieves high per image classification accuracies with minimal training, is robust to hyperparameter changes, and requires only modest computational resources. Subsequently, we show that this approach can recognize a wide variety of subcellular structures and extract biologically meaningful data.

Related work

Image analysis pipelines typically proceed either by first extracting pre-defined descriptors from an image's raw pixel values or by learning features using training data. The TDA image features we introduce are in the former category. While many different descriptors have been defined in the past, including those that capture some topological information,[1,11,12] they do not use persistent homology. Our features are based on persistent local homology, which may be used to learn stratified structures,[13,14] such as the intersections of actin filaments and for supervised image segmentation.[15]

Image segmentation models produce pixel masks for images that split images into regions with similar behavior. Models where the training data consist of images paired with "ground truth" masks have been successful for many applications, including medical image segmentation.[16] These approaches assume that the ground truth is known by experts, however, who must draw training masks by hand. Weakly supervised segmentation models instead use training data where images and labels are provided, but not masks. CNN-based models exist but require substantial datasets to train or fine-tune.[17]

Many machine learning approaches using both pre-defined[18–20] and supervised[2,21] features have been previously proposed and implemented for analysis of microscopy images. Largely speaking, these methods focus on whole-image classification or supervised segmentation. A major application, for instance, is to train a supervised model to segment pixels containing cells from the background of an image. Our method uses a weakly supervised approach to explore microscopy image datasets where ground truth segmentations are not already known and therefore cannot be specified by experts.

Here, we consider several microscopy image datasets depicting the actin cytoskeleton where we desire subcellular level information about morphology changes. Our previous publications[22–25] detail numerous ways to extract information from microscopy images of the actin cytoskeleton using fluorescence intensity values of localized actin probes. These include measuring whole-cell or local fluorescence intensity, performing line scan analysis, and combining intensity-based thresholding with morphology filters to segment and quantify specific types of actin structures. All these methods require one to choose a particular actin structure for analysis, and thus introduce a potential source of bias. For example, one might decide that an actin perturbation most strongly affects the lamellipodia and focus efforts there based on previous work rather than on objective criteria. In addition, traditional analysis methods based on fluorescence intensity cannot quantify spatial properties of actin ultrastructure. Intensity values capture actin levels in particular locations, but do not differentiate between regions with similar actin content organized into distinct geometries.

RESULTS AND DISCUSSION

Topological feature extraction from fluorescence microscopy data

To extract topological information, images were first masked by automatic intensity thresholding to reduce pixel values outside of the cell to zero. Images were then divided into uniform radius patches (Figure 1Ai). A subset of high-intensity pixels was selected per patch together with evenly spaced points along the boundary (Figure 1Aii). Neighboring points were progressively connected if they were within an increasing distance of each other (Figure 1Aiii–vi). The resulting sequence of simplicial complexes was then used to generate a persistence landscape (Figure 1Aviii, B), which encodes sets of birth-death pairs (Figure 1Avii) generated from the appearance and disappearance of specific homology features,[7] including connected components and their holes (Figure 1Aiii–vi).

To interpret the intrinsic characteristics of the resulting high dimensional vectors (Figure 1B), t-distributed stochastic neighbor embedding (t-SNE)[26] was used to reduce image summaries to a single score. The scores were subsequently scaled to range from -1 to 1 and a color gradient was mapped to the values. These values were then mapped to the pixels of the corresponding uniform radius patch so that the result can be visualized on the input image as a mask (Figure 1C). These masks show that this method extracts physiologically relevant features of the cytoskeleton by demarcating the large actin superstructure known as the lamellipodia from the cell interior (Figure 1C).

Image classification based on topological features

We then constructed an image classification pipeline to test the discriminatory power of our TDA descriptors. We restricted ourselves to linear support vector regression (SVR) classifiers, rather than more sophisticated methods such as deep neural networks, to try and make it clear that our results depended mainly on our TDA descriptors and not on the machine learning method. The datasets we used are summarized in Table S1. Each dataset contained images in two manually assigned classes. We used these classes as ground truth for training and testing the SVR classifiers. For all datasets, two-class patch classification models were trained using persistence landscapes generated from image patches as input.

In general, SVR classifiers define a hyperplane that optimizes the separation between training data with different ground truth class labels[27] and assign a score to each patch and persistence landscape. Once trained, a classifier predicts patches with score <0 as one class, while those with score >0 are in the other. This

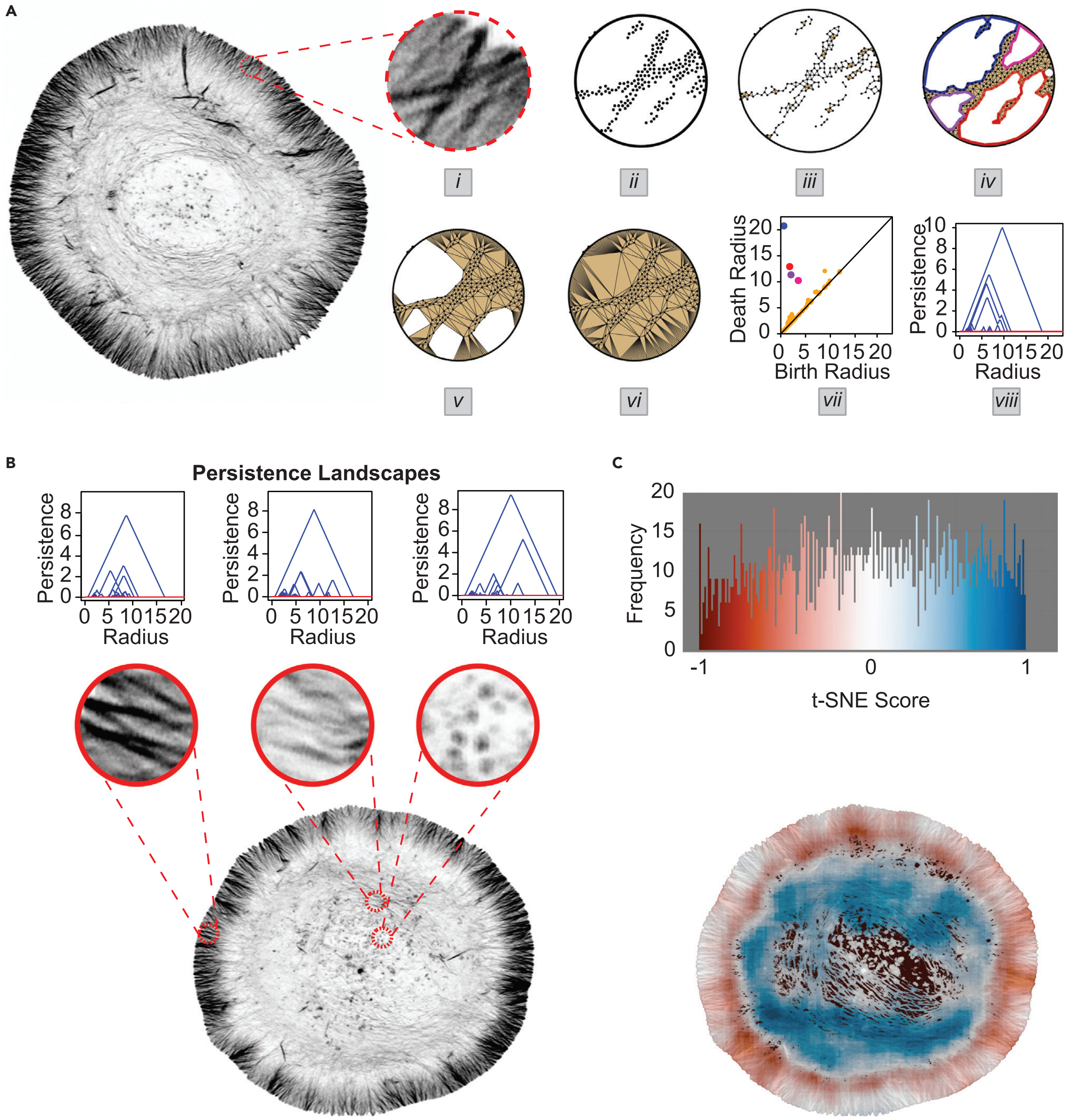
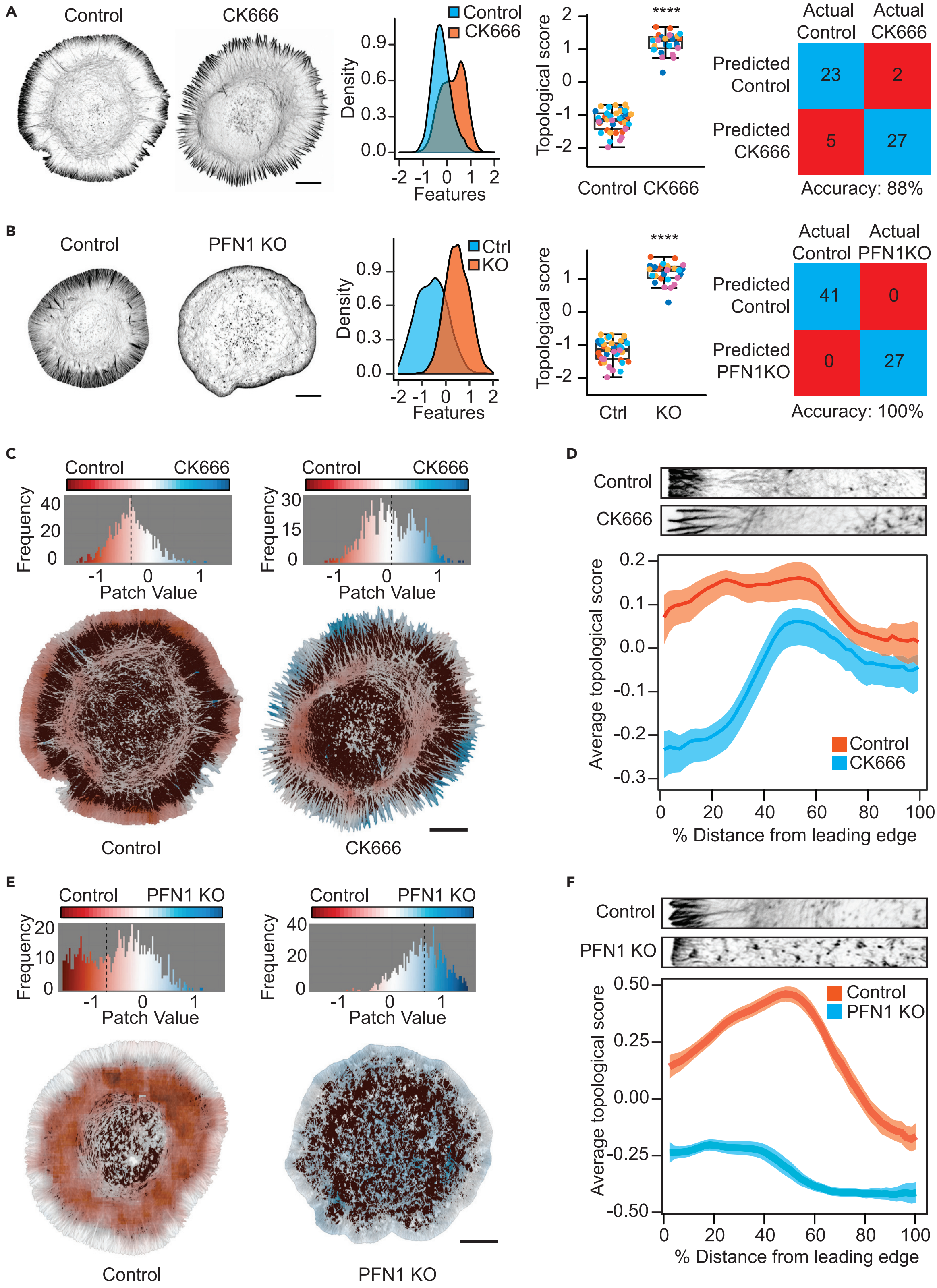


Figure 1. Persistent homology-based image analysis pipeline to extract topological features

(A) Computation of a persistence landscape from a selected image patch. (i) Representative image patch of 75 pixel radius. (ii) Points are sampled from the spatial coordinates of fluorescence signals. (iii–vi) Sequence of simplicial complexes generated by connecting neighboring points within a distance that increases from (iii) to (vi). (iv) Four cycles that represent persistent homology classes in degree one are colored. (v) The pink cycle has been filled in. (vi) All cycles have been filled in. (vii) Persistence diagram plotting the birth radius and death radius of each of the persistent homology classes in degree one. Colored points correspond to the colored cycles in (iv). (vii) Persistence landscape gives a vector encoding of the persistent homology.

(B) Persistence landscapes generated from each selected patch in the input image. Three representative patches and their landscapes are shown.

(C) Histogram of t-SNE scores generated from persistence landscapes. Colorized pixels representing t-SNE scores overlaid on the original image as a mask.



(legend on next page)

prediction may differ from the ground truth class. Since our SVR classifiers were trained using patches, we also used them to define classifiers for whole images. To predict a class for an image, the scores for the image's patches were averaged, and a class was assigned corresponding to whether the average score was <0 or >0.

We used 5-fold cross-validation to assess classifier performance (Figure S1). The images in a dataset were separated into five approximately equal data groupings or "folds" with training and testing at an 80/20 ratio (Figures S1A–S1C). The number of correctly identified images out of the total number was visualized as a confusion matrix (Figure S1D). To assess performance for the entire pipeline, we performed 5-fold cross-validation 50 times for each dataset and recorded both testing accuracy and other common performance metrics (Tables S2–S5).

Two main hyperparameters were chosen to run the pipeline: the radius in pixels of the subsampled patches and the number of patches sampled per image. A uniform selection of pixel radius 75 was made for all datasets based on biological considerations, namely the size in pixels of relevant subcellular structures. The number of patches sampled per image was chosen so that the total number of pixels in all patches was twice the number of pixels in the image.

Using TDAExplore to detect cellular features

This combination of TDA and machine learning (TDAExplore) was then used to detect changes to the architecture of the actin cytoskeleton after induction of both genetic and chemical perturbations to major actin assembly factors (Table S2). We found TDAExplore was able to robustly identify which cells were challenged with perturbations to the actin cytoskeleton (Figures 2 and S2). This included using a small-molecule inhibitor (CK-666) against the actin nucleator Arp2/3 (Figure 2A), CRISPR-Cas9 knockout of the gene for the actin monomer binding protein profilin 1 (PFN1 KO) (Figure 2B), shRNA knockdown of the actin filament severing protein cofilin 1 (cofilin KD) (Figures S2A and S2D) and the monomer binding protein thymosin β-4 (Tβ4 KD) (Figures S2B and S2E), and mislocalization-based inhibition of the Mena/VASP actin polymerases (Figures S2C and S2F).

The 100% image accuracy for the PFN1 KO dataset (Figure 2B) is likely indicative of both the strong effects of the perturbation and its penetrance since the perturbation involved complete disruption of the gene. Likewise, the relatively weaker training performance of the Tβ4 and cofilin 1 knockdown datasets (84% and 91%, respectively) (Figures S2A and S2B) or the Mena/VASP mislocalization data (81%, Figure S2C) could potentially be explained by the cell-to-cell variability of shRNA knockdown or the degree of expression of the FP4-mito DNA construct. Perturbation strength may also play a role since the CK-666 drug treatment, which was expected to have 100% penetrance, did not achieve 100% accuracy (Figure 2A).

We conducted several tests to validate TDAExplore's robustness. To ensure that TDAExplore was not overfitting input data by detecting non-relevant features, we trained an SVR classifier by randomly assigning half the control KO images from our PFN1 KO dataset to a group (group 1) and half to another (group 2). The resulting patch scores were indistinguishable between groups (Figure S3A), indicating that TDAExplore could not detect differences within the same experimental group. Since our datasets contained insufficient images to perform hyperparameter tuning, we instead recorded TDAExplore's performance with various hyperparameter choices and observed only modest performance differences for most of our datasets (Figures S3B and S3C). One advantage of TDAExplore is its high performance using small datasets (on average 60 images per set), made feasible by assessment at the patch level. The possibility that changes to dataset size could alter image accuracy was tested by reducing high-performing datasets by half. When the number of images was reduced from 68 to 34, 100% accuracy for the PFN1 KO classifier was maintained (Figure S3D). Because all actin cytoskeleton perturbation datasets were generated from the same cell line, the control conditions for our PFN1 KO, cofilin KD, Tβ4 KD, and CK-666 datasets should have indistinguishable features. When controls from all experiments were assigned scores by an SVR classifier trained on the PFN1 KO dataset, their patch scores were not statistically different from each other (Figure S3E). This revealed that bona fide changes to actin architecture are required to make distinctions between datasets and that TDAExplore's trained classifiers were robust enough to accurately classify control cell datasets that were imaged months apart from each other.

Figure 2. TDAExplore analysis of the actin cytoskeleton

(A) Performance evaluation for classification of cells treated with the Arp2/3 inhibitor CK-666 or its inactive control CK-689. N = 25 and 32 for CK-689 and CK-666, respectively. From left to right, distribution of patch features, topological score per cell, and confusion matrix displaying classification summaries. Topological scores are generated through five separate rounds of testing; rounds are designated by color.

(B) Performance evaluation for classification of control and PFN1 KO cells, n = 41 and 27, respectively. From left to right, distribution of patch features, topological score per cell, and confusion matrix displaying classification summaries. Topological scores are generated through five separate rounds of testing; rounds are designated by color.

(C) Distribution of patch values after treatment with CK-689 and CK-666, where CK-689 patches are values <0 and colored red, while CK-666 is classified as patch values > 0 and colored blue. White represents intermediate values. Below are computed feature masks of patch values for representative images from CK-689- and CK-666-treated cells.

(D) Average topological score based on distance from the leading edge to the cell center. Transparent bands depict 95% confidence intervals. Representative cell regions from the leading edge to center are shown above for comparison.

(E) Distribution of patch values for control and PFN1 KO cells where control patches are values <0 and colored red, while PFN1 KO is classified as patch values > 0 and colored blue. White represents intermediate values. Below are computed feature masks of patch values for representative images from control and PFN1 KO cells.

(F) Average topological score based on distance from the leading edge to the cell center. Transparent bands depict 95% confidence intervals. Representative cell regions from the leading edge to center are shown above for comparison. Box and whisker plots in (A and B) denote 95th (top whisker), 75th (top edge of box), 25th (bottom edge of box), and 10th (bottom whisker) percentiles and the median (bold line in box). ****p $\leq$ 0.0001. p values were generated by a two-sided permutation test using the mean score. Scale bar represents 10 μm.

To test whether TDAExplore can detect other structures using the same type of microscopy, we altered mitochondrial morphology using the oxidative uncoupler carbonyl cyanide *m*-chlorophenylhydrazone. The classifier still performed well with an 81% accuracy between groups (Figures S4A and S4B). We also tested if TDAExplore was broadly applicable by using it on published datasets (summarized in Table S3). Datasets were selected because they represented different modes of microscopy, size, and features that were imaged, as well as for having been previously used to assess classification performance of feature-based and CNN classifiers. The latter allowed for a direct comparison with TDAExplore. TDAExplore performed comparably well with these methods, including on data from low-resolution or non-fluorescent images (Tables S3 and S5; Figures S5A and S5G).

Comparison of TDAExplore with transfer learning

To further assess how TDAExplore compares with standard models in the field, we used CNNs to assess our imaging data. While CNNs typically require large datasets,[28] transfer learning can be applied to small datasets.[29] Transfer learning is a machine learning method in which a model trained for a certain task is reused as the starting point for a different task. We used the 71-layer-deep Xception CNN architecture for transfer learning. First, we obtained a publicly available pre-trained version of the model that was trained by its maintainers on more than a million images from the ImageNet database.[30] We then updated its training using patch data from our PFN1 KO and CK-666 datasets. Since TDAExplore uses patches for training, patches were generated from the PFN1 KO and CK-666 datasets using the same hyperparameters (patch radius and number of patches per image) as for TDAExplore. These patches were divided into training (75%) and testing (25%) groups and fed to the neural network. The models were then tested and achieved accuracies of 97.10% (PFN1 KO) and 72.36% (CK-666). TDAExplore obtained somewhat lower average accuracies of 80.9% (PFN1 KO) and 69.7% (CK-666).

It is not surprising that the CNN model outperformed TDAExplore for patch classification. The Xception architecture has roughly 20 million trainable parameters, while TDAExplore has roughly 20,000. However, TDAExplore required less training time (Figure S6) and appeared better suited for segmentation tasks on our datasets. To test this, we used the weights assigned by the CNN's transfer learning model to create score masks for the PFN1 KO dataset. TDAExplore correctly identified regions of the control cell containing punctate actin structures that were similar to the knockout cell, while the CNN model did not (Figure S7). Some similarities between the two datasets were expected, since knocking out PFN1 affected most, but not all, cellular actin structures.[25] It was also anticipated that this particular cytoskeletal topology would be associated with the PFN1 KO group, since an abundance of actin puncta in the cell interior was a hallmark of PFN1 KO cells.[25]

Image segmentation with TDAExplore reveals subcellular differences between image groups

Next, we used TDAExplore to extract spatial information at the subcellular level for our CK-666 and PFN1 KO datasets. In a process similar to the previously discussed t-SNE mask generation, patch scores generated from an SVR classifier trained on persistence landscapes were mapped back to their pixels of origin so that topological scores could be visualized on the input image as a mask. This provides a visualization of the spatial effect of each perturbation with red colored pixels predicted as CK-689/control KO and blue pixels predicted as CK-666/PFN1 KO (Figures 2C and 2E). To test whether the process of mask production is sensitive to changes from random sampling, a PFN1 KO classifier was trained five times and masks were generated for each classification run. Masks were consistent across training sessions, indicating that random sampling did not have a significant effect on mask generation (Figure S3F).

To further quantify spatial components of the topological score, the center of each cell was defined, and patch values were assessed based on distance to the center (Figures 2D and 2F). Arp2/3 inactivation with the small molecule inhibitor CK-666 causes a striking rearrangement of actin architecture at the cell boundary without changing total polymerized actin levels.[25,31] Our feature mapping recapitulated these results and showed that the average topological score for CK-666-treated cells deviated most strongly from their controls near the cell boundary (Figures 2C and 2D). PFN1 KO (Figures 2E and 2F) and depletion of cofilin (Figures S8A and S8B) also showed differential effects based on subcellular location. The largest topological score difference between control and PFN1 KO cells was approximately 50% of the distance from the boundary to the cell center, while the largest difference in topological scoring between cofilin KD cells and control was at the cell boundary. The deviation in topological scoring did not map strongly to specific subcellular locations for Tβ4 KD cells, although there was a slightly larger difference at the cell interior than at the edge (Figures S8C and S8D).

TDAExplore computational requirements

To understand how accessible TDAExplore would be to users with different computational resources, we collected usage statistics with our PFN1 KO dataset (Figures S7A and S7B) for multiple hyperparameter selections. Our results indicate that, with 8 processors, 5-fold cross-validation with patch radius 75 pixels and twice as many total pixels in all the patches as in the image would take less than 13 min and 5 GB of memory. The pipeline can utilize additional processors to decrease computation time.

In summary, TDAExplore combines TDA and machine learning for segmentation and feature extraction of high-resolution images. It is unbiased, relying on minimal user input for image exploration, segmentation, and classification. It can differentiate between multiple types of perturbations over a wide range of subjects without overfitting. Importantly, it also visualizes features at the sub-image level. Thus, TDAExplore should be well suited to extract novel quantitative information from imaging data in a wide variety of applications.

EXPERIMENTAL PROCEDURES

Resource availability

Lead contact

Further information and requests for resources and reagents should be directed and will be fulfilled by the lead contact, Eric Vitriol (evitriol@augusta.edu).

Materials availability

No unique reagents were generated in this study.

Data and code availability

Imaging data used to generate this manuscript's figures and tables and scripts for image analysis have been deposited at Zenodo: https://doi.org/10.5281/zenodo.5486202. TDAExplore is available as a command line program: https://doi.org/10.5281/zenodo.5487090 and as an R package: https://doi.org/10.5281/zenodo.5486681 at Zenodo. All data, scripts, and code listed above is publicly available. Additional information is available from the lead contact upon reasonable request.

Cell culture

All cells used in this study were Cath.-a-differentiated (CAD) cells (originally purchased from Sigma-Aldrich). CAD cells were cultured in DMEM/F12 medium (Gibco) supplemented with 8% fetal calf serum, 1% L-glutamine, and 1% penicillin-streptomycin. Prior to imaging, cells were plated on coverslips coated with 10 μg/mL laminin (Sigma). No new cell lines were generated in this study and cell line generation is detailed elsewhere. For the generation of cofilin KD refer to Vitriol et al.[32] For details on the generation of thymosin β-4 KD refer to Lee et al.[22] For PFN1 KO, Arp2/3 inactivation, and Mena/VASP sequestration refer to Skruber et al.[25]

Immunofluorescence

For visualization of actin filaments, cells were fixed with 4% electron microscopy grade paraformaldehyde (Electron Microscopy Sciences) for 10 min at room temperature (RT) and then permeabilized for 3 min with 1% Triton X. Cells were then washed three times with PBS and actin filaments were stained with Alexa Fluor 488 phalloidin or Alexa Fluor 568 phalloidin (diluted 1:100, Life Technologies) for 30 min at RT in immunofluorescence staining buffer. For visualization of mitochondria, cells were fixed with 4% electron microscopy grade paraformaldehyde (Electron Microscopy Sciences) for 10 min at RT and then permeabilized for 5 min with 0.1% Tween 20. Cells were washed three times with PBS and incubated with an antibody to TOM-20 (rabbit, 1:500, CST) at 4°C overnight. Cells were then washed twice in PBS and incubated with (goat anti-rabbit Alexa Fluor 568, 1:1,000) for 2 h and then washed three times in PBS before mounting with Prolong Diamond (Life Technologies).

Microscopy and image processing

Images were acquired with a Nikon A1R+ laser scanning confocal microscope with a GaAsP multi-detector unit using an Apo TIRF 60× 1.49 NA objective. Deconvolution-based super-resolution confocal microscopy 10 was performed using zoom settings higher than the Nyquist criteria, resulting in oversampled pixels (0.03 μm). Confocal z stacks were created and then deconvolved with Nikon Elements software using the Landweber algorithm (15 iterations, with spherical aberration correction) to create images with approximately 150 nm resolution.[33] For input into the analysis pipeline, three-dimensional z stacks were reduced to a single image using maximum intensity projections. For analysis of the actin cytoskeleton, only the first 10 frames from the bottom of the cell were included in the maximum intensity projection so that only the ventral portions of the cells were analyzed. For analysis of mitochondria, the entire z stack was included in the maximum intensity projection. Cell boundaries were then identified by creating a mask based on intensity-based thresholding. All pixels outside of the cell boundary were assigned a value of zero.

Third-party datasets

We used image datasets BBBC 013v1, 014v1, 015v1, and 016v1 from the Broad Bioimage Benchmark Collection[34] and Binucleate and Liver Gender datasets from the Image Informatics and Computational Biology Unit (IICBU) Biological Image Repository[35] to validate our TDA approach. BBBC data are images of human osteosarcoma cells (U2OS) from high-throughput microscopy assays. Only the GFP channel images were analyzed. If the data contained multiple images from the same well, only one image per well was used for the analysis to mitigate batch effects. Subsets of data with only two classes of phenotypes were selected using the procedure described by Uhlmann et al.[20] IICBU data contain images from *Drosophila melanogaster* cells and murine liver tissue. Only the red color channel was analyzed for murine liver tissue images.

Extracting persistence landscapes from images

Gray scale images were first summarized by sampling patches and computing persistence landscapes as described in Figure 1. This method optimizes computational demands while maintaining analysis quality. Patches were selected from each image by repeatedly sampling patch centers from a probability distribution constructed on the image's pixels. The probability distribution was biased toward higher intensity pixels to prioritize patches centered in non-background regions. The selection also integrated a geometric criterion: newly sampled centers were removed if they were within 2 pixels of a previously sampled pixel. This method was an alternative to grid-based approaches used for CNNs, and minimizes patch overlap while focusing on regions of interest. Because it involved a probability distribution, this step was not deterministic.

Each selected patch was converted to a set of two-dimensional points by taking the coordinates of the top 2.5% of pixels by intensity while observing a similar geometric criterion. This resulted in point samples that captured the most intense regions in patches while avoiding over-sampling. Each patch sample was augmented by adding a circle of points along the patch's circular boundary. This enabled efficient computation of persistent local homology, which captures features like straight lines in addition to holes.[36] Alpha complex persistence diagrams for 0° and 1° homology were computed using the R package TDA,[37] which wraps the software GUDHI[38] for such computations. Persistence landscapes were computed using the R package tda-tools[39] which wraps the persistence landscapes toolbox.[40] The first 50 landscapes were retained for each degree of homology, and each was discretized using 201 values for a total of 10,050 dimensions per homology degree in the final output vectors and 20,100 dimensions in total.

Other featurization methods

We use persistence landscapes[7] to convert persistence diagrams into vectors, which are then used for machine learning. There are many other ways to do this[41] and each of these may be used with our method in place of persistence landscapes. We chose persistence landscapes for two advantages. First, they do not lose any information—one may recover the persistence diagram from the persistence landscape.[7] Second, in contrast to methods that use a parameter to determine a level of smoothing,[42–44] persistence landscapes are parameter free. We note that, for computational reasons, we use a discrete form of the persistence landscape[40] that evaluates the persistence landscapes on a chosen grid, which is equivalent to requiring the coordinates of the points of the persistence diagram to lie on the same grid.

SVR on patch landscapes

Two class (control versus experimental group) SVR models were trained on a diverse range of datasets. For each dataset, images were used to generate patches and corresponding persistence landscapes. Persistence landscapes were labeled based on treatment. Every model training session consisted of first splitting the set of images into a testing set (approximately 20% of the data) and a training set (approximately 80% of the data). The labeled patch landscapes from training images were used to train a linear SVR model (L2 regularized and L2 loss) via the R wrapper for Liblinear.[45] The cost parameter C was selected using the package's built-in heuristic on the training landscapes. Per-class cost weights were used to penalize misclassification of training points in the smaller class more heavily. Accuracy on patches was assessed by comparing labels assigned to the testing data by the model to the actual labels. The trained SVR model assigned a score to each patch landscape. These scores were averaged for each image to assign aggregate scores to all training and testing images. A second SVR was trained similarly to the first, but with the training image scores and labels. Image accuracy was assessed by comparing predicted image labels with actual labels for the testing set.

Assessing model performance

A single round of model assessment on a dataset consisted of first generating persistence landscapes from the dataset, then performing 5-fold cross-validation by training and testing both patch and image level SVR models from the persistence landscapes. The relatively small size of the datasets induced substantial variance in accuracy across folds. The total number of correctly classified testing images across folds was reported, rather than average testing accuracy across folds.

Since patch selection had a non-deterministic component, performance on each dataset was assessed by conducting 50 independent rounds of model assessment with the same hyperparameters and recording the average accuracy across the 5 cross-validation folds for each round. Table S1 reports summary statistics for these accuracies on each dataset. Standard deviations for image classification accuracy were uniformly less than 5% across all datasets, indicating low variance induced by random sampling.

Receiver operating characteristic curves (ROCs) and their corresponding area under curve (AUC) scores, precision, recall, and F1 scores were also recorded across the 50 independent rounds of model assessment on each dataset. Table S1 indicates which label in each dataset was considered ''positive'' when computing these scores. Testing results from all 5 cross-validation folds were aggregated to compute these scores for each round of model assessment. Tables S4 and S5, and Figures S9 and S10 report summary statistics for these performance metrics.

Hyperparameter selection

The radius of patches (in pixels) and the number of patches sampled per image were hyperparameters in the persistence landscape generation procedure. The number of patches per image was selected by specifying the ratio of the total number of pixels in all patches to be sampled from an input image to the number of pixels in the image, which was called the ''patch ratio.''

The standard practice of tuning hyperparameters by splitting datasets into hyperparameter training, model training, and model testing sets was infeasible given the small number of images in the datasets. Instead, a uniform patch radius of 75 pixels and patch ratio of 2 was used for all models. The radius was initially selected based on the size of biologically relevant features in the PFN1 dataset, while the patch ratio of 2 was selected arbitrarily.

To verify that this choice of hyperparameters did not produce misleading results, model performance using a range of hyperparameter values was assessed. For each dataset, two single rounds of model assessment were performed for each choice of hyperparameters with patch radii taking values of 30, 60, 90, 120, 150, and 180, and patch ratios taking values of 0.33, 0.67, 1.00, 1.33, 1.67, and 2.00. In total there were 36 combinations of hyperparameters, each assessed twice, for 72 single rounds per dataset. Altering the patch radius or ratio did not significantly affect image accuracy for our CK-666 dataset (Figure S3B). More generally, we also found that classification accuracy with patch radius of 75 pixels and patch ratio of 2 was not outlying for any dataset (Figure S3C).

Generating image masks from patch scores

The SVR models and t-SNE dimensionality reduction allowed the assignment of scores to image patches based on the patches' persistence landscapes. To visualize these scores on the original images, a modified version of kernel convolution was used. Patches and persistence landscapes were computed for an image, and each pixel in the mask was assigned the average score of patches containing the pixel. For t-SNE-based scoring, mask pixels were then scaled to values between -1 and 1. Two thousand patches with a radius of 75 were sampled in total. Masks based on SVR classification were generated for all images in each dataset. The number of patches and pixel radius were the same as those used for training the model.

Generating line summaries from image masks

To further explore the spatial distribution of topological differences identified by the SVR models, image masks were converted to line summaries for actin-labeled datasets. The centroid of each image and the distance of a mask's pixels from its centroid were computed. Distances were scaled between 0 and 1, and the average mask intensity for pixels with distances in bins of width 5% (0%–5%, 5%–10%, etc.) recorded. The resulting line summaries contained 20 scores, one for each bin. For each image class the average line summary and confidence bands containing 95% of scores were computed. Line summaries were visualized by linearly interpolating between the 20 data points.

Computational performance assessment

Computational performance was assessed by conducting 5-fold cross-validation rounds on the PFN1 dataset for a range of hyperparameter values. Selected patch radii values were 25, 50, 75, 100, and 125. Patch ratio values were 0.4, 0.8, 1.2, 1.6, and 2. In total there were 25 combinations of hyperparameters, each assessed once. Computations were performed using the University of Florida Research Computing's ''HiPerGator'' cluster. Parallelization was utilized with 24 computation threads (reported as ''CPU's'') on a single Intel Gold 6142 2.6 GHz processor. Resource usage statistics were collected using the monitoring software REMORA.[46] Figure S6 records the results. Resource demands scale with the total number of persistence landscapes computed, which in turn is directly proportional to the patch ratio divided by the patch radius squared (see Equation S1).

Comparison with transfer learning

To compare TDAExplore with standard models in the field, we used CNNs. While CNNs typically require large datasets,[28] transfer learning can be applied to small datasets.[29] Transfer learning is a machine learning method in which a model trained for a certain task is reused as the starting point for a different task. We used the 71-layer-deep Xception CNN architecture for transfer learning. First, we obtained a publicly available pre-trained version of the model that was trained by its maintainers on more than a million images from the ImageNet database.[30] We then updated its training using patch data from our PFN1 KO and CK-666 datasets. Since TDAExplore uses patches for training, 150 × 150 patches (474 patches per image) were generated from the PFN1 KO and CK-666 datasets. These patches were divided into training (75%) and testing (25%) groups and fed to the neural network. The training was run in 5 epochs for both the PFN1 KO and CK-666 datasets, with a total run time of 3,213 and 2,958 s, respectively, on a single GPU. Additional metrics of performance (ROC, AUC, precision, F1 score, and recall) are reported in the supplemental information (see Figure S11 and Table S6).

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.patter.2021.100367.

ACKNOWLEDGMENTS

Computations were performed using the University of Florida Research Computing's supercomputer ''HiPerGator.'' This project was supported by a National Institutes of Health (NIH) Pathway to Independence Award (R00 NS087104) and an NIH Maximizing Investigators' Research Award for Early Stage Investigators (R35 GM137959) to E.A.V. This research was partially supported by the Southeast Center for Mathematics and Biology, an NSF-Simons Research Center for Mathematics of Complex Biological Systems, under National Science Foundation grant no. DMS-1764406 and Simons Foundation grant no. 594594. This material is based upon work supported by, or in part by, the Army Research Laboratory and the Army Research Office under contract/grant no. W911NF-18-1-0307. We would like to thank Maurice Swanson (University of Florida) for introducing P.B. and E.A.V. in the hopes that our work would complement each other, which ultimately led to this project. P.E. would like to thank Mark McCurry for helpful comments on a draft of this manuscript.

AUTHOR CONTRIBUTIONS

Conceptualization, P.E., P.B., and E.A.V.; methodology, P.E., K.S., N.M., P.B., and E.A.V.; formal analysis, P.E., K.S., and N.M.; investigation, P.E., K.S., N.M., J.B.H., T.-A.R., P.B., and E.A.V.; resources, K.S., T.-A.R., P.B., and E.A.V.; writing – original draft, P.E., N.M., K.S., P.B., and E.A.V.; writing – review & editing, P.E., N.M., K.S., P.B., and E.A.V.; visualization, P.E., K.S., and N.M.; project administration, P.B. and E.A.V.; supervision, P.B. and E.A.V.; funding acquisition, P.B. and E.A.V.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: June 30, 2021
Revised: August 31, 2021
Accepted: September 20, 2021
Published: October 12, 2021

REFERENCES

1. Tian, D. (2013). A Review on Image Feature Extraction and Representation Techniques. Int. J. Multimedia Ubiquitous Eng *8*, 385–396.
2. Moen, E., Bannon, D., Kudo, T., Graf, W., Covert, M., and Van Valen, D. (2019). Deep learning for cellular image analysis. Nat. Methods *16*, 1233–1246.
3. Carpenter, A.E., Jones, T.R., Lamprecht, M.R., Clarke, C., Kang, I.H., Friman, O., Guertin, D.A., Chang, J.H., Lindquist, R.A., Moffat, J., et al. (2006). CellProfiler: image analysis software for identifying and quantifying cell phenotypes. Genome Biol. *7*, R100.
4. Kensert, A., Harrison, P.J., and Spjuth, O. (2019). Transfer learning with deep convolutional neural networks for classifying cellular morphological changes. SLAS Discov. *24*, 466–475.
5. Sahlol, A.T., Kollmannsberger, P., and Ewees, A.A. (2020). Efficient classification of white blood cell leukemia with improved swarm optimization of deep features. Sci. Rep. *10*, 2536.
6. Edelsbrunner, H., and Harer, J. (2008). Persistent homology—a survey. In Surveys on Discrete and Computational Geometry Contemporary Mathematics, J.E. Goodman, J. Pach, and R. Pollack, eds. (American Mathematical Society), pp. 257–282.
7. Bubenik, P. (2015). Statistical topological data analysis using persistence landscapes. J. Machine Learn. Res. *16*, 77–102.
8. Carlsson, G. (2009). Topology and data. Bull. Amer. Math. Soc. *46*, 255–308.
9. Nguyen, D.D., Cang, Z., and Wei, G.-W. (2020). A review of mathematical representations of biomolecular data. Phys. Chem. Chem. Phys. *22*, 4343–4367.
10. Pun, C.S., Xia, K., and Lee, S.X. (2018). Persistent-homology-based machine learning and its applications—a survey. SSRN J.
11. Reed, T.R., and Dubuf, J.M.H. (1993). A review of recent texture segmentation and feature extraction techniques. CVGIP: Image Understand. *57*, 359–372.
12. Kumar, G., and Bhatia, P.K. (2014). A detailed review of feature extraction in image processing systems. In 2014 Fourth International Conference on Advanced Computing & Communication Technologies (IEEE), pp. 5–12.
13. Stolz, B.J., Tanner, J., Harrington, H.A., and Nanda, V. (2020). Geometric anomaly detection in data. Proc. Natl. Acad. Sci. U S A *117*, 19664–19669.
14. Bendich, P., Wang, B., and Mukherjee, S. (2012). Local homology transfer and stratification learning. In Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, Y. Rabani, ed. (Society for Industrial and Applied Mathematics), pp. 1355–1370.
15. Hu, X., Fuxin, L., Samaras, D., and Chen, C. (2019). Topology-preserving deep image segmentation. arXiv.
16. Tajbakhsh, N., Jeyaseelan, L., Li, Q., Chiang, J.N., Wu, Z., and Ding, X. (2020). Embracing imperfect datasets: a review of deep learning solutions for medical image segmentation. Med. Image Anal. *63*, 101693.
17. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (IEEE), pp. 2921–2929.
18. Caicedo, J.C., Cooper, S., Heigwer, F., Warchal, S., Qiu, P., Molnar, C., Vasilevich, A.S., Barry, J.D., Bansal, H.S., Kraus, O., et al. (2017). Data-analysis strategies for image-based cell profiling. Nat. Methods *14*, 849–863.
19. McQuin, C., Goodman, A., Chernyshev, V., Kamentsky, L., Cimini, B.A., Karhohs, K.W., Doan, M., Ding, L., Rafelski, S.M., Thirstrup, D., et al. (2018). CellProfiler 3.0: next-generation image processing for biology. PLoS Biol. *16*, e2005970.
20. Uhlmann, V., Singh, S., and Carpenter, A.E. (2016). CP-CHARM: segmentation-free image classification made accessible. BMC Bioinformatics *17*, 51.
21. von Chamier, L., Laine, R.F., Jukkala, J., Spahn, C., Krentzel, D., Nehme, E., Lerche, M., Hernández-Pérez, S., Mattila, P.K., Karinou, E., et al. (2021). Democratising deep learning for microscopy with ZeroCostDL4Mic. Nat. Commun. *12*, 2276.
22. Lee, C.W., Vitriol, E.A., Shim, S., Wise, A.L., Velayutham, R.P., and Zheng, J.Q. (2013). Dynamic localization of G-actin during membrane protrusion in neuronal motility. Curr. Biol. *23*, 1046–1056.
23. Vitriol, E.A., McMillen, L.M., Kapustina, M., Gomez, S.M., Vavylonis, D., and Zheng, J.Q. (2015). Two functionally distinct sources of actin monomers supply the leading edge of lamellipodia. Cell Rep. *11*, 433–445.
24. Skruber, K., Read, T.-A., and Vitriol, E.A. (2018). Reconsidering an active role for G-actin in cytoskeletal regulation. J. Cell Sci. *131*, jcs203760.
25. Skruber, K., Warp, P.V., Shklyarov, R., Thomas, J.D., Swanson, M.S., Henty-Ridilla, J.L., Read, T.-A., and Vitriol, E.A. (2020). Arp2/3 and Mena/VASP require profilin 1 for actin network assembly at the leading edge. Curr. Biol. *30*, 2651–2664.e5.
26. van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-SNE. J. Machine Learn. Res. *9*, 2579–2605.
27. Statnikov, A., Wang, L., and Aliferis, C.F. (2008). A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinformatics *9*, 319.
28. Khan, A., Sohail, A., Zahoora, U., and Qureshi, A.S. (2020). A survey of the recent architectures of deep convolutional neural networks. Artif. Intell. Rev.
29. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). A survey on deep transfer learning. In Artificial Neural Networks and Machine Learning—ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4–7, 2018, Proceedings, Part III Lecture Notes in Computer Science, V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, eds. (Springer International Publishing), pp. 270–279.
30. Chollet, F. (2017). Xception: deep learning with depthwise separable convolutions. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (IEEE), pp. 1800–1807.
31. Rotty, J.D., Wu, C., Haynes, E.M., Suarez, C., Winkelman, J.D., Johnson, H.E., Haugh, J.M., Kovar, D.R., and Bear, J.E. (2015). Profilin-1 serves as a gatekeeper for actin assembly by Arp2/3-dependent and -independent pathways. Dev. Cell *32*, 54–67.
32. Vitriol, E.A., Wise, A.L., Berginski, M.E., Bamburg, J.R., and Zheng, J.Q. (2013). Instantaneous inactivation of cofilin reveals its function of F-actin disassembly in lamellipodia. Mol. Biol. Cell *24*, 2238–2247.
33. Wilson, T. (2011). Resolution and optical sectioning in the confocal microscope. J. Microsc. *244*, 113–121.
34. Ljosa, V., Sokolnicki, K.L., and Carpenter, A.E. (2012). Annotated high-throughput microscopy image sets for validation. Nat. Methods *9*, 637.
35. Shamir, L., Orlov, N., Mark Eckley, D., Macura, T.J., and Goldberg, I.G. (2008). IICBU 2008: a proposed benchmark suite for biological image analysis. Med. Biol. Eng. Comput. *46*, 943–947.
36. Hatcher, A. (2009). Algebraic Topology, 1st ed. (Cambridge University Press).
37. Fasy, B.T., Kim, J., Lecci, F., and Maria, C. (2014). Introduction to the R package TDA. arXiv https://arxiv.org/abs/1411.1830.
38. Maria, C., Boissonnat, J.-D., Glisse, M., and Yvinec, M. (2014). The GUDHI library: simplicial complexes and persistent homology. In Mathematical Software—ICMS 2014 Lecture Notes in Computer Science, H. Hong and C. Yap, eds. (Springer Berlin Heidelberg), pp. 167–174.
39. (2014). Jose bouza GitHub-jjbouza/tda-tools: pipeline used internally for Peter Bubenik's TDA group at UF. https://github.com/jjbouza/tda-tools.
40. Bubenik, P., and Dłotko, P. (2017). A persistence landscapes toolbox for topological statistics. J. Symbolic Comput. *78*, 91–114.

41. Barnes, D., Polanco, L., and Perea, J.A. (2021). A comparative study of machine learning methods for persistence diagrams. Front. Artif. Intell. *4*, 681174.

42. Reininghaus, J., Huber, S., Bauer, U., and Kwitt, R. (2015). A stable multi-scale kernel for topological machine learning. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (IEEE), pp. 4741–4748.

43. Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., et al. (2017). Persistence images: a stable vector representation of persistent homology. J. Machine Learn. Res. *18*, 1–35.

44. Le, T., and Yamada, M. (2018). Persistence Fisher kernel: a riemannian manifold kernel for persistence diagrams. arXiv.

45. Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: a library for large linear classification. J. Machine Learn. Res. *9*, 1871–1874.

46. Rosales, C., Gómez-Iglesias, A., and Predoehl, A. (2015). Remora: a resource monitoring tool for everyone. In Proceedings of the Second International Workshop on HPC User Support Tools—HUST '15 (ACM Press), pp. 1–8.