MODELING AND ANALYSIS OF MITOCHONDRIAL DYNAMICS USING DYNAMIC SOCIAL NETWORK GRAPHS

by

Marcus D. Hill

(Under the Direction of Shannon Quinn)

Abstract

Graphs are ubiquitous data structures that eloquently express complex relationships, while simultaneously capturing underlying patterns that exist within data. These properties make graphs preferred for modeling complex systems, such as the dynamics of mitochondria. Traditionally, analysis of mitochondrial dynamics required manual inspection of microscopy imagery by specialists, which was a tedious and error prone process. These manual and strenuous tasks combined with the dramatic increase in the volumes of microscopy imagery being generated has created a strong demand for automated approaches for modeling subcellular organelles. Yet, modeling mitochondria is far from a trivial task because they do not adhere to any predetermined shape, but rather they are amorphous, diffuse structures. In this work, we provided a novel-graph based methodology for modeling and analyzing the morphology and dynamics of mitochondria. Ultimately, in our pursuit of gaining more insights regarding the dynamics of mitochondria, we have developed temporal and spatial anomaly detection techniques and novel strategies for conducting mitochondrial dynamics classification.

Index words: Anomaly Detection, Graph Representation Learning, Object Tracking, Object Detection, Clustering, Time-Series Analysis, Organellar Modeling

MODELING AND ANALYSIS OF MITOCHONDRIAL DYNAMICS USING DYNAMIC SOCIAL NETWORK GRAPHS

by

Marcus D. Hill

B.S., University of Georgia, 2017

A Dissertation Submitted to the Graduate Faculty of the
University of Georgia in Partial Fulfillment of the Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2022

MODELING AND ANALYSIS OF MITOCHONDRIAL DYNAMICS USING DYNAMIC SOCIAL NETWORK GRAPHS

by

Marcus D. Hill

Major Professor: Shannon Quinn

Committee: Liming Cai

Sheng Li

Electronic Version Approved:

Ron Walcott

Dean of the Graduate School

The University of Georgia

May 2022

Dedication

To the memory of my beloved grandmother, Deborah Theresa Hill-Curry.

Acknowledgments

This work is not just a reflection of my scholastic journey but also a reflection of all of the love and support I received over the last few years. I would like to thank every person that has contributed – even in the slightest way – to me making it to this point in life. There are a few individuals and groups that have been paramount to this achievement, and I would like to specifically thank them.

First and foremost, thank you, God, for your love and guidance. You have been my rock and foundation through the many highs and lows of this journey. Second, I would like to thank my committee for their mentorship. I was blessed with a great committee, and I will never be able to say that enough. In particular, thank you, Dr. Quinn, for guiding me from start to finish.

Next, I have to recognize my friends that have made these last few years a memorable experience. My "Lab 307" friends made coming to campus an exciting, and loud, experience every time. The many laughs we shared balanced out all the stress and hard nights of grad school. My brothers in Sigma, especially "Wholesome Team", made sure I lived and did not let great moments pass me by, while also providing wisdom, advice, and support whenever needed. "The Trap" made UGA feel like home and its members have been with me from the very beginning. I have to give a special mention to Neelima for being one of the most integral pieces to my support system and a great collaborator.

Last, but not least, I must honor my family and loved ones. The following people were there every step of the way and words will never fully express my gratitude: Marvin Hill, Geraldine Hill, Jasmine Huff, Brandon Hill, Jimmy Huff, Algerine McGirt, Willie McGirt Sr., Reginald McGirt, Lelola McGirt, Phillip McGirt, Willie McGirt Jr., and Andrew McGirt. I am grateful for the love and many forms of support given by Mayah Williams.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Graphs are ubiquitous data structures that eloquently express complex relationships, while simultaneously capturing underlying patterns that exist within data. These properties make graphs preferred for modeling various phenomena across different domains. Notably, graphs have modeled social communities [4], [9], internet networks [60], and transportation routes [60]. Another significant, yet challenging, area for graphs to be applied is towards the modeling of biological structures, specifically subcellular organelles.

Subcellular organelles are microscopic structures responsible for the proper functioning of cells, and abnormalities regarding their morphology and behavioral patterns are indicators of poor cellular health [33], [108]. Traditionally, analysis of organellar dynamics required manual inspection of microscopy imagery by specialists, which was a tedious and error prone process [108]. These manual and strenuous tasks combined with the dramatic increase in the volumes of microscopy imagery being generated has created a strong demand for automated approaches for modeling subcellular organelles [20]. Yet, modeling these structures is far from a trivial task because some organelles do not adhere to any predetermined shape, but rather they are amorphous, diffuse structures. Additionally, these difficulties are further exacerbated by the presence of infections and foreign invaders because many pathogens specifically target organelles and alter their morphology and motility patterns [80][29]. Due to such challenges, many modern imaging

techniques fail to sufficiently capture significant aspects of more amorphous structures, like mitochondria, but are capable of accurately modeling structures with more rigid and well-defined shapes, like nuclei. Fortunately, graph-based modeling approaches do not share these limitations.

In this work, we will describe in depth the graph-based modeling approach that we have utilized to model mitochondria and the downstream analysis tasks that we have performed to elucidate mitochondrial dynamics. Our novel contributions:

- Developed methodologies for identifying temporal and spatial anomaly detections of mitochondria in fluorescent microscopy imagery.
- Engineered multiclass datasets for mitochondrial dynamics classification.
- Developed an approach for classifying implicitly encoded mitochondrial dynamics using dynamic social network graphs.
- Established and implemented dynamic node social network graphs for organellar modeling.
- Developed an approach for classifying mitochondrial dynamics explicitly encoded in time-series datasets.

1.2 Significance

Subcellular organelles are responsible for performing vital roles within cells, and if these specialized structures deviate from performing their intended functions then the host could suffer serious, if not fatal, repercussions. Understanding the behaviors of these structures is critical for addressing various health issues. Ideally, having a mechanism for evaluating the health of a cell is desired to preemptively identify and mitigate problems. As a result, a useful organelle to assess is mitochondria because various pathogens target this structure in an effort to maintain an environment that allows them to replicate and thrive [80][29]. A notable example of this behavior has been demonstrated by the Mycobacterium tuberculosis pathogen [33], which is the etiologic agent for tuberculosis in humans [24].

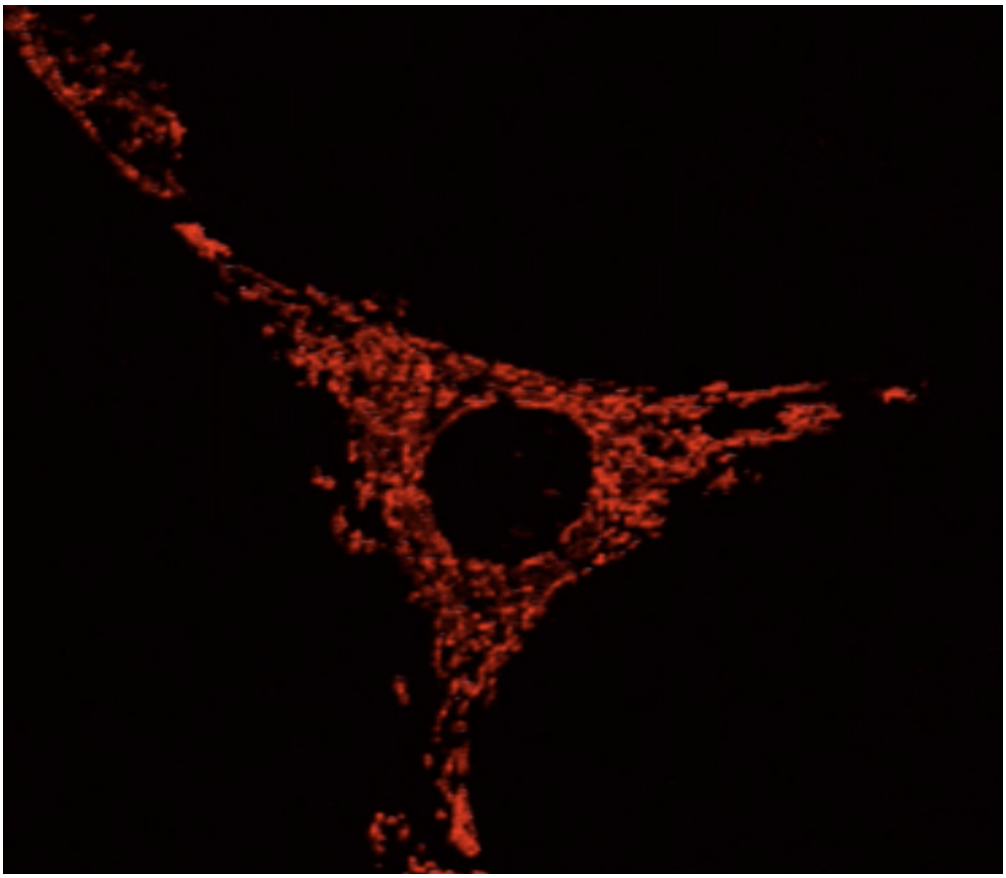

Figure 1.1: Mitochondria in a HeLa cell.

Prior to the coronavirus pandemic, tuberculosis was the leading cause of death from a single infectious agent [42]. In the year 2020, approximately 1.5 million people died as the result of a tuberculosis infection, and 5.8 million people were reported as being newly diagnosed with the disease [42]. Understanding the relationship between this disease and the cells it infects is necessary to reduce the number of new cases per year. Thus, elucidating the dynamics of mitochondria may be key to developing new medical treatments and therapies to address tuberculosis infections and other deadly diseases.

Mitochondrial functions, such as energy metabolism, lipid synthesis, and autophagy regulation are linked to the dynamics of the organelle [80]. Pathogens seek to inhibit such functions by inducing fission and fusion events to alter the organelle's morphology, and, ultimately, reprogram the immunological

response of the host [80]. Therefore, knowledge of the morphological changes of mitochondria is crucial for understanding a host's response to infections or cellular invaders. Mitochondrial dynamics have been assessed from various perspectives, including microscopy image segmentation [20][108], cellular classification [35][59][75][84][108], and motion analysis [108]. Prior approaches [50][92][89][18] utilized standard image processing techniques to learn mitochondrial textures and extract features from the imagery to perform such downstream analysis tasks. However, mitochondrial dynamics can be explored even further by modeling the organelle using dynamic graph structures.

Graph-based approaches are capable of performing similar classification and motion analysis tasks, while providing richer insights regarding the structure and relationships of mitochondria because graphs can characterize the mitochondria at various levels of granularity. Additionally, graph-based approaches can highlight the temporal relationships that exist between clusters of mitochondria, which can subsequently be leveraged to identify when morphologically significant events occur as well as the specific portions of the organelle involved.

1.3 Data

In our efforts to demonstrate the morphological spectrum that mitochondria undergo, we have amassed a collection of confocal imaging videos of live HeLa cells [57] fluorescently tagged with the protein DsRed2-Mito-7[1] for the experiments that we describe later in this work (1.1). We maintained three distinct groups of cells: a group that was not exposed to any external stimulant, referred to as our control group; a group that was exposed to listeriolysin O (llo), a pore-forming toxin, to induce mitochondrial fragmentation; and a group that was exposed to mitochondrial-division inhibitor 1 (mdivi) to induce mitochondrial fusion. Live imaging videos of each cell were recorded with a Nikon A1R confocal microscope. The imaging occurred in an environment that maintained 37 degrees celsius and 5% CO_2. Every imaging video consists of at least 20,000 frames, of dimensions 512x512, captured at 100 frames per second. In all of our imagery,

[1] DsRed2-Mito-7 was a gift from Michael Davidson (Addgene plasmid # 55838 ; http://n2t.net/addgene:55838 ; RRID:Addgene_55838)

each red “dot” depicts a single mitochondrion within a cell. For scale, the length of mitochondria is typically between 500 nm to 1 mm or greater, and the average diameter is approximately 500 nm [67][26].

Chapter 2

History of Mitochondria

Mitochondria are double membrane-bound organelles present in all eukaryotic cells. They are powerhouses that drive ATP production through aerobic respiration and have served as a driving force for evolution [77][37][15]. These organelles are thought to have evolved from endosymbiotic bacteria that was engulfed by a cell. Lynn Margulis published this theory in her work On the Origin of Mitosing Cells in 1967, a view that was also widely held by other researchers in the 20th century [87]. This idea was initially controversial but research in later decades proved that the origin of these organelles was distinct from their eukaryotic host's lineage [65][87]. Phylogenomic analyses have mapped the history of eukaryotes to Archeal descent [65][30]. The new discovery of a group of Archaea, called Asgards, helped establish the provenance of eukaryotic cells [30][64]. This theory persists to be a point of contention in the biological domain but is nonetheless widely accepted. Mitochondria, also through modern analyses, are known to have originated from alphaproteobacteria, but studies to determine their exact ancestor remain to produce conflicting results [65][30][64][87].

Mitochondria has been researched thoroughly in the last fifty years and it has show that these crucial organelles not only act as the cell's powerhouse but also play a large role in cell physiology, apoptosis, radical production, aging, innate immunity, and pathology [77][37][15]. The host cell provides the mitochondria with food and oxygen which the mitochondria uses to convert into chemical energy through a function called oxidative phosphorylation. [86] To aid and perform in their tasks, mitochondria undergo

dynamic cycles of fission and fusion to maintain shape, distribution and size. Additionally, their dynamic structure and morphological changes are crucial for many cellular processes and allow them to respond to cellular needs by adapting to nutrient availability and to maintain the metabolic state of the cell. Naturally, cells that demand higher energy contain larger numbers of mitochondria, which divide using their own strand of DNA within a cell. Rapid fission and fusion events allow for redistribution of mitochondria to accommodate cell needs. Mitochondrial fusion is a process of one mitochondrion breaking apart into two mitochondrial bodies. Mitochondrial fusion is the physical merging of two distinct mitochondria to create one elongated mitochondria. Fusion enables complementation, a process where two mitochondrial bodies with different defects inside the same organelle can merge to create a single functioning mitochondrion [86]. This helps mitigate stress as partially damaged mitochondrion can fuse with other damaged mitochondria. Fission is needed to create new mitochondria but also aids in the removal of damaged mitochondria. This facilitates apoptosis during high levels of cellular stress and contributes to quality control. Fragmentation can modify other aspects of the organelle's function including susceptibility to mitochondrial permeability transition, respiratory properties of the electron transport chain, and reactive oxygen species production [77][37]. This division process is important in remodeling and rearrangement of mitochondrial networks that is crucial for cell division when it is necessary to transport mitochondria to daughter cells during mitosis and meiosis. Mitochondria regularly go through these morphological transitions regulated by dynamic processes of membrane fission and fusion in response to both cellular and environmental stresses [77]. These mitochondrial dynamic events are essential for cell activity and directly and indirectly associated with mitochondrial maintenance, regulating cellular activity and play a crucial role in signaling pathways [37][15]. They occur in a constant balanced manner to aid in the metabolic needs of the cell [94]. Correct regulation of these fusion and fission events is essential given the pivotal role mitochondria play in cell physiology.

Research has shown that disruptions in these processes affect normal development and cell functions. Irregularities in these morphologies have been linked to neurodegenerative diseases, including Parkinson disease [98], Alzheimer disease [1] [95] and Huntington's disease [21]. Dysfunction in mitochondria

can account for generation of free radicals and deficiencies in energy supply, calcium buffering or rapid apoptosis all of which can contribute to neurodegenerative disorders [94]. Understanding the dynamics of mitochondria provides researchers with an understanding of the pathophysiology of mitochondrial disorders and other diseases linked to mitochondrial dysfunction. Knowledge on mitochondrial dynamics allows scientists to gain more insight into their relationships with the cell cycle and with mitochondrial responses to pathophysiological perturbations, create new therapies, analyze their role in cell survival and death, and understand their relationship to proteins that determine the maintenance of mitochondrial ultrastructure and its close interactions with the endoplasmic reticulum. [44][5]

Chapter 3

Modeling Mitochondria via Dynamic Social Network Graphs

3.1 Background:

Fluorescence microscopy is a critical tool for researchers to understand the biological processes of the cell and analysis of microscopy images is the backbone of life sciences. However, screening microscopic slides is a laborious, manual process that can often lead to subjective classification [25]. Automation of this process can contribute to more reliable results and speed up diagnosis and treatment development processes. Image segmentation offers a way to extract cellular, nuclear and tissue components and cell classification can detect anomalous behavior, categorize unhealthy cells [105]. Methods such as these can help automate bio-analysis tasks.

Image segmentation is an image processing technique that aids automation in analyzing images. Segmentation of biological components such as cells and organelles is a fundamental step in biological studies. The dynamics of cells and cell structures can indicate the physiological state of the cell and lead to insights into the effects of pathogens on cells [105]. Medical imaging has produced several image processing methods in recent decades, many of which help capture biologically relevant information on cell dynamics. Segmenting cells is a difficult task as cells can have diverse shapes, image quality of the cell can be lacking,

cells can overlap with neighbor cells [27]. Segmentation of microscopic images refers to the process of finding the boundaries of cells or cell structures.

One of the most well known segmentation methods is the watershed method [7]. Watershed segmentation maps a grayscale image as a topographic landscape with ridges and valleys. The gray values of pixels or their gradient magnitude defines the elevation of the landscape. Based on the 3D representation, the image is transformed into catchment basins through the watershed transform. A catchment basin contains all the minimum value pixels. It decomposes an image and assigns each pixel to a region or a watershed. Intensity thresholding is another type of segmentation in which a single parameter known as intensity threshold determines whether a pixel falls into the background or foreground. This method takes in a grayscale image and outputs a binary image with only black and white pixels [78]. The two of these methods do pose some disadvantages as they can result in inaccurate cell boundaries and can over-segment the image [78][100].

Despite their disadvantages, several state-of-the-art cell segmentation frameworks use both these methods and are proven effective in segmenting microscopy images. The frameworks include Cellsegm [50], SMASH [92], ImageJ [89], and CellProfiler [18]. Cellesgm and SMASH utilize watershed thresholding and a few other traditional algorithms such as Otsu's thresholding algorithm [72]. Occasionally, segmentation frameworks would also include machine learning algorithms for further downstream analysis. But these models suffer from over-segmentation due their use of low level segmentation methods [100].

Recent deep learning advancements have led to several methods to perform cell classification and modeling effectively. The rise of convolutional neural networks and other deep learning architectures has created several new benchmarks and has shifted automated classification of cells and cellular structures. Researchers have adapted deep learning architectures for cell and nuclei segmentation and detection. For example, Zhu et al. [109] use fully connected networks (FCN) to detect and count cells. Fujita and Han [36] proposed a MASK R-CNN architecture to detect and segment cells. Lu et al. [62] proposed an architecture called WBC-Net that utilizes UNET++ and ResNet to segment and identify white blood cells. These architectures have an end-to-end segmentation and tracking pipeline allowing one architecture to

both segment and detect cells. Traditional methods on the other hand have a poor generalization ability and require manual parameter tuning, giving deep learning architectures an advantage [90][109][36].

Unfortunately, these models cannot be generalized across all cell types thus cell classification remains to be a challenging task. Many small blob and particle detection methods have been applied for cell detection but most of these methods cannot deal with irregular shaped cells and are only effective for the single cell category they are trained on. Almost all methods only deal with classifying and detecting specific kinds of cells [109].

Additionally, even fewer methods exist that help effectively track and analyze subcellular organelles. Automatic quantification of mitochondria is a very difficult task as they are amorphous structures that have a diverse range of shapes. Researchers initially performed qualitative assessment of mitochondrial morphology but this led to changes in morphology going undetected [108]. A few methods have been developed to track the morphologies of mitochondria and assess their anomalous dynamics.

Oztel et al. [73] proposed a deep learning approach for automated segmentation of mitochondrial structures. They use a CNN architecture consisting of four convolutional layers, three pooling layers and one fully connected layer. This method applies pixel level classification to segment mitochondria out of high resolution electron microscopy images. These images are different from what we use in our problem context. Although very useful, this method is only useful for segmenting mitochondria and not for tracking or classifying mitochondrial morphologies [73].

Iqbla et al. [90] adapted GoogLeNet-22 [96], a CNN architecture, to distinguish between normal mitochondria and mitochondria affected by drugs. In this method, cells are posed and the convolutional neural network was used to extract features from TPFP (two-photon excited fluorescence probe) compound images and map them to feature space [90]. Mitometer [58] is a software package that allows for automated segmentation and spatiotemporal tracking of mitochondria in live-cell fluorescence microscopy time-lapse images. It is able to identify changes in mitochondrial morphology only using the pixel size and time between the frames. For classification, Mitometer's algorithm compares the track areas before and after fission or fusion and adds a track as a candidate if the area difference is below 1 standard

deviation. It performs segmentation by assigning all true pixels in the final binary mask to a mitochondrion. All spatially connected true pixels in the final mask are determined to belong to an individual mitochondria, whereas pixels separated by false pixels belong to different mitochondria.

MitoMo [108] is another automated image/video processing and machine-learning software that is designed to work with fluorescent images. The pipeline can process images with any mitochondrial-targeted dye and segment mitochondria using global or adaptive thresholding and de-clumping procedures. Other software such as CellProfiler can also be used to specifically segment the organelles. Finally classification is performed using machine-learning algorithms, K-nearest neighbor (KNN) [34] and Naïve Bayes algorithms [28].

Although these software and methods are very useful and insightful, they do not tackle the same problem as OrNet and are limited to specific types of mitochondria.

3.2 Methods: Constructing Dynamic Social Network Graphs

We developed a Python framework known as OrNet[1] (Organellar Networks) [31] to model mitochondria as dynamic social network graphs. A dynamic social network is a temporal graph $G = \{g_1, \ldots, g_n\}$, where each subgraph g_t is represented as a set of vertices V and edges E that encode explicit and implicit information regarding the relationships of the entities being modeled at a discrete point in time t. In the context of organellar modeling, these graphs represent the evolving relationship of an organelle's subcellular structures across time. Specifically, we modeled mitochondrial clusters depicted in fluorescent microscopy videos as nodes within such graphs, and defined the edges as dynamic, weighted connections to quantitatively characterize the relationship between clusters. The edge weights are floating-point values that adjust according to the morphological changes demonstrated by the mitochondria in a microscopy video. Construction of these graphs requires transforming the mitochondrial cluster representations from

[1] M. Fazli*, **M. Hill***, A. Durden, R. Mattson, A. T. Loy, B. Reaves, A. Courtney, F.D. Courtney, C. Chennubhotla, and S. Quinn, "Ornet-a python toolkit to model the diffuse structure of organelles as social networks," Journal of Open Source Software, 2020. * indicates authors made an equal contribution.

pixels in an image space to nodes and edges within a topological space through a series of steps, including image segmentation and object tracking.

3.2.1 Cell Segmentation

The first step in our graph construction process is isolating every cell in our corpus of data into their own individual videos. Fluorescent microscopy videos may contain multiple cells within one video, so we perform this step to ensure that the graphs constructed do not represent biologically infeasible relationships, such as connections between mitochondrial clusters that exist in different cells; therefore, if a microscopy video only contains a single cell this step can be skipped. Our approach isolates, or extracts, each cell by performing image segmentation, meaning we identify the pixels in each video frame that are associated with each unique cell and create segmentation masks. Segmentation masks are, essentially, 2-dimensional arrays that match the shape of the input image, where each element in the mask contains either a value of '0' or '1'. Pixels labeled '1' indicate the spatial location of a foreground object of interest, while all other pixels are labeled '0' to indicate that they represent either background objects or other foreground objects that are not of interest. In order for us to isolate each cell into their own individual videos, we create segmentation masks for each unique cell on every frame of a video. For example, a microscopy video that captures the activities of 5 different cells would have 5 segmentation masks for every frame of that video; one mask for each unique cell. Thus, each cell has a collection of masks associated with it that identifies its spatial locations in every frame of the video, and we can leverage these collections of masks to separate each cell into their own distinct videos where the foreground only contains that specific cell. For simplicity, the number of mask collections managed for each video can be reduced to one by making one mask for each frame, where '0' still denotes background objects and each cell identified in the foreground is assigned a unique integer label; we took this approach in our framework.

Our cellular extraction process consists of manually crafting a segmentation mask for the first frame of videos using the ITK-SNAP [107] tool, performing contour tracking to identify the trajectories of the cells as they move about the scene over time, then finally creating individual videos for each cell. Optional

steps that we also perform when processing videos through this process is constraining the number of frames and normalizing pixel intensities in the video. In our experiments, we constrained the length of the videos because we noticed minimal activity in our cells after a certain period of time, so we focused our attention on the frames where the organelles were most active. Pixel intensities were normalized, specifically with the median normalization image processing technique, to minimize noise present in the imagery and reduce any potential effects of varying lighting conditions. Ultimately, the end result of this step is c single-cell videos, where c denotes the number of cells identified in the original microscopy video.

3.2.2 Mitochondrial Cluster Detection and Tracking

The next step in our graph construction process is identifying the mitochondrial clusters present in our videos. A single mitochondrion is virtually imperceptible in fluorescent microscopy imagery; however, when many of them are densely concentrated in a region, the cluster that they form is visible due to the organelle being fluorescently tagged with a specific color. Our goal is to represent these mitochondrial clusters as nodes within a dynamic social network graph. Higher concentrations of fluorescently tagged mitochondria clustered together emit brighter colors than regions sparsely populated with mitochondria, which emit duller hues of the tag color. Our approach leverages that property to detect clusters by identifying the local pixel intensity maximas, or peaks, on each frame of the videos; intuitively, brighter pixel intensities correspond to regions of denser concentrations of mitochondria. Computationally, these peaks are identified by applying a maxima filter to the original video frames and creating images that are dilated versions of the original frames. The original frames are then compared to their dilated versions, and pixel coordinates that share the same intensity value in both images are returned as peaks. Essentially, the local intensity peaks give a rough estimate to the spatial locations of the mitochondrial clusters as well as the number of clusters present in the imagery, but the shapes and sizes of the clusters are still unknown.

Our strategy for learning the shapes and sizes of the mitochondrial clusters depicted on each frame of the microscopy videos is by clustering the pixels of those frames. However, clustering image pixels to represent these structures is no trivial task because mitochondria clusters are spatially diffuse, amorphous

structures, meaning they do not adhere to any well-defined rigid shape. Therefore, more common clustering techniques, such as k-means clustering [66] and k-nearest neighbors [34] are rendered ineffective. K-means clustering, which is a clustering technique that assigns each data point to the closest of k possible clusters based on the distance between the point and each candidate cluster's centroid, is not well-suited for identifying mtiochondrial clusters because it's clustering strategy partitions the 2-dimensional data space according to fixed, circular shapes and the morphology of mitochondria is not constrained to any such shape. Additionally, k-means clustering tends to produce clusters of similar size, which is not always appropriate when modeling mitochondria because their subcellular structures can vary in size. K-nearest neighbor clustering, a technique that assigns cluster membership labels to data points according to the label of their k nearest neighbors, is also not suitable for this context because no pixels are labeled in advance. Thus, our approach must be able to cluster unsupervised data, leverage the densities represented by the magnitude of pixel intensities, and be able to produce clusters of variable sizes and shapes (figure 3.1).

We utilized a probabilistic expectation-maximization algorithm, known as Gaussian Mixture Models (GMM) [14], to address the strict requirements necessary to detect mitochondrial clusters. GMMs are probabilistic models that represent subpopulations, or clusters, within a data space, and the subpopulations are represented by gaussian distributions that are referred to as mixture components. It is rare to know the true membership assignments of data points to subpopulations, which is analogous to unlabeled datasets in machine learning contexts. GMMs can learn membership assignments through an iterative process of assigning data points to mixture components and evaluating the likelihood that each data point would originate from such a probability distribution, then updating the parameters of the mixture components, specifically the means and covariances of their gaussian distributions, to maximize the likelihood of such membership assignments. The parameters of each mixture component can be distinct, meaning each subpopulation can have varying shapes and sizes, which is necessary to appropriately model mitochondrial structures.

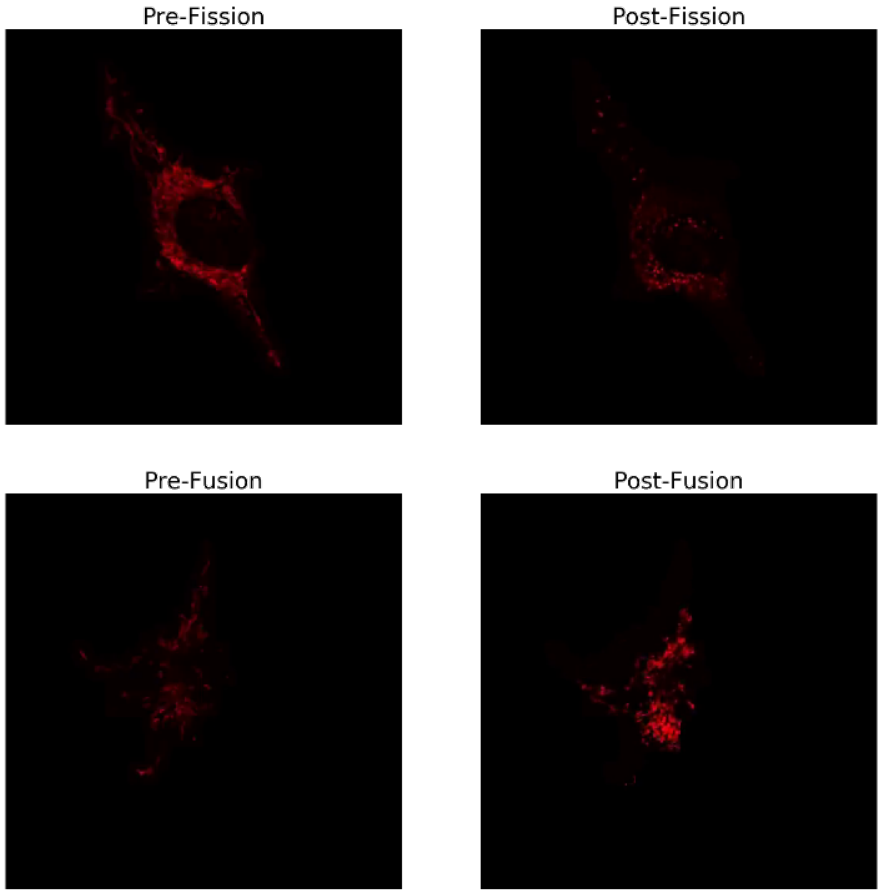

Figure 3.1: Mitochondrial Dynamics

Our process for applying GMMs to our microscopy videos is by first initializing models on each frame of the video, then performing the expectation-maximization learning process until the components converge to localize the spatial locations of the mitochondrial structures. The number of mixture components utilized is equal to the number of local peaks found on the first frame of the video. The GMM model applied to the first frame of the video is initialized with the means of the mixture components being set to the coordinates of the local peaks, the covariance matrices are set to the variance of the 8 pixel intensities surrounding their component's mean, and the weights of the components is a normalized vector of the

pixel intensities corresponding to the local peak coordinates. Once the parameters of the first GMM have converged, they are used to initialize the parameters of the GMM corresponding to the subsequent frame. The recursive initialization pattern continues until the last frame of each microscopy video, creating an explicit dependency that allows for tracking the trajectories of the mitochondrial clusters across a video.

3.2.3 Graph Construction

The converged mixture components represent the spatial locations and shapes of the mitochondrial clusters depicted on each frame of a video, and the relationships of these probability distributions are used to construct our dynamic social network graphs. We designate the mixture components to serve as the nodes in the dynamic networks because they quantitatively describe the mitochondrial clusters. Since the nodes of the graph are probability distributions, we can characterize the relationship between pairs of nodes by modeling the changes in their similarity over time. In essence, we can form weighted edges between the nodes of the graph by computing similarity scores between their corresponding probability distributions. There exist several metrics to measure the similarity between probability distributions, and we elected to use the Hellinger Distance (Eq. 3.1) [45] because it is a bounded distance function with a maximum distance of 1, making it simple to transform the metric into an affinity measure by computing 1 minus the distance value. By assigning the mixture components as the nodes of our graphs and repeating this edge construction process for every frame of the video, we create a dynamic graph that captures the subtle changes between mitochondrial clusters that reflect the state of the organelle as it undergoes morphology-altering events.

$$H^2(P,Q) = 1 - \frac{det({\Sigma_1}^{\frac{1}{4}})det({\Sigma_2}^{\frac{1}{4}})}{det(\frac{\Sigma_1+\Sigma_2}{2})^{\frac{1}{2}}} \exp\{-\frac{1}{8}(\mu_1-\mu_2)^T(\frac{\Sigma_1+\Sigma_2}{2})^{-1}(\mu_1-\mu_2)\} \tag{3.1}$$

3.3 Conclusion

Quantitatively modeling mitochondria's dynamics is a difficult task due to its formless morphology. To address this challenge, we developed a Python framework, known as OrNet, that represents organellar structures as nodes within dynamic social network graphs. The process of transforming the mitochondrial representations from pixels in an image space to nodes in a topological space required isolating each individual cell from the original fluorescent microscopy videos into single-cell videos, detecting local pixel intensity maxima to estimate the quantity and spatial locations of mitochondrial clusters in the videos, clustering the pixels of each video frame using probabilistic mixture models, before finally constructing dynamic graphs that model the relationships that exists between the probability distributions representing the mitochondrial clusters.

Chapter 4

Identifying Anomalies in Mitochondrial Dynamics

4.1 Introduction

Our purpose for modeling mitochondrial structures was to elicit new insights regarding the manner in which this organelle responds to changes in its environment, and to do so in an automated manner that addresses some of the modern challenges of analyzing large-scale amounts of data. Thus, as an initial exploration into understanding such dynamics we sought to characterize what are normal, healthy behavioral patterns from abnormal behaviors of mitochondria by leveraging our dynamic social network graphs. In particular, we developed a novel spectral graph theory methodology[1] for identifying anomalous behaviors by assessing the dynamics patterns of mitochondria in fluorescent microscopy videos after exposure to morphology-altering chemicals.

Upon perturbation, mitochondria generally demonstrate one of two morphological responses: fission or fusion. Fission is the process of mitochondrial clusters fragmenting, or breaking apart into smaller structures, with the intention of removing damaged mitochondria [51][13]. Conversely, when mitochon-

[1] **M. Hill**, M. Fazli, R. Mattson, M. Zain, A. Durden, A. T. Loy, B. Reaves, A. Courtney, F. D. Quinn, S. C. Chennubhotla, and S. P. Quinn, "Spectral Analysis of Mitochondrial Dynamics: A Graph-Theoretic Approach to Understanding Subcellular Pathology," in Proceedings of the 19th Python in Science Conference, 2020.

dria are stressed they undergo fusion, which is the process of multiple structures joining together to form a single entity to increase connectivity and improve the function of the organelle's network [51][106]. A balance between fission and fusion events are vital to maintaining a healthy cell [94]. As a result of fission and fusion sequences, the morphology of mitochondria transitions between many states along its spectrum, and this series of transitions is the phenomenon referred to as mitochondrial dynamics [32]. Mitochondrial dynamics were the focus of our research because various diseases intentionally seek to disrupt the fission-fusion balance to create more hospitable environments for their pathogens to survive and replicate [80]. Being able to identify unhealthy dynamics could prove vital for many tasks, including early detection of life-threatening infections and developing therapies and treatments. The traditional method for identifying perturbed mitochondrial structures is via manual inspection of microscopy imagery by specialists, which is labor-intensive, error-prone and scaled poorly to handle large volumes of data [32]. Thus, creating a demand for automated approaches for modeling and analyzing mitochondrial dynamics [20].

The major benefits provided by our automated mitochondrial dynamics analysis methodology are unsupervised, quantitative means to identify when significant morphological events occurred in a microscopy video and which subcellular structures were affected. To provide such benefits, we framed the goals of our approach as temporal and spatial anomaly detection tasks. We considered these efforts as anomaly detection tasks because the significant fragmentation and fusion experienced by mitochondria after exposure to external stimuli are extreme deviations from the morphological patterns observed in healthy, undisturbed cellular environments and are a rarity because of the low frequency of which these patterns are observed. We developed these techniques by analyzing the dynamics patterns of three different cell groups: a group that was exposed to any external stimuli, or our control group; a group that was exposed to the pore-forming toxin Listeriolysin O to induce fragmentation [2], which is referred to as our LLO group; and a group that was exposed to Mitochondrial division inhibitor-1 to induce fusion [32], that we referred to as our Mdivi group.

4.2 Background

Anomaly detection encompasses a broad range of techniques that have and have various applications to several different fields. [49] offers the following definition for an anomaly: An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs. Anomaly detection is a critical task that indicates abnormal running conditions and performance degradation [49]. In the computational biology field, outlier detection is used to detect abnormalities in cell and bodily functions. Automating the task of such detections will reduce manual effort and speed up the process of understanding effects of pathogens on the body and aid in the development of drug therapies. Several research and surveys have been conducted on anomaly detection techniques. Broadly, the available methods can be categorized into three different groups. Unsupervised clustering determines anomalies with no prior knowledge of the data and is processed as a static distribution flagging anomalies by using remote points. Classification requires pre-labelled data and models both normal and abnormal data points. The method is most often used for static data that has set labels for anomalous data. Lastly, semi-supervised learning methods work by modeling only anomalous data or only normal data. The key idea behind this task is to teach the algorithm normal class points and have it recognize abnormalities. This is a semi-supervised technique as pre-labelled data is required for either the anomalous data or the normal data [49].

In our problem context, we are approaching anomaly detection through a graph based approach. Graphs are extremely versatile as they can model static and dynamic data. Particularly, modeling dynamic graphs is very useful in detecting anomalies but unfortunately research in outlier detection on graph based data is lacking.

4.3 Methods: Temporal and Spatial Anomaly Detection using Spectral Graph Theoretics

4.3.1 Spectral Decomposition

We performed spectral decomposition on special matrix representations of the state graphs contained within dynamic social networks to highlight the changes in connectedness of the mitochondrial networks over time. Understanding the connectedness of mitochondrial networks is essential for analyzing the organelle's dynamics because sudden, drastic changes regarding the connectivity of our dynamic graphs are good indicators of disruptions in mitochondrial dynamics due to the established relationship between the topology of the state graphs and the morphology of the organelle.

The first step in analyzing the connectedness of each state graph is to compute a special representation of them, known as graph Laplacians. The initial representations of our state graphs are the data structures known as adjacency matrices. An adjacency matrix of a graph is a $|V|x|V|$ matrix A that indicates whether a connection exists between every pair of vertices i and j in the graph by making the value of the corresponding matrix elements equal to non-zero values, where V denotes the set of vertices. For undirected graphs the adjacency matrix is symmetric (i.e., A_{ij} and A_{ji} are equal); this property is not guaranteed for directed graphs because they can contain asymmetric connections. For weighted graphs, the non-zero value used to represent the connection is the edge weight that defines the connection between the two vertices. Since the state graphs in dynamic social networks are weighted, undirected, and fully-connected their adjacency matrices are symmetric matrices where each element contains the similarity score between the related mixture component pairs. The adjacency matrices can be utilized to derive other matrices that are also necessary for computing graph Laplacians, known as the degree matrices. A degree matrix is a $|V|x|V|$ diagonal matrix D such that the values of the diagonal are the number of edges adjacent to each respective vertex, or the sum of each corresponding row in the adjacency matrix. We can leverage the adjacency and degree matrices of each state graph to compute the graph Laplacians through a straight-

forward matrix subtraction operation defined in Eq. 4.3, where A is the adjacency matrix and D is the degree matrix of the graph. In our anomaly detection experiments, we utilize the normalized version of the graph Laplacians (Eq. 4.2) because they ensure that the values of the Laplacian matrix fairly represent both the number of connections of each node as well the size of the weights of the connections by, essentially, dividing the unnormalized graph Laplacians by the degree matrix. Graph Laplacians, like adjacency and degree matrices, contain connectivity information regarding graphs, and they are useful because they are considered positive-semi definite matrices, meaning the scalar value that results from multiplying any real-valued vector and its transpose by the symmetric matrix will always be a non-negative value. That property is significant because it ensures that when we factorize our graph Laplacian matrices during the spectral decomposition process one of the factors, a vector of scalar values, will only contain non-negative values.

$$L = D - 1 \tag{4.1}$$

$$L = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \tag{4.2}$$

After computing the normalized graph Laplacian representations of every state graph, we factorized them into eigenvalue vectors and eigenvector matrices using the spectral graph theory technique spectral decomposition. Eigenvalues and eigenvectors are canonical forms of matrices (Eq. 4.3): eigenvectors are a finite set of vectors that when multiplied by a matrix will undergo a linear transformation that results in the vector being oriented in the same direction but scaled by a factor; eigenvalues are the scalar values.

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v} \tag{4.3}$$

In the context of spectral graph theory, eigenvalues, or the spectrum, of a graph are of particular interest because these values characterize various structural properties of the graphs, such as the number of connected components of the graph [22], which are commonly leveraged to partition the graphs into

meaningful sub-communities, or clusters [69]. In our context, we will leverage these succinct, quantitative representations of our graph states to highlight anomalous structural activity of mitochondrial networks.

4.3.2 Temporal Anomaly Detection

Knowledge of when morphological events occur is invaluable in the pursuit of understanding mitochondrial dynamics. This information is highly valued because possessing the ability to indicate precise moments in time when a morphology-altering event is occurring enables richer analysis of the behaviors of pathogens and other cellular invaders. Furthermore, automated event detection approaches can either completely remove the need for any manual inspection of microscopy imagery, or in more critical settings where human supervision is required, significantly reduce the amount of manual labor necessary because the quantity of videos that must be reviewed can be reduced to only the video frames that precede events through the frames that immediately succeed the events. Thus, we addressed the need for such an automated approach by developing a temporal anomaly detection methodology for identifying significant structural changes in organellar networks.

The core idea of our temporal anomaly detection approach is modeling the variation in connectivity of dynamic social networks as time-series sequences of their spectrums and identifying points in time that correspond to video frames where significant structural changes of the mitochondria's morphology is visible. Intuitively, the morphology of healthy mitochondria is relatively unchanging, so sudden or drastic changes in the connectivity of our graph is an indication that a significant event is occurring. Therefore, our goal was to develop an approach that could identify periods of relative stability from moments of instability, which can vary from brief events to prolonged periods. Additionally, our approach had to be robust because structural changes can vary depending on the type of perturbation; fission events can fragment mitochondrial structures to the point that only a small fraction of structures remain by the end of a video, which starkly contrasts a fusion event. Our solution to such requirements was to utilize a fundamental statistics concept that detects outliers in data, z-scores, to detect extreme deviations in our time-series sequences.

The first step in our anomaly detection process was constructing the time-series sequences of the dynamic spectrum. That process consisted of computing the eigenvalue vectors of each state graph, ordering the vectors according to their size, and aligning the vector elements across time according to their ordering. The number of eigenvalues, and by extension eigenvectors, varies by microscopy video and is dependent on the number of nodes of the dynamic graph. For example, a dynamic graph that contains 15 nodes will have the same number of eigenvectors, which can be represented as a single matrix, and also 15 scalar eigenvalues, which can be represented as a single vector. Next, we sorted the elements of each eigenvalue vector in descending order: largest eigenvalues are the first elements of the vectors and the smallest eigenvalues are the last elements in the vectors. Once the eigenvalue vectors of each state graph were sorted, we constructed |V| number of time-series sequences that model the changes in eigenvalue magnitudes according to their rank in sorted order. For example, one of the time-series sequences models the changes in the magnitude of the largest eigenvalue of each state graph across time, and another sequence models the changes in the second-largest eigenvalues over time; ultimately, each time-series sequence models the magnitudinal changes of a specific rank of eigenvalues.

Next, we analyzed the collection of time-series spectrum sequences for each dynamic social network graph, and highlighted points in time where the sequence significantly diverges from the local trend using z-score outlier detection. The connectivity information of each state graph is encoded across all of the eigenvalue scalar values, so to represent the connectivity information into a single descriptive value we performed a mean aggregation over each spectrum. The mean aggregation step allowed us to effectively reduce our multiple time-series sequences into a single time-series sequence for analysis. To detect timepoints of anomalous morphological behavior, we computed a moving average of the aggregated sequence by utilizing a sliding window to capture the local trends in connectivity, then identified timepoints where the sequence deviated more than a predetermined number of standard deviations away from the moving average. The intuition behind this approach relies on the statistical notion that data points that differ significantly, or exist a far distance away in terms of standard deviations (z-score), from the mean of normally distributed data could indicate a behavioral change in the phenomenon that produced the data. It

is possible that such extreme deviations from the mean could occur by chance, but is not likely. In context of our temporal anomaly detection problem, we assume that our aggregated spectrum values are normally distributed, and we believe that significant deviations from the local trends captured by the mean of the sliding window of aggregated time-series is a good indicator of timepoints where a significant change in the connectivity of the dynamic social network occurred, which biologically means a disruption in the dynamics of a mitochondrial network. In our experiments, we explored the effectiveness of this technique on dynamic social networks that modeled various mitochondrial perturbations.

4.3.3 Spatial Anomaly Detection

Identifying the location of structural changes in mitochondrial networks is just as invaluable as knowing when the changes occurred. The ability to pinpoint exact spatial locations of change is useful for performing qualitative assessments of microscopy imagery because only a subset of the image is highlighted, which can greatly reduce the quantity of data to be evaluated and allows for more concentrated analysis of meaningful regions. Such an ability is especially desired for assessing the behaviors of mitochondria because the organelle is a spatially diffuse structure, so it is challenging to visually track or even detect all significant structural changes. We responded to such a need by developing a spectral graph theory-based methodology for indicating regions of mitochondria demonstrating significant structural changes in fluorescent microscopy videos.

The core idea of our spatial anomaly detection task is identifying which mitochondrial clusters demonstrated the most variability in their relationships with other clusters. For larger mitochondrial networks, naively comparing and contrasting all cluster relationships, which are modeled through the edges in the dynamic social network graphs, could become computationally expensive. As a result, we sought a more efficient approach: we leveraged the connectivity information contained within the eigenvector matrices of the state graphs. Each row in these eigenvector matrices correspond to a specific node in the dynamic graphs and, essentially, serve as fixed-sized vector representations, or embeddings, of the vertices in a latent space. The embedding vectors are not to be confused with the eigenvectors: each embedding vector is

composed of elements from each eigenvector, which are typically represented as columns in an eigenvector matrix. The node embeddings are significant because they are meaningful vector representations of the mitochondrial clusters that encode all cluster relationships.

We were able to identify the mitochondrial clusters that underwent the significant structural changes by assessing the variability of each node embedding temporally. We computed the variability (Eq. 4.4) of each node's relationships by calculating the euclidean distance between their embedding vectors across time. Intuitively, this approach identifies the mitochondrial clusters whose relationships changed the most over the duration of the microscopy video. Once the mitochondrial clusters whose structural relationships changed the most were identified, we highlighted their spatial regions in the original image domain using a bounding box, as demonstrated in figure 4.1. Ultimately, we deem this methodology a spatial anomaly detection approach because when it is leveraged in conjunction with our temporal anomaly detection techniques we were able to highlight the mitochondrial regions that demonstrated the most significant structural changes at the times when morphology-altering events were occurring.

$$\sum_{t=0}^{n-1} d(v_t, v_{t+1}) \tag{4.4}$$

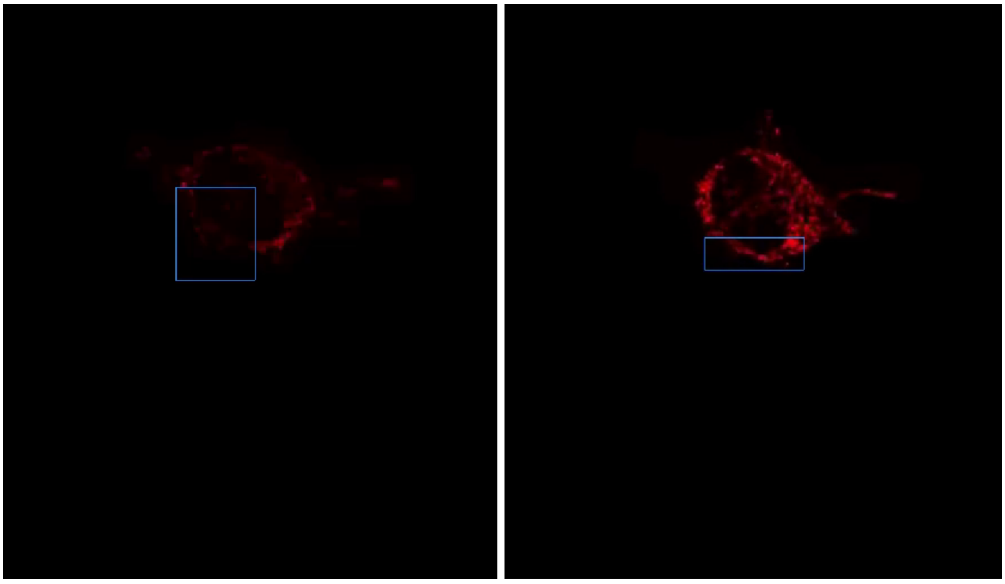

Figure 4.1: Bounding box visualizations of a mitochondrial (Mdivi) structure on the first and last frame.

4.4 Experiments

We performed qualitative assessments to evaluate the efficacy of our anomaly detection methodologies. The goals of our assessments were to determine whether our measurements were truly indicators of anomalous organellar activity, to assess the effectiveness of this approach at detecting different types of morphological events, and to learn the effects of the various hyper-parameters on our approach to provide general heuristics for usage. Our assessment of our temporal anomaly detection methodology consisted of manually inspecting the microscopy video frames that were indicated as anomalous to see if structural changes occurred by our approach, while our spatial anomaly detection focused on seeing if the regions highlighted demonstrated structural changes.

4.4.1 Temporal Anomaly Detection Assessment

First, to ensure that our approach was accurately capturing the mitochondrial dynamics trends, we visualized the spectrum values of each state graph alongside a plot of the corresponding z-score signals (figure 4.2). We evaluated the plots to see whether the z-score signal plots were responding accordingly to changes in the spectrum values; in essence, we were gauging whether timepoints where the spectrum values sharply increased or decreased corresponded to z-scores with greater values, regardless of the sign of the z-score. We generated several of those plots for each microscopy video with different hyper-parameter values to measure the effects of the hyper-parameters on the z-score signal plots. Our temporal anomaly detection methodology has two hyper-parameters that a user must specify a value for: the window size, which determines the number previous time points in the sequence to compute the mean of; and the threshold size, which is the cutoff value to declare time points as anomalous if the z-score exceeds the value in either the positive or negative directions. We experimented with integer window sizes ranging from 2 through 50, and floating-point threshold values ranging from 2.0 through 3.0.

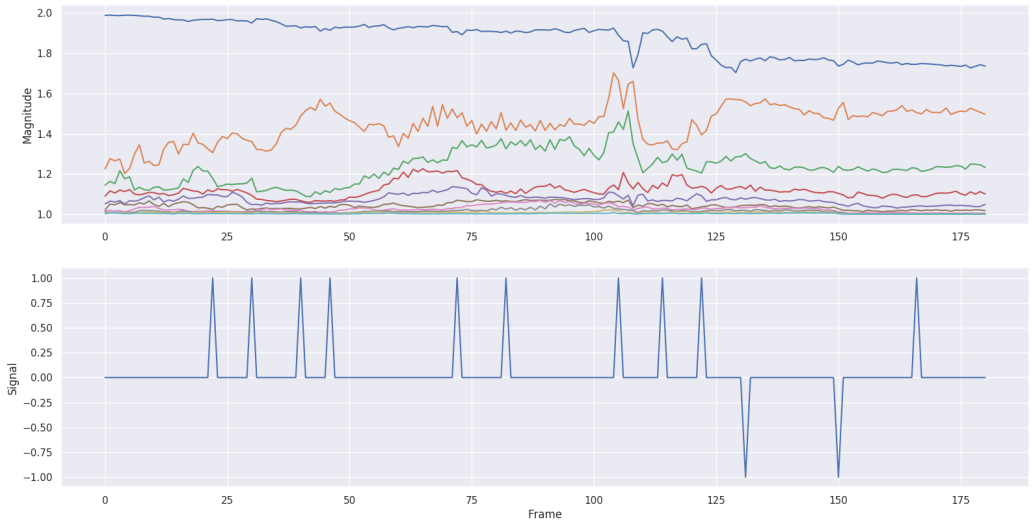

Figure 4.2: Graph Spectrum Plots

After ensuring the detection signals aligned with the spectrum time-series sequences, we assessed whether the morphology of the mitochondria demonstrated any significant changes in the frames corresponding to the timepoints declared anomalous. Specifically, we evaluated LLO videos to see whether significant fragmentation occurred, Mdivi for fusion, and in the control videos we did not expect to see much, if any, activity declared anomalous due to those cells not being exposed to any external stimuli.

4.4.2 Spatial Anomaly Detection Assessment

The primary goals of our spatial anomaly detection assessments were to ensure that mitochondrial structures were being tracked properly and that the spatial regions highlighted captured structures undergoing significant change. Although this assessment is similar to that which we performed to evaluate our temporal anomaly detection methodology, in the sense that we inspected the microscopy videos in our corpus, it is much more concentrated: the focus of these assessments were to identify anomalous morphological behavior on specific regions of the imagery, rather than whether anomalous behavior occurred at all.

Our initial evaluations focused on the sizes of the bounding boxes to ensure that structures were being fully highlighted. We identified the mitochondrial clusters that demonstrated the most significant changes in terms of connectivity with the other mitochondrial clusters and drew bounding boxes around the spatial regions that represented those structures in the microscopy videos. The size and coordinates of the bounding boxes were derived from properties of the gaussian mixture components that corresponded to the mitochondrial structures. We utilized the means of the gaussian components to serve as the center coordinates of the bounding boxes. The size of the bounding boxes were left as a hyper-parameter to be provided by the user, and was a major point of our evaluations. The size of the bounding boxes required assessment because a benefit of spatial anomaly detection is being able to dramatically reduce the search space, so boxes too large still leaves a large area for manual inspection.

Once we assessed whether reasonable sizes were being utilized for the bounding boxes, we focused our evaluations on the tracking ability of this approach. Our assessments needed to ensure that each box corresponded to the same structures over the duration of the video, especially since our underlying mechanism for detecting mitochondrial structures relied upon a probabilistic model. Thus, we visually tracked the trajectories of the bounding boxes to see if any identity swapping was occurring. Finally, we evaluated the content of the bounding boxes for mitochondrial clusters undergoing significant morphological changes. Due to the expansive, diffuse structure of mitochondria, it is likely that morphological events are occurring in multiple regions of a microscopy video scene. Additionally, even healthy mitochondria must undergo balanced sequences of fission and fusion events to maintain a healthy cellular environment. Therefore, many, if not most, mitochondrial clusters, depending on cell type, will demonstrate a degree of morphological activity, but our approach must be able to identify structures that experienced significant structural change. We qualitatively defined significant structural change as seeing substantial amounts of fragmentation and fusion: in terms of fragmentation we must see structures reduced to smaller components or even portions of it are no longer visible in the imagery; conversely, in terms of fusion, we must see many smaller structures fusing together to form few larger structures. Visually, significant fragmentation events typically start with structures emitting brighter hues that darken as the event transpires, while

significant fusion events demonstrate the opposite. Ultimately, we perform such evaluations for each microscopy across all different cell types in our videos to understand the effectiveness of our approach.

4.5 Results and Discussion

After rigorous experimentation, we concluded that both our temporal and spatial anomaly detection methodologies are effective at performing their intended functions, but we recognize that the extent at which they are effective requires much thought and experimentation with hyper-parameter values. The effectiveness of our approaches, especially temporal anomaly detection, is context dependent, meaning parameters values that work for one scenario may not be directly applicable in other cases; however, we believe general behaviors and trends that we noticed will still occur during the assessment of other fluorescent microscopy imagery. Thus, we seek to provide heuristics for utilizing these methodologies to achieve meaningful results.

4.5.1 Temporal Anomaly Detection Results

The two hyper-parameters that required tuning in our temporal anomaly detection experiments were the window size and the anomaly, or outlier, detection threshold value. Our experiments indicated that the window size had a more dramatic effect on the performance of the detector. Specifically, window sizes too small (figure 4.3) cause the detector to declare an excessive number of timepoints as anomalous, while window sizes too large (figure 4.4) cause the detector to rarely declare timepoints as anomalous. The implications of these two extremes is that for too small window sizes nearly every frame will be declared anomalous, which leaves a large search space to filter through for actual points of time where significant morphological activity is occurring; while windows sizes that are too large pose the risk of missing most morphological events because any events that occur during the initial learning period, or the first frames of the video that are less than the window size, will be overlooked and because of the large number of timepoints that are aggregated in the windows the impact of local disturbances to the dynamics will also

likely be ignored. For our videos, we found that a window size of 20 worked best; a reasonable number of timepoints were declared anomalous per video.

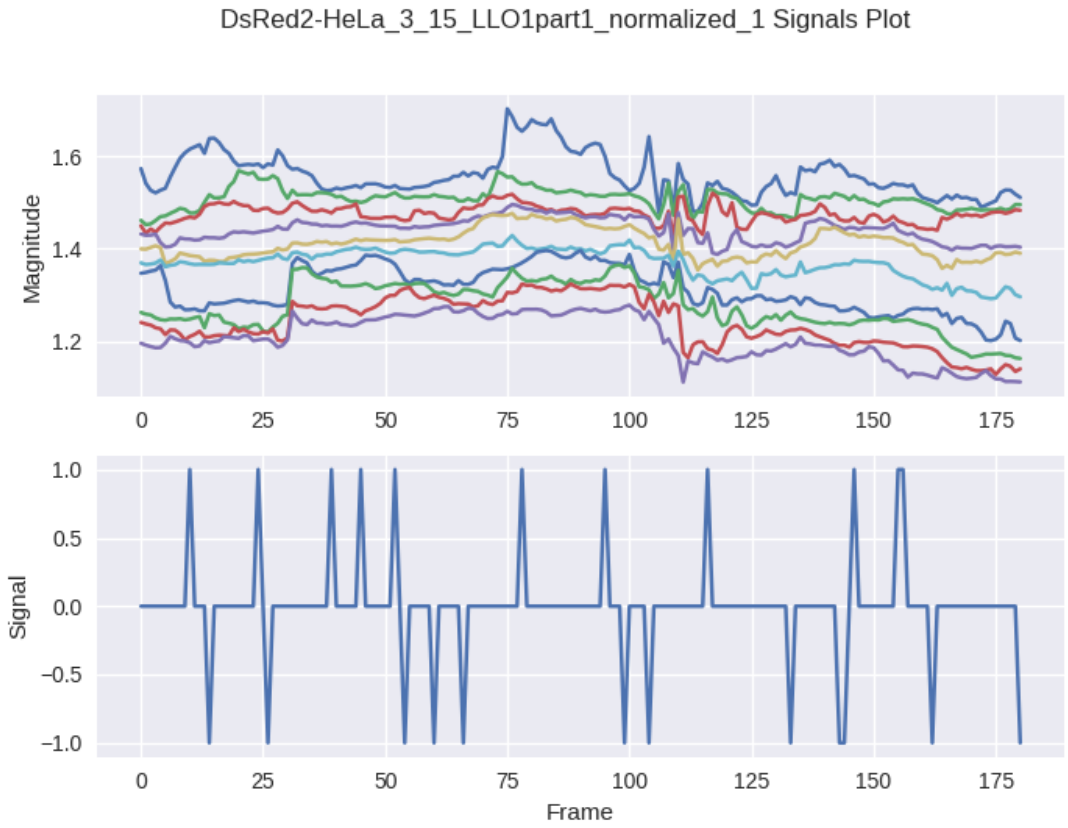

Figure 4.3: Effects of a small window size (window=2).

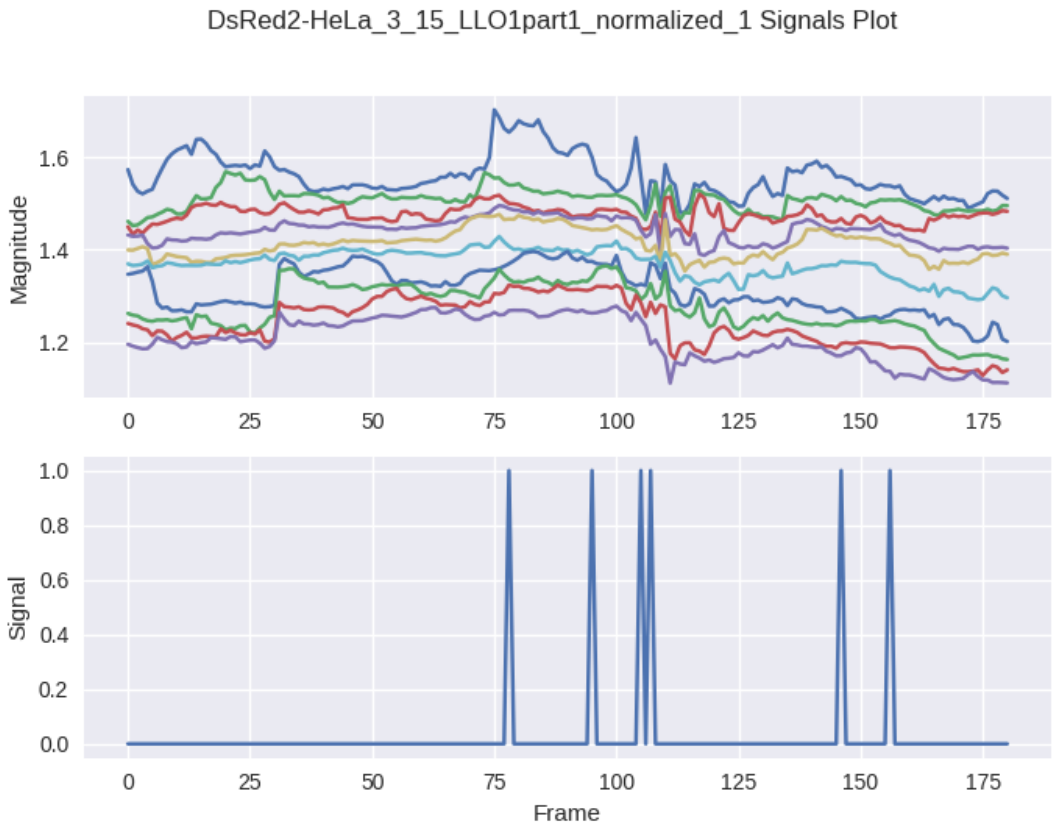

Figure 4.4: Effects of a large window size (window=50).

Although less extreme in terms of its impact on the ability of the temporal anomaly detectors, the threshold value still retained a certain effect on the detector. Intuitively, increasing the threshold values decreases the number of timepoints that will be declared as anomalous because the aggregated spectrum values must be a more extreme distance away from the means of the sliding window, while decreasing the threshold value increases the number of points that will be declared as anomalous because their aggregated spectrum values do not have to be as far away from the mean of the sliding window to be indicated as anomalous. We experimented with the range for the detector being between 2.0 and 3.0, because instances of normally distributed data are typically no more than 2 standard deviations away from the mean according to the empirical rule, and found values closer to 2.0 to work best for our context.

Threshold values closer to 3.0 cause the detector to rarely declare any timepoints as anomalous (figure 4.5), while 2.0 declared a reasonable amount (figure 4.6). Threshold values around 2.5 also served well (figure 4.7); however, they would at times miss smaller, but still significant, trends in the spectrum sequences.

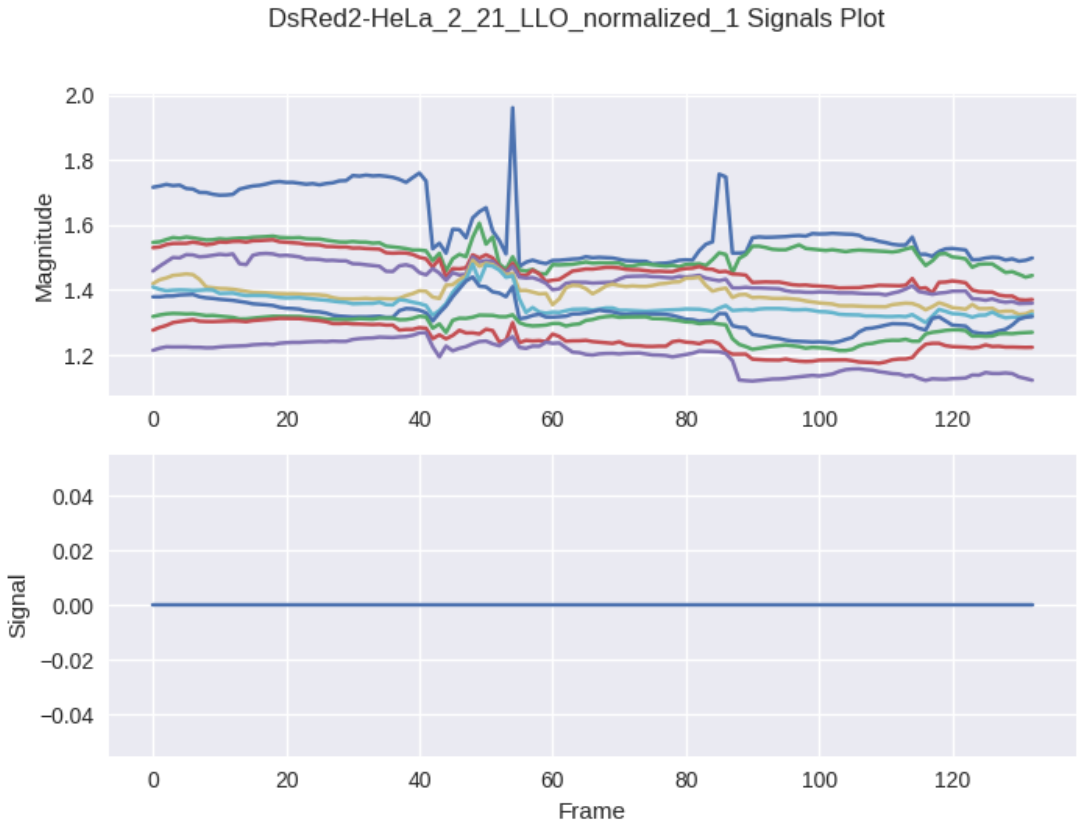

Figure 4.5: Large outlier threshold value (threshold=3).

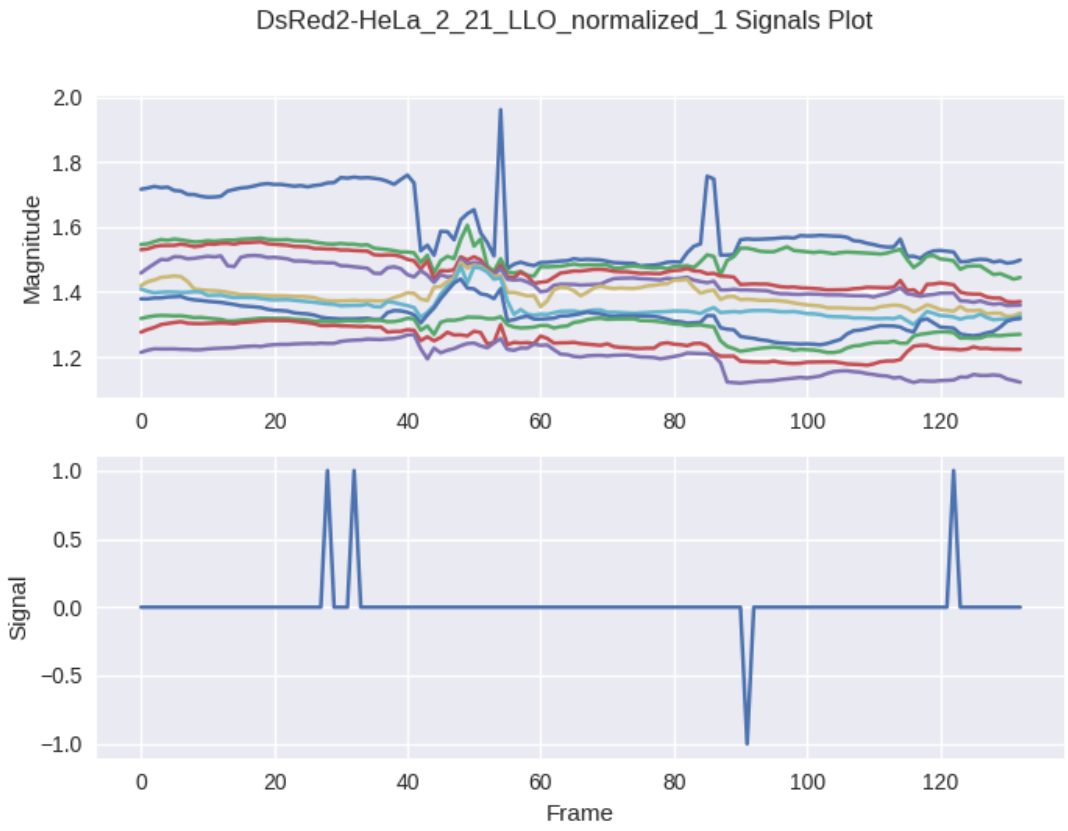

Figure 4.6: Small outlier threshold value (threshold=2).

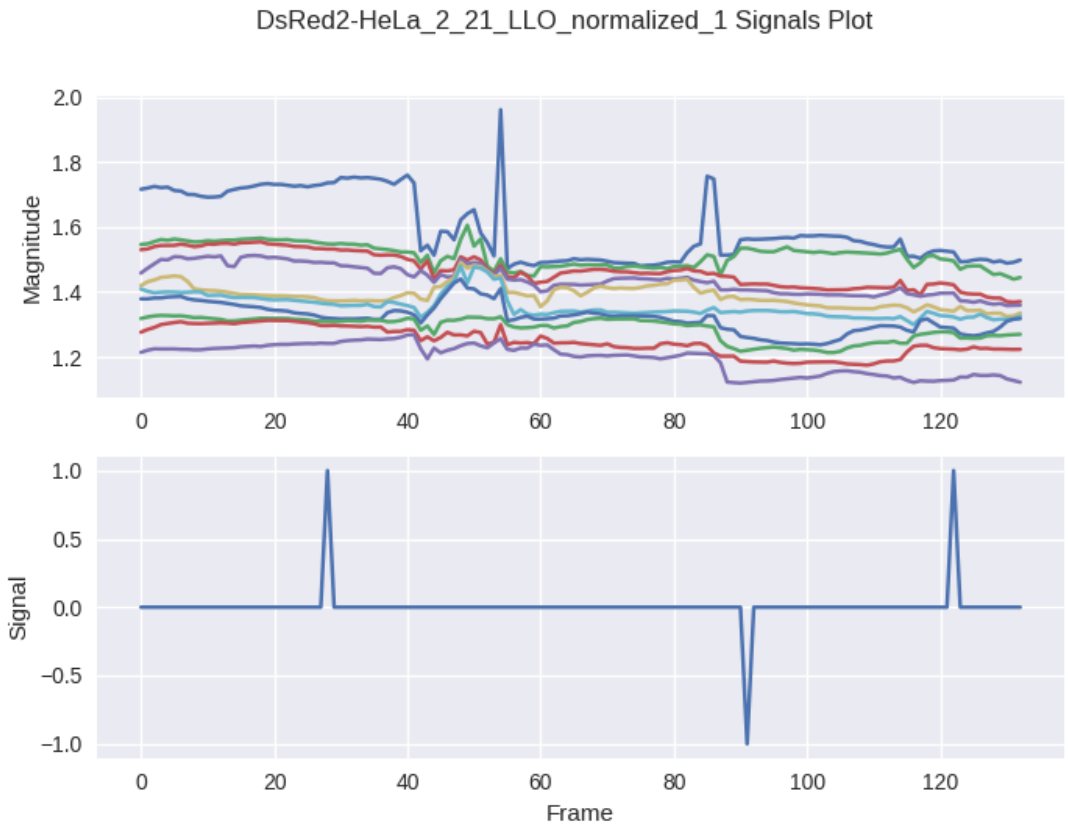

Figure 4.7: Medium outlier threshold value (threshold=2.5).

Upon inspecting the microscopy imagery that corresponded with anomalous time points, we concluded that significant morphological changes were visible when utilizing properly tuned parameters for the detectors. A benefit of this approach is that the timepoints declared anomalous typically were not multiple timepoints corresponding to the same local events. Thus, when we inspected the imagery we noticed significant morphological changes of the mitochondria between each anomalous timepoint (figure 4.8). However, for poorly tuned detectors we recognized that frames corresponding to relatively close anomalous time points would not demonstrate significant morphological differences. Thus, with properly tuned hyper-parameters our temporal anomaly detection methodology can be a useful approach to supplement mitochondrial analysis approaches.

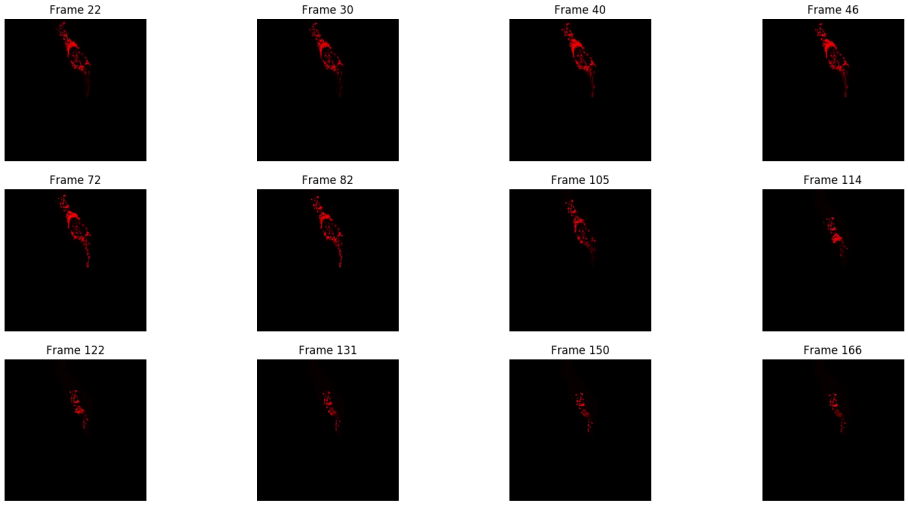

Figure 4.8: Frames that correspond to timepoints declared anomalous.

4.5.2 Spatial Anomaly Detection Results

The only hyper-parameter that we experimented with in terms of spatial anomaly detection was the size of the bounding boxes that highlighted the spatial regions in microscopy regions that corresponded to the most significant structural changes of mitochondria. We allowed the users to specify the number of standard deviations away from the mean in each direction to define the boundaries of the boxes. We found that 3 standard deviations away from the mean were sufficient for capturing the entirety of mitochondrial clusters. Intuitively, this parameter value was decided in accordance to the property that nearly all normally distributed data lie within 3 standard deviations of the mean. Our major concern was producing bounding boxes that were excessively large, because such boxes would still leave a large search space requiring a large amount of manual labor to thoroughly inspect. However, this proved not to be an issue.

Additionally, a useful property that we noticed during our qualitative assessments was that the bounding boxes were responsive to morphological changes of the mitochondria. In our assessments, we manu-

ally identified structures in the videos and visually tracked the bounding boxes to ensure that each box remained with the same cluster that it initially highlighted, rather than switching to highlight another mitochondrial cluster. The results of such assessments were that the bounding boxes not only properly tracked the structures but the size of the boxes also adapted to the morphological events.

Unlike the hyper-parameters for our temporal anomaly detection approach, we are more comfortable suggesting the parameter values we used in our experiments as a good starting point for other contexts. The reasoning behind such a suggestion is that the best parameter value that worked for our data was derived from statistical properties of the underlying probability distributions that modeled the organellar structures in our dynamic social network graphs; therefore, we anticipate similar results in other contexts. There is, however, a limitation to the spatial anomaly detection methodology: bounding boxes can overlap. Due to the spatial location and size of the bounding boxes being derived from properties of our mixture components, they also can experience overlap just like probability distributions do. As a result, highlighting multiple regions of mitochondrial clusters at one time using these bounding boxes can clutter regions and make it difficult for users to assess the structural changes occurring in that region. As a result, in our analysis we only highlighted non-overlapping bounding boxes, typically in a ranked manner according to the amount of variance displayed by their eigenvectors over time. This enabled for clear identification of regions demonstrating significant morphological change.

4.6 Conclusion

The morphology of mitochondria is perturbed in distinct ways by the presence of bacterial or viral infections in the cell, and modeling these structural changes can aid in understanding both the infection strategies of the pathogen, and cellular response. Modeling mitochondria poses many challenges because it is an amorphous, diffuse subcellular structure. Yet, dynamic social networks are well-suited for the task because they are capable of representing the global structure of mitochondria by flexibly modeling the many local clusters present in the cell. This extensible modeling approach enables the spatiotemporal relationships of the mitochondrial clusters to be explored using theoretic graph techniques. We proposed

quantitative spatial and temporal anomaly detection methodologies that could be utilized in conjunction with traditional qualitative metrics to elucidate mitochondrial dynamics. We ultimately hope to use these spectral analytics and the OrNet software package to conduct large-scale genomic screens of Mycobacterium tuberculosis mutants, in an effort to build a deeper understanding of how the pathogen invades cells and induces cell death at the genetic level. This work is one of the first steps toward that ultimate goal.

Chapter 5

Evolving Dynamic Social Networks with Density-Based Object Tracking

5.1 Introduction

Modeling mitochondria using dynamic social network graphs proved to be effective as we were able to produce novel methods for analyzing the dynamics of the organelle by assessing its motility patterns using temporal and spatial anomaly detection techniques. However, a limitation of this approach is that all mitochondrial clusters identified on the first frame of a video are assumed to persist for the entire duration of the video. Such a modeling approach does not account for mitochondrial clusters that cease to exist due to a fission event or the emergence of clusters due to the fusion of sparse, previously undetected mitochondria.

Thus, we implemented a significant restructuring of the object tracking process of our graph construction methodology that enables vertices the ability to dynamically enter or exit the graph to better represent the morphology of mitochondria.

We accomplished the restructuring of our dynamic social networks by decoupling the object detection task from the object tracking task. Specifically, we utilized the density-based clustering methodology established by the Hierarchical DBSCAN (HDBSCAN) algorithm [17] to identify the spatial locations of mitochondrial clusters on each frame of a microscopy video before utilizing a combinatorial optimization technique, known as the Kuhn-Munkres algorithm (also commonly referred to as the Hungarian algorithm) [56], to match the identity of the clusters between frames.

5.2 Background

5.2.1 Object Detection

Object detection is a computer vision technique that identifies objects in an image and indicates their spatial locations, typically by drawing bounding boxes around the object. This is different from image recognition as object detection predicts the location of the object and then assigns the label. This started out as a supervised learning problem but in recent years, large architectures have been able to learn from large datasets and evaluate several benchmarks.

Object detection is one of the most challenging problems in computer vision and researchers have developed several novel methods to address different problems in different fields. The boom in development of deep learning for image classification gave rise to several unique object detection techniques. Object detection algorithms have been popular since the early 2000's but these models were built as an ensemble of hand-crafted feature extractors and as a result were slow, had average metrics and performed poorly on unseen data. Object detection has constantly remained to be a prominent topic that has seen the most growth out of any field in computer vision in the last two decades. The reintroduction of CNN's and creation of similar architectures furthered the research of computer vision. Generally, object detection techniques can be classified into two different groups: Traditional methods and Deep learning methods.

Most early object detection methods were based on handcrafted features. Despite the lack of effective image representation and limited computing resources, a few notable methods emerged in the early 2000's

that began the object detection boom. The most prominent models of the time were the Viola-Jones [99] and HOG Detectors [23]. Ground-breaking for its time, the Viola-Jones Detector was one of the earliest models capable of detecting human faces in real-time. The detector used sliding windows, a method in which all possible locations and scales are scanned in an image to see if any window contains a human face. The Viola-Jones Detector provided three novel ideas: integral image, feature selection and detection cascades. The integral image is an idea to speed up the convolution process by making each window independent of its window size. computational method to speed up box filtering or convolution process. Unlike previous methods that used Haar basic filters, Viola-Jones Detector used Adaboost to select a small set of features from a large set of random pools. Detection cascades are a multi stage detection paradigm designed to reduce computational overhead. Histogram of Oriented Gradients (HOG) was another feature descriptor approach that was designed to be computed on a dense grid of uniformly spaced cells and use overlapping local contrast normalization to improve accuracy. The main idea of the HOG detector was to identify the location and orientation of edges in local image regions, or the cells, and construct a histogram of the orientation counts.

After traditional methods became more saturated, development in object detection methods plateaued until the reintroduction of Convolutional Neural Networks (CNNs) in 2014. The object detection architectures took off at unprecedented speed and branched out into two different groups: CNN based two-stage detectors and CNN based one-stage detector [61]. The main difference is One-stage detectors have high inference speeds and two-stage detectors have high localization and recognition accuracy [61]. Convolutional Neural Networks were able to learn high level feature representations of an image, thus restarting a new boom in the creation of object detection methodologies. However, early CNN-based detection approaches struggled to detect multiple objects of various quantities, shapes and sizes. Girshick et al. [41] proposed a method, R-CNN, to extract a set of object proposals by selective search. Then each proposal image was rescaled to a fixed size and fed into a CNN model that had been trained on ImageNet to extract features. Then a basic machine learning classifier such as SVM could be used to recognize object categories. Despite being novel and advanced, R-CNN did suffer a few drawbacks including slow training

time and having a fixed selective search algorithm. To combat the speed disadvantages of R-CNN, Fast R-CNN [40] was developed as an improvement that allowed users to simultaneously train a detector and a bounding box regressor under the same network configurations. This method increased detection, making it 200 times faster than R-CNN. This speed is a result of feeding one region proposal to a CNN every time. But even this model still is limited by the proposed image detection speed. These object detection algorithms all use regions to localize objects within the image and are two-stage architectures. Alternatively, one stage detectors predict bounding boxes in a single-step without using region proposals. YOLO (You Only Look Once) [83] was the first one-stage detector system in the deep learning era. In YOLO, a single convolution network is applied to the full image and it finds bounding boxes and the class probabilities for them. This model was largely successful and the author of YOLO proposed two other versions which further increased the detection accuracy while simultaneously maintaining the high detection speed. YOLO however does trade off the high detection speed for a drop in localization accuracy compared to two-stage detectors.

5.2.2 Clustering

Clustering is a data mining technique used to group similar objects to form clusters [71]. It is a machine learning technique often used for pattern recognition, grouping data based on a particular criteria, and classification to name a few. [71] surveys a few clustering techniques including Partitioning technique, Hierarchical technique, Grid-Based technique, Density –Based technique, Divide and Conquer, Random Sampling. The most common application of Partition clustering is kNN. This method is a clustering technique in which n objects are partitioned into k clusters of similar objects but such that the objects in each cluster are different. Two conditions apply to this method: 1. Each cluster must contain at least one object and 2. Every object must be associated with at least one cluster.

Hierarchical clustering is partition clustering that occurs in either a top down or bottom up approach. In a top-down approach, all data points start in one cluster and splits are applied as one moves down the hierarchy. In a bottom-up approach, every data point begins in its own cluster and they are recursively

combined as one moves up the hierarchy. However there are a few drawbacks to this method, the main issue being unable to preserve quality criteria [71][81]. If clusters are not split or merged correctly in a single event, it may result in clusters that do not capture the data properly. A few algorithms have tried to combat the problem by using methods, such as picking non spherical clusters that vary in size, using a tree structure of cluster objects, and using the links of different points rather than the distance between them [71][81].

Grid based clustering, like the name suggests, uses a grid data structure and converts the object space into a finite number of cells to form a grid structure. This technique has a fast processing time, independent of the number of data points, but dependent on only the number of cells in each dimension in the quantized space. Because it is query independent, it can easily parallelize resulting in its fast computation time [71][54]. STING is an example of a grid based approach that divides spatial area into rectangular cells. It follows a top down approach to answer spatial queries and removing irrelevant cells [71][54]. A disadvantage to this technique is its lack of diagonal boundary (both vertical and horizontal boundaries are set so the cluster does not get too large) which may affect the accuracy. Other grid-based clustering algorithms include OptiGrid (Optimal Grid-Clustering), which tackles the "curse of dimensionality" problem by separating the clusters by hyperplanes proving to be a convenient method for high-dimensional data, and GridClus (Grid Clustering), which uses a multidimensional grid data structure to organize the space surrounding the patterns rather than organizing the patterns themselves [71][54][88]. This is advantageous to the algorithms which use dissimilarity criteria as this can deal with very large amounts of data by creating a grid structure to maintain the distribution [71][54][88].

Density based clustering is a clustering approach that forms clusters of densely gathered objects separated by sparse regions of low density data objects. These clusters can be of arbitrary shape and allow the method to handle noise by easily filtering out noisy data points [71][54][10]. DBSCAN, OPTICS and DENCLUE are the most widely used density-based clustering algorithms. DBSCAN is an algorithm based on user-defined parameters and proposes the density-connectivity between two objects and defines a cluster as a maximum set of density-connected objects. It assumes that clusters are dense regions and

are separated by regions of low density [71][54]. It can cluster large spatial datasets by calculating the local density of the data points. The main advantage to this method is it is robust to outliers and is not limited to creating a limited number of clusters specified by the user. DBSCAN requires two parameters: epsilon, the radius of the circle to create around each datapoint and pinpoints, minimum number of data points required. The number of clusters is not required as it produces the clusters on a density basis [10]. DBSCAN however has a disadvantage of limited parameter discovery and cannot always obtain optimal clusters. OPTICS fixes this problem of parameter discovery by extending on DBSCAN. It is nearly identical to DBSCAN but addresses parameter discovery by ordering the datapoint linearly so that the closest points become neighbors in the ordering. By generating an ordering of data points representing density-based clustering, OPTICS can explore large parameters and give good results [71][54][10]. DENCLUE, another popular density-based clustering algorithm forms clusters based on kernel density estimation. DENCLUE builds on Schnell's algorithm which defines clusters by local maxima of the density estimate. Data points are assigned to clusters by hill climbing using Gaussian kernels. Points assigned to the same maximum are grouped into a single cluster. The main idea behind DENCLUE clustering comes from the influence function which describes the impact of a data point on its neighborhood. The overall density function is calculated as the sum of the influence of all the data points [46].

5.3 Methods: Dynamic Node Graphs

5.3.1 Object Detection

Mitochondrial structures do not adhere to any predetermined shape and can appear in a variety of formations, so we utilized a density-based clustering algorithm to detect the subcellular structures because of their ability to find clusters of arbitrary shapes within data by indicating regions of higher density relative to the remainder of the data space [10]. In particular, we utilized the HDBSCAN algorithm because of its ability to extract flat clusters from our data in a manner that does not require the user to decide the number of clusters in advance [17]. This property of the algorithm is useful because mitochondria can

undergo a series of morphological transformations that can cause the number of its structures to fluctuate over time. Ultimately, we reframed the object detection phase of our dynamic graph process as a pixel clustering problem.

The core of the HDBSCAN approach is performing single linkage-clustering, which is an agglomerative clustering technique where every instance starts in its own cluster, and subsequently, the closest pairs of clusters are merged together in a hierarchical manner until only one cluster remains. However, single-linkage clustering is sensitive to noise, so the first step in the clustering process is transforming the data in a way that highlights regions where many instances are close, or dense, from regions where instances are distant from each other, or sparse. For this to be accomplished, a new symmetric distance metric is proposed that exaggerates the distance between sparse points and all other points, known as the $d_{mreach-k}$ (5.1). This metric relies upon a density estimate for each point, which is computed by finding the distance between a data point and its kth-nearest neighbor (Eq. 5.2). However, to find the nearest neighbors of a point, which in this context is an image pixel, we must first establish a metric, d(a, b), that can express distances between pixels. In image processing literature, there exist several distance metrics that incorporate both the spatial and intensity information of pixels, and we utilized the unnormalized city block metric (Eq. 5.3) due to its speed [39].

$$d_{\text{mreach-k}} = \max\{\text{core}_k(a), \text{core}_k(b), d(a, b)\} \tag{5.1}$$

$$\text{core}_k(a) = d(a, x_k), \text{where } x_k \text{ is the } k^{th} \text{ nearest neighbor of instance } a \tag{5.2}$$

$$\text{city block}(A_{ij}, A_{lm}) = (|i - l| + |j - m|) + |A_{ij} - A_{lm}| \tag{5.3}$$

The next significant goal of the algorithm is to create the clustering hierarchy. This is accomplished by (1) constructing a complete mutual reachability graph, where the vertices are the pixels and the edges between pixels are computed using the $d_{mreach-k}$ metric; (2) computing the minimum spanning tree

of that graph; (3) sorting the edges of the minimum spanning tree in increasing order to find the pairs of pixels to be merged to form clusters in the bottom-up manner of single-linkage clustering. This step highlights different regions of interest and noise in the image depending on the density level at which the clustering hierarchy is examined. Although clusters are identified at this step in the algorithm, a user would have to define a single fixed density threshold to extract a set of flat clusters.

Fortunately, the HDBSCAN approach has provided a mechanism for extracting flat clusters of varying densities from the clustering hierarchy based on the stability of each cluster. This is vital for modeling mitochondrial structures undergoing morphological changes because it is unlikely that every structure depicted in the microscopy imagery will consist of the same density of mitochondria. However, before flat clusters are extracted, a denoising operation is applied that simplifies the clustering hierarchy.

The denoising operation involves traversing the clustering hierarchy in a top-down manner to determine the levels where significant splits occur. For a split to be considered significant, the two resulting clusters must both be above a minimum cluster size, which is determined in advance by the user of the algorithm. If a group of instances fail to meet the minimum cluster size after a split, then that split is not considered to be significant, and the instances in that group are considered to be noise; if the size of the remaining group of instances is above the threshold, then it will retain its identity as a cluster. In context of the pixel clustering problem, performing this denoising operation highlights the prominent structural objects in the imagery by reducing the number of noisy clusters from the hierarchy.

The final step of this clustering algorithm is extracting a set of flat clusters, of varying densities, from the denoised clustering hierarchy according to their stability. The stability of a cluster is measured by the duration of its existence in the hierarchy before it is involved in a significant split. A new measure, λ (Eq. 5.4), is introduced to assess such durations. Each cluster is assigned two scores λ_{birth} and λ_{death} to indicate when the cluster was formed in the hierarchy and when it was split into smaller clusters, respectively. Additionally, each instance in a cluster has a score λ_p, that is defined as the λ-score where the instance splits off from the cluster and its value is between λ_{birth} and λ_{death}.

$$\lambda = 1/\text{distance}(\text{cluster}_x, \text{cluster}_y) \tag{5.4}$$

Quantitatively, the stability of a cluster (Eq. 5.5) is defined as the summation of every instance's λ-score minus the λ_{birth}-score of cluster. Once the stability score for every cluster in the denoised clustering hierarchy has been computed, the flat clusters to be extracted are identified by recursively traversing the hierarchy tree starting from the leaves and identifying clusters whose stability is greater than the sum of its children's stability. Ultimately, the flat clusters identified on each frame are mitochondrial structures of varying densities.

$$\text{stability} = \sum_{p \subset \text{cluster}} (\lambda_p - \lambda_{\text{birth}}) \tag{5.5}$$

5.3.2 Object Tracking

After performing object detection, each frame has a set of mitochondrial structures that are identified, and a core aspect of our modeling approach is elucidating the temporal relationships between these structures. Our goal is to capture the dynamics of organelles by highlighting the spatial and behavioral changes of their subcellular structures. By utilizing the Kuhn-Munkres algorithm to perform object tracking on the mitochondrial structures identified during the object detection phase, we were able to satisfy our goal.

The Kuhn-Munkres algorithm was designed to efficiently solve a fundamental combinatorial optimization problem, known as the assignment problem [56]. The objective of the assignment problem is to find the minimal cost solution of assigning agents to a set of tasks, where each agent has a cost associated with performing each task and cannot be assigned to more than one task. Prior works [8] [103] [76] [38] have demonstrated that this algorithm can be utilized for object tracking purposes.

The algorithm consists of representing all entities involved as a NxN cost matrix, where the rows are the agents, the columns are the tasks, and each cell in the matrix is the cost value between a specific agent and a task. Through a series of matrix operations, including row reduction, column reduction, scalar addition,

and scalar subtraction, the cost matrix is transformed into a new matrix where the cost of optimal matches between agents and tasks is zero. Thus, making it straight-forward to discover the optimal assignments. In our context, we reframed the assignment problem as an object tracking task with the goal of matching the identities of mitochondrial structures between frames. In doing so, we were able to indicate the spatial locations of the subcellular structures over time as the organelle responds to environmental changes. We adapted the algorithm for our purposes by recursively declaring the mitochondrial structures from a given frame t as the agents, the mitochondrial structures from frame t + 1 as the "tasks", and the cost values equal to the intersection-over-union (IOU) scores (Eq. 5.6) of the image contours of the mitochondrial structures between frames. Intuitively, our tracking approach relies on the notion that mitochondrial structures demonstrate subtle morphological and spatial changes between subsequent frames **Durden18**; therefore, each unique structure is expected to have the highest IOU score with itself in subsequent frames.

$$\text{IOU}(x, y) = \frac{\text{area}_x \cap \text{area}_y}{\text{area}_x \cup \text{area}_y} \tag{5.6}$$

5.3.3 Graph Construction

We constructed state graphs for a given frame t by defining the mitochondrial clusters identified by our object detection and tracking phases as the vertices, and by establishing unweighted edges between all pairs of vertices to create a fully-connected graph. However, the edges can become weighted by computing the euclidean distance between cluster centroids. We saved the inclusion of edge weights for future analysis works.

5.4 Experiments

The goal of restructuring the dynamic social networks was to create improved dynamic graphs that better reflected the morphological changes of mitochondria through more intuitive topological changes. We believe it to be far easier for novices and experts alike to identify the effects of fission and fusion events

on the morphology of mitochondria by assessing the persistence of nodes across time, rather than interpreting such effects through analysis of graph connectivity. Additionally, we believed that by modeling mitochondrial dynamics using this simpler, more intuitive method, that we could likewise use simpler analysis techniques to achieve equal, if not greater, performance with regards to anomaly detection. Thus, to fully understand the effectiveness of this approach, we experimented with various hyper-parameters to craft graph construction heuristics that can be followed to achieve biologically relevant graphs, and we analyzed the characteristics of these dynamic node graphs to evaluate whether or not they encoded enough mitochondrial dynamics information to detect morphologically significant events.

5.4.1 Hyper-parameters

Since our primary mechanism for identifying mitochondrial structures is a density-based clustering algorithm, we must define connections between the parameters of the algorithm and the biology of the mitochondria. Specifically, we sought to avoid constructing biologically inaccurate models that identify structures that are visibly distinct and separated by a significant distance as one mitochondrial cluster and models that disregard relevant structures as noise. The major hyper-parameters that we evaluated were a thresholding value to segment the background from the foreground, the minimum membership size for a grouping of data points to be identified as a valid cluster, and a thresholding value that enforces a strict minimum distance between clusters.

As a data preprocessing step, we performed background thresholding to reduce the effect of noise in our object detection phase. In this context, noise is defined as random variations of pixel intensities or artifacts that originate due to the environment in which the imagery was captured. This preprocessing step is crucial for two reasons: first, our mitochondrial modeling approach relies on being able to accurately detect mitochondrial structures in the imagery to form the dynamic graphs, so mistaking artifacts in the videos as mitochondria will severely impede the results of mitochondrial dynamics analyses; and second, large portions of microscopy imagery is background content, so by identifying those regions and excluding them during the clustering process we can drastically improve the speed of our detection phase. Thus, the

threshold value utilized must be selected with care because the thresholding step plays a significant role in this approach. We judged threshold values according to the amount of noise they reduced in the images.

The minimum cluster size parameter establishes the fewest number of data points that can constitute a valid cluster, and it is utilized during the flat cluster extraction step of the clustering process. Intuitively, this step is directly related to the number of clusters that will be returned from the clustering process; a smaller value for this parameter results in more clusters because the number of instances required to create legal clusters is low, while larger values restricts the number of possible clusters because only fraction of the potential clusters are eligible to be extracted. This parameter is significant in our organellar modeling context because it is one of the most straight-forward ways to manage the fidelity of our modeling approach. As a result, we experimented with this parameter to find values that ensured that distinct mitochondrial structures were modeled as their own entities, rather than being identified as components of a larger, biologically inaccurate cluster that combines spatially distant structures.

An additional clustering parameter that we experimented with to prevent constructing biologically inaccurate models is a distance threshold, epsilon [63], that explicitly enforces a minimum distance between clusters; clusters that fall below that threshold distance of each other are merged together into a single cluster. We saw utility in this parameter, especially in this context, because mitochondria has a diffuse subcellular structure, meaning not all of its components are contiguous, some are relatively distant from other mitochondrial structures. However, clustering models that are not properly configured for the specific context in which they are employed can easily assign a single label to many sub-structures that would be better modeled as distinct entities. Thus, we investigated whether utilization of epsilon could enable us to achieve dynamic graph models that are more reflective of the mitochondrial morphology depicted in the microscopy imagery.

5.4.2 Temporal Anomaly Detection

Dynamic node graphs present a simpler means for conducting temporal anomaly detection of mitochondrial dynamics than our original dynamic social network graphs. Our original temporal anomaly detection

methodology relied upon identifying timepoints that correspond to significant changes in the connectivity of the dynamic graphs, which was represented by the graph spectrum and eigenvectors that were computed by performing spectral decomposition. However, performing spectral decomposition can become computationally expensive for large graphs. Our proposed solution is to still utilize a z-score detection model to assess the presence of mitochondrial structures over time.

The underlying idea of our dynamic node graphs model is that nodes, like the mitochondrial structures they represent, should be allowed to dynamically enter or exit the graph to better represent the morphological changes of mitochondria; thus, by tracking the quantity of nodes present across time we are able to establish a time-series sequence for evaluation. In a similar manner to the temporal anomaly detection experiments that we conducted for our original dynamic social network graphs, we visually inspected the cluster-count sequences alongside the z-score outlier signal to see if dynamics trends were being captured, and we experimented with the window size and the outlier detection threshold parameters to see if the trends identified in the cluster-count sequences corresponded to morphologically significant events.

5.5 Results and Discussion

5.5.1 Hyper-parameters

Determining a suitable set of hyper-parameter values for this modeling approach was challenging because each parameter that we experimented with was sensitive and could drastically affect the clustering process. Therefore, we had to first understand the effects of each individual parameter in terms of producing clusters that were biologically accurate, before exploring the effects of the parameters when combined.

The background thresholding parameter was essential for accurate subcellular modeling because many factors from the environment in which the imagery was collected caused many artifacts that would be detected and mistaken for actual mitochondrial structures . The composition of the fluorescent microscopy videos we utilized was bright, red-emitting mitochondrial structures in the foreground against a solid dark

background. Naively, segmenting out only the background by utilizing all pixels whose intensity values were greater than '0', produced a high number of clusters, many of which corresponded to noise and artifacts (figure 5.1). Therefore, a more robust mechanism for thresholding had to be utilized. Through experimentation, we found an automatic thresholding technique proposed by Ridler and Calvard [85] that performs well at reducing the noise in the imagery. The technique finds a suitable threshold for segmenting out the foreground objects by (1) arbitrarily selecting a value to separate image into two groups, background and foreground; (2) computing the means of both groups; (3) computing the average of the means; (4) assigning the average of the means as the new thresholding value; and (5) iteratively repeat the thresholding and averaging steps until the threshold converges. We computed fundamental statistics regarding the threshold values identified by the technique for each of our cell groups (table 5.1), and we found that the average and median threshold value for each group was approximately '6'; the average and median of all videos in our corpus was '6'. Ultimately, we utilized that value to segment out the foreground from the background for every frame in our dataset, and the noise was significantly reduced when clustering (figure 5.2).

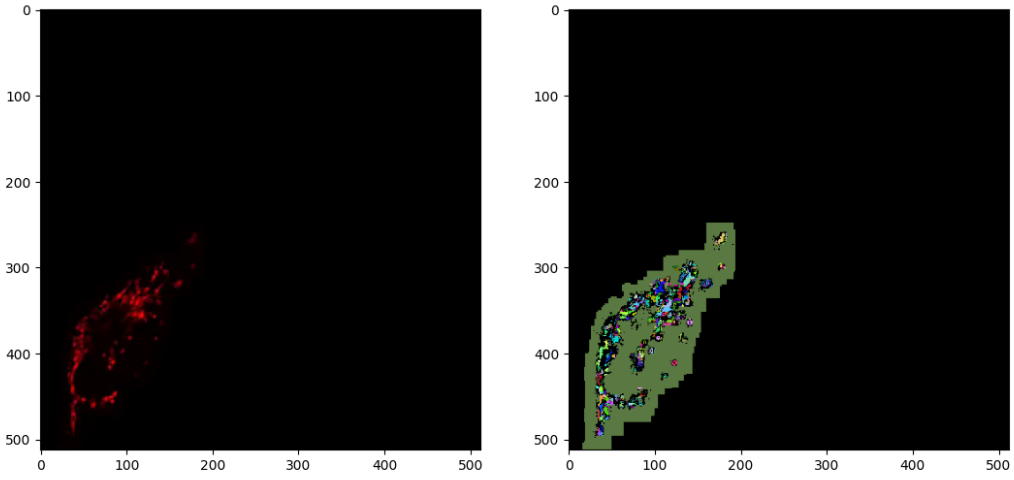

Figure 5.1: Background segmented from the foreground (threshold=0).

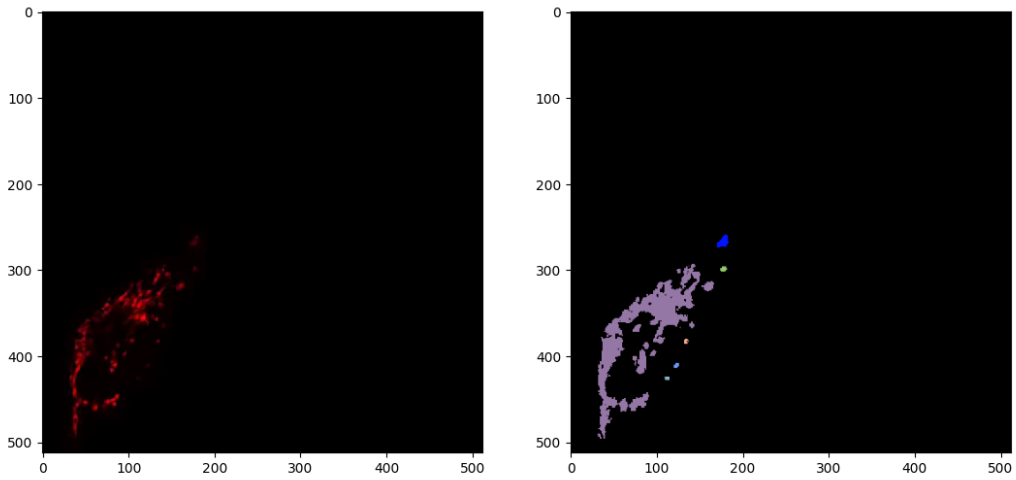

Figure 5.2: Background segmented from the foreground (threshold=6).

Table 5.1: Ridler and Calvard Background Threshold Statistics

	Min	Mean	Median	Max
Control	0	5.38	6.0	14
LLO	0	6.67	6.0	22
Mdivi	0	5.65	6.0	16
All Videos	0	6.14	6.0	22

The major phenomenon we wanted to observe by experimenting with the minimum cluster size parameter was ensuring that the graphs constructed modeled the mitochondria accurately. We acknowledged that parameter values too small would cause a massive number of clusters, and would decompose structures that we visually identified as standalone entities into smaller clusters (figure 5.3); conversely, parameter values that are too large cause structures that are spatially separated to be identified as a single mitochondrial cluster (figure 5.2). We found the absolute minimum value for the parameter, '2', decomposed structures. The default value, '5', would label mitochondrial structures that are spatially distant as one cluster (figure 5.2). Therefore, we found that for our corpus of data, we achieved the best clustering results visually and in terms of anomaly detection results with values below the default.

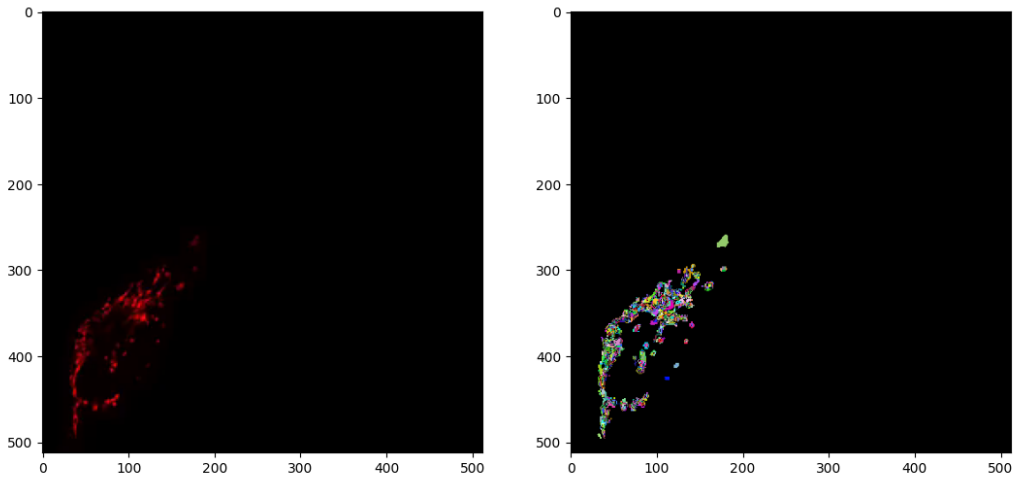

Figure 5.3: The effects of small minimum cluster parameter values (min_cluster_size=2).

The final parameter we experimented with was the minimum cluster distance threshold to explicitly enforce distinct mitochondrial structures being labeled as unique clusters. In our pursuit of accurately modeling mitochondrial structures, we recognized that adjusting the minimum cluster size parameter too high causes meaningful components of structures to be labeled as noise, and small values cause large structures to be decomposed into sub-components. We sought to alleviate both issues and create clusters that correspond to distinct entities. We found that the epsilon parameter works best with small values for the minimum cluster size. By increasing the value from the default, '0', more clusters are merged, but set the parameter too high, such as '5.0' or more, and spatially distant clusters are merged. Visually, the clusters that we generated with this parameter did not make a significant difference for most scenarios.

5.5.2 Temporal Anomaly Detection

Identification of time points that correspond to significant morphological changes in mitochondria is possible using dynamic node graphs. We found that the cluster-count sequences are more intuitive and

informative for our anomaly detection purposes than the graph spectrum sequences. In terms of intuition, it is straight-forward to understand the information being conveyed by visualizing the sequences: upwards or downwards trends indicate the increase or decrease, respectively, in the number of mitochondrial clusters detected over time and sudden shifts indicate a significant change in the number of clusters identified. Additionally, the line-plot curves of the cluster-count sequences are not as "smooth" as the graph spectrum plot curves; they are more "jagged", and we believe this corresponds to the more minor series of fission and fusion events that even healthy mitochondria experience. Such information enables richer analysis of mitochondrial dynamics.

Another major benefit of this temporal anomaly detection approach is the magnitude of shifts in the cluster-count sequences. Due to cluster-count sequences diverging sharply at times of morphological events, it is easier to isolate more significant events by increasing the outlier detection threshold value. In our previous approach, increasing the threshold parameter to a relatively high value, such as '3', can cause the detector to not identify any timepoints as anomalous, even though significant morphological activity occurs in the video (figure 5.4). Utilization of the same parameter for the current approach highlights only the most significant morphological events in a microscopy video (figure 5.5). However, we recognize that achieving the best results is still dependent on selecting the best hyper-parameters for both generating the graphs and for our analysis models.

The hyper-parameters that construct dynamic node graphs that appear to resemble the mitochondrial structures the closest do not necessarily perform the best at downstream analysis tasks, such as temporal anomaly detection. After experimenting with various combinations of hyper-parameters, we found that selecting a high background threshold, a low value for the minimum number of clusters, and a relatively high value for the distance threshold results in clusters that appear to resemble the mitochondrial structures (figure 5.6); specifically, we utilized a background threshold value of "22", which is the largest threshold value identified by the Ridlder and Calvard threshold technique for any frame in our dataset, a minimum cluster size of "3", and an epsilon value of "6". Although the values produced accurate cluster visualizations, the corresponding cluster-count sequence plot demonstrated significant oscillations in terms of the

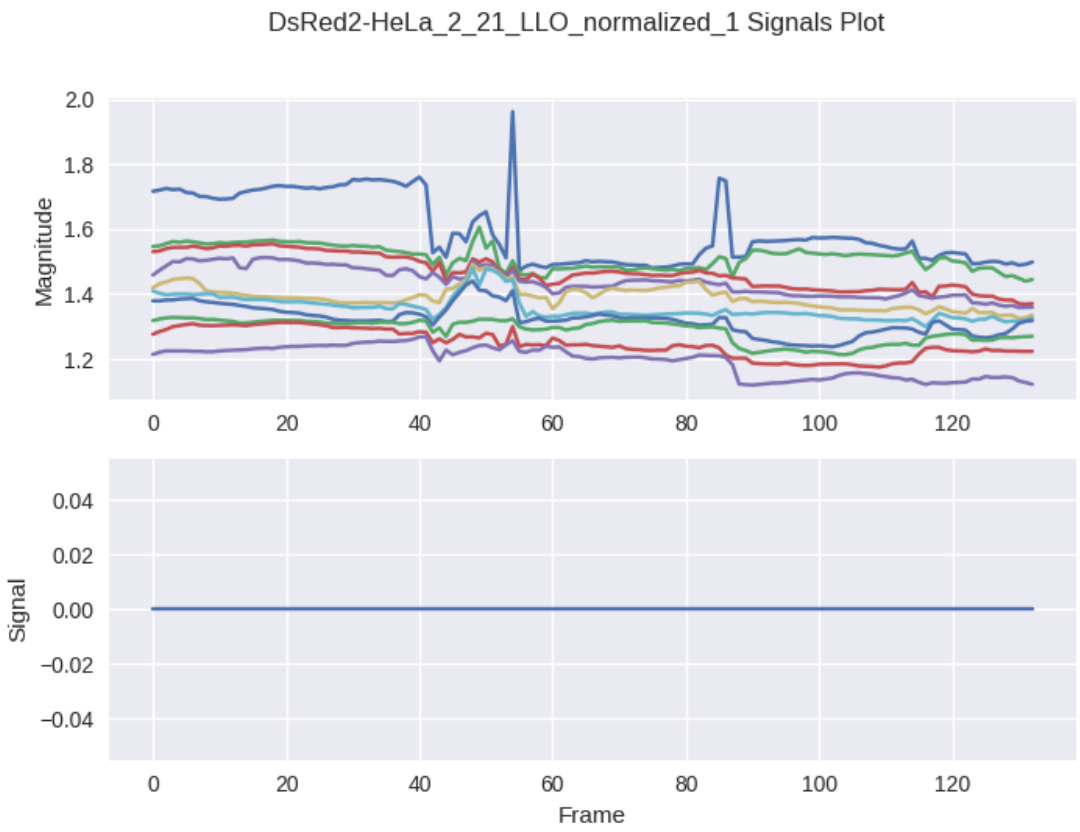

Figure 5.4: Increasing the threshold too high causes the detector to miss significant events captured by the graph spectrum (threshold=3).

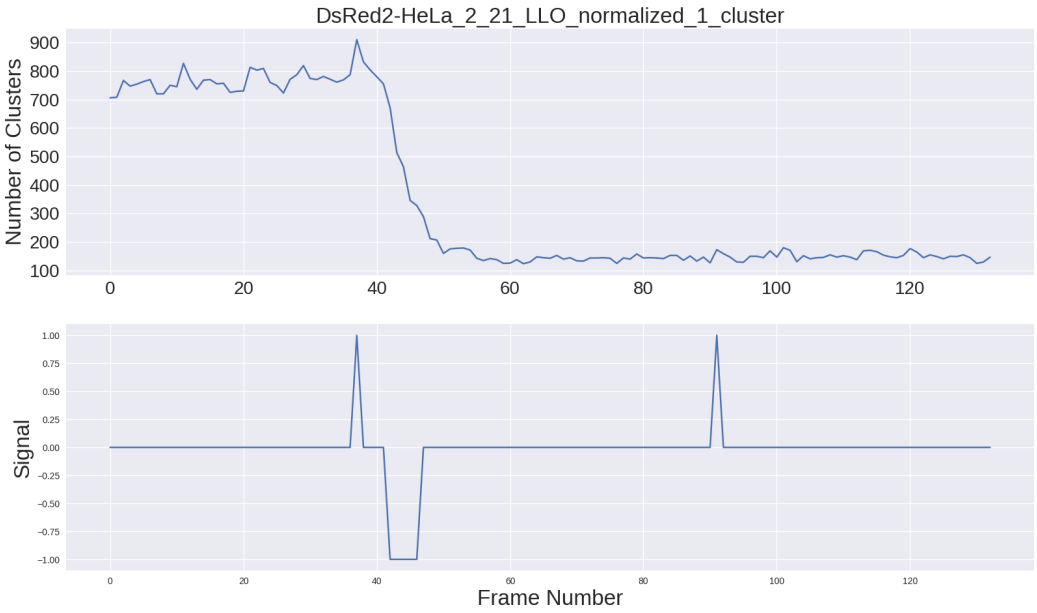

Figure 5.5: Isolating significant morphological events by increasing the threshold (threshold=3).

number of detected clusters, which severely impeded effectiveness of the temporal anomaly detector. Yet, assigning the value “6” for the background threshold value and the value “2” for the absolute minimum cluster size allowed us to construct the dynamic node graphs that created the best cluster-count sequences for analysis. Therefore, the best parameter values for temporal anomaly detection for our context appears to be the average threshold value returned by the Ridler and Calvard technique across all frames for the background threshold parameter and a low minimum cluster size. We believe that those parameters are a good place to start for other organellar analysis experiments, but acknowledge adjustments are likely necessary depending on the context of the data.

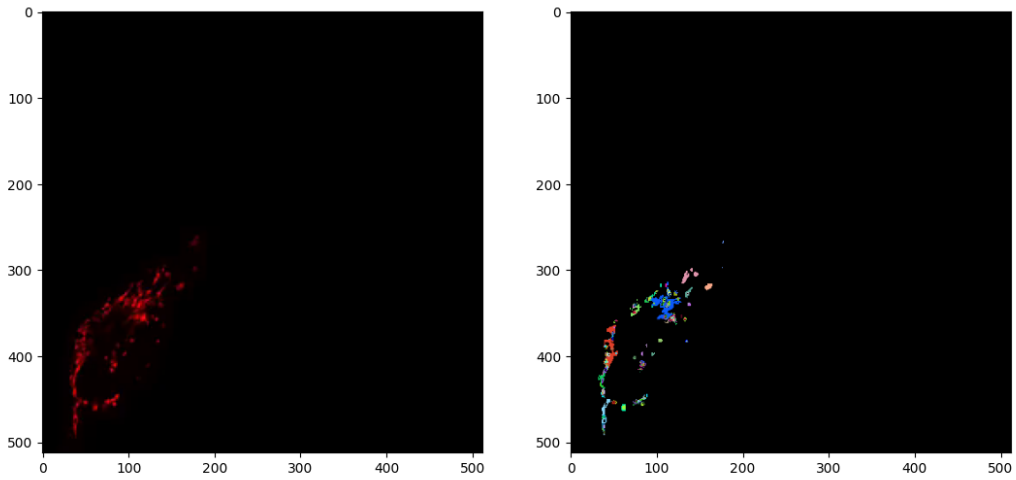

Figure 5.6: Clusters that accurately resemble the mitochondrial structures.

5.6 Conclusion

Modeling mitochondria using dynamic node graphs that reflect the morphological changes of the organelle through topological changes in the graph is a challenging, yet rewarding, strategy. Our utilization of a density-based clustering algorithm to identify mitochondrial structures in fluorescent microscopy video frames required careful consideration of various hyper-parameters to ensure that resulting graphs are biologically accurate and are a useful representation for downstream analysis tasks, such as temporal anomaly detection. Through our experiments, we were able to understand the effects of hyper-parameters with regards to modeling mitochondrial clusters and identify a general heuristic for finding parameters to aid in temporal anomaly detection. In essence, we improved upon our earlier dynamic social network graph concept.

Chapter 6

Classification of Morphological States

6.1 Introduction

We have demonstrated that dynamic social networks are sufficient representations for modeling the dynamics of mitochondria to identify anomalous behaviors, but there still exists more questions to explore, such as "What additional information is being encoded by these representations?", "How useful would this information be in other analysis tasks?", and "What are meaningful ways to extract this information?" In our attempt to explore these questions for new insights, we focused our attention on evaluating the usefulness of dynamic social network graphs as mitochondrial representations for mitochondrial dynamics classification.

We define mitochondrial dynamics classification as being able to distinguish between various morphological states of mitochondria. In our experiments, we have framed this problem as a graph classification task because we represent the morphological states of mitochondria demonstrated in microscopy videos as state graphs within a dynamic social network graph. However, such a classification is not a trivial task because it requires extracting and encoding the information contained within the graphs into representations that can serve as input to a classifier. Encoding such information is challenging because graphs

can contain vast amounts of information both explicitly and implicitly. Therefore, we made the focus of this work to be evaluating the effectiveness of dynamic social network graphs as input representations for mitochondrial dynamics classification and identifying strategies that best leverage the spatio-temporal information encoded within the dynamic graphs.

6.2 Background

6.2.1 Graph Classification

Graph data allows us to capture the complexity of various systems and model them as graphs. Graph classification has many practical applications in many different domains including bioinformatics, networks, recommender systems and transportation systems to name a few. The ability to discriminate between graphs, graph nodes, or graph edges offers a novel solution to several problems. Graph classification tasks can be divided into three main groups. Node classification categorizes nodes into different classes, but can also be a problem of completing attribute values of some nodes. Edge prediction classifies the relationships between nodes or finds hidden relationships between nodes in a graph. This task can help predict the evolution of the graph structure and whether certain nodes will interact. Lastly, graph level tasks entail classifying graphs or subgraphs into different classes. This task also includes community detection, the grouping and labeling of subgraphs. Graph classification has many applications in bioinformatics and computational biology. Graphs are able to model complex biological systems and this task can help discriminate between normal and anomalous events [43].

The biggest challenge in graph classification however lies in the fact that real world graphs are often very large and complex. They contain noise that adds little to no information about the system being modeled. Classifiers designed to distinguish between graphs must be able to filter the noise in the graphs. Graph theoretic approaches have offered solutions to these problems even before the emergence of deep graph-based methods [43].

Before the emergence of Graph Convolutional Networks and other Graph Neural Network architectures, graph kernels remained to be an efficient and robust approach to graph classification. The idea of a graph kernel method begins by extracting the statistics or features and then using these features as input to a standard machine learning classifier. Important graph statistics include node degree, node centrality, clustering coefficients, closed triangles, etc. These statistics help define graph level features as they are often an aggregate of node-level statistics. Graph kernel methods are approaches to designing features for graphs or implicit kernel functions that can be used in machine learning models. The main challenge with this method often lied in developing a kernel function that was computationally fast and efficient [43]. Kernel methods such as random walk kernels and cyclic pattern kernels often seemed inefficient at graph classification tasks [55][52]. Random walk kernels like the name suggests counts the number of common walks in the different graphs to distinguish them. It computes the common walks of two graphs and two walks are common if the walks are equal in both labels and length. The similarity of the graphs is measured based on the number of 'common walks' two graphs have. Similarly, cyclic-pattern kernels detect cyclic patterns in different graphs and use a similarity function to find the likeness of two graphs [55].

One of the most powerful and well known kernels is the Weisfeiler-Lehman kernel which improves on the bag of nodes method by using iterative neighborhood aggregation. The main idea is to extract node-level features that contain more information than their local ego graph (the subgraph containing the node, its neighbors and all the edges between the nodes in its neighborhood) and aggregate these features into the graph level representation. With the aggregated node labels, a compressed, short new label is created. The three steps of sorting, compressing and relabeling are repeated h times or until the labels differ from the two graphs. The method itself is based on the Weisfeiler-Lehman test of isomorphism on graphs. It maps the original graph to a sequence of graphs and their node attributes contain the topological and label information [43][91].

Deriving from the idea of bag of nodes, Bhuiyan et al. [11] proposed two different representations to use for graph classification: bag of vertices and bag of partitions. This representations are designed

to combat two problems: the complexity of finding discriminatory subgraphs or computing topological metrics and scarcity of graph data resulting in poor performance. The representation considers a graph as a bag of uniform random subset of vertices and each vertex in the set is set to become an independent subset of a graph classification data set, thus inflating the size of training data and improving performance. A graph is essentially mapped to a bag of multiple vertices or a bag of multiple partitions (subgraph instances) so that each of the instances in a band is a new data point for training. The label from the parent graph is inserted by the instances in each bag. The same logic can be followed for partition-induced subgraphs where each subgraph becomes a data instance and inherits the label from the original graph.

With the recent boom in deep learning and neural network architecture, graph theoretic approaches and representations have made huge contributions to the field. Previously, the user of a model needed to extract features encoding structural information about a graph but recent methods allow to automatically encode graph structure into low dimensional embeddings using deep learning techniques and dimensionality reduction [6][104]. An increasing number of systems use graph representation and creating models that capture the information and knowledge of graph data is an extremely important problem. Modeling graphs using deep learning models have various applications such as predicting the role of people in social network graphs, predicting drug interactions and recommendation systems.

The central problem in machine learning on graphs is finding a way to incorporate the graph's information into a machine learning model [6]. This is a challenging method we need to encode high dimensional, non-Euclidean information into a feature vector [6]. Like previously discussed, traditional methods rely on graph statistics, kernel function and engineered features to represent graphs. There are several disadvantages to these methods as such features are inflexible, do not always provide enough topological information about graph structure, and can be a time consuming and expensive process. Recent deep learning methods have advanced in learning representations and automatically encoding structural information about graphs by mapping nodes or subgraphs as points in low dimensional vector space. Furthermore, these approaches optimize this mapping so that mappings in the embedding space capture

the topology of the graph. Then the optimized embeddings can be used as feature inputs for downstream machine learning tasks [104].

In recent years, graph deep learning has seen a boom and the rise of several different architectures. Graph neural networks have branched into four main groups to accommodate different problem statements: Recurrent Graph Neural Networks (RecGNNs), Convolutional Graph Neural Networks (ConvGNNs),Graph Autoencoders (GAEs), and Spatial-temporal Graph Neural Networks (STGNNs) [104].

6.2.2 Time-series Analysis

A time series is a sequence of observations taken sequentially in time [102]. Time series analysis is concerned with techniques for analyzing the dependence of adjacent observations and the method of extracting meaningful summary and statistical information from points arranged in chronological order [70][102]. This helps draw predictions for future behaviors from past data patterns. Time series analysis has a number of applications, such as climate modeling [68], biological sciences [93], and finance [3].

Generally, an observed time series can be decomposed into four different patterns: trend, seasonal, cyclic and irregular [53]. A trend pattern results from either long-term increase or decrease in data. A seasonal pattern exists when the time series is affected by seasonal patterns. A cyclic pattern is a variation when data exhibits rises and falls that are not of a fixed period and a cycle usually constitutes at least two years. Irregular patterns consist of a variation of anomalous observations in a time series [53]. Furthermore, series can fall into two categories of either stationary or non-stationary data. Stationary series is a series in which the mean and variance of all the data points remain constant with time. In a non-stationary series, the mean and variance change over time. Non-stationary data is unpredictable, sporadic and often indicates relationships between variables when none exists. Thus it cannot be modeled and forecasted properly and must be transformed into stationary data to produce consistent and accurate results. Differencing is a method often applied to non-stationary data to transform it into stationary data. Differencing helps stabilize the mean and creates a new series that calculates the change between each observation in the original series [53][70][102].

The first well known time series method was published in the 1920's when Undy Yule applied harmonic analysis and regression to perform a time series analysis of sunspots [70]. The work of Yule, Gilbert and others published the first application of the autoregressive models in the 1920's and 1930's. Research in the 1950's and 1960's gave rise to other methods such as exponential smoothing [16] by researchers Robert Goodell Brown, Charles C. Holt and Peter Winters and Kalman Filters [101] by Rudolf E. Kalman [70] [102]. Exponential smoothing assigns time series data exponentially, increasing weights for the newer data. This assigns the newer data more weight and is seen to hold more relevancy. Advantages of this method include its ability to predict accurate forecasts, adjust to noise and are not limited to stationary data. On the other hand, the smoothing technique this method uses does not always capture drastic variations such as peaks and troughs in trends and is only beneficial for predicting short term trends [53][70].

Autoregressive Moving Average (ARMA) models were popularized by George E. P. Box and Jenkins in their book Time series analysis: Forecasting and control in the 1970's. This book proposed the Box-Jenkins method, which applies autoregressive moving average or autoregressive integrated moving average models to find the best fit for time series data. ARMA models are used to describe weakly stationary stochastic time series in terms of two polynomials, one for autoregression and the second for the moving average [70][102].

Autoregressive model is a regression model that produces an output based on a linear combination of input values. Moving average captures the average change in data over time and creates a new series using the raw observations in the original time series. ARMA models merge autoregressive models and moving average methods to model a time series. An extension of the ARMA models that incorporates integration is the Autoregressive Integrated Moving Average (ARIMA) model.. This model is a combination of autoregression, moving average and differencing. A common myth in the field is that ARIMA models are more general than exponential smoothing. Exponential smoothing requires two levels of differencing to make data stationary whereas ARIMA models only require one level of differencing to make the series stationary [53][102].

Traditional methods have focused on harnessing parametric models to analyze time series data; the recent machine learning boom has provided a way for models to learn temporal dynamics in a purely data driven manner. Well known deep learning architectures in recent times have produced very successful models for image classification, natural language processing and reinforcement learning [53]. These architectures are able to learn complex data representations and automate feature engineering and model design. Thus, several neural network architectures with time series applications have emerged. The flexibility and computational speed of neural networks pose advantages to applying them to time series data. Researchers have adapted convolutional neural networks, originally designed for image data, to time series data by utilizing casual convolutions. A causal convolution is a type of convolutional filter designed to ensure only past information is used for forecasting and the model cannot violate the ordering in which the data is modeled [12].

However, standard convolutional layers can be computationally intensive when dealing with long term dependencies. To solve this problem, researchers have proposed using dilated convolutions. Dilated convolutions are a type of convolution that help 'inflate' the kernel by inserting holes between the kernel elements. They allow the model to access a broad range of history when forecasting. Recurrent neural networks (RNNs) [74] have been used for sequence modeling and contain an internal memory state which keeps track of a summary of past data. This memory state helps the model learn from previous iterations during training and use processed past data, making them ideal models for time series analysis. But, RNNs are known to suffer from the limitations of learning large sequence data due to the vanishing gradient problem. Long Short-Term Memory networks (LSTMs) [48] were developed to address these limitations by adding a separate memory pipeline. This improved gradient flow within the network and the use of a cell state allowed it to store long-term information through a series of gates. RNNs and Bayesian filters are similar as they both maintain a hidden state that is recursively updated over time. RNNs however can be as simple or complex as needed while having full control of the parameters. The evolution of attention mechanisms in recent years with Transformer [97] architectures has significantly improved learning with long-term dependency. Although Transformers were created for natural language processing applications,

their attention layers prove very useful for time series analysis. The attention layers in transformer methods are able to pick up information from important time steps, even if it is from data that is several time steps back, by aggregating temporal features using dynamically generated weights.These deep learning methods have the ability to model both discrete and continuous targets.

Hybrid models on the other hand combine traditional methods with a neural network architecture. They utilize deep neural networks either to encode time varying parameters for non-probabilistic parametric models or to produce the distribution parameters used by probabilistic models. These deep architectures are used in conjunction with probabilistic generative models like Gaussian processes [82] and linear state space models for temporal dynamics or Kalman filters for inference. These methods are advantageous for small datasets where deep learning can overfit. With hybrid models, neural network training is done using prior information from domain experts and reduces the hypothesis space of the network.

6.3 Methods: Graph-level Representations

We explored two different ways to approach classifying different morphological states of mitochondria: classifying the final state of the organelle after a significant morphological event has occurred and classifying the organelle according to the dynamics patterns demonstrated over the course of the event. Due to the mitochondrial dynamics being represented as dynamic social networks, this is, ultimately, a graph classification task. By characterizing the mitochondria according to its final graph state, the problem can be formulated more like traditional graph classification problems, while characterizing the mitochondria according to their dynamics patterns is more akin to time-series classification tasks. Thus, our work examines the effectiveness of both approaches.

6.3.1 Manually Aggregated Node Feature Statistics

Classifying the morphology of mitochondria according to its representation after a morphologically significant event has occurred is a straight-forward approach to address the classification task. In the context of

dynamic social networks, such an approach means classifying the final state graph of a model. Intuitively, we believed that the final state graph of our networks, specifically those that performed object tracking using Gaussian Mixture Models, would implicitly encode meaningful dynamics information that reflected the morphological event that transpired. Thus, we sought to aggregate the node information of the final state graphs into graph-level embeddings that could be utilized as inputs to classifiers.

Our mechanism for constructing graph-level embeddings was by aggregating node feature information using descriptive statistical measures. The features we created to characterize the nodes of our graph were the quantitative properties of the mixture components that represented the mitochondrial clusters in the original image space. Specifically, we characterized each node by the parameters, or means and covariances, of the probability distribution and the area of the spatial region in the image that corresponds to the mixture component as a node-level feature vector. To generate a graph-level representation that incorporates information from each node in the graph, we performed an aggregation by leveraging a statistical measure, such as the mean, median, minimum, or maximum of each feature. In essence, we computed an element-wise aggregation of the node feature vectors to craft a single graph-level feature vector that characterizes the entire graph. Since we sought to characterize the morphology of the after an event, we only had to perform such a feature aggregation using the information contained in the final state graph, but we did, however, have to perform these aggregations for every microscopy in our corpus of data to generate a classification dataset.

6.3.2 Cluster Count Time-Series Sequences

A more sophisticated manner of classifying the morphological state of mitochondria is by characterizing it according to its dynamics patterns. For this approach, we leveraged the properties of our dynamic node social network graphs because our anomaly detection experiments demonstrated that significant morphological information is contained within these structures. The core idea of this approach is to characterize the morphology of mitochondria by the topological changes of our dynamic graph. In other words, we represented the dynamics of mitochondria by the number of clusters detected over time. Therefore, the

graph-level representation of each dynamic node social network graph is the sequence of cluster-counts. As demonstrated by the results of our temporal anomaly detection experiments, the quantity of mitochondrial clusters detected during the duration of the video reflect the morphological event that has transpired and creates distinct trends in the cluster-count sequence, and we believed that the sequence trends are distinct enough to classify morphological states.

6.4 Experiments

We conducted binary classification for all pairs of cell groups to understand whether dynamic social network graphs encode the mitochondrial dynamics in distinct ways that reflect the morphological event experienced by the organelle. These classification tasks allow us to inspect whether our graphs are sufficient for distinguishing between morphologically-altered mitochondria from healthy, unperturbed mitochondria, as well as determine whether the graphs are useful for being able to distinguish the morphological effects of different stimuli from each other. Our datasets include 113 single-cell microscopy videos: 53 LLO, 31 Mdivi, and 21 control videos.

We performed all of the binary classifications from both the classical graph classification and the time-series classification perspectives using the same process: (1) generate the graph-level representation, (2) oversample the data to account for class imbalances, (3) train a classifier using a portion of the real data and the synthetic data generated from oversampling, and (4) finally evaluate the performance of the classifier on testing set of data that consists of only real data representative of both classes. We generated manually-aggregated node feature statics to serve as the graph-level representation for our classical graph classification approach and cluster-count time-series sequences for the time-series approach according to the methodology described in section 6.3. Due to there being nearly twice as many LLO instances as any other class in our dataset we utilized the Synthetic Minority Oversampling Technique (SMOTE) [19] to generate synthetic instances to aid our classifiers. We explored the effects of withholding various amounts of real instances for the test sets, and the effect of oversampling the majority class too. For our classical graph learning approach, we utilized K-Nearest Neighbors (KNN) [34], Decision Tree (DT) [79],

and Random Forest (RF) [47] classifiers, while we employed a Recurrent Neural Network (RNN) [74], specifically a Long Short-Term Model (LSTM) [48] to address our temporal classification needs. For our experiments, we favored interpretable models to establish baseline understandings of the information contained within our dynamic social network graphs; we did, however, utilize more sophisticated classifiers, such as the LSTM networks, that are regarded as "black box" algorithms, so for those models we employed simpler architectures. Finally, we left out 15 real instances of each class for a testing set and evaluated the performance of each classifier in terms of accuracy, precision, recall, and F1 scores.

6.5 Results and Discussion

The results of our classification experiments demonstrated that our dynamic graph approaches contain enough mitochondrial dynamics information to successfully distinguish between mitochondria that have undergone significant morphological changes as the result of fission and fusion events (Figure 6.1, Figure 6.2, Figure 6.3). The best accuracy achieved for each of the binary classification tasks was around 75%. We observed that neither of the two approaches completely outperformed the other in all areas. Each classification approach had limitations that must be considered when classifying mitochondrial morphologies.

Satisfactory results were achievable when classifying the final graph state of mitochondria using the manually aggregated node statistics as the graph-level representation; however, very specific conditions had to be met for successful outcomes. The manually aggregated node feature datasets were sensitive to the oversampling strategies we employed and would only achieve above random guessing accuracy results, which in this context is around 50% accuracy, when only the minority class was oversampled. This severely limited the number of instances we could train our classifiers on due to 15 of the real instances from both classes being set aside for the training dataset. Therefore, generating too many synthetic instances of the minority classes would create new imbalance issues. During the LLO-control and Mdivi-control classification problems all classes were represented with only 38 instances, consisting of real and synthetic data, in the training set, while the LLO-Mdivi task were only able to represent both classes with 16 instances, also consisting of real and synthetic data. Additionally, the descriptive statistical measure that was utilized

to aggregate the node features to achieve the best results varied depending on the classifier and problem context, meaning no single aggregation method universally outperformed the others, good results were typically the result of a good combination of classifier and aggregation strategy. Thus, it is difficult to indicate a single aggregation to achieve the best results; all strategies must be examined to ensure the best results.

The time-series classification approach that focused on classifying the morphology of the mitochondria according to their dynamics patterns did not suffer from any of the limitations of the other approach, but it did fail to sufficiently classify the Mdivi group from the control group (table 6.1, table 6.2, table 6.3). The results of the LSTM classifier on the cluster-counts time-series sequences achieved near random guessing level results, which is not surprising due to the large similarity between sequences of both groups. Majority of the models developed for the other approach also struggled with classifying these two different groups apart; however, a few were able to achieve satisfactory results. Although this approach was able to properly classify the LLO group from the control and the LLO group from the Mdivi group, failing to classify the Mdivi group from the control group could be signs that this approach could not sufficiently distinguish between healthy and perturbed mitochondria in a critical context.

Ultimately, each approach has its own strengths and weaknesses that must be considered when classifying the morphologies of the various mitochondrial groups. We believe the time-series approach was the overall best model due to its ability to generate good results without having to evaluate various aggregation strategies to possibly achieve good results, which could be costly in terms of time for large-scale datasets. However, for smaller datasets or contexts that are not time-sensitive, experimenting with node feature aggregation strategies and classifiers combinations may yield the best results.

Table 6.1: Control vs. LLO

	Accuracy	Precision		Recall		F1	
		Control	LLO	Control	LLO	Control	LLO
KNN (median)	0.63	0.61	0.59	0.52	0.68	0.56	0.63
DT (max)	0.69	**0.87**	0.63	0.44	**0.93**	0.57	**0.75**
RF (max)	0.68	0.83	0.62	0.45	0.91	0.57	0.74
Cluster Count LSTM	**0.74**	0.72	**0.79**	**0.79**	0.69	**0.75**	0.73

Table 6.2: Control vs. Mdivi

	Accuracy	Precision		Recall		F1	
		Control	Mdivi	Control	Mdivi	Control	Mdivi
KNN (mean)	0.70	0.78	0.67	0.59	0.82	0.66	0.73
DT (max)	0.70	0.75	0.67	0.57	0.83	0.64	0.76
RF (median)	**0.76**	**0.88**	**0.70**	**0.61**	**0.91**	**0.71**	**0.79**
Cluster Count LSTM	0.48	0.47	0.45	0.53	0.43	0.47	0.42

Table 6.3: LLO vs. Mdivi

	Accuracy	Precision		Recall		F1	
		LLO	Mdivi	LLO	Mdivi	LLO	Mdivi
KNN (mean)	0.60	0.60	0.60	0.62	0.57	0.60	0.57
DT (min)	0.71	0.65	**0.88**	**0.93**	0.49	0.76	0.62
RF (min)	**0.74**	0.69	0.86	0.90	0.58	**0.78**	0.69
Cluster Count LSTM	**0.74**	**0.81**	0.71	0.66	**0.82**	0.72	**0.76**

6.6 Conclusion

Dynamic social network graphs have demonstrated the ability to encode significant amounts of mitochondrial dynamics by successfully distinguishing between various groups of the cells that underwent various morphological changes as a response to different external stimuli. Such results provide an empirical means of justification for these modeling and analysis approaches. The results highlighted the importance of feature aggregation techniques in constructing graph-level representations to serve as input to classifiers. Although our results demonstrated that both classical machine learning and deep learning classifiers are able to learn mitochondrial dynamics information from these implicit and explicit representations, there still exists much room for improvement.

Chapter 7

Conclusion

Mitochondria are an essential component of understanding cellular responses because this organelle is actively targeted by pathogens and cellular invaders to disrupt the functionings of the cell. Thus, proper modeling and analysis of the morphological perturbations experienced can provide new insights regarding the infection strategies of pathogens. Our work explores the process of modeling these structures in an automated manner using quantitative techniques.

Specifically, we developed a Python framework, OrNet, that models mitochondrial structures as dynamic social network graphs. Performing such a modeling feat was no trivial task because mitochondria are amorphous, spatially diffuse structures. Yet, we were able to transform the pixel representations of mitochondria in fluorescent microscopy videos to nodes of dynamic graphs in a topological space. Our initial mechanism for constructing such graphs was by detecting and tracking dense regions of mitochondrial structures using probabilistic clustering techniques.

We explored the expressive capabilities of our dynamic social network graphs by developing anomaly detection methodologies to identify when the morphology of mitochondria underwent significant changes due to an external event. We leveraged the connectivity information of our dynamic graphs to identify when and where morphology-altering events were occurring in live microscopy videos. We are able to successfully develop quantitative means for indicating such anomalous behaviors through application of spectral graph theory techniques.

Although our initial modeling approach proved to be successful and useful in terms of anomaly detection tasks, we believed we could improve our mitochondrial dynamics encoding abilities by restructuring our dynamic social network graphs to utilize dynamic nodes, rather than edges. Dynamic node social networks were challenging to construct, they were better suited for the temporal anomaly detection task.

An additional downstream analysis task that we performed to understand the expressive capabilities of our dynamic social network graphs was mitochondrial dynamics classification. Classifying the morphological states of mitochondria perturbed by various external stimuli provided us another significant analysis tool that could be utilized to understand mitochondrial responses to environmental disruptions and an empirical means of justifying our dynamic social network modeling approach.

Ultimately, we have demonstrated that dynamic social network graphs are useful models for capturing the dynamics of mitochondrial structures. Additionally, we have developed several methodologies for automated analysis of the dynamics information encoded in our graphs. Our hope is that such analysis techniques will be utilized in future works to further elucidate mitochondrial dynamics and aid in the development of medical treatments and therapies.

Bibliography

[1] F. Akhter, D. Chen, S. Yan, and S. Yan, "Mitochondrial perturbation n alzheimer's disease and diabetes," *Progress in molecular biology and translational science*, 2017. DOI: 10.1016/bs.pmbts.2016.12.019.

[2] M. Akita, M. Suzuki-Karasaki, K. Fujiwara, *et al.*, "Mitochondrial division inhibitor-1 induces mitochondrial hyperfusion and sensitizes human cancer cells to trail-induced apoptosis," *International journal of oncology*, vol. 45, Aug. 2014. DOI: 10.3892/ijo.2014.2608.

[3] T. G. Andersen, T. Bollerslev, P. F. Christoffersen, and F. X. Diebold, "Volatility forecasting," National Bureau of Economic Research, Working Paper 11188, Mar. 2005. DOI: 10.3386/w11188. [Online]. Available: http://www.nber.org/papers/w11188.

[4] L. Backstrom and J. Leskovec, "Supervised random walks: Predicting and recommending links in social networks," in *WSDM '11: Proceedings of the fourth ACM international conference on Web search and data mining*, 2011, pp. 635–644. DOI: 10.1145/1935826.1935914.

[5] P. Bernardi, "Looking back to the future of mitochondrial research," *Frontiers in Physiology*, vol. 12, Apr. 2021. DOI: 10.3389/fphys.2021.682467.

[6] L. Berton, A. Lopes, and D. Vega-Oliveros, "A comparison of graph construction methods for semi-supervised learning," Jul. 2018, pp. 1–8. DOI: 10.1109/IJCNN.2018.8489524.

[7] S. Beucher and C. Lantuéjoul, "Use of watersheds in contour detection," vol. 132, Jan. 1979.

[8] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," *IEEE International Conference on Image Processing (ICIP)*, 2016.

[9] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node classification in social netwroks," *Social Network Data Analytics*, pp. 115–148, 2011. DOI: 10.1007/978-1-4419-8462-3_5.

[10] P. Bhattacharjee and P. Mitra, "A survey of density based clustering algorithms," *Frontiers of Computer Science*, 2021.

[11] M. Bhuiyan and M. Hasan, "Representing graphs as bag of vertices and partitions for graph classification," *Data Science and Engineering*, vol. 3, Jun. 2018. DOI: 10.1007/s41019-018-0065-5.

[12] A. Borovykh, S. Bohte, and C. Oosterlee, "Dilated convolutional neural networks for time series forecasting," *Journal of Computational Finance*, Mar. 2017. DOI: 10.21314/JCF.2019.358.

[13] E. Bossy-Wetzel, M. Barsoum, A. Godzik, R. Schwarzenbacher, and S. Lipton, "Mitochondrial fission in apoptosis, neurodegeneration and aging," *Current opinion in cell biology*, vol. 15, pp. 706–16, Jan. 2004. DOI: 10.1016/j.ceb.2003.10.015.

[14] C. Bouman, M. Shapiro, Ncsa, G. Cook, and S. Borman, "Cluster: An unsupervised algorithm for modeling gaussian mixtures," Jan. 1995.

[15] M. Brand, A. Orr, I. Perevoshchikova, and C. Quinlan, "The role of mitochondrial function and cellular bioenergetics in ageing and disease," *British Journal of Dermatology*, vol. 169, no. s2, pp. 1–8, 2013. DOI: https://doi.org/10.1111/bjd.12208. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/bjd.12208. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/bjd.12208.

[16] R. G. Brown, *Statistical forecasting for inventory control*. McGraw/Hill, 1959.

[17] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining*, 2013.

[18] A. Carpenter, T. Jones, M. Lamprecht, *et al.*, "Cellprofiler: Image analysis software for identifying and quantifying cell phenotypes," *Genome biology*, vol. 7, R100, Feb. 2006. DOI: 10.1186/gb-2006-7-10-r100.

[19] N. Chawla, K. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority oversampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[20] J. Chen, L. Ding, M. P. Viana, *et al.*, "The allen cell structure segmenter: A new open source toolkit for segmenting 3d intracellular structures in fluorescence microscopy images."

[21] C. T. Chu, "Mitochondria in neurodegeneration," *Current Opinion in Physiology*, p. 100 532, Apr. 2022. DOI: 10.1016/j.cophys.2022.100532.

[22] F. R. K. Chung, "Spectral graph theory," 1996.

[23] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, vol. 2, Jun. 2005.

[24] G. Delogu, M. Sali, and G. Fadda, "The biology of mycobacterium tuberculosis infection," *Mediterranean Journal of Hematology and Infectious Diseases*, 2013.

[25] B. S. Deshmukh and V. H. Mankar, "Segmentation of microscopic images: A survey," in *2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies*, 2014, pp. 362–364. DOI: 10.1109/ICESC.2014.68.

[26] S. Detmer and D. Chan, "Functions and dysfunctions of mitochondrial dynamics," *Nature reviews. Molecular cell biology*, vol. 8, pp. 870–9, Dec. 2007. DOI: 10.1038/nrm2275.

[27] S. Dimopoulos, C. Mayer, F. Rudolf, and J. Stelling, "Accurate cell segmentation in microscopy images using membrane patterns," *Bioinformatics*, vol. 30, pp. 2644–2651, Sep. 2014. DOI: 10.1093/bioinformatics/btu302.

[28] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine Learning*, vol. 29, Feb. 1998. DOI: 10.1023/A:1007413511361.

[29] R. K. Dubey, "Assuming the role of mitochondria in mycobacterial infection," *International Journal of Mycobacteriology*, 2016.

[30] M. Embley, "The archaebacterial origin of eukaryotes," *Comparative Biochemistry and Physiology Part A: Molecular & Integrative Physiology*, vol. 153, S46–S47, Jun. 2009. DOI: 10.1016/j.cbpa.2009.04.498.

[31] M. Fazli, M. Hill, A. Durden, *et al.*, "Ornet-a python toolkit to model the diffuse structure of organelles as social networks," *Journal of Open Source Software*, 2020.

[32] A. Ferree and O. Shirihai, "Mitochondrial dynamics: The intersection of form and function," *Advances in experimental medicine and biology*, vol. 748, pp. 13–40, Jun. 2012. DOI: 10.1007/978-1-4614-3573-0_2.

[33] K. Fine-Coulson, S. Giguere, F. D. Quinn, and B. J. Reaves, "Infection of a549 human type ii epithelial cells with mycobacterium tuberculosis induces changes in mitochondrial morphology, distribution and mass that are dependent on the early secreted antigen, esat-6," *Microbes and Infection*, 2015.

[34] E. Fix and J. Hodges, "Discriminatory analysis: Nonparametric discrimination: Consistency properties," *International Statistical Review*, vol. 57, Dec. 1989. DOI: 10.2307/1403797.

[35] G. M. Fogo, A. R. Anzell, K. J. Maheras, *et al.*, "Machine learning-based classification of mitochondria morphology in primary neurons and brain," *Scientific Reports*, 2021.

[36] S. Fujita and X.-H. Han, "Cell detection and segmentation in microscopy images with improved mask r-cnn," in Feb. 2021, pp. 58–70, ISBN: 978-3-030-69755-6. DOI: 10.1007/978-3-030-69756-3_5.

[37] C. A. Galloway, H. Lee, and Y. Yoon, "Mitochondrial morphology-emerging role in bioenergetics," *Free radical biology and medicine*, 2021. DOI: https://doi.org/10.1016/j.freeradbiomed.2012.09.035.

[38] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3d traffic scene understanding from movable platforms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[39] V. D. Gesu and V. Starovioitov, "Distance-based functions for image comparison," *Pattern Recognition Letters*, 1999.

[40] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169.

[41] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision adn Pattern Recognition*, 2014.

[42] "Global tuberculosis report 2021," *Geneva: World Health Organization*, 2021.

[43] W. Hamilton, "Graph representation learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 14, pp. 1–159, Sep. 2020. DOI: 10.2200/S01045ED1V01Y202009AIM046.

[44] T. Harmuth, C. Prell-Schicker, J. Weber, *et al.*, "Mitochondrial morphology, function and homeostasis are impaired by expression of an n-terminal calpain cleavage fragment of ataxin-3," *Frontiers in Molecular Neuroscience*, vol. 11, Sep. 2018. DOI: 10.3389/fnmol.2018.00368.

[45] E. Hellinger, "Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen.," *Journal für die reine und angewandte Mathematik*, vol. 1909, no. 136, pp. 210–271, 1909. DOI: doi:10.1515/crll.1909.136.210. [Online]. Available: https://doi.org/10.1515/crll.1909.136.210.

[46] A. Hinneburg and H.-H. Gabriel, "Denclue 2.0: Fast clustering based on kernel density estimation," vol. 4723, Sep. 2007, pp. 70–80, ISBN: 978-3-540-74824-3. DOI: 10.1007/978-3-540-74825-0_7.

[47] T. Ho, "Random decision forests," vol. 1, Sep. 1995, 278–282 vol.1, ISBN: 0-8186-7128-9. DOI: 10.1109/ICDAR.1995.598994.

[48] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, 1997.

[49] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, pp. 85–126, Oct. 2004. DOI: 10.1023/B:AIRE.0000045502.10941.a9.

[50] E. Hodneland, T. Kögel, D. Frei, H.-H. Gerdes, and A. Lundervold, "Cellsegm - a matlab toolbox for high-throughput 3d cell segmentation," *Source code for biology and medicine*, vol. 8, p. 16, Aug. 2013. DOI: 10.1186/1751-0473-8-16.

[51] A. Horn, S. Raavicharla, S. Shah, D. Cox, and J. Jaiswal, "Mitochondrial fragmentation enables localized signaling required for cell repair," *The Journal of Cell Biology*, vol. 219, Apr. 2020. DOI: 10.1083/jcb.201909154.

[52] T. Horvath, T. Gartner, and S. Wrobel, "Cyclic pattern kernels for predictive graph mining," Aug. 2004, pp. 158–167. DOI: 10.1145/1014052.1014072.

[53] R. J. Hyndman and G. Athanasopoulos, "Forecasting: Principles and practice," OTexts, 2013.

[54] B. Ingale and P. Fulare, "Review of algorithms for clustering random data," *International Journal of Computer Science and Information Technologies*, 2014.

[55] U. Kang, H. Tong, and J. Sun, "Fast random walk graph kernel," *Proceedings of the 12th SIAM International Conference on Data Mining, SDM 2012*, pp. 828–838, Apr. 2012. DOI: 10.1137/1.9781611972825.71.

[56] H. W. Kuhn, "Hte hungarian method for the assignment problem," *Naval Research Logistics Quaterly*, 1955.

[57] J. Landry, P. Pyl, T. Rausch, *et al.*, "The genomic and transcriptomic landscape of a hela cell line," *G3 (Bethesda, Md.)*, vol. 3, Mar. 2013. DOI: 10.1534/g3.113.005777.

[58] A. Lefebvre, D. Ma, K. Kessenbrock, D. Lawson, and M. Digman, "Automated segmentation and tracking of mitochondria in live-cell time-lapse images," *Nature Methods*, vol. 18, Sep. 2021. DOI: 10.1038/s41592-021-01234-z.

[59] A. P. Leonard, R. B. Cameron, J. L. Speiser, *et al.*, "Quantitative analysis of mitochondrial morphology and membrane potential in living cells using high-content imaging, machine learning, and morphological binning," *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research*, vol. 1853, no. 2, pp. 348–360, 2015.

[60] J. Leskovec and A. Krevl, *Snap datasets: Stanford large network dataset collection*, Jun. 2014. [Online]. Available: `http://snap.stanford.edu/data`.

[61] A. Lohia, K. Kadam, R. Joshi, and D. Bongale, "Bibliometric analysis of one-stage and two-stage object detection," *Library Philosophy and Practice*, Feb. 2021.

[62] Y. Lu, X. Qin, H. Fan, T. Lai, and Z. Li, "Wbc-net: A white blood cell segmentation network based on unet++ and resnet," *Applied Soft Computing*, vol. 101, p. 107 006, Mar. 2021. DOI: `10.1016/j.asoc.2020.107006`.

[63] C. Malzer and M. Baum, "A hybrid approach to hierarchical density-based cluster selection," Sep. 2020, pp. 223–228. DOI: `10.1109/MFI49285.2020.9235263`.

[64] J. Martijn, J. Vosseberg, L. Guy, P. Offre, and T. Ettema, "Deep mitochondrial origin outside the sampled alphaproteobacteria," *Nature*, vol. 557, May 2018. DOI: `10.1038/s41586-018-0059-5`.

[65] W. F. Martin and M. Mentl, "The origin of mitochondria," *Nature Education*, 2010.

[66] J. McQueen, "Some methods for classification and analysis of multivariate observations," *Computer and Chemistry*, vol. 4, pp. 257–272, Jan. 1967.

[67] K. Mitra and J. Lippincott-Schwartz, "Analysis of mitochondrial dynamics and functions using imaging approaches," *Current protocols in cell biology / editorial board, Juan S. Bonifacino ... [et al.]*, vol. Chapter 4, Unit 4.25.1–21, Mar. 2010. DOI: `10.1002/0471143030.cb0425s46`.

[68] M. Mudelsee, "Trend analysis of climate time series: A review of methods," *Earth-Science Reviews*, vol. 190, Dec. 2018. DOI: `10.1016/j.earscirev.2018.12.005`.

[69] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Adv. Neural Inf. Process. Syst*, vol. 14, Apr. 2002.

[70] A. Nielsen, *Practical Time Series Analysis*. O'Reilly Media, Inc., 2019.

[71] Nisha and P. J. Kaur, "A survey of clustering techniques and algorithms," *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 304–307, 2015.

[72] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. DOI: 10.1109/TSMC.1979.4310076.

[73] I. Oztel, G. Yolcu, I. Ersoy, T. White, and F. Bunyak, "Mitochondria segmentation in electron microscopy volumes using deep convolutional neural network," Nov. 2017, pp. 1195–1200. DOI: 10.1109/BIBM.2017.8217827.

[74] B. A. Pearlmutter, "Gradient calculations for dyanmic recurrent neural networks: A survey," *IEEE Transactions on Neural Networks*, 1995.

[75] J.-Y. Peng, C.-C. Lin, Y.-J. Chen, *et al.*, "Automatic morphological subtyping reveals new roles of caspases in mitochondrial dynamics," *PLoS Computational Biology*, 2011.

[76] A. A. Perea, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu, "Multi-object tracking through simultaneous long occlusions and split-merge conditions," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[77] M. Picard, O. S. Shirihai, B. J. Gentil, and Y. Burelle, "Mitochondrial morphology transitions and functions: Implications for retrograde signaling?" *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, vol. 304, no. 6, R393–R406, 2013, PMID: 23364527. DOI: 10.1152/ajpregu.00584.2012. eprint: https://doi.org/10.1152/ajpregu.00584.2012. [Online]. Available: https://doi.org/10.1152/ajpregu.00584.2012.

[78] B. Preim and C. Botha, "Image analysis for medical visualization," in Dec. 2014, pp. 111–175, ISBN: 9780124158733. DOI: 10.1016/B978-0-12-415873-3.00004-3.

[79] R. Quinlan, *C4.5: Programs for Machine Learning*. Jan. 1993, vol. 1, ISBN: 1-55860-238-0.

[80] E. Ramond, A. Jamet, M. Coureuil, and A. Charbit, "Pivotal role of mitochondria in macrophage response to bacterial," *Frontiers in Immunology*, 2019.

[81] Y. Rani and H. Rohil, "A study of hierarchical clustering algorithm," *International Journal of Information and Computation Technology*, vol. 3, pp. 1115–1122, Jan. 2013.

[82] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*, O. Bousquet, U. von Luxburg, and G. Rätsch, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 63–71, ISBN: 978-3-540-28650-9. DOI: 10.1007/978-3-540-28650-9_4. [Online]. Available: https://doi.org/10.1007/978-3-540-28650-9_4.

[83] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.

[84] Y. Reis, M. Bernardo-Faura, D. Richter, *et al.*, "Multi-parametric analysis and modeling of relationships between mitochondrial morphology and apoptosis," *PLoS ONE*, 2012.

[85] T. Ridler and S. Calvard, "Picture thresholding using an iterative selection method," *IEEE Transactions on Systems, Man, and Cybernetics - TSMC*, vol. 8, pp. 630–632, Aug. 1978. DOI: 10.1109/TSMC.1978.4310039.

[86] A. Roger, S. Muñoz-Gómez, and R. Kamikawa, "The origin and diversification of mitochondria," *Current Biology*, vol. 27, R1177–R1192, Nov. 2017. DOI: 10.1016/j.cub.2017.09.015.

[87] L. Sagan, "On the origin of mitosing cells," *Journal of theoretical biology*, 1967. DOI: 10.1016/0022-5193(67)90079-3.

[88] E. Schikuta, "Grid-clustering: An efficient hierarchical clustering method for very large data sets," *Proceedings of 13th International Conference on Pattern Recognition*, vol. 2, 101–105 vol.2, 1996.

[89] C. Schneider, W. Rasband, and K. Eliceiri, "Nih image to imagej: 25 years of image analysis," *Nature Methods*, vol. 9, Jul. 2012. DOI: 10.1038/nmeth.2089.

[90] M. Shahid Iqbal, S. El-Ashram, S. Hussain, *et al.*, "Efficient cell classification of mitochondrial images by using deep learning," *Journal of optics*, Jan. 2019. DOI: 10.1007/s12596-018-0508-4.

[91] N. Shervashidze, P. Schweitzer, E. Jan, V. Leeuwen, K. Mehlhorn, and K. Borgwardt, "Weisfeiler-lehman graph kernels," *Journal of Machine Learning Research*, vol. 1, pp. 1–48, Jan. 2010.

[92] L. Smith and E. Barton, "Smash - semi-automatic muscle analysis using segmentation of histology: A matlab application," *Skeletal muscle*, vol. 4, p. 21, Nov. 2014. DOI: 10.1186/2044-5040-4-21.

[93] D. Stoffer and H. Ombao, "Editorial: Special issue on time series analysis in the biological sciences," *Journal of Time Series Analysis*, vol. 33, Sep. 2012. DOI: 10.1111/j.1467-9892.2012.00805.x.

[94] J. Suárez-Rivero, M. Villanueva, P. Cruz-Ojeda, *et al.*, "Mitochondrial dynamics in mitochondrial diseases," *DISEASES*, vol. 5, Dec. 2016. DOI: 10.3390/diseases5010001.

[95] V. Sukhorukov, N. Mudzhiri, A. Voronkova, T. Baranich, V. Glinkina, and S. Illarioshkin, "Mitochondrial disorders in alzheimer's disease," *Biochemistry (Moscow)*, vol. 86, May 2021. DOI: 10.1134/S0006297921060055.

[96] C. Szegedy, W. Liu, Y. Jia, *et al.*, "Going deeper with convolutions," Jun. 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594.

[97] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[98] P. Villace, R. M. Mella López, and D. Kortazar, "Mitochondria in the context of parkinson's disease," *Neural Regeneration Research*, vol. 12, Mar. 2017. DOI: 10.4103/1673-5374.200802.

[99] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," vol. 1, Feb. 2001, pp. I–511, ISBN: 0-7695-1272-0. DOI: 10.1109/CVPR.2001.990517.

[100] Z. Wang, "Cell segmentation for image cytometry: Advances, insufficiencies, and challenges," *Cytometry Part A*, vol. 95, Dec. 2018. DOI: 10.1002/cyto.a.23686.

[101] G. Welch and G. Bishop, "An introduction to kalman filter," in *SIGGRAPH 2001*, 1995.

[102] G. T. Wilson, "Time series analysis: Forecasting and control, 5th edition, by george e. p. box, gwilym m. jenkins, gregory c. reinsel and greta m. ljung, 2015. published by john wiley and sons inc., hoboken, new jersey, pp. 712. isbn: 978-1-118-67502-1," *Journal of Time Series Analysis*, vol. 37, no. 5, pp. 709–711, 2016. DOI: https://doi.org/10.1111/jtsa.12194. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jtsa.12194. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/jtsa.12194.

[103] N. Wojke, A. Bewley, and D. Paulus, "Simple online realtime tracking with a deep association metric," *IEEE International Conference on Image Processing (ICIP)*, 2017.

[104] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, pp. 1–21, Mar. 2020. DOI: 10.1109/TNNLS.2020.2978386.

[105] C. Xiao, X. Chen, W. Li, *et al.*, "Automatic mitochondria segmentation for em data using a 3d supervised convolutional network," *Frontiers in Neuroanatomy*, vol. 12, Nov. 2018. DOI: 10.3389/fnana.2018.00092.

[106] R. Youle and A. Bliek, "Mitochondrial fission, fusion, and stress," *Science (New York, N.Y.)*, vol. 337, pp. 1062–5, Aug. 2012. DOI: 10.1126/science.1219855.

[107] P. A. Yushkevich, J. Piven, H. Cody Hazlett, *et al.*, "User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *Neuroimage*, vol. 31, no. 3, pp. 1116–1128, 2006.

[108] A. Zahedi, V. On, R. Phandthong, *et al.*, "Deep analysis of mitochondria and cell health using machine learning," *Scientific Reports*, 2018.

[109] R. Zhu, D. Sui, H. Qin, and A. Hao, "An extended type cell detection and counting method based on fcn," Oct. 2017, pp. 51–56. DOI: 10.1109/BIBE.2017.00-79.