Novel Approaches to Time-Lapsed Microscopy Image Analysis and Detection of Biological Agents

By

Stephen Gunther Hummel

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

Chemical and Physical Biology

May, 2014

Nashville, Tennessee

Approved:

Vito Quaranta, M.D.

Carlos Lopez, Ph.D.

Thomas Yankeelov, Ph.D.

ACKNOWLEDGEMENTS

First, I would like to thank Dr. Vito Quaranta, Dr. Lourdes Estrada, and Dr. Darren Tyson for giving me the opportunity to attend Vanderbilt University and to work in the Quaranta Laboratory. I am truly humbled from the opportunity and learned a tremendous amount from you all. I hope the work described in this thesis will have broad applications and help progress research in the lab. I would also like to thank my thesis committee, Dr. Vito Quaranta, Dr. Carlos Lopez, and Dr. Thomas Yankeelov, for your guidance and comments throughout this project.

Specifically in terms of this work I would like to thank Shawn Garbett and Dr. Darren Tyson. Your efforts, patience, and discussion of problems as I slowly worked through the project was tremendously helpful. Our discussions were always insightful and useful. I hope that the algorithm is something that you can use but there is always benefit to having summer students track cells.

To the other members of the Quaranta lab, good luck. Several of you will be graduating soon and have put in a tremendous amount of energy and effort into your research. You are making a difference and providing novel insights into understanding the systems of cancer biology. To the students who are just beginning keep your head up and remember you can learn something from every setback, just keep it all in perspective.

I would also like to thank my family. Krissy and Asher you made my time at Vanderbilt University productive and fun. Thank you for allowing me to bring my work home with me. I look forward to our future adventures.

The opportunity to work on this project would not have been possible without funding from the United States Army Advanced Civil School Program. The program has covered my tuition expenses and afforded me the opportunity to become an instructor at the United States Military Academy at West Point.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

This Thesis Dissertation is comprised of four chapters that, respectively: 1) Examine the potential value of studying cellular dynamics and heterogeneity in the context of emerging biological warfare threats; 2) Review both the advantages and disadvantages of current methods that track cells using time-lapse live-cell microscopy; 3) Propose a novel algorithm to measure and track changes in cellular behavior dynamically; and 4) Highlight a novel semi-automated algorithm developed to identify and track cellular focal adhesions overtime in 3D.

Background

Biological warfare has existed for centuries, with one of the earliest known examples occurring in 1155 when Emperor Frederick Barbarossa poisoned water wells with cadavers in the siege of Tortona, Italy.[1] Such incidents have continued throughout the ages. In 1972, the Convention on the Prohibition of the Development, Production, and Stockpiling of Bacteriological (Biological) and Toxin Weapons and their Destruction was signed and adopted for enforcement by the United Nations Office for Disarmament Affairs.[2] This treaty aims to prevent the development of offensive[3] biological weapon (BW) agents and eliminate existing stockpiles; however, it only applies to those 170 nation-states that signed the convention and does not affect the actions of the 23 non-signatory states, such as Israel, Chad, and Kazakhstan,[4] or independent groups and individuals that seek to employ such weapons.

The 2001 anthrax letters demonstrated that the 1972 BW convention limits only one aspect of the problem. Weapons of mass destruction (WMD), once previously under the sole control of nation-states, now could be maintained and deployed by an individual, albeit possibly in smaller quantities than could be produced by a nation-state. In 2010, it was concluded that these letters, which were sent to political leaders and media outlets across the United States, constituted a terrorist attack[5] and were sent by Dr. Bruce Ivins, a trained microbiologist employed by the Department of Defense.[6] In April and May of 2013, two separate ricin letter attacks were allegedly carried out by individuals who, with little to no scientific experience and support, were able to create a biological agent, albeit one that may not have had the potency of an effective weapon.[7] Compared to the 2001 anthrax letters, the separate 2013 ricin letters illustrate a transition in BW production from the trained individual to the layman, as it has been alleged that the first set of letters was sent by a karate instructor from Tupelo, Mississippi,[8] and the second set from a part-time actress / housewife from Dallas, Texas, who pleaded guilty to sending the letters on

December 11, 2013.[9] These recent incidents demonstrated that a relatively low level of sophistication and technological knowledge was no bar to deployment of a WMD.[10]

A 2005 Washington Post article by Steve Coll and Susan Glasser presciently stated that "one can find on the web how to inject animals, like rats, with pneumonic plague and how to extract microbes from infected blood . . . and how to dry them so that they can be used with an aerosol delivery system, and thus how to make a biological weapon. If this information is readily available to all, is it possible to keep a determined terrorist from getting his hands on it?"[11]

The ability of non-scientists to create and deploy a biological weapon highlights the emergence of a new threat, the "biohacker." "Biohacking" is not necessarily malicious and could be as innocent as a beer enthusiast altering yeast to create a better brew. Yet the same technology used by a benign biohacker can easily be transformed into a tool for the disgruntled and disenfranchised[12] to modify existing or emerging biological warfare agents and employ them as bioterrorism.

The United States Military defines a biological warfare agent as "a microorganism that causes disease in humans, plants, or animals or the degradation of material."[13] Biological agents are classified as pathogens, toxins, bioregulators, or prions.

Pathogens are disease-producing microorganisms, such as bacteria, rickettsiae, or viruses.[14] These can be either naturally occurring or altered by random mutation or recombinant DNA techniques. Toxins are defined as poisons formed as a specific secreting product in the metabolism of a plant or animal such as snake venom.[15] Bioregulators, such as enzymes and catalysts, are compounds that regulate cell processes and physiological activity. Bioregulators are necessary and found in the human body in small quantities however introduction of excess quantities can cause malaise or death.[16] Prions are proteins that can cause neurodegenerative diseases by converting the normal amino acid sequence into another prion in humans or animals.[17] The most notable prion caused the 1996 mad cow epidemic in England.

Biological agent weapons, unlike conventional weapons or other WMD, have the potential to create a runaway uncontrollable event. The damage of a bomb or artillery shell is constrained by the blast radius. The effects of chemical and nuclear WMD dissipate over time, albeit with a broad range of half-lives, environmental diffusivities, and ease of decontamination. In contrast, BW are microorganisms that upon dissemination could proliferate exponentially within a single host, linger, and spread from one host to another. Hence BW have the potential to be unbounded in both space and time. The hosts themselves serve as potent amplifiers for the agent. Common to all BW agents is the existence of a lag time between time of infection and onset of symptoms. This lag time or incubation time allows infected individuals to be asymptomatic and continue with their normal lives,[18] increasing the potential for spreading.

The Defense Advanced Research Projects Agency (DARPA) commissioned a JASON study in 2003 to examine the best means to detect, identify, and mitigate the effects of a biological agent release within the United States.[19] The study emphasized that current technologies and those expected to be developed within the next five years would not accomplish a nationwide blanket of biosensors. Instead, sensors that are currently available should be used at critical locations according to a pre-established "playbook."[20] Outside the range of these critical nodes, biosurveillance against a bioterrorism event would be accomplished through medical surveillance. The essential component of such surveillance would be the "American people as a network of 288 million[21] mobile sensors with the capacity to self-report exposures of medical consequence for a broad range of pathogens."[22] As a result of the H1N1 flu pandemic, the 2012 National Strategy for Biosurveillance further reiterates the findings of the JASON report and calls for medical biosurveillance to move beyond chemical, biological, radiological and nuclear (CBRN) threats. This expansion increases medical surveillance to examine a "broader range of human, animal, and plant health challenges,"[23] in an effort to improve early detection of emerging diseases, pandemics, and other exposures.

Medical biosurveillance, however, has an intrinsic limitation: it is entirely dependent on the self-reporting of symptoms and illnesses, which only occurs after an incubation period. This time lag is the window of opportunity for malicious activity by the biohacker aimed at increasing the damage and spread of BW effects. For instance, delayed onset of symptoms and ease of international travel enable an individual from the United States to be anywhere in the world within a few hours of BW exposure, potentially infecting hundreds if not thousands along the way. From the biohacker's disturbed point of view, a highly virulent pathogen with short incubation interval and rapid mortality may not be as desirable as a less virulent one, which will allow the infected individuals to travel greater distances before exhibiting symptoms or dying. A biohacker possesses several strategies to maximize the BW incubation period to evade or alter the medical biosurveillance network.

Many biological warfare agents are naturally occurring around the world or easily derived from plants and could be transformed by biohacking. The advent of modern technologies enables the biohacker to employ one or a multitude of strategies to increase the tactical or strategic effectiveness of a biological agent. These strategies have been broadly classifed as "Wolf in Sheep's Clothing," "Trojan Horse," "Spoof," "Fake Left," and "Roid Rage."[24]

A "Wolf in Sheep's Clothing" occurs when a biological organism or toxin is modified through genetic engineering so that it can be expressed in an active form but does not present the normal native epitopes.[25-26] In a "Trojan Horse," a biohacker maintains the epitope of a non-threatening agent but re-engineers the active component of the toxin to increase its biological threat but not the detectability. The "Spoof" occurs when a benign agent is modified to express epitopes distinctive of a known toxin in order

to trigger an unnecessary protective response by the target parties (the local, state, or federal government), while the delivering party (the biohacker) can afford to remain unencumbered. The "Fake Left" is a means to modify through selection or genetic engineering the method of transmission of an organism e.g., from fluid to airborne, to facilitate dispersion of an agent in a target population. The "Roid Rage" strategy potentiates the effects of a common virus by expressing the deadly viral components of another virus, such as altering the flu virus to express the RNA sequence of Ebola.

Any of these strategies could be used separately or in conjunction. These strategies also do not require large or sophisticated laboratories to accomplish.[27] As noted above, beer home brewing hardware may be sufficient to culture some bioagents. Moreover, a plethora of scientific data is at the biohacker's disposal. For example, research by Herfst et al published in Science in 2012 on the avian flu virus (A/H5N1) highlights the five specific genetic modifications required to transmit the virus from ferret to ferret, a highly relevant model since ferrets are susceptible to the same flu viruses as humans.[28] Such information provides a framework for biohackers to implement their strategy.

Modifications to a known pathogen could potentially render treatments useless as well as lead to increased numbers of casualties as the agent requires longer identification time. Our current understanding of bio-weapon agents is depicted in figure 1, highlighting in two dimensions the natural proximity and overlap of both physical characteristics and biological effects of known bio-weapon agents from a multi-dimensional phase space. Such proximity evokes the potential for a biohacker to blur the identification of a agent. For example, the closeness between Staphylococcal Enterotoxin B (SEB) and Ricin demonstrate an opportunity for a bio-hacker to swap either their epitopes or functions.

Due to the preponderance of available information, technology, and equipment, preventing the emergence of a biohacker is a major challenge. Foiling a successful large scale BW attack relies on the ability to *rapidly* detect the presence of and identify a biological weapon agent, thus leading to rapid treatment of those infected. Currently however there exists a significant lag time between point detection methods, positive identification and treatment.

Current Methods and Issues

As a 2006 Strategic Study on Bioterrorism highlights, "early detection of an attack or outbreak of a disease is crucial in order to confine the spread and to deploy the most effective response mechanisms, including medical countermeasures."[29] Current detection methods are categorized as either point-detection or medical bio-surveillance. Their main drawback is temporal lag, which enables a BW agent to spread. In the case of a point detector, for instance, a potential threat agent must be collected, transported, and tested, taking up to 96 hours (dependent on the proximity of sample collection and laboratory). A publication from Lawrence Livermore National Laboratories highlights that "detecting

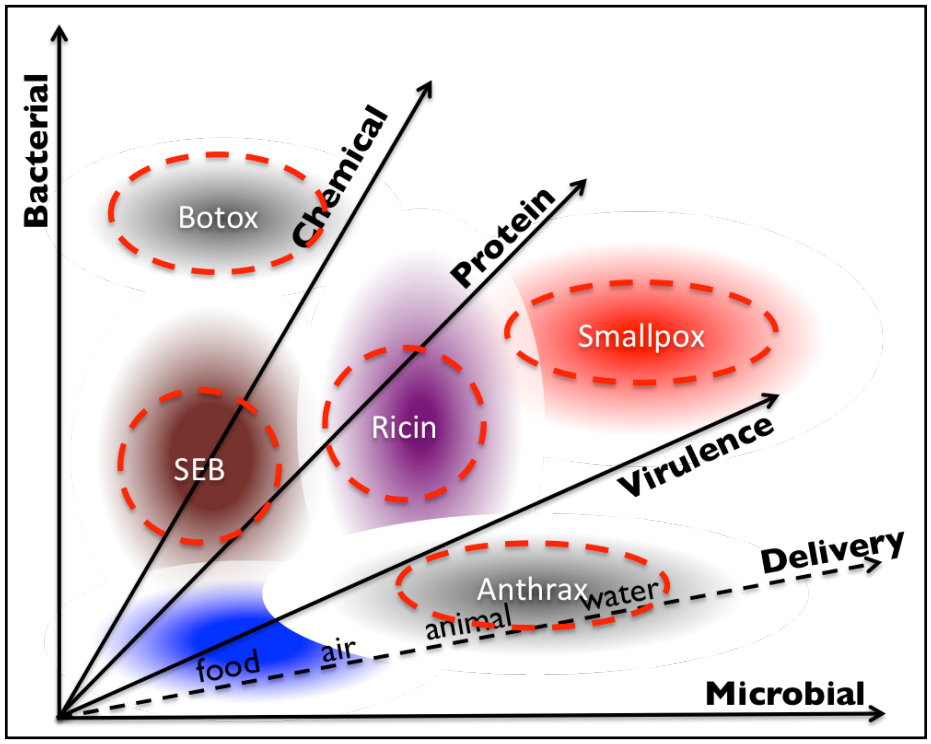


Figure 1. This phase space diagram is a 2D representation of a multi-dimensional array showing several known bio-weapon agents; Botulism Toxin (Botox), Staphylococcal Enterotoxin B (SEB), Ricin, Smallpox, and Anthrax and their relationship to each other in terms of classification. Shown but not labelled in the blue is the phase space for Russian engineered Anthrax. This figure was adapted from a presentation by John Wikswo.[30]

viable pathogens involves several labor- and time-intensive steps, such as pipetting, centrifuging, plating, and colony counting...confirmed results can take several days to obtain."[31]

A 2003 study commissioned by the Defense Advance Research Projects Agency (DARPA) and conducted by the Jason Committee highlighted that current technologies as well as technologies expected to be developed within the next five years would not provide a nationwide blanket of biosensors. Instead sensors that are currently available should be used at critical locations according to the pre-established "playbook." Outside of the range of these critical nodes, bio-surveillance against a bioterrorism event will be accomplished through medical surveillance. The critical component of medical bio-surveillance is the "American people as a network of 288 million mobile sensors with the capacity to self-report exposures of medical consequence to a broad range of pathogens."[32] The delayed onset of symptoms and detection, identification, and verification difficulties of biological agents confer advantages to the enemy, highlighted in figure 2. This determination is potentially out-dated as the technology and associated costs from 2003 have changed.

The Joint Biological Point Detection System (JBPDS), a continuous environmental aerosol monitor, is currently available for point detection surveillance. These devices collect samples over a four hour interval and then the sample is transported to a "central laboratory" for analysis; highlighting the existing best case scenario for detection to identification. Further limiting to this system is the database of pathogens. The JBPDS is only capable of examining the pathogens such as anthrax, tularemia, plague, and brucellosis that are present in the database. Hence if a pathogen is not being specifically in the designated database then no alarm for a potential pathogen is activated.

Further complicating the current point detection systems is their static nature of the chemical and biological agent database. The database is tied to looking for specific markers that are present but figure 3 indicates such markers may or may not be present. A biohacker could be using one of the previously mentioned strategies modify an agent so that the signature of the agent as seen through its gene expression, or genotype, is different yet the resulting agent and effects, phenotype, are the same.

One proposed method to overcome current lag time in the detect-to-treat scheme is through the employment of a mobile polymerase chain reaction (PCR). However, PCR suffers from two major limitations. The first is the critical requirement of maintaining sample purity, as PCR is extremely sensitive to contamination and would require a clean-room environment (e.g., laminar flow hoods) that is difficult to implement in the field or combat environments. Another is that PCR would require a database of known pathogens for identification, and if the sampled pathogen is not in the database or has been modified by a biohacker then little to no information is garnered.

Additionally, PCR and other current detection methods seek a unique identifiable characteristic of the pathogen such as an epitope. These markers however could be transiently expressed by a biohacker

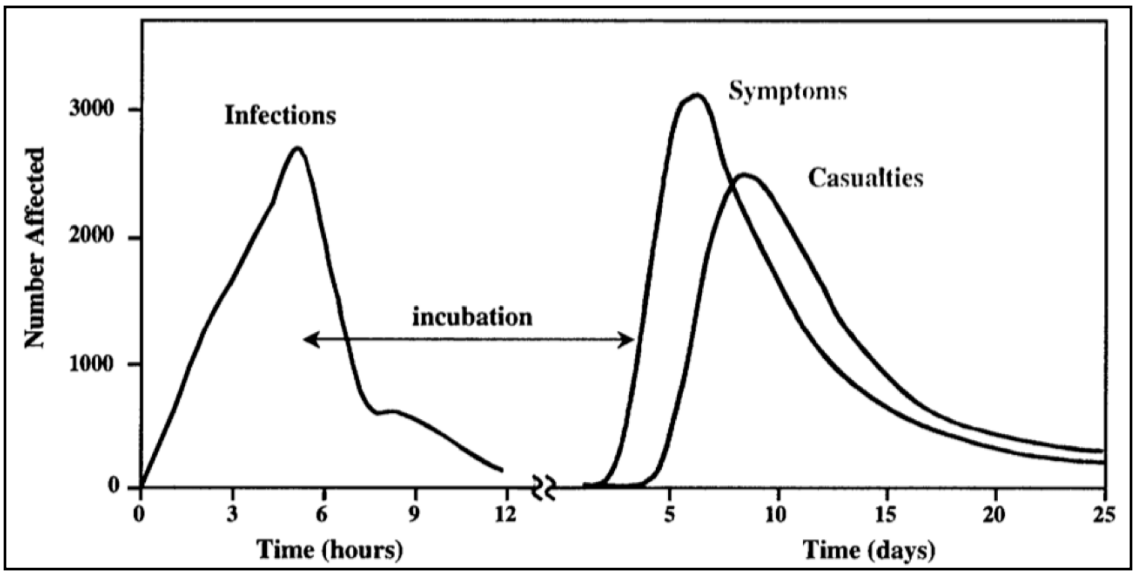


Figure 2. Nominal timeline for a bioterrorism event. There is a substantial lag time between the time of infection and the onset of symptoms, and development of full-blown disease. Adapted from Figure 2 of the Jason Committee report.[33]

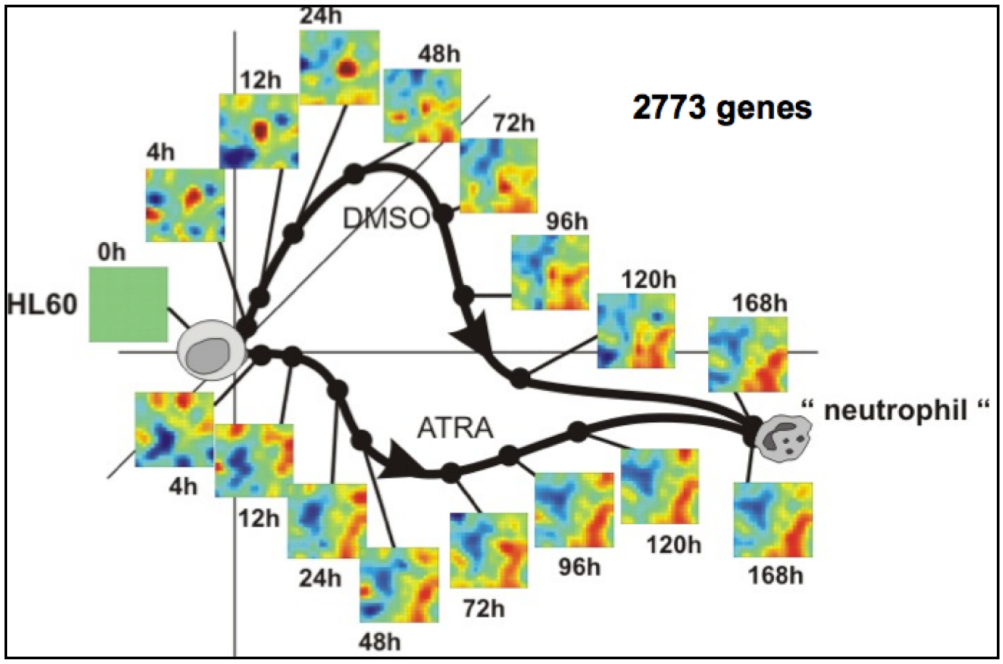


Figure 3. Path to the neutrophil as seen through Gene Expression Dynamics Inspector (GEDI) highlights that under different conditions there are two separate means and 2773 genes that transform a HL-60 cell into a neutrophil. It is such a path that demonstrates static observation may lead the researcher to missing the crucial endpoint. Additionally this path highlights that a "bio-hacker" could modify the biological weapons agent to avoid detection and yet the end result is the same. Adapted from Huang *et al* (2005) and Eichler *et al* (2003).[34-35]

and so identification at a single time point could result in misidentification. Figure 4 is a temporal series following the release of a bio-weapons agents that has been modified by a bio-hacker. Through clinical bio-surveillance, a large number of cases with flu-like symptoms would present themselves at hospitals and clinics and a growing number of associated casualties would occur a few days after the release (B). Currently, identification is accomplished using a standard solid-phase peptide assay that identifies receptor binding epitope that is unique to Anthrax (C) and is being expressed by the casualty causing agent. Treatment is initiated using antibiotics such as Cipro. Treatment however does not abate the worsening symptoms which progress from flu-like symptoms to pustules. The epitope that led to the initial Anthrax identification is also no longer visible. The Cipro treatment had no effect on the worsening smallpox symptoms as physicians began seeing these physical symptoms emerge (D). Missing within the static identification techniques such as DNA sequence or epitopes presentation are the actual biological effects of the BW agent.

Though the potential to identify biological agents through their DNA exists, the practice is not perfect and such methods ignore other components that could be used for identification. Work by Eklund *et al* (2009), which measured the rate of glucose consumption, oxygen concentration, lactate changes and the acidification rate, highlight that changes are not stochastic but, rather, effects are dynamic and different between the BW agents. Figure 5 highlights heterogeneous response of different BW agents, Ricin and Anthrax, on different cell types in time. More specifically, it is a dynamic response that potentially creates unique signatures for an agent that can be measured in real-time. This level of quantifiable difference poses the question if BW agents can be identified through their dynamics and heterogeneous effects on cells and highlights the significant amount of information available for understanding the BW agents mechanism of action modified by a "bio-hacker."

Dynamics and Heterogeneity

The impact of BW agents on human health directly percolates from organ failure and tissue destruction, but is ultimately defined by the toxic effects on cellular functions, with the most severe being cell death. Measuring response to biological agent at the cell level is therefore paramount to assessing BW potential damage and subsequent treatment strategies. Figure 6 highlights the centrality of the cell between small molecules and organism. Current assays suffer from limitations in measuring two critical aspects of cellular response to perturbations: dynamics and heterogeneity.

Dynamics refers to the rate of change in the size of a cell population. Current assays do measure change of size in a perturbed cell population, but not in terms of rates. Rather, they are largely based on a single fixed time-point measurement (at 72- or 96-hour after treatment). This static metric is quite useful to estimate potency of a toxic substance, but dynamics of response remain vague at best because they are

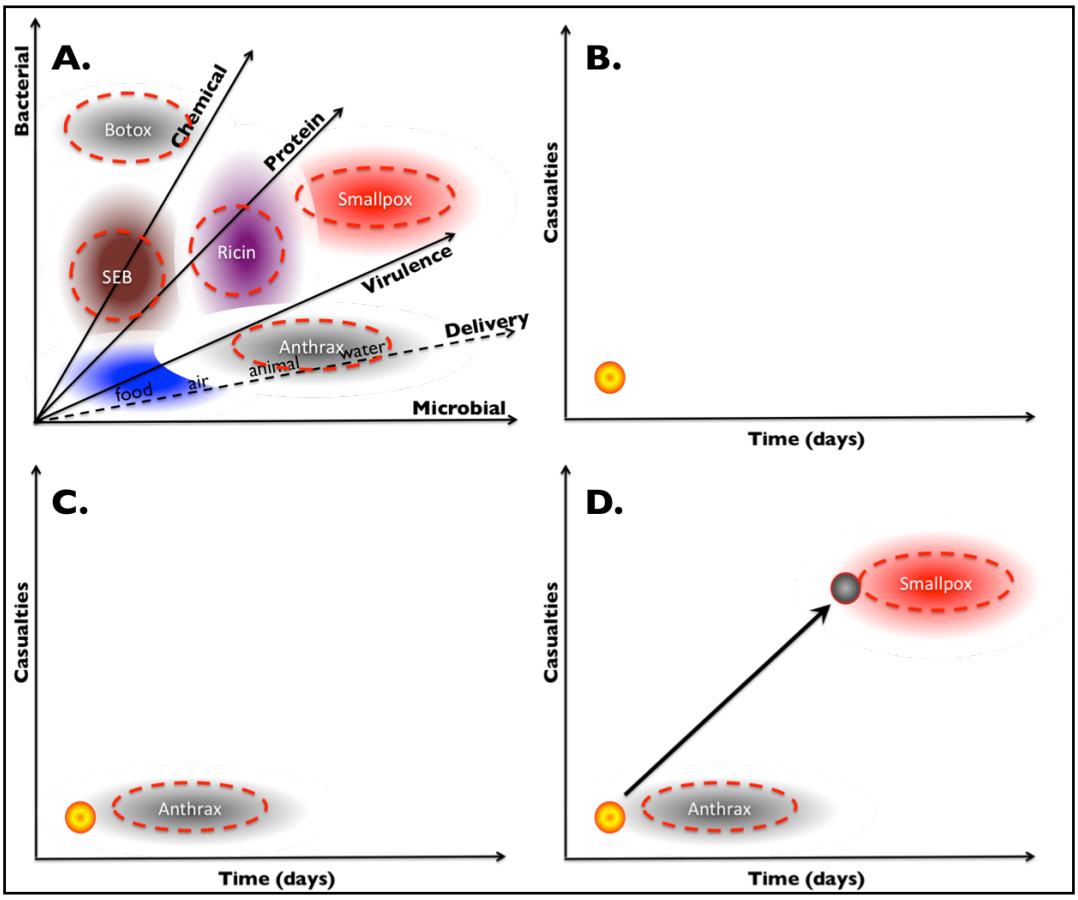


Figure 4. The consequences of misidentification as a "bio-hacker" modifies the BW agent through transient expression to be identified as a different agent leading to the wrong treatment.

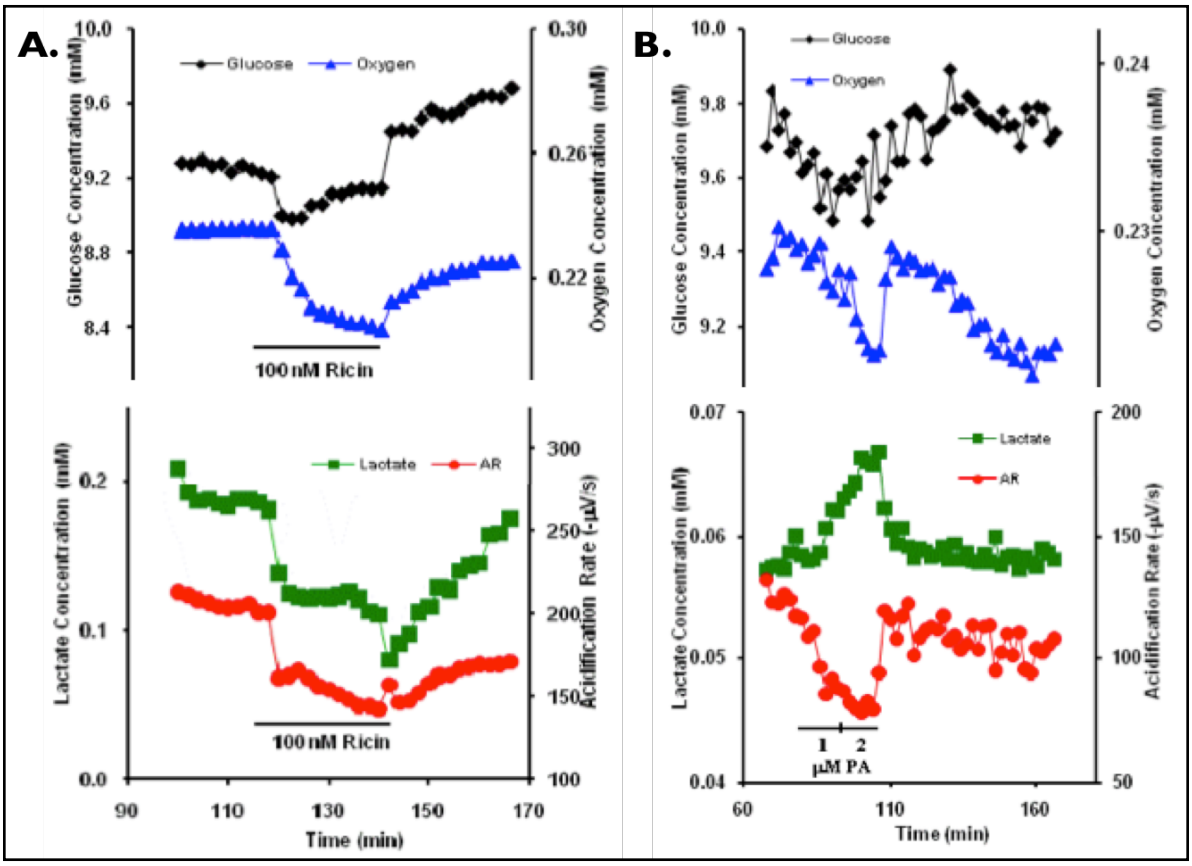


Figure 5. Glucose, Oxygen, Lactate, and Acidification response to (A) 100nM Ricin in neuroblastoma cells and (B) 1 and 2 uM Anthrax in macrophages. Adopted from Eklund *et al.*[36]

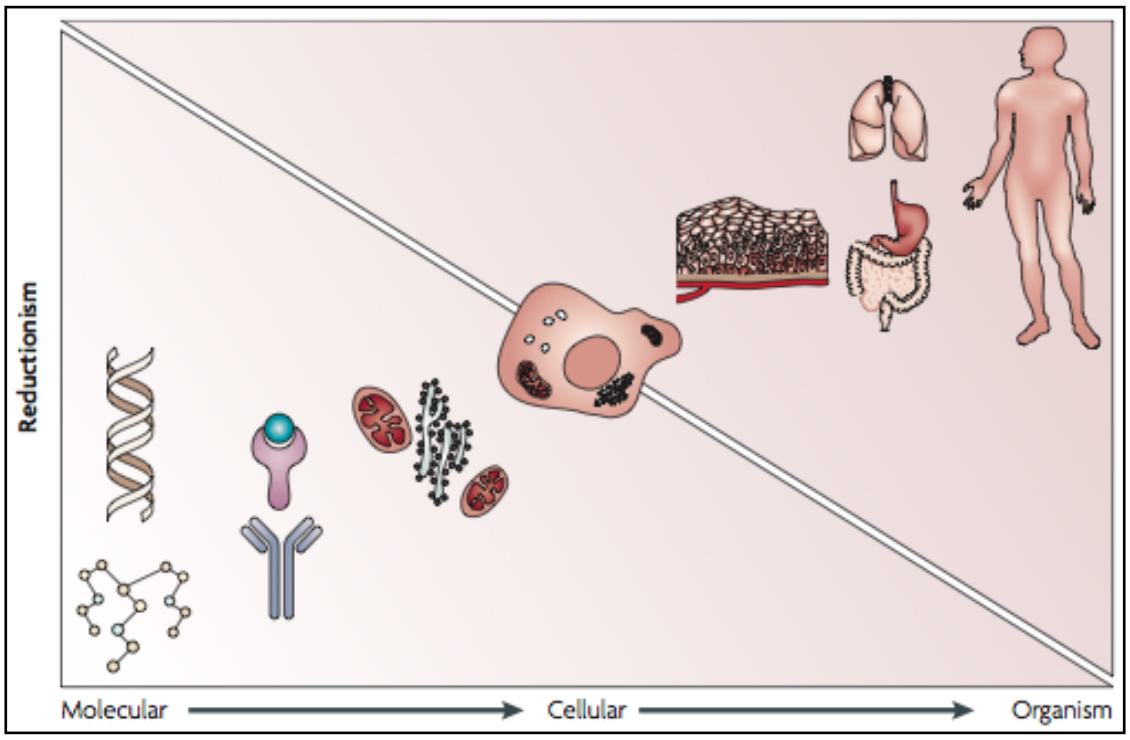


Figure 6. Mutli-scale visualization of molecules to organism. Highlighted at the center of the figure is the role of the cell which is the link between small molecules and tissues, organs, and the organism. Adopted from Anderson and Quaranta, 2008.[37]

by necessity based on the unverified assumption that the measured change in population size will carry on indefinitely. This is a grave limitation because accurate estimates of rate response are absolutely necessary for predictive models for depth and spreading of the effects of a BW agent.

Heterogeneity exists at all biological scales from genes to cells to populations. In a broad context, understanding variation has been advocated in genetics and proteomics to improve therapeutics and vaccine production. Heterogeneity also occurs in the cell-to-cell response to BW agents, even when cells are relatively homogeneous as in a differentiated tissue (e.g., epidermis or liver). This variation becomes key to the tissue, or organ, damage depth and capacity to recover. Current assay generally measure response of a perturbed cell population as averages. For example, inhibition of population cell growth can be due to a varying combination of individual cells that enter a reduced division rate, quiescence, or death, in response to a perturbation. By taking an average measurements and disregarding deconvolution in actual cell fates, precious information on BW agent effects is lost. Quantification of single-cell heterogeneity that shapes the overall cell population response to a BW agent would provide potential means of identification of that agent.

The centrality of the cell in the scale of systems biology has brought forward advanced methods to detect and accurately measure in continuous time phenotypic changes at the single-cell level. This emerging technological capability represents a pivotal opportunity to measure both the dynamics and heterogeneity of response to BW agents, and reclassify their effects in terms of accurate predictive mathematical models based on measurements taken at the time of a BW exposure, prior to the presentation of symptoms.

Historically, the caged canary has been synonymous with early warning mechanisms. For example, in coal mines dangerous gases such as methane and carbon monoxide would kill the canary first, providing an early warning mechanism to alert personnel to evacuate the threatened area. In the 21st century, by reducing the scale of the canary from an organism to a cell, it should be possible to rapidly forecast the magnitude of the threat posed by exposure to a BW agent by measuring dynamics and heterogeneity of single-cell cell response.

The BioDigital Canary

The BioDigital Canary (BDC), figure 7, is a next generation bio-detector that incorporates multiple, orthogonal (*i.e.*, mutually complementary) quantitative measurements of cellular stress using state-of-the-art time-lapse live-cell fiber optic microscopy, mass spectrometry, and NMR spectroscopy that combined measure nuclear morphology, division, apoptosis (cell death) rates, migration/motility, lipid changes, choline, and glutamine. Combined, the dynamic and variable changes induced by a BW agent are unique to that agent and such signatures lead to real-time identification of the agent.

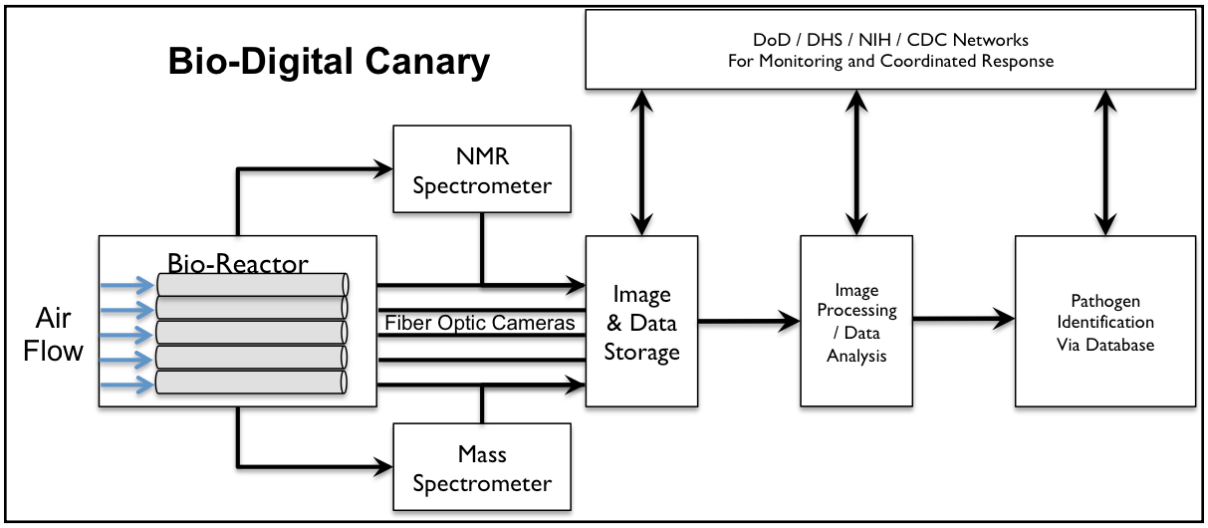


Figure 7. Schematic of the Bio-Digital Canary. Highlighted in the schematic are the critical components of the BDC to include the bio-reactor, measuring devices, and the data handling components.

In the BDC, cultured human cells act as a biological sensor. While the idea of using cells as biosensors is not new, its implementation in the field has become feasible only in the last few years. Recent advances in cell culture have enabled automated, self-feeding, stable 3D cell culture systems. A 2005 study demonstrated that cells cultured in bioreactors, similar to the Bio-Digital Canary reactors, can be maintained for over a year while exhibiting minimal morphological changes;[38] thus supporting deployable and sustainable applications which require minimal maintenance.

The next-generation bioreactor provides reliable identification of cellular stress based on changes in cell morphology, motility, and cell death/proliferation rates both across the system and among individual cells in response to exposure to a biological agent. By quantitatively characterizing these multivariate alterations, the bioreactor generates a "signature" that is unique to a particular agent. The signature is simultaneously and continuously compared to an existing database for rapid identification. In case of unknown/emerging agents, preliminary assessment of potential mechanism of action is feasible. The cell bioreactor, detectors and associated software for data analysis comprise a self-contained system that is adaptable and scalable.

The data detectors included in the BDC (i.e., lapsed fiber optic microscopy, mass spectrometry, and NMR spectroscopy) are each based on well established technology, but never before deployed in an integrated, mutually supportive manner. The crux of the BDC, as currently designed, resides in the analysis of the time-lapsed image microscopy which must not only accurately quantify the morphological features and changes of those features for each cell, but also track the cell and its lineage over time. The incorporation of time-lapsed image microscopy also provides a link between the population of cells and the behavior of that population at the single cell level.

The following work illustrates a novel approach to cellular tracking using time-lapsed image microscopy. This work will explore previous methods highlighting their successes and limitations and introduce a novel integer programming based algorithm for image processing and cellular tracking.

CHAPTER II

REAL-TIME ANALYSIS OF TIME-LAPSE LIVE-CELL MICROSCOPY

Technological advances have led to the development of high-throughput microscopes that are capable of capturing high resolution images over an extended period of time. Such technology enables researchers to study morphological or molecular features (e.g., *via* fluorescent protein reporters) over time. As Georgescu *et al* (2006) highlighted in 2011, an important limitation of time-lapse live-cell microscopy is the ability to track cell populations through time for the purpose of exploring cell ancestry.[39] This is largely due to the fact that current cell tracking algorithms are based on particle tracking, and do not consider cell division. Live cell populations in culture exhibit a doubling time of approximately 20-30 hours. Accurate identification of siblings is absolutely required for cell ancestry tracking with low error.

Tracking cells manually is extremely time consuming, tedious, and error prone. For example if an experiment lasts 7 days and images are captured every 15 minutes, there are 672 images with approximately 100 cells in the initial image. Consequently, a minimum of 67,200 cells must be tracked over the course of the experiment. This cell quantity also needs to take into account variability of cell fate in response to perturbations, i.e., cells that are dying and cells that are dividing as well as their movement. Hence the task of manual tracking is not trivial.

Integer Programming

Integer programming, also known as the "Traveling Salesman" problem, has been around for centuries, the concept being that a salesman with a list of cities and known distances must determine the shortest possible route that enables the salesman to visit each city exactly once and return to the original city. This "optimization problem" has been applied to a plethora of problems including cell tracking.

Al-Kofahi *et al* (2006) reported using integer programming to develop an automated process for tracking proliferative lineages as well as cell motility in murine neural progenitor cells. This method was the first of its kind to incorporate morphological features for cellular tracking. By measuring morphological features such as size, shape, location, motility, and migration, Al-Kofahi *et al* calculate a probability distribution for each option of a specific cell in a subsequent image. The integer programming function attempts to minimize probability density function which in turn generates the path of a cell over time.

These authors tracked murine neural progenitor cells through time to test their integer programming algorithm. Though the automated nature of Al-Kofahi's *et al* work greatly diminishes the

time required to manually track cells, the algorithm is not accurate nor scalable due to the low number of cells (ten) being tracked in each image and the unverified assumptions used to build the probability distribution functions. Consequently the exact method used by Al-Kofaji *et al* was not scalable to images with high cell density.

Large Datasets

Large datasets can be viewed in two ways: either large numbers of cells on the images, or a large number of images that require processing. Large datasets are computationally expensive and require time for processing. Subsequently, development of automated tracking techniques is accomplished using smaller datasets, the assumption being that if the items are tracked using a few cells over a short period of time then the principles are easily scalable to larger datasets. This assumption however is incorrect since options drastically increase as the dataset become large. Tracking 10 cells over 10 images might provide 100 potential options over the course of the image set. However, it is not the same as tracking 100 cells over 100 images which would provide 10,000 data points.

If the integer programming is set up to use every object in the subsequent frame as a potential match for a previous object, then for each image generates 99! by 200 matrix for integer programming. Such large matrices are computationally and time expensive.

Assumptions

Pivotal to Al-Kofahi *et al*'s[40] method is the probability distributions for their parameters. Their probability distribution of the changes in morphological parameters is based on a truncated normal distribution which ultimately penalizes cells that had little to no change. This change is calculated for parameters such as distance travelled and changes in morphology. The significance of this assumption will be illustrated later when highlighting changes in "high-confidence" cell tracks which follow a half-normal distribution and not a normal distribution.

CellAnimation

CellAnimation is a modular framework for microscopy assays developed by Walter Georgescu at Vanderbilt University and published in 2011.[41] The core program of CellAnimation is its tracking capabilities. For tracking and subsequent identification of mitotic events, CellAnimation uses a modified nearest neighbor algorithm with heuristic thresholds. The algorithm incorporates the distance travelled and change in cellular area to correctly identify cells from one image to the next.

Prior to tracking, images are imported into the assay and undergo a rapid segmentation program that identifies cells using a watershed algorithm and object size. The graphical user interface provides

ease of use and the algorithm is capable of handling large datasets as well as a platform for correcting tracks. The underlying modified nearest neighbor tracking algorithm used not only the linear distance between potential objects but also incorporates a change in cellular area parameter. This parameter uses a threshold to ensure there are no drastic (> 20%) changes in area of a cell from one image to the next.

The CellAnimation algorithm has one significant limitation: its ability to detect mitotic events is not optimal. As stated above, correctly identifying siblings is paramount to the accuracy of long term tracking and cell ancestry recognition.

Detecting Mitotic Events

CellAnimation uses a series of "if/and" classifier statements to determine if a cell is mitotic. These statements are based on area, eccentricity, and minimum time for cytokinesis. The steps of the CellAnimation process, highlighted in figure 8, are efficient but frequently fail to detect mitotic events. Several test runs demonstrated a 50% rate of missed mitotic events. This is unacceptable because one missed mitotic event introduces a wrong tracking trajectory, which generates incorrect cell lifespans and ancestry. That is, instead of being considered siblings, of the two newborn cells one continues its lifespan to an unrealistic length, the other is considered an altogether new cells entering the image field.

An example of a missed mitotic event using CellAnimation is highlighted in figure 9. The left and right images are from frame 21 and 22, respectively, of a control experiment using MCF10A cells. The numbers indicate the CellAnimation tracks. In frame 21 two cells are identified as 53 and 71, respectively. In the next frame, following a correctly identified mitotic event, cell 71 is properly split into cells 147 and 2124. In contrast, as a consequence of a missed mitotic event, cell 53 continues on its lifespan and cell 146 "magically" appears. Such missed mitotic events are further illustrated by the cartoon in figure 10.

The disruptive effect of missed mitotic events is apparent in the changes in cellular lifespan. A correctly detected mitotic event will yield two new lifespans whereas a missed mitotic event yields one new, short lifespan and one long continued lifespan. This information subsequently yields incorrect cellular ancestry and diminishes the value of the automated microscopy data. Preliminary examination of the CellAnimation mitotic events yielded missed events nearly 50% of the time. These missed events tend to occur as cells become more confluent.

Attempts were made to correct the mitotic event portion of the CellAnimation algorithm. However, the structure of the code itself proved to be a major hindrance. Additionally, there was no guarantee that modifications to the heuristic method in CellAnimation would yield better detection of mitotic events.

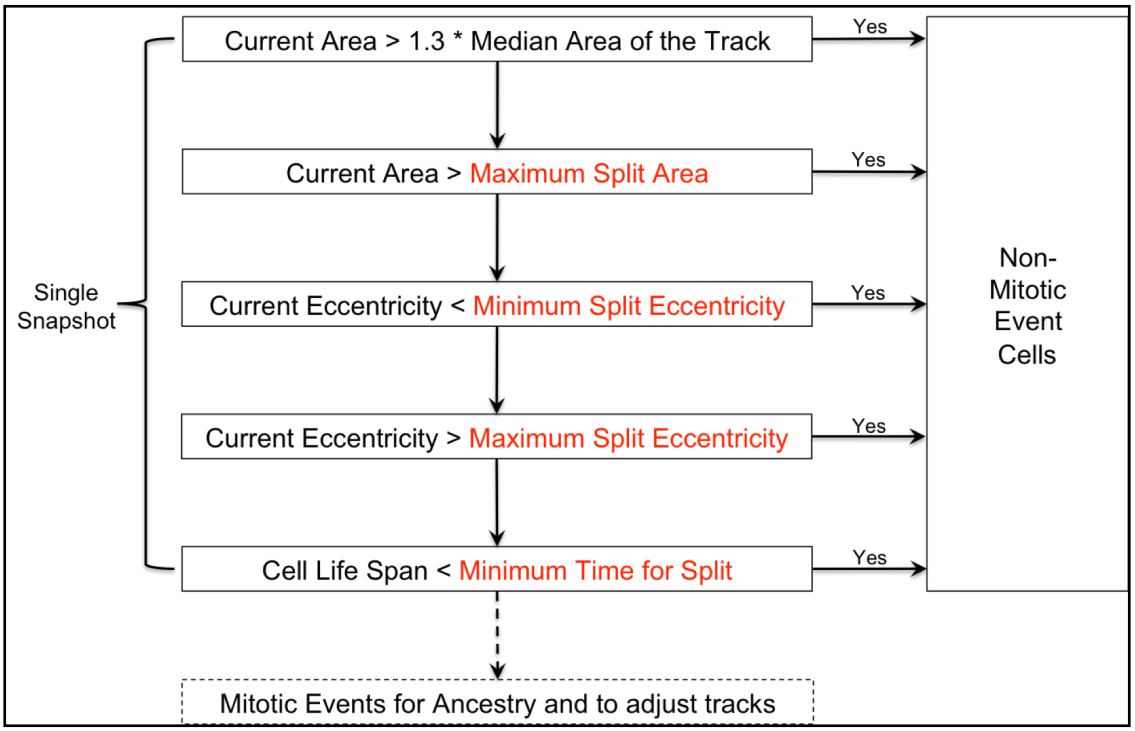


Figure 8. Steps for CellAnimation's means to detect mitotic events. The diagram highlights the "if/and" classifier statements used by CellAnimation to detect mitotic events.

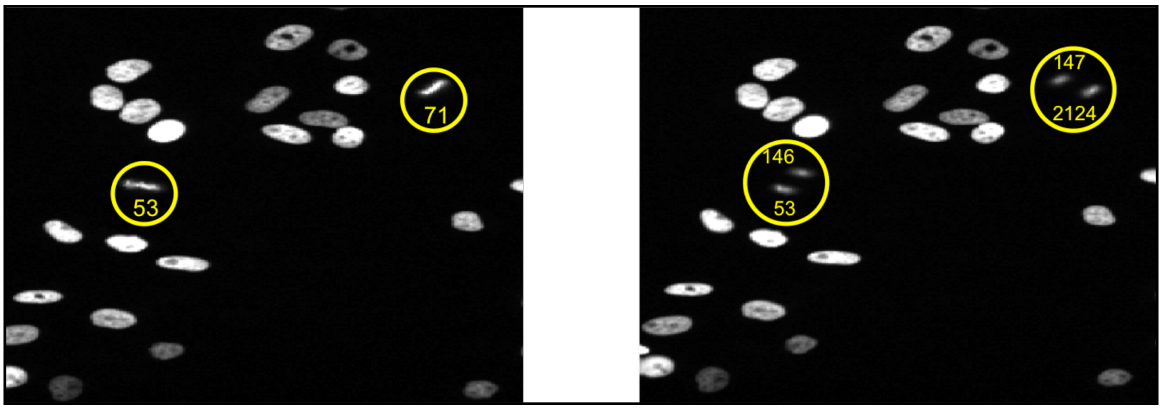


Figure 9. Example of missed mitotic event in sequential frames. Cells labelled 53 and 71 divide into two daughter cells. The labeling for the daughter cells, 147 and 2124, reflect two new tracks while the daughter cells of 53 are 146 and 53. The cell 53 track would thus have a longer lifespan than is accurate.

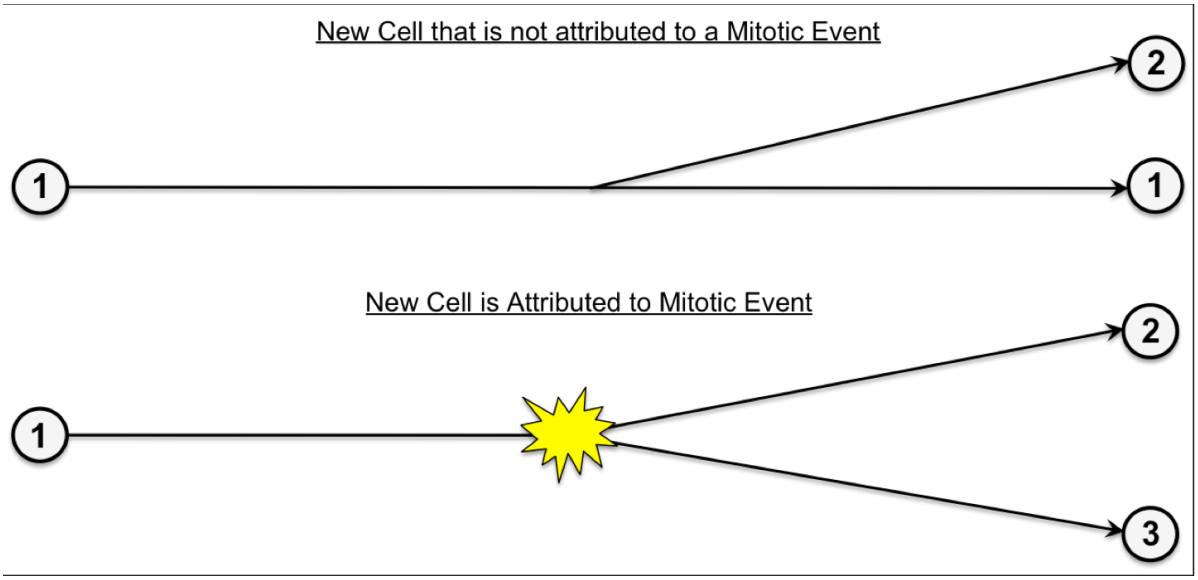


Figure 10. Cartoon of missed mitotic events in CellAnimation highlighting the continuation of one cell while one cell simply appears.

CHAPTER III

A NOVEL APPROACH

Concept

Cells exhibit the potential to undergo different fates in response to perturbations, such as BW agents. Cells are capable of dividing, dying, or entering quiescence depending on the signals and conditions of their microenvironment. The diversity of cell fates is a major challenge to tracking cells over time.

Different cell fates are compounded by changes in motility, cell morphology and division rates, highlighting the challenge of accurately tracking cells particularly on a large scale over extended time periods. To harness this data types, we sought to combine the power of integer programming with the ability to handle large datasets. There exist three distinct and novel components to our approach: 1) segmentation is revised using a user generated training set for better identification of cells, 2) a k-nearest neighbor algorithm is used to generate "high confidence tracks", and 3) assumptions about the tracks are automatically scrutinized within the program.

The schematic in figure 11 highlights these three distinct components of our Time-Lapsed Live-Cell (TL2C) tracking. The first process is image segmentation, the second component assigns values to objects and tracks them over time, and the third is the review process to visualize the output.

All segmentation and tracking was accomplished using the 2012b MathWorks MATLAB (MathWorks, Natick, MA) platform. The MATLAB platform enabled ease of development by the incorporation of functions already pre-established. Processing time for completing a full image stack increases with the number of images, number of cells per image, and decreases with the processor speed.

Segmentation

The segmentation code was developed by Shawn Garbett and Sam Hooke in the Quaranta Lab at Vanderbilt University. It is a five step process that uses a combination of MATLAB scripts, R code, and a graphical user interface for detailed segmentation review. R is a free statistical computing and graphics environment developed by Bell Laboratories. R is compatible with a variety of platforms including UNIX, MAC, and Windows.[42]

The first step in the segmentation process is the "Naive Segmentation." Using a script file titled, "LocalNaiveSegment" in MATLAB the image files are loaded and then processed through the "NaiveSegment" function. This function segments using several functions: "Top Hat," "Noise Threshold," "Background Threshold," and "Fill Holes." The "Top Hat" function removes background objects bigger

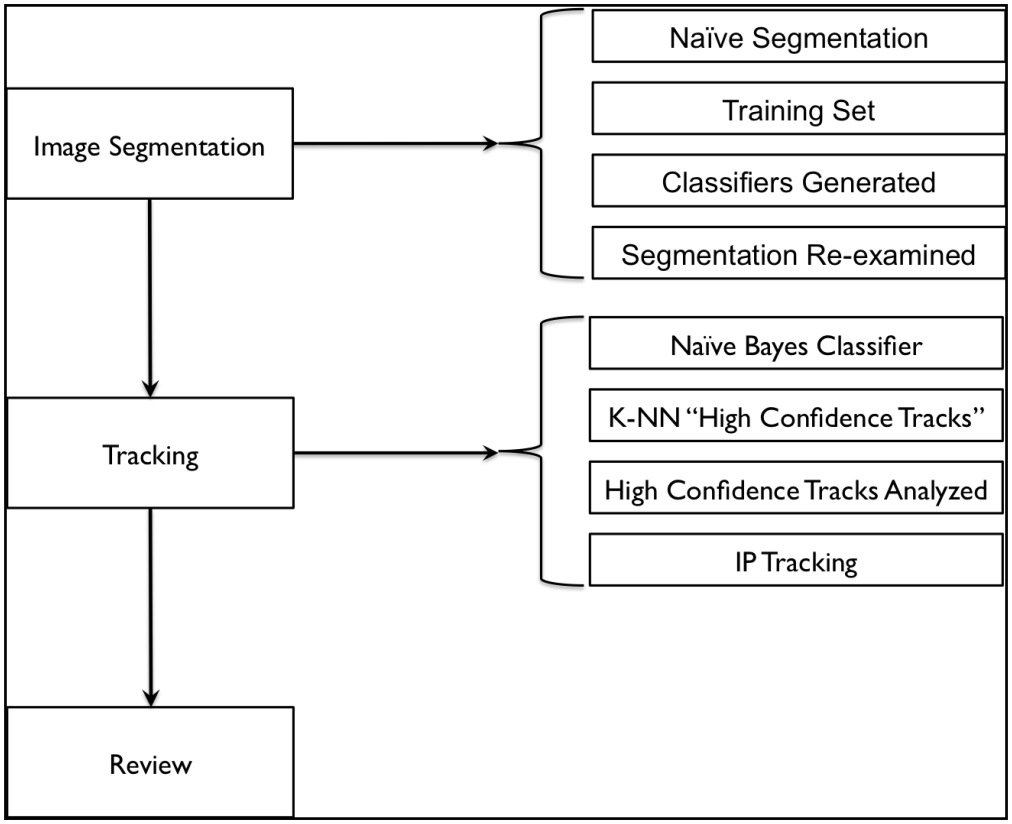


Figure 11. Schematic of over-arching processes and sub-processes of the proposed novel Tracking Algorithm.

than 50 pixel areas. Noise Threshold then uses a 3 pixel circle to remove noise. Background Threshold uses a 0.2 level which means that 20% below normal is off and 80% is real. Finally the image "holes" generated by the removing of noise and background thresholding are filled in accordingly by "Fill Holes". This first pass segmentation is completed for all images in a stack and the output data is saved as both a comma separated file and a MATLAB supported ".mat" file. The output data also includes the physical properties of the cells within each frame. These properties include centroid location, area, perimeter, major axis length, minor axis length, eccentricity, convex area, solidity, intensity, and filled area.

Using the "Segment Review" graphical users interface function (figure 12), the user is then able to review how well the Naive Segmentation function processed the image stack. The images are loaded into the interface and the user is then able to visualize an image and select individual cells/objects and relabel them. The cells/objects at this point are determined by the user to be either "nucleus," "mitotic," "under-segmented," "over-segmented," or "debris." The term under-segmented refers to two or more cells not segmented properly so that an aggregate of cells is shown as a single object. The complement to under-segmentation is over-segmentation where a single cell is separated into two or more cells. During the segment review process the user is expected to filter through images and select at least 200 of each category. However, it is often difficult to find that many mitotic cells within a image stack, depending on experiment conditions. Upon completion of the review process the data is saved as a comma separated file and includes the cells / objects physical characteristics as well as the classifier determined by the user in the review.

The comma separated file is then analyzed in R through Perl import script. Within the R environment the Perl script extracts the segment review data in order to generate classifiers using the cell's / objects physical characteristics.

Following the model generation in R, the images are reprocessed using three functions, "LocalNaiveSegment," "LocalFinish," and "LocalGMMSegment." The local naive segmentation reprocesses the images through the same NaiveSegmentation function and the cells / objects are identified once again. The local finish method then uses a watershed algorithm based on a transformation distance to re-segment the objects that are under-segmented before the re-segmented objects are re-identified using the "Top Hat" function.

Finally the "LocalGMMSegment" function begins to process the images. This function measures the mean and standard deviation of the cell / object for the following morphological features: area, eccentricity, minor axis length, and solidity for the objects in the training set derived from the user segment review. This data is used to create a Gaussian mixture model (GMM). Then each object in the image is reviewed separately and the physical properties (area, eccentricity, minor axis length, and solidity) are measured. Likelihoods are then generated using these physical parameters to determine the

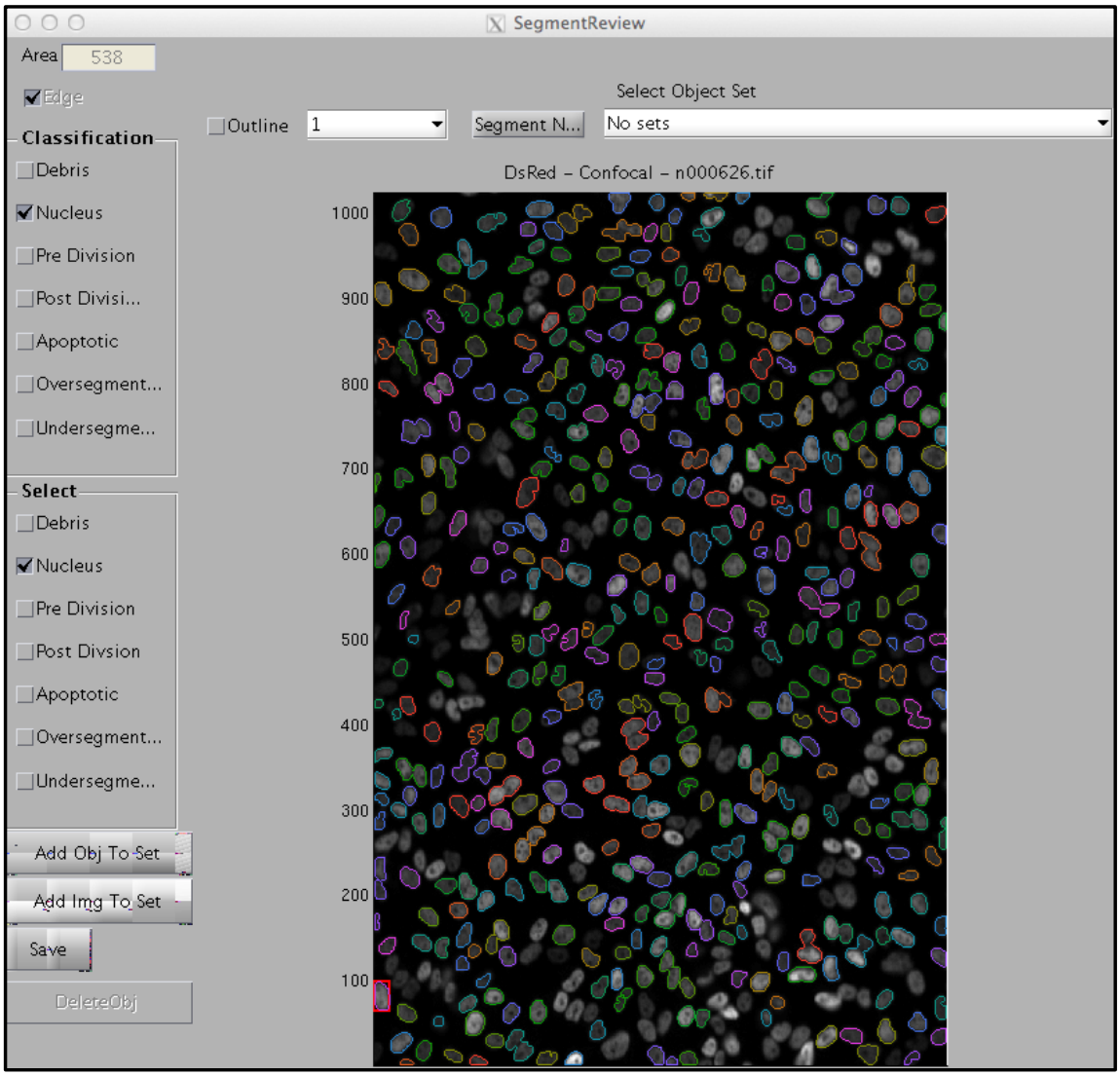


Figure 12. Example of an image being undergoing the Segment Review function where the user can generate a training set of what is correctly segmented, what is under-segmented, over-segmented, debris, and dividing cells.

cell category (debris, nucleus, and mitotic). Once a likelihood is determined, the cell is re-segmented if necessary and all of the cell parameters are measured and added to the output file. Upon completion of processing of all objects in all images, a .mat file is generated for each image in the stack. These .mat files and the embedded information can then be used to track cells throughout an image stack.

Tracking

Following segmentation, the corresponding .mat files are imported by the tracking algorithm function for processing. Each .mat file is read in, and data is arrayed into a matrix by image number and object number. Each object has corresponding morphological characteristics that are also arrayed in the matrix. These characteristics include area, perimeter, major axis length, minor axis length, convex area, eccentricity, intensity and solidity. Additionally, the classifier generated by the manual portion of segmentation is also imported into the tracking function and used to determine if a cell is mitotic.

The imported data is then processed through a series of functions labelled: Naive Bayes Classifier, High Confidence Tracks, Potential Matches, Probability Density Function Calculations, Integer Programming Array Construction, Binary Integer Programming, Match Indexing to Track Generation, and Fractional Proliferation. Each of these functions is highlighted further below.

Naive Bayes Classifier

In simplistic terms, a Naive Bayes Classifier is a supervised machine learning technique, *i.e.*, a probabilistic classifier based on applying Bayes' theorem. This theorem is highlighted in equation 1. The classifier is based on the concept that states are mutually exclusive and exhaustive, i.e., at least one state must occur and no two states can occur at the same time. Hence, in detecting mitotic events, a cell is classified as either mitotic or non-mitotic given the parameters of A which in this application are physical characteristics of the cell.

The Naive Bayes Classifier uses the concept presented in equation 1 in MATLAB and functions in two steps. First the algorithm conducts supervised learning by combining known data (area, intensity, major axis length, minor axis length, solidity, and perimeter) with known responses into a model ("dividing" or "non-mitotic"). This model uses the separate physical parameters provided to calculate the probability that an object is in a certain mutually exclusive category, such as a letter or number, based upon the values of the parameters and the classification. Ideally, the parameters are 'separable'; i.e., the two states do not have overlapping values. This concept is highlighted by the cartoon (figure 13) and illustrated further by the MCF10A data in figure 14. The model, or predictive values based on the classifier and probabilities, is then applied to an unknown dataset with the same parameter categories to determine the state of the unknown.

$$\Pr(B \mid A) = \frac{\Pr(A \mid B) \times \Pr(B)}{\Pr(A \mid B) \times \Pr(B) + \Pr(A \mid \overline{B}) \times \Pr(\overline{B})}$$

Equation 1. Generalization of Bayes' Rule. This equation specifies the probability of B given A where B is the prevalence in a population. Adopted from Rosner's Fundamentals of Biostatistics.[43]

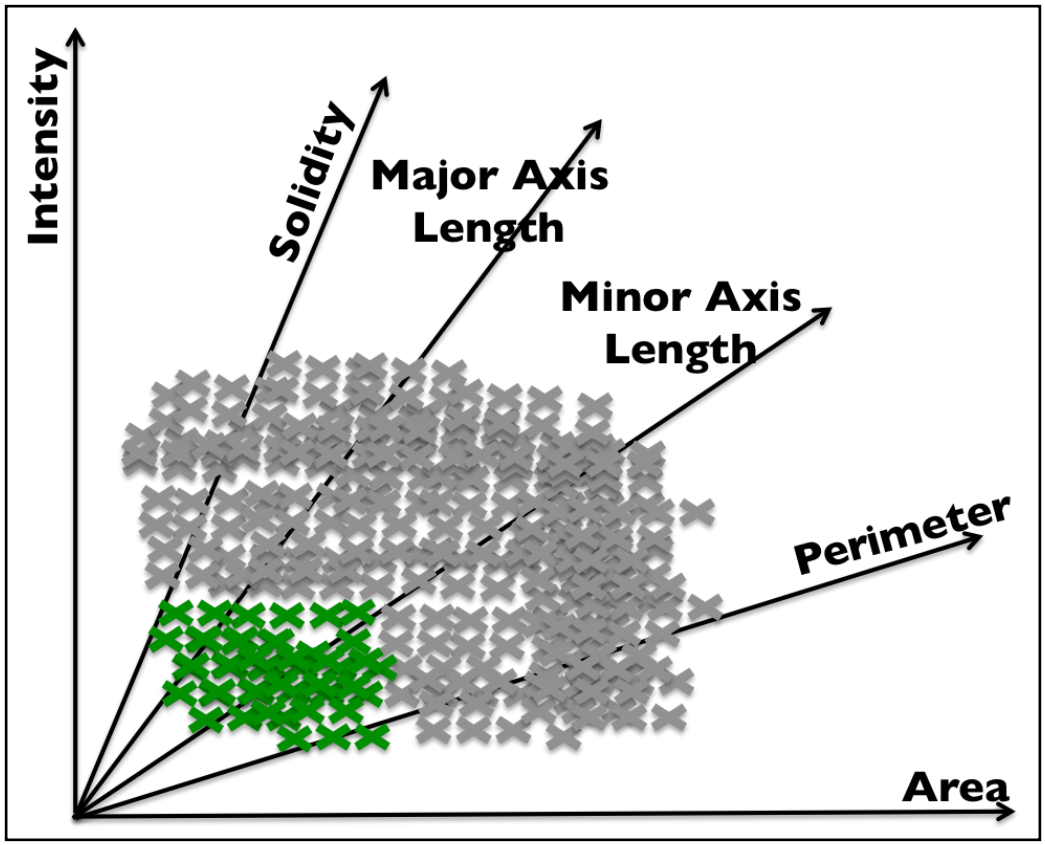


Figure 13. Cartoon representation of physical characteristics and cell phase. The green "x" represents a cell undergoing mitosis while the grey "x" represents a non-mitotic cell. This is meant to be a snapshot of cell cycle and characteristics and is not indicative of quiescent cells.

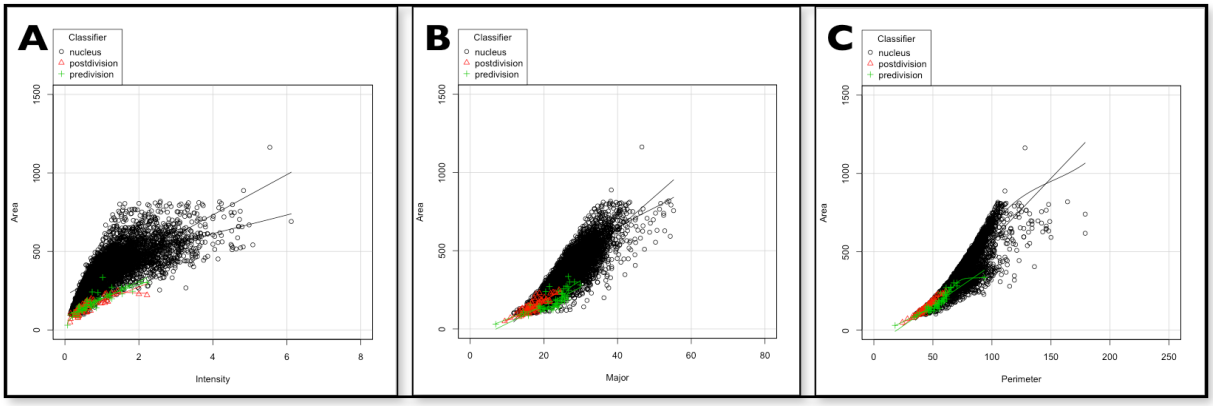


Figure 14. Comparison on the morphological parameters; Area versus Intensity (A), Area versus Major Axis Length (B), and Area versus Perimeter (C). This data is derived from the manually reviewed segmentation and the black circles are those labelled "nucleus" while the green and red dots represent cells that are mitotic; either pre- or post-division. This figure highlights that certain parameters provide better separation (A and B) for dividing and non-dividing cells whereas perimeter overlaps between the categories.

This process classifies objects based on the probability that said object is similar to another object that is already classified. In terms of cellular tracking the Naive Bayes Classifier is used to determine if cells in an image are mitotic, either pre-division or post-division. This determination is based on the probability distributions generated on the cells morphology when comparing cells that were identified during the manual segmentation to be either mitotic and segmented properly. For the Naive Bayes Classifier used in the tracking algorithm there are 6 specific morphological features used: area, major axis length, minor axis length, eccentricity, convex area, and intensity. These parameters were selected using a general linear model (GLM) and calculated in R using the binomial. These selected parameters had a p-value > 0.001 as indicated in figure 15.

The Naive Bayes Classifier seeks to exploit the dynamics of cell morphology as the cell progresses through the cell cycle. Cells tend to be smaller and more elongated near the point of mitosis so by using the data generated from Segment Review, it is possible to identify the population of cells in an image that would potential be mitotic.

This process within the tracking function analyzes every cell in every image and labels them either "dividing" or "nucleus." These designators from the Naive Bayes Classifier do not actually mean a cell is dividing or not; rather it identifies the cell to be examined for a possible mitotic event during the binary integer programming. The classifier tags generated within each image for each cell are maintained in the matrix throughout the tracking process.

High Confidence Tracks

The High Confidence Track function in the tracking algorithm uses MATLAB's internal nearest neighbor algorithm to generate "high-confidence tracks." The nearest neighbor algorithm categorizes the query points based on their distance to points in a training set. The distance is calculated using the Euclidean distance (equation 2) given an mx-by-n data matrix x. The cells in the query set are then indexed to the known data from the previous image. The cells with the shortest distance are the nearest neighbor and linked to form the initial tracks.

A high confidence track is crudely defined as a cell that survives the length of the image stack without dividing or dying. The nearest neighbor algorithm compares the changes in the cell's X and Y coordinates as well as the changes in the cells area and eccentricity as the images progress from t to t+1. After each comparison the cells are indexed to assign cells to corresponding tracks and ensure there is no redundancies in identifications within the same frame. Once the indexing is complete the nearest neighbor progresses to the next frame until the last frame.

Once complete, the nearest neighbor tracks are sorted and the cells that meet the "high confidence criteria" become subsets. As mentioned, the criteria for high confidence are that cells do not

```
Call:
glm(formula = my.model, family = "binomial", data = Xy)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.6334   0.0001   0.0019   0.0192   1.9656

Coefficients: (1 not defined because of singularities)
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      9.34485   11.64687   0.802 0.422351
Area             0.34090    0.06637   5.137 2.80e-07 ***
MajorAxisLength  2.20443    0.48110   4.582 4.60e-06 ***
MinorAxisLength  5.37529    0.86517   6.213 5.20e-10 ***
Eccentricity    12.92092    2.86952   4.503 6.71e-06 ***
ConvexArea      -0.20791    0.05715  -3.638 0.000275 ***
FilledArea           NA         NA      NA       NA
EquivDiameter   -8.77682    1.15756  -7.582 3.40e-14 ***
Solidity       -11.83785   10.97386  -1.079 0.280708
Perimeter       -0.01347    0.14994  -0.090 0.928406
Intensity       -4.77242    0.54212  -8.803  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1197.0  on 4588  degrees of freedom
Residual deviance:  469.9  on 4579  degrees of freedom
AIC: 489.9

Number of Fisher Scoring iterations: 11
```

Figure 15. Highlights the general linear model calculations comparing the cellular morphological parameters between dividing and non-dividing cells.

$$d^2_{st} = (x_s - y_t)(x_s - y_t)'$$

Equation 2. Euclidean distance equation used by MATLAB. Adopted from MATLAB code documentation.[44]

divide and do not leave the frame over the specified period of time. The high confidence tracks are then processed to measure the changes in each cell of a track from frame to frame. The changes in position (distance travelled), area, eccentricity, major axis length, minor axis length, solidity, and eccentricity are calculated and used to generate the probability distributions of potential cells in the integer programming portion.

The data from the high confidence tracks illustrates the half-normal assumption that was previously neglected by Al-Kofahi *et al* (2006) in their work. Figure 16A plots the distribution of changes calculated for each cell frame-to-frame throughout the "high-confidence tracks." Figure 16B compares the measured data plotted in 16A to the theoretical quartiles of a half normal distribution.

The parameters selected for the probability distributions were also analyzed to determine relationship between the variables. The analysis is highlighted in figure 17 and used the Spearman correlation to ensure that the variables are independent. This was a concern since the distributions are based on changing morphological features and the influence of cell size could prevent the selected morphological parameters from being independent.

Potential Matches

The Potential Match function generates a list of all possible tracks for a cell in an image in the next sequential image. This list generation is accomplished for every cell in every image. The function uses MATLAB's internal "range search" function. The radius of the range search uses the product of the user input "range multiple" and the average distance. The average distance is calculated from the distance each cell travels from one image to the next in the high confidence tracks.

The potential match function determines the potential matches for all images, and then arrays the data in a large matrix which is passed to the next function. This process is highlighted in figure 18.

Probability Density Function Calculations

The probability density function (PDF) is used in the binary integer programming to calculate the minimized matches with the binary integer programming. In order to generate the PDF values, this function first calculates the differences between the cell in image t and the potential matches in image $t+1$ for the following features: distance travelled, area, eccentricity, major axis length, minor axis length, solidity, and intensity.

The differences for each feature is then used to calculate the negative log likelihood which is highlighted in equation 3. The sum of all the negative log likelihoods for each potential match then becomes the PDF value for those matches.

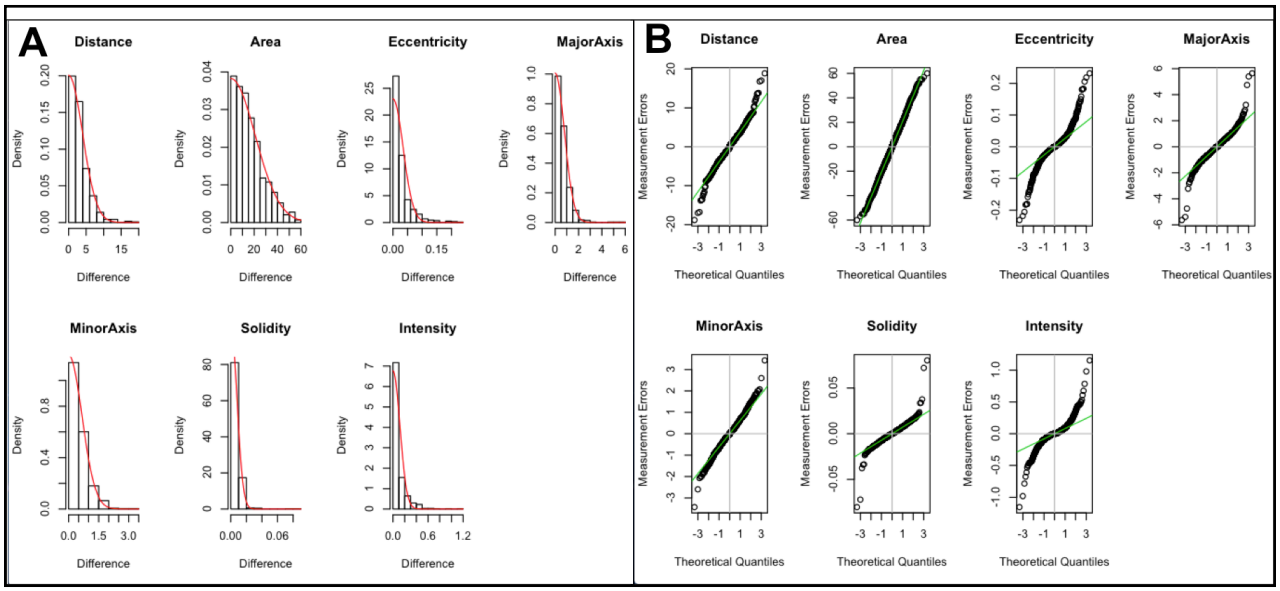


Figure 16. Plots of changes in morphological features derived from the high confidence tracks.

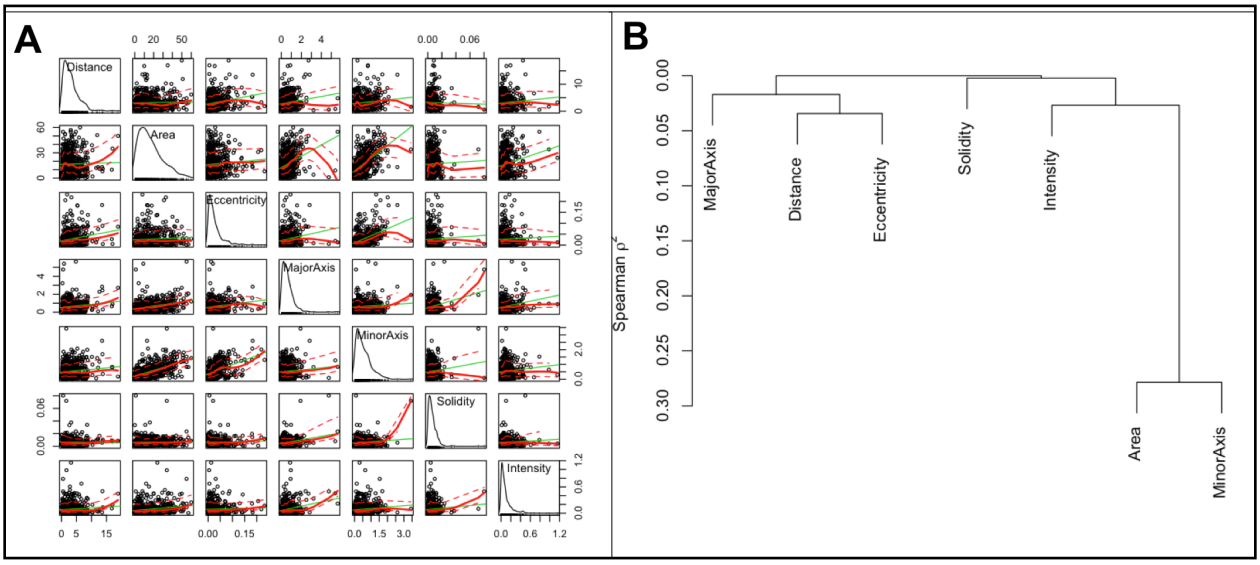


Figure 17. Relationship between the morphological features of the cells which contribute to the probability distribution function used to determine the best track in the integer programming.

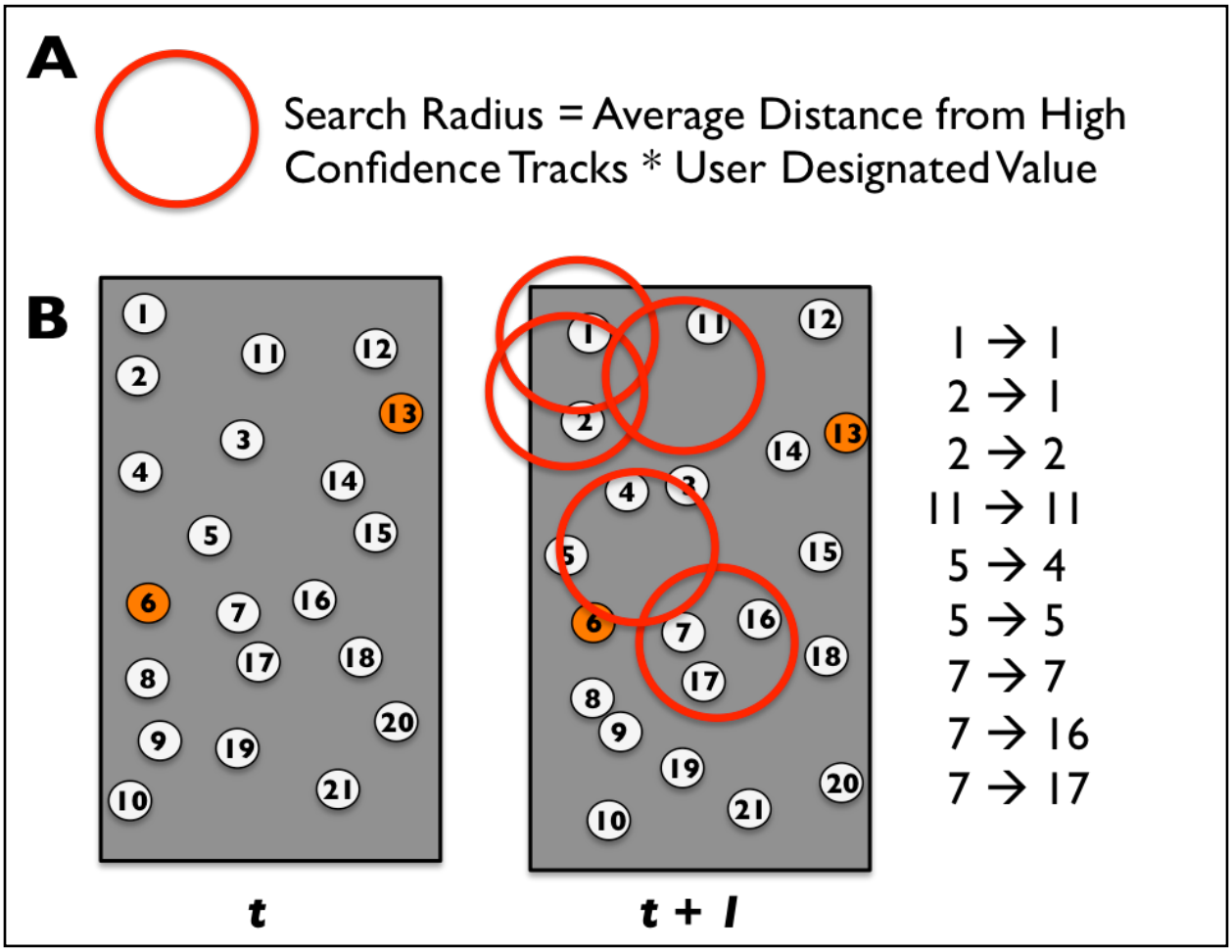


Figure 18. Cartoon the highlights the use of the search radius to generate the list of potential matches. A illustrates the radius is the product of the Average Distance generated from the high confidence tracks and a user designated value. B illustrates how the search radius is centered on the cells *X, Y* coordinates from image *t* in image *t + 1*. A list of all corresponding cells is then generated for the cell from image *t*.

```
% Negative Log Liklihood Calculations
Distance_NLL = -log(2/(AverageDistance*pi)) + Distance.^2/(AverageDistance*AverageDistance*pi);
Area_NLL = -log(2/(MeanAreaChange*pi)) + AreaDiff.^2/(MeanAreaChange*MeanAreaChange*pi);
Eccentricity_NLL = -log(2/(MeanEccentricityChange*pi)) + EccDiff.^2/(MeanEccentricityChange*MeanEccentricityChange*pi);
MAL_NLL =-log(2/(MeanMALChange*pi)) + MALDiff.^2/(MeanMALChange*MeanMALChange*pi);
MIL_NLL = -log(2/(MeanMILChange*pi)) + MILDiff.^2/(MeanMILChange*MeanMILChange*pi);
Solidity_NLL = -log(2/(MeanSolidityChange*pi)) + SolidityDiff.^2/(MeanSolidityChange*MeanSolidityChange*pi);
Intensity_NLL = -log(2/(MeanIntensityChange*pi)) + IntensityDiff.^2/(MeanIntensityChange*MeanIntensityChange*pi);
PDF_Sum = Distance_NLL + Area_NLL + Eccentricity_NLL + MAL_NLL + MIL_NLL + Solidity_NLL + Intensity_NLL;
```

Equation 3. The equations highlighting above are extracted from the Tracking Algorithm in order to calculate the negative log likelihoods for the cellular parameters as well as their sum to create the probability distribution function.

Integer Programming Array Construction

The Integer Programming Array function uses potential matches to generate an array for each step in the image sequences so that there is an array for the objects for image *t* to *t+1*. This array process can be seen in figure 19. The array function also reads through all cells that were previously labelled "dividing" to determine if a cell potentially has two daughter cells in the subsequent image. The array function then generates the necessary construct and adds the daughter PDF values together. There is a weighting factor given to the potentially mitotic PDF value. This value is user designated and future plans are to elucidate a consistent value for the weighting factor. For the MCF10A cells under these conditions the weighting factor is 1/8th. This factor however needs to be examined further and validated since it may be a function of cell type and motility.

Binary Integer Programming

The Binary Integer Programming function uses the internal MATLAB algorithm. This algorithm, highlighted in equation 4, uses the linear programming based branch-and-bound method. The algorithm searches for a feasible solution and then updates the solution matrix before moving onto the next step or branch in the step up. As the algorithm progresses, it continually verifies the previous selection by ensuring there is not another feasible solution. This process is illustrated in figure 20.

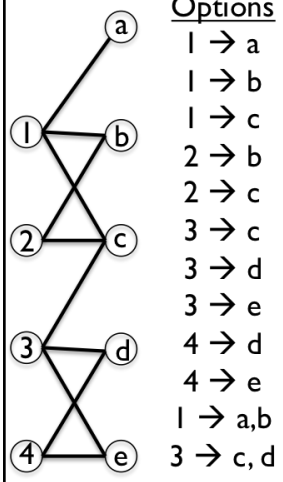
This function progresses sequentially from one frame to the next until complete. The output for this function is a binary list where a "1" indicates a selected cell and a "0" represents the non-selected. The output is equal in length to the array and the selected values correspond to the potential match list generated previously.

Match Indexing to Track Generation

This function generates the cellular tracks for an image stack. The function uses the potential match list, the binary integer programming output, and potential mitotic data to generate the tracks. The binary output is indexed against the potential match list to generate a list of all selected cells from one frame to the next.

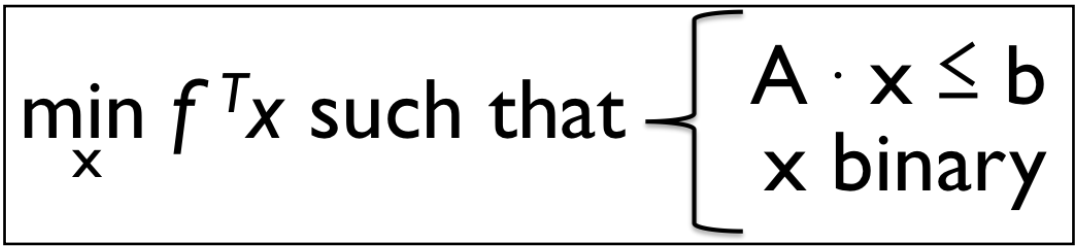
If a cell is identified as having undergone a mitotic event from the indexing process, then the parent and daughter IDs and morphological features are extracted and entered into the track matrix. The cells that come from a mitotic event are labelled in the information category with either "111222" or "222111." Additionally if a cell appears in a frame that is not the first frame then it is labelled "333333." Cells appearing could come from the outside of the focal plane or from the edge of the image.

Once the track list is completed for all of the images, then the tracks for each cell are given a "global identification number." This is a sequential and arbitrary ID to keep track of the cells through the



1	2	3	4	a	b	c	d	e
1	0	0	0	1	0	0	0	0
1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0
0	1	0	0	0	1	0	0	0
0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1
0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	0	1
1	0	0	0	1	1	0	0	0
0	0	1	0	0	0	1	1	0

Figure 19. Translation of options to binary matrix. The options highlighted in the left side of the cartoon are translated into an array and then a binary matrix. The top row highlights the separate portions of the matrix where known tracks are compared to the potential options from another frame.



Equation 4. MATLAB Binary Integer Programming Algorithm.[45]

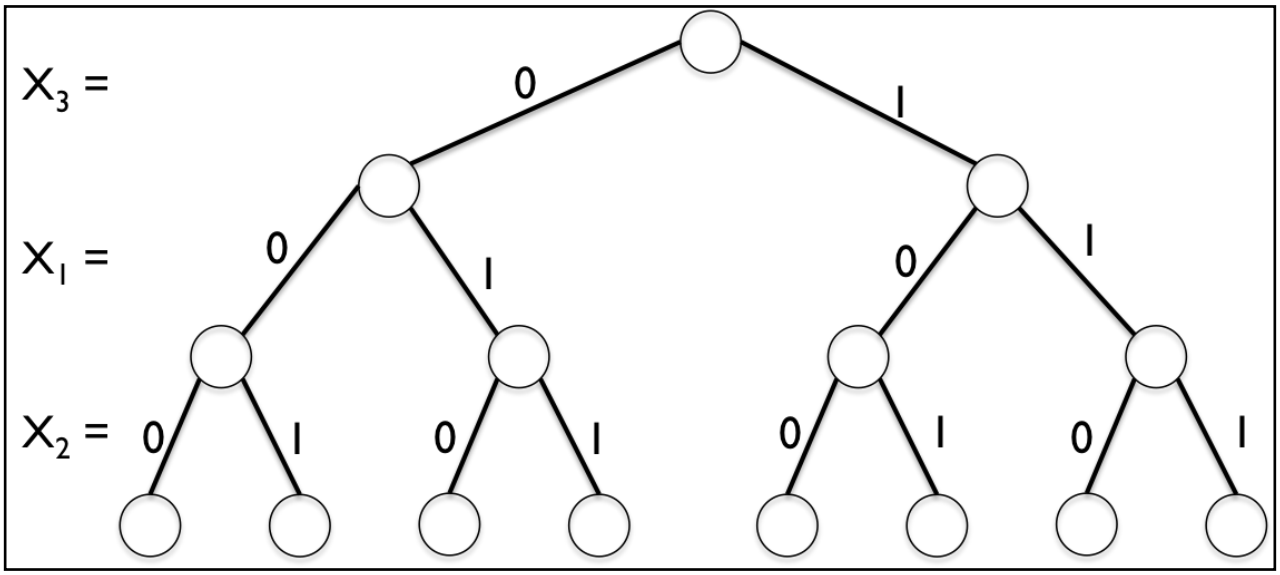


Figure 20. Branching tree example for MATLAB's Binary Integer Programming. This figure was adopted from and modified from MATLAB's 2013 Binary Integer Programming documentation.[46]

image stack. This ID also enables analysis of the tracks and enables fixing of tracks if issues are identified. The tracks are written as a comma separated file that can be used for further analysis.

Fractional Proliferation

The Fractional Proliferation function is a function which reads through the track list to extract mitotic information from the cells for data analysis. The function reads through the track information for the "111222" and "222111" tags. The function also extracts the cells with the "333333" tag to see if those cells are from potential mitotic events that were missed. The extracted data is then reformatted to data construction used by Tyson DR., *et al* in their 2012 Nature Methods paper.[47] The Fractional Proliferation information is used to both validate the test and as potential marker for changes in cellular behavior as the result of a perturbation. The output of this section is a comma separated file that can be used for further analysis.

Performance - Speed

The tracking algorithm was run through 2013b MATLAB by MathWorks (Natick, MA) on a Apple Macbook running with a 2.9 GHz Intel Core i7 processor with a 8GB memory. The speed of the algorithm was tested under various conditions.

Increasing the number of images increases the overall run time, but the proportion of run time for each function remains the same. This result is highlighted in figure 21 where 21a highlights the overall runtime versus the image stack while 21b highlights the proportion of each function in the overall time.

Subsequently the frame skip option was examined. By increasing the frame skip option from 1 to 2 the run time is halved for each image stack length. This result is illustrated in figures 21c and 21d which highlight the total run time and the proportion of the function run, respectively.

Altering the range multiple for the search radius increased the overall run time. This function effects were examined using an image stack of 400 images and increased from 5 to 10 and then to 15. The range multiple is a user designated value that is multiplied against the average distance travelled by the cells in the high confidence tracks. This radius is used to search for potential matches in the potential function of the code. Figure 22 highlights the increasing overall run time (a) as well as the increasing time for the potential function (b).

Performance - Accuracy

The accuracy of the algorithm was examined in three categories: cell counts, tracks, mitotic events.

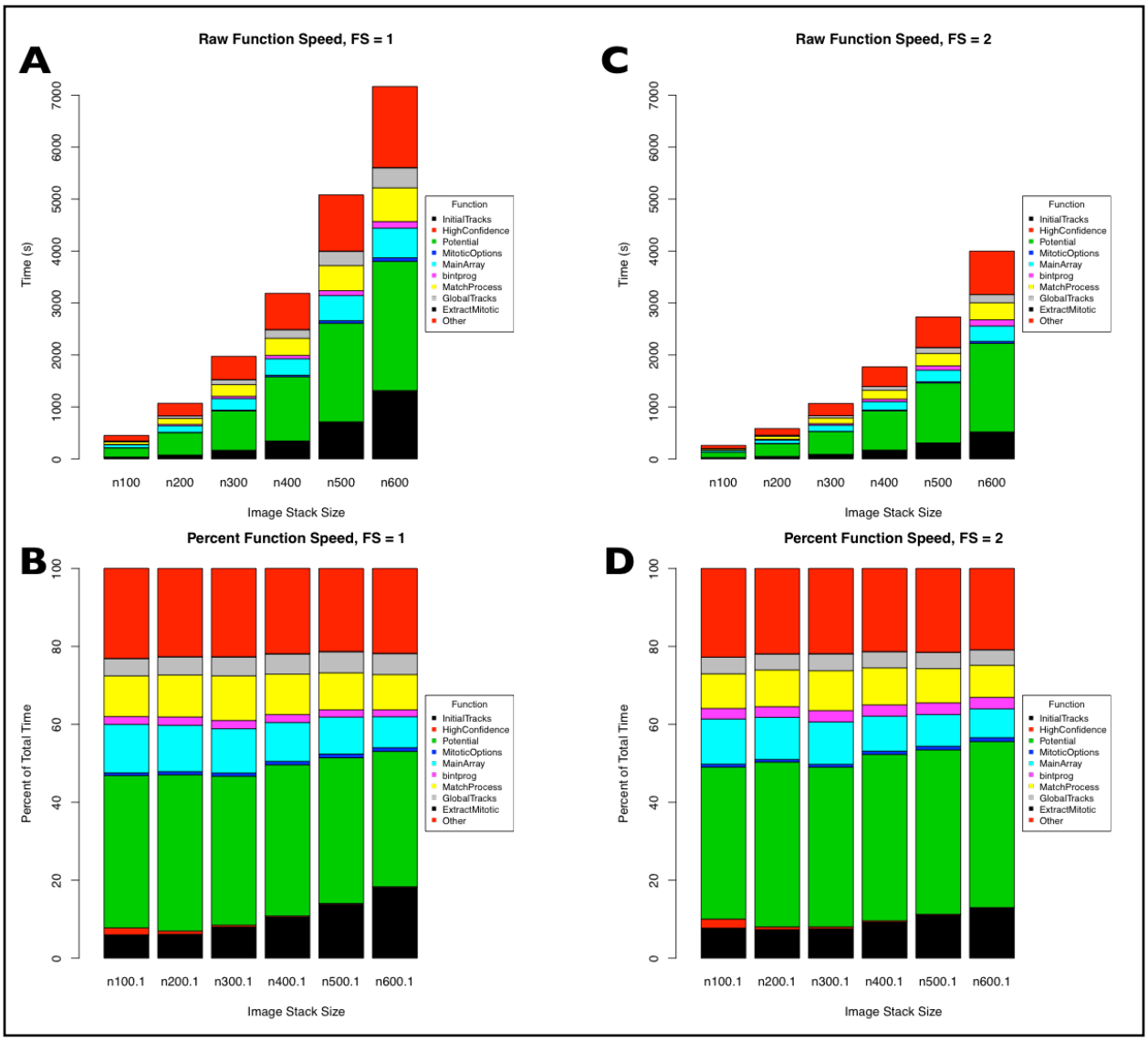


Figure 21. Algorithm run time on 2.9 GHz Intel Core i7 processor with 8 GB memory. A and B highlight the run time at 100 image stack intervals. The relative proportions of each function in the code remain relatively unchanged. Increasing the frame skip option C and D however decreases the overall time by one halve while the proportions remain the same.

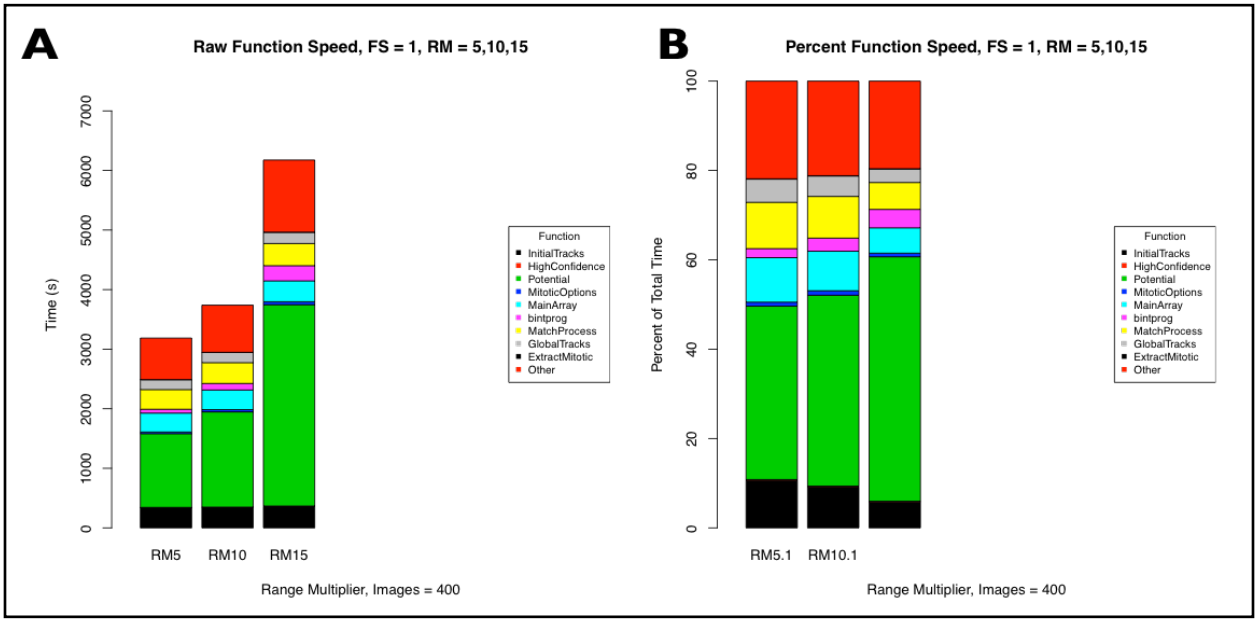


Figure 22. Highlights the effects of increasing the range multiplier in the code from 5 to 10 and to 15. The range multiplier is used in conjunction with the average distance travelled for each cell in the high confidence tracks to calculate the search radius in the next frame to find potential matches.

The accuracy of cell counts was evaluated by comparing ImageJ analysis of the original images, counts of the post segmentation from SegmentReview, and the post-integer programming analysis. ImageJ is a java based image processing tool developed by the Research Services Branch of the National Institute of Health in Bethesda, Maryland.[48] The cell count comparison was conducted using a 10 frame interval through the 655 image test set. The image test set is a control well of MCF10A cells treated with DMSO. The results highlighted in figure 23 illustrate that there is little difference between the ImageJ analysis of the original images, the post-segmentation analysis and the post-integer programming analysis in the first 400 frames. After the 400th frame, which is approximately 40 hours into the experiment, the cells become confluent, i.e., they are touching and overlapping in places, significantly increasing difficulty of the segmentation task. The trends of the post-segmentation and the post-integer programming cell counts are the same, indicating that the variation is likely due to the cells moving out of frame along the edges since the integer programming removes tracks that move out of the frame. The difference between image J analysis and the segmentation is likely do to the image J algorithm.

Due to cell count variation at the higher frame numbers, the tracks through the first 400 frames were examined. The track information was plotted in R Project Statistical Software (University of Auckland, New Zealand). The tracks are initially plotted in three dimensions, so that the cell tracks are shown by the *X, Y*-location and the frame number being the depth. Figure 24 highlights the tracks generated from the first 400 images.

Upon further examination of the tracks, it was determined that some tracks were not correct. Some tracks ended prematurely creating shorter tracks than expected in the experimental conditions. The tracks were then subsetted into four groups based on track length based on the number of image frames. These groups were frames 1 to 5, frames 6 to 10, frames 11 - 15, and frames > 16. The number of tracks in each subset and the proportion of each subset are highlighted in figure 25. By increasing the range multiplier, the total number of tracks was decreased while the proportion of long tracks (> 16 frames) was increased as shown in figure 25B. This result indicates a relationship between the broken tracks and the range multiplier suggesting that some of the broken tracks are due to cells moving out of the search radius during the potential function of the algorithm. An additional cause of broken tracks is the result of cells moving in and out of the frame either on the focal plane or from the edges of the image.

To examine the location of the broken tracks, the short tracks were plotted in 3D, shown in figure 26, using a 400 image stack with a range multiplier of 20. In the 3D rotatable image, it is clearly recognizable that most of the short tracks are near the edges of the image (unfortunately this is poorly captured by the 2D printed version).

The lifespans of the cells was also examined in order to ensure the detection of mitotic events. Results are highlighted in figure 27 and are derived from the Fractional Proliferation output data

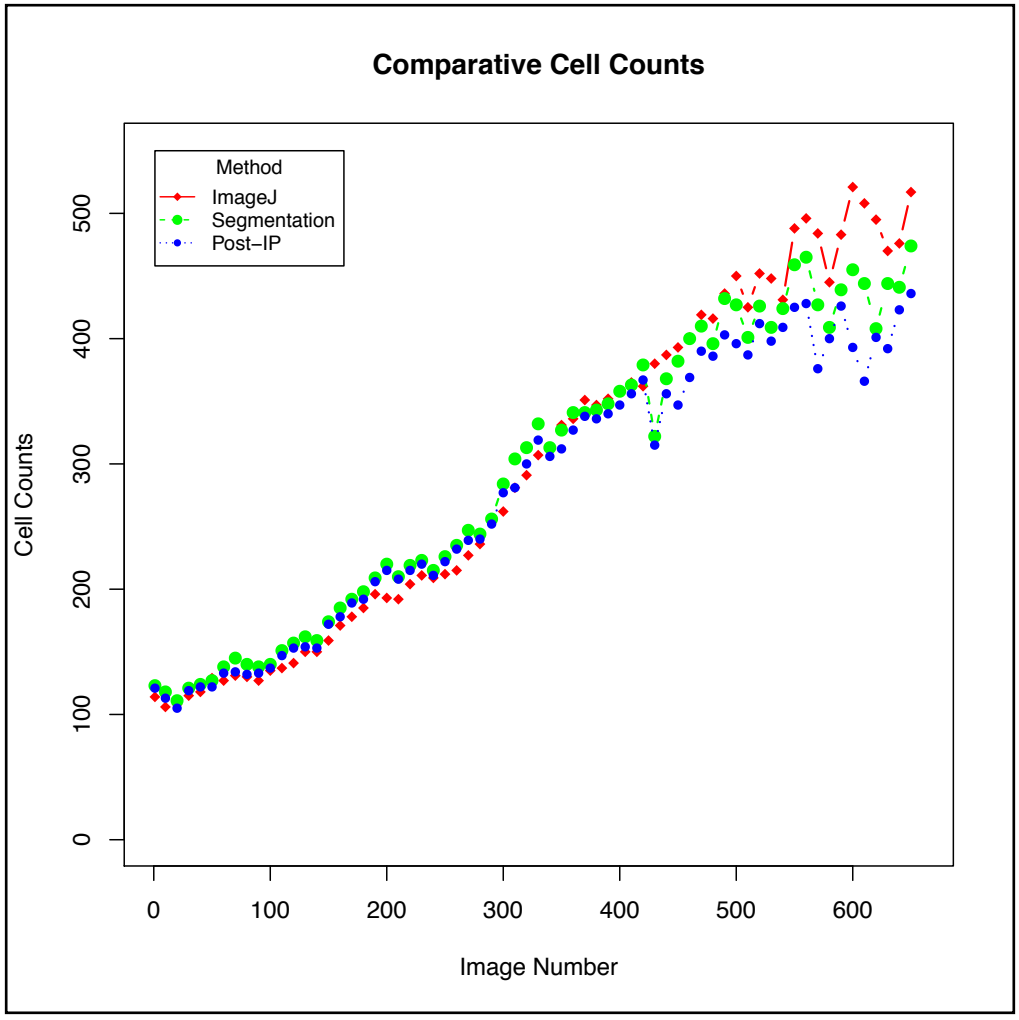


Figure 23. Comparative cell counts using three different methods / sources. ImageJ processed the raw images for cell counts. Segmentation added the objects found in each image following SegmentReview processing and Post-IP used the results of the tracking to count the cells.

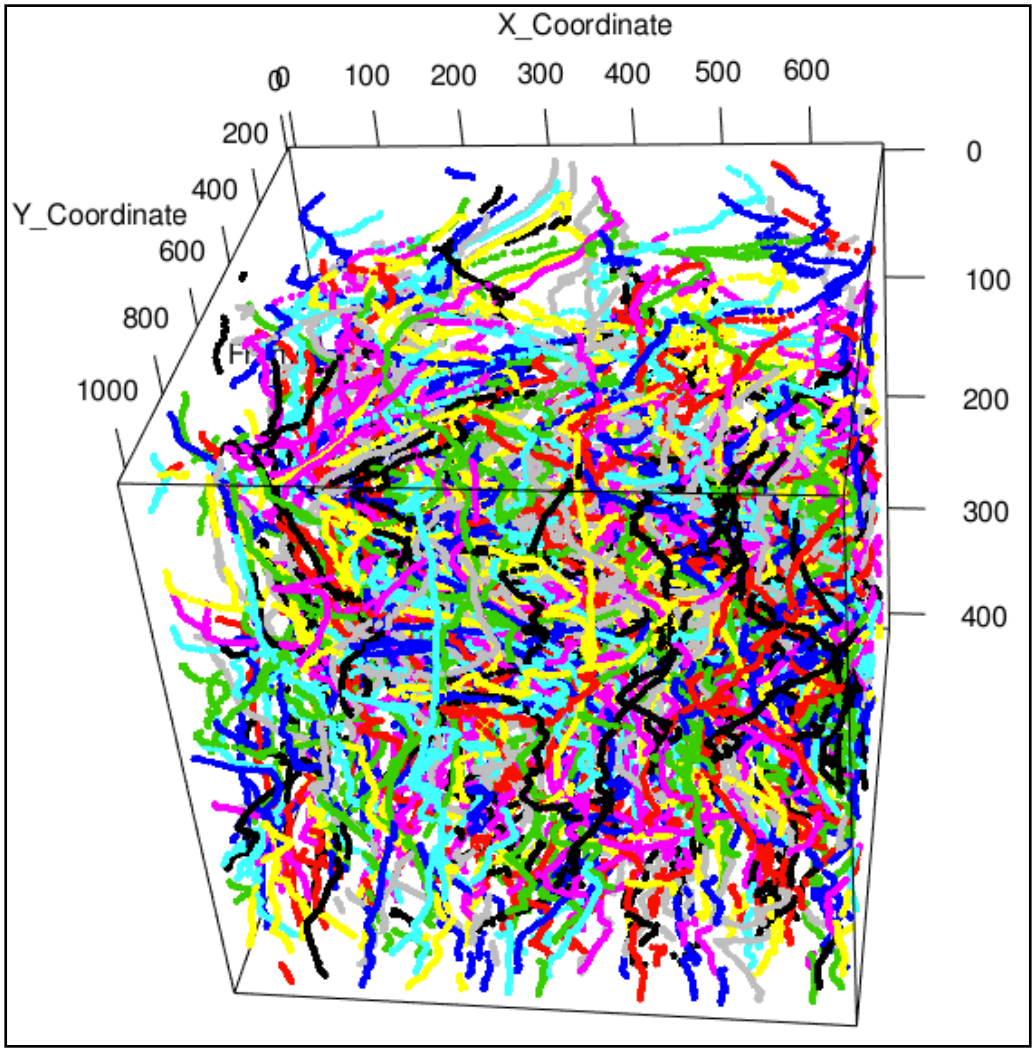


Figure 24. The tracks for cells in images 1 through 400 plotted in 3D. Each cell / track is plotted using a different color. The individual tracks in this image are difficult to distinguish but highlights the amount of information in one image stack.

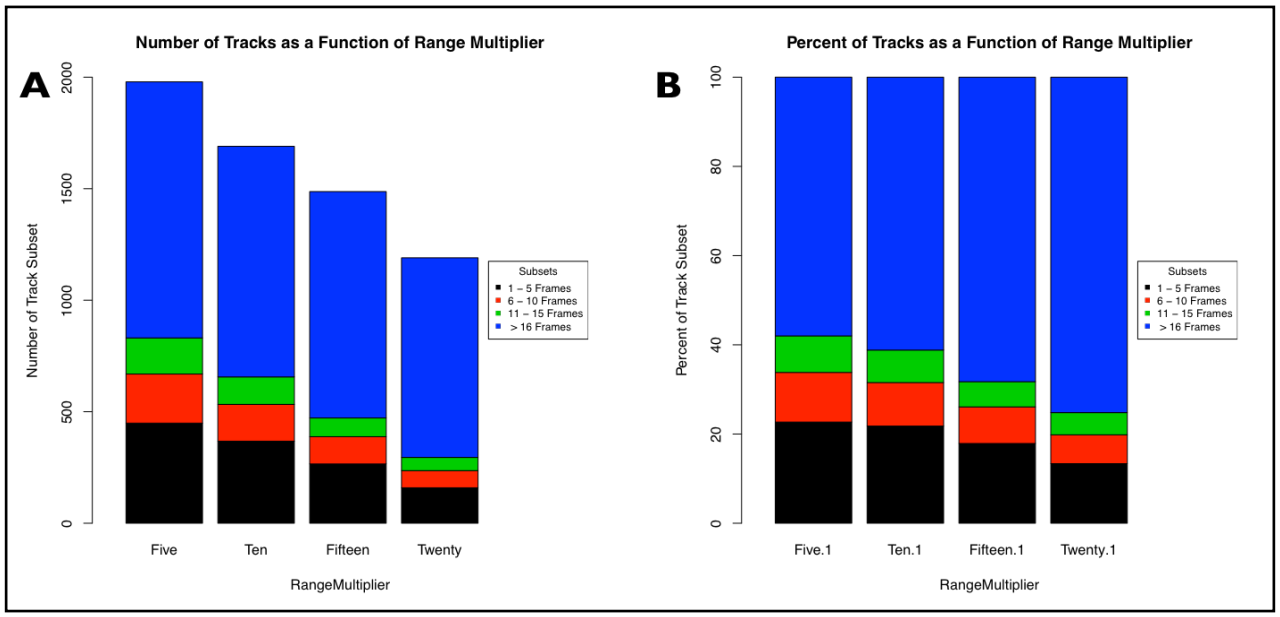


Figure 25. Highlights the number of tracks (a) and the percentage of tracks (b) in four different subsets as a function of the range multiplier. The blue highlights the shorted subset 1 - 5 frame lifespan, the green is 6 - 10 frame lifespan, the red is 11 - 15 frame lifespan, and the black is the lifespan greater than 16 frames. The overall number of tracks deceases with increasing range multiplier suggesting that the a faster moving cells are being identified with a larger search radius.

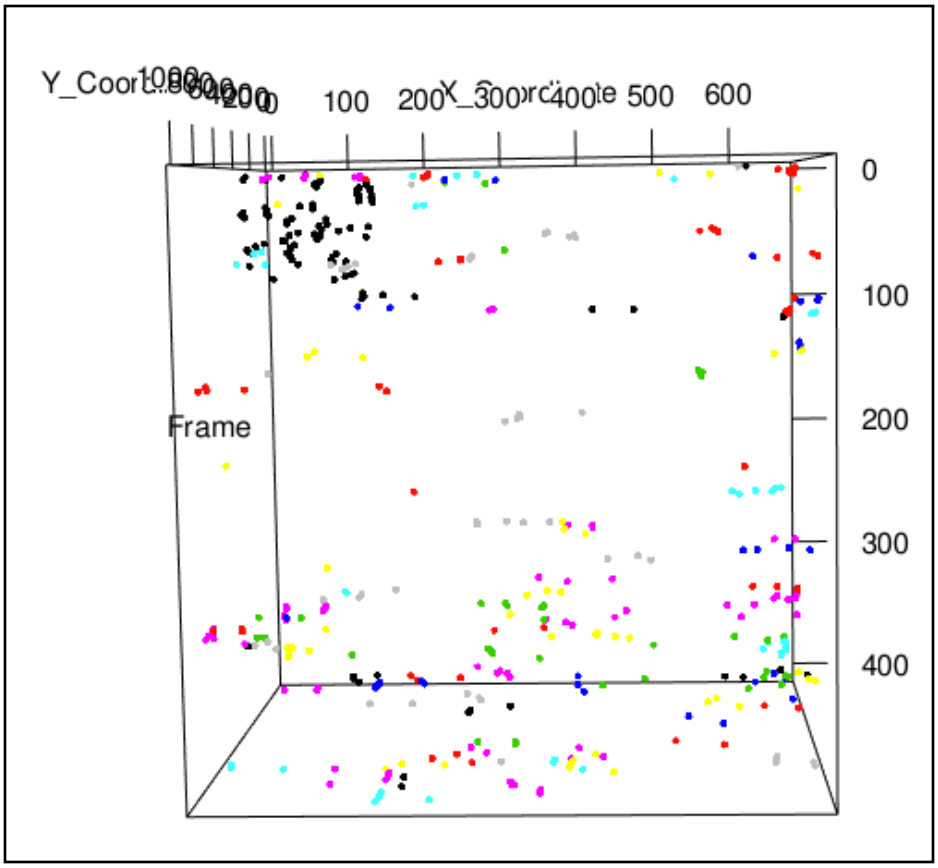


Figure 26. A three dimensional plot of the shortest track subset (lifespan 1 - 5 frames). The images shows the *X*- and *Y*- location of the centroid while the z-axis is the frame number. The high number of short tracks in the first 100 frames suggest that the non-confluent state is allowing the cells to move outside of the range search and in the later images the cells are becoming confluent and tracks are intersecting and breaking down at times. This was tracked using the 20 value of the range multiplier.

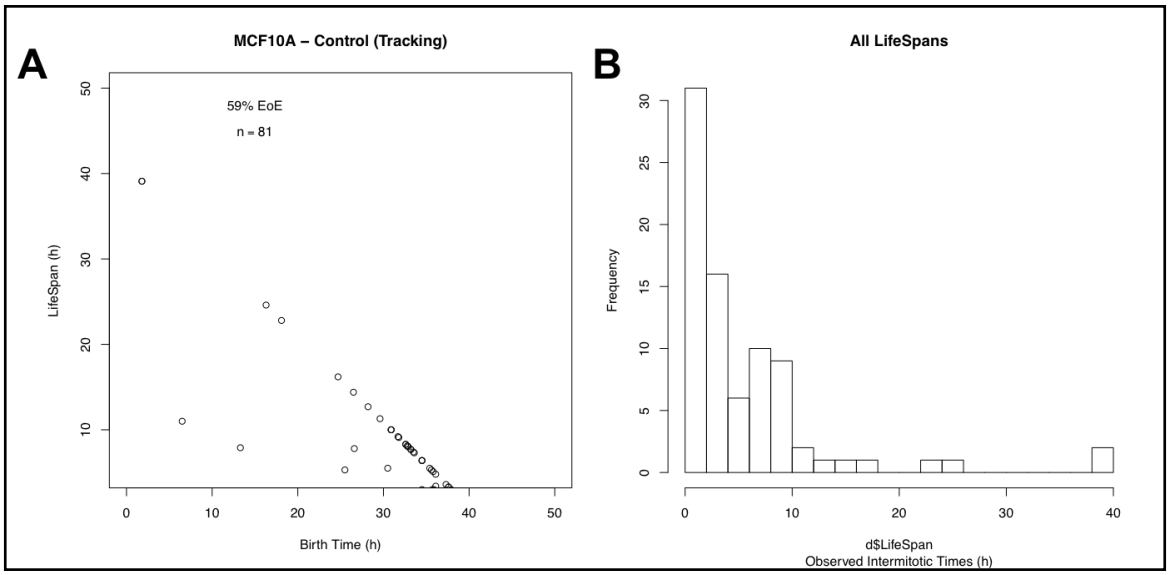


Figure 27. Highlights data extracted regarding mitotic events from the tracking information. This data is incomplete due to the short run of the program. The program only ran out to 400 frames instead of 655 which was due to the confluence of the cells.

generated by the tracking code. Due to the shortened length of the movie to 400 images, this data is incomplete. If the image stack were able to progress the complete life span information would be automatically generated for comparison with manually tracked data.

Future Directions

A critical shortfall for the current state of the algorithm is the lack of visualization tools. MATLAB presents an opportunity to build a graphical user interface in order to visualize the post algorithm tracks and to modify the tracks as necessary. Due to time constraints, the user interface was not developed, though the code is open and manageable for future development. This user interface could also be tied into the SegmentReview GUI so that only one single user interface is presented. The GUI should also help refine the short tracks.

Initial development has begun on an additional function for user initiated merging of tracks. Currently this function extracts basic information about tracks from the overall tracks matrix. This function looks for unique track IDs, as well as cell locations in the start and end frames. The function then removes all cells that are within 5% of the image edge which prevents the attribution of cells that move in and out of the image over time. The function then moves through each frame, looking for tracks that are both ending and beginning. Using IDs, track details are extracted from the overall track matrix and plotted in 3D scatter plots. The plot in figure 28 only consists of tracks that are starting in the selected frame as well as the tracks that end in the previous frame. To assist in visualization, a red circle is drawn around the first starting track and a blue ring around the last cell of the ending track. This feature enables a user to rapidly see potential matches. The radius of the circle equals the previously used search radius.

At each frame the user is prompted with several questions regarding the potential broken tracks and is asked to provide ID numbers and potential mitotic events. This allows the function to continue short tracks that are identified by the user and introduce missed mitotic events. The identified information is then processed through the tracks matrix so that the tracks are properly updated.

This function could easily be further improved in the future through the incorporation of a Bayesian algorithm to select the potential short track matches, thus reducing the user interaction.

Image stacks from a larger repertoire of experiments also need to be analyzed to validate the use of the algorithm on different cells and/or conditions.

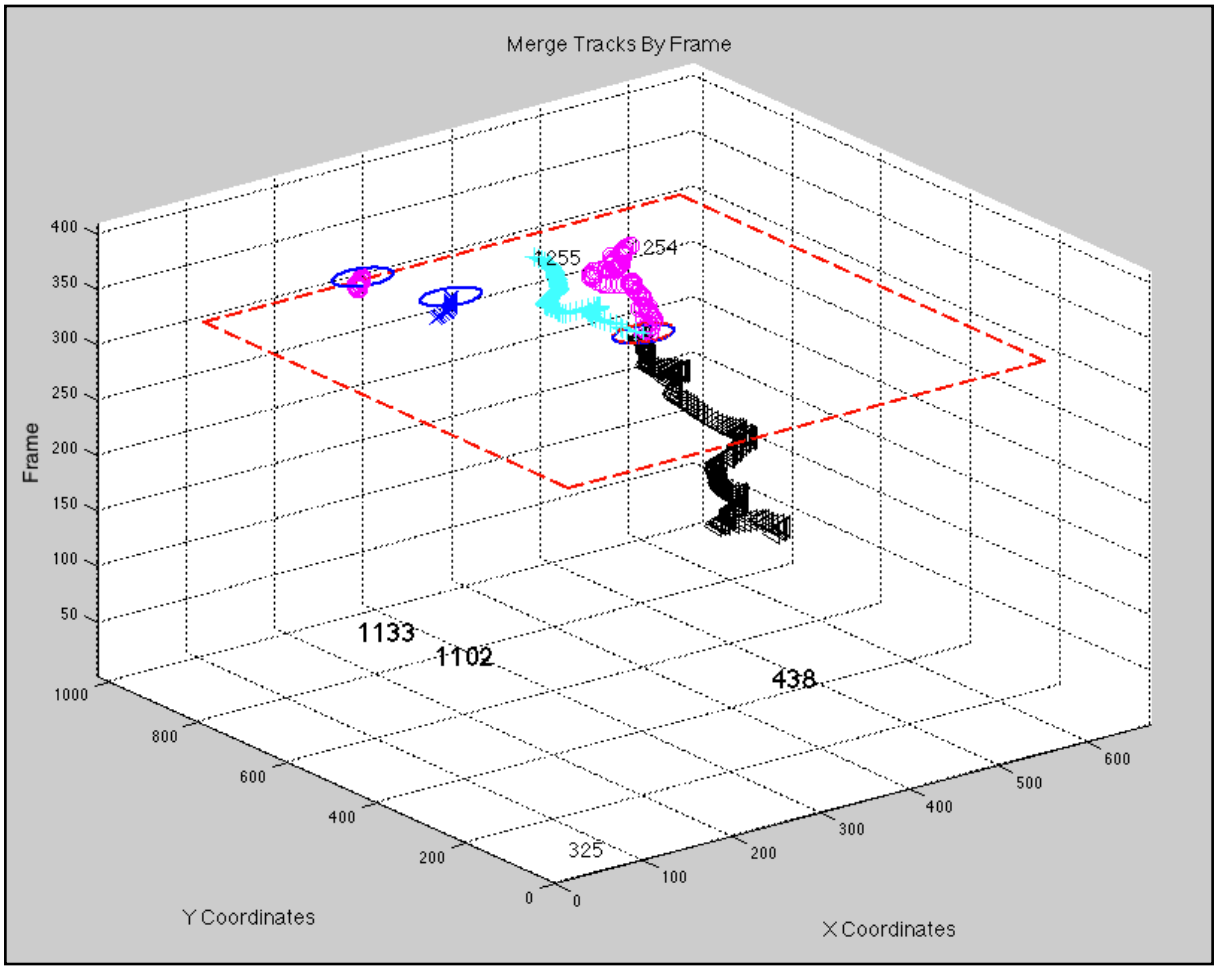


Figure 28. Screen shot of the current merge plot function which enables users to review and correct the short tracks. This image depicts a missed mitotic event where track ID 438 divides into daughter cells 1254 and 1255. The initial algorithm missed the event and track 438 ended and the two daughter tracks abruptly began.

CHAPTER IV

AUTOMATED IDENTIFICATION OF FOCAL ADHESION IN 3D

Background

The following is a collaboration between the Vanderbilt Laboratories of Vito Quaranta and Donna Webb for the purpose of applying concepts of automated cell tracking to the identification of subcellular structures. In this example, we focus on focal adhesions.

Cell migration plays a role in a variety of physiological processes, such as wound healing, and diseases, including cancer metastasis. Due to the multitude of proteins and protein complexes involved, its molecular underpinnings remain incompletely understood. A major class of protein complexes involved in cell migration is focal adhesions (FAs), which are "organized aggregates of specialized proteins distributed at the basal surface of adherent cells."[49] Understanding structure-activity relationship of FAs is a major challenge. For instance, the size of FAs appears to regulate cell speed, whereby cells with small FAs move rapidly while cells with large FAs move slowly.[50] While this function of FAs in cell motility can be investigated by genetic manipulations and pharmacological interventions[51, 52], a significant obstacle is the requirement of manual quantification of FA size and abundance, a time consuming and error prone task. To overcome this challenge we developed a computational pipeline to automatically process time-lapse *z*-stack fluorescence microscopy images for identification and characterization of FAs.

Algorithm

Software

The algorithm was developed using the MATLAB by MathWorks (Natick, MA) platform.

Image Processing

The automated detection of focal adhesions in three dimensions (3D) is accomplished using two scripts/functions in MATLAB. The overall process of the scripts is illustrated in figure 29. The first script conducts the image processing, detects the FAs, measures their x- and y-locations, intensity and area, produces a rough FA track through the slices using MATLAB's k-nearest neighbor algorithm, and provides a comma separated file with the image number, z-plane number, x-location, y-location, area, intensity, index / track identification, and distance from that previous slice.

The analysis and processing of 3D microscopic images is not trivial due to the user option to change the number of slices in a 3D image, and to the intensity variation in each plane. Consequently, it

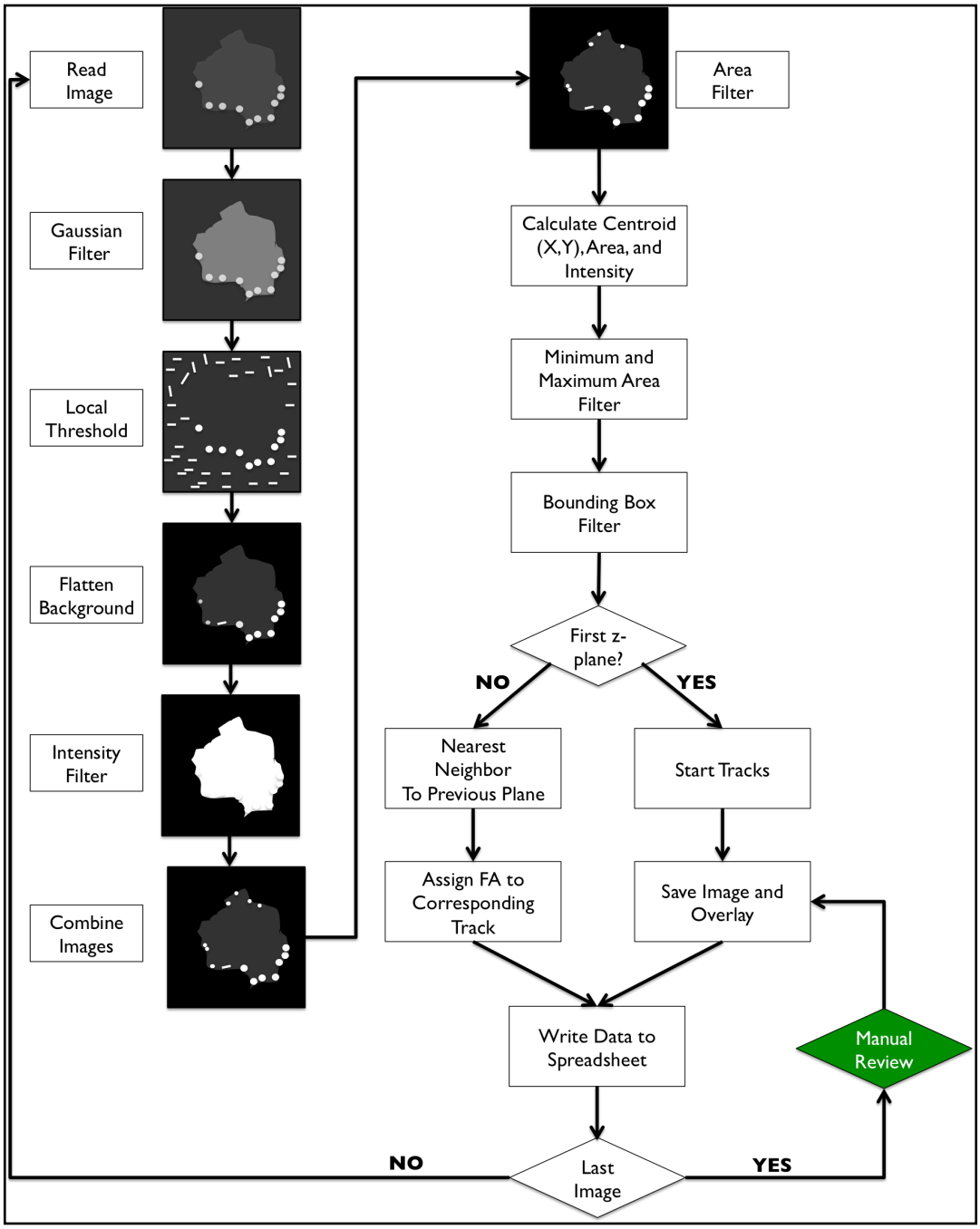


Figure 29. Schematic of the image processing and focal adhesion identification process for 3D confocal microscopy images.

is necessary to process images with a varying number of 3D slices and a varying intensity. This issue is highlighted in figure 30. The z-plane provides the researcher with the ability to visualize and analyze morphological cellular characteristics at varying depths as compared to the single plane of two dimensional (2D) images. While multiple planes provide an advantage in visualization, they complicate image processing.

Each image in a time series has parameters, *X*, *Y*, *Z* which indicate the images width, height, and depth, respectively. Using a 'for loop' for the length of the image stack (start frame to end frame), each image is read and analyzed to determine the number of *z*-planes in the 3D image. The images are then normalized by the following formula: multiplying the minimum pixel value by the maximum pixel value which is then divided by double the maximum minus the minimum pixel value in the image, equation 5.

The output is then filtered using a series of Gaussian and local average filters. The Gaussian filter creates a symmetric gaussian filter using user set size and standard deviation (for the images shown, filter used a size of 5 pixels with a standard deviation of 2.5 pixels). The average local filter used a disk shaped filter of 10 pixels. The average brightness of the image was also determined for each plane and multiplied by the brightness intensity threshold which is also a user specified value. We observed that a ten percent change in the intensity threshold can lead to either no FAs or hundreds of pixelated debris throughout the images. In the images shown this value was determined to be 1.25. This brightness threshold value is used to select all pixels of greater value within the image slice. This image is then saved as image 1.

A frequency filter is then used to attenuate the signal of the image. The script allows for the user to either specify a "low pass" or "high pass" filter (for the images shown the low pass filter was used). In such a filter, a distance matrix is generated where the distance calculated using the pixel strength of the original image. The values in the distance matrix are then attenuate by the following formula: dividing the values by the cut off frequency, a user input value of 10, and the remaining value is then raised to twice the filter order. The full equation can be seen in Equation 6. The product of the frequency filter is then saved as image 2.

The two separate images are then flattened by dividing the intensity threshold image over the frequency filtered image. The resulting quotient is image 3 or "combined images" and processed to identify objects within the image. Objects smaller than the user specified value were removed from the image, ideally leaving viable objects. All of the slices in an image and images in an image stack are processed using the same steps and settings.

One shortcoming is that the threshold intensity at one depth or *z* plane may not provide the most ideal intensity at another. The wrong intensity threshold could and does potentially leave artifacts within the image. However, if small enough (depending on the user settings) these artifacts can be cleared using the minimum area filter.

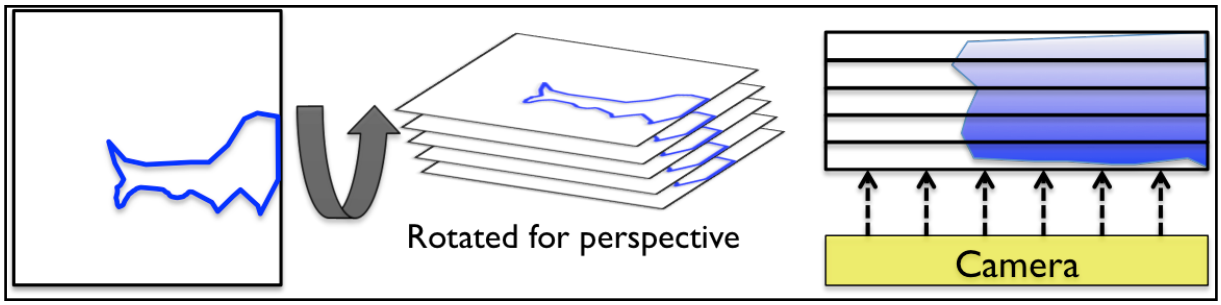


Figure 30. Schematic illustrating the potential for intensity attenuation at varying depths in 3D images.

```
% Normalizing the Guassian Filtered Image
int_class='uint16';
max_val=double(intmax(int_class));
img_raw=Image;
img_dbl=floor(double((img_raw-min(img_raw(:))))*max_val./double(max(img_raw(:))-min(img_raw(:))));
switch(int_class)
    case 'uint8'
       Image=uint8(img_dbl);
    case 'uint16'
       Image=uint16(img_dbl);
    otherwise
        Image=[];
end
Image1 = Image;
```

Equation 5. Normalization equation used to normalize each slice of a 3D image stack for identifying and tracking focal adhesions.

```
CutOffFreq=10;
FilterOrder=6;
FilterType='LowPass';

% Frequency Filter
img=Image;
original_img_sz=size(img);
img_sz(1)=2^nextpow2(2*original_img_sz(1)-1);
img_sz(2)=2^nextpow2(2*original_img_sz(2)-1);
%calculate the square distance matrix from the center point
dist_matrix=false(img_sz(1:2));
dist_matrix(floor(img_sz(1)/2)+1,floor(img_sz(2)/2)+1)=true;
dist_matrix=bwdist(dist_matrix).^2;
%arrange it so it's in the proper form for fft2
dist_matrix=ifftshift(rot90(dist_matrix,2));
%setup the filter
cutoff_freq=CutOffFreq;
filter_order=FilterOrder;
filter_type=FilterType;
switch(filter_type)
    case 'LowPass'
        butterworth_filter=1./(1+ (dist_matrix./cutoff_freq).^(2*filter_order));
    case 'HighPass'
        butterworth_filter=1-1./(1+ (dist_matrix./cutoff_freq).^(2*filter_order));
end
%filter the image in the frequency domain
nr_slices=original_img_sz(3);
img_filtered=zeros(original_img_sz);
for i=1:nr_slices
    slice_fft=fft2(double(img(:,:,i)),img_sz(1),img_sz(2));
    filtered_slice=real(ifft2(butterworth_filter.*slice_fft));
    img_filtered(:,:,i)=filtered_slice(1:original_img_sz(1),1:original_img_sz(2));
end
Image2=img_filtered;
```

Equation 6. Filter equation used to process slices in a 3D image stack in order to identify and track focal adhesions.

Focal Adhesion (FA) Identification

A processed image consisting of processed *z* planes of "combined images" is then analyzed to identify the centroid location, area, and intensity. This is done using the region properties function in MATLAB to determine the area and *x*- and *y*- coordinates of the centroid. The accumulation array function is used to calculate the raw intensity. The accumulation array adds the pixels in the designated object. It is important to understand that the intensity values are a function of the thresholding values such that increasing the threshold decreases the intensity counts. The area data is then used to provide another level of filtration on the objects to help determine the FA locations in a cell. With the user determined parameters of minimum and maximum FA area, the objects are filtered leaving only those that meet the criteria. Each object is then indexed in the binary image of identification.

The selected objects in each slice are then compared to the next slice in the z plane enabling a user to compile the FA 3D properties. The variation in FA complexes in terms of both 3D size and location can then be compared to the speed at which a cell is moving.

The output of this first process is a comma separated file that allows the user to manually curate the FA tracks. Ideally, the k-nearest neighbor search conducted in z-planes for a cell would generate perfect tracks through the slices and from image to image. However, depending on the depth in the z-plane and the dynamics of cell movement, FAs may be constantly appearing, disappearing, and changing in size. The manual curation step provides the user a quick tool with information regarding the potential FAs and allows the user to ensure that the FAs of a track in a slice are matching the FAs at a greater depth. This also allows the research to validate the FA tracks in time.

Outputs

The FA data is then saved and exported as a comma separated file with the headers, "Counts", "Image Number", "Slice", "CentroidX", "CentroidY", "Area", "Intensity", "IndexID", and "Distance." This file enables the user to review the focal adhesions by reviewing the area, intensity, location, and distance to nearest FA in the previous slice of an image. The user can then curate the index identification to ensure accuracy before the file is imported into the next function.

Focal Adhesion Visualization

Using the intensity threshold image and the data embedded in the user curated comma separated file, an output image is created with FAs and identification numbers. Each FA is drawn on the image in blue and the identification number is drawn in red. The FAs are identified as correct by using the area in the user curated file and comparing that value to the values embedded in the processed image. If the areas match then that FA is drawn on the image slice. This process is repeated for every slice of an image

and until the list of FAs is exhausted. Each slice is then written in order into a tagged image file (TIF) stack for that image file name.

The image stack (images with z-planes) are exported to a user designated output location.

Opportunities for Improvement

There are two critical areas that could be improved within the code: the intensity thresholding in the z plane, and the user curated portion.

Currently the algorithm is designed as a single-intensity threshold that fits all images and slices in a stack and time sequence. This, however, could be improved by using varying intensity thresholds, since this would almost certainly lead to better plane-specific segmentation (what works for z-plane equal to one does not necessarily work for z-plane equal to five) and enhanced FA identification accuracy.

The other potential for improvement is to minimize the necessity for manual curation by maximizing automation. Such improvement could be done using a either a Naive Bayes Classifier or through integer programming. For a Naive Bayes Classifier there would need to be background information describing a correctly identified FA and these properties would need to be separable from incorrect FAs. Several possible parameters include area, intensity, solidity, major axis length, and minor axis length.

Additionally, a binary integer programming step could improve the tracking of FAs not only through slices but also through time. The binary integer programming could use the same physical properties as the Naive Bayes Classifier to calculate the probability distribution in order to determine the best option for the FA tracks.

Results of 3D Focal Adhesion Algorithm

The results of the semi-automated FA process are promising and figure 31 highlights one example image. The 3D image has 5 slices which were processed sequentially and analyzed using the methods highlighted in figure 29. The slices indicate that there is one primary FA that meets the area and intensity thresholds in all slices of the image while figure 32 shows a FA that appears in slice 1 but not in slice 2. These differences highlight the requirement for user input, as well as curation based on knowledge of the experimental conditions. The FA in slice 1 may really exist in the subsequent slices but the user designated intensity thresholds could have caused the intensity and subsequent size to be washed out with the background. This possibility is suggested by the visible gray hue that surrounds the cell. The intensity is also attenuated with height/depth of the image as the cell, media, and debris absorb light from the camera.

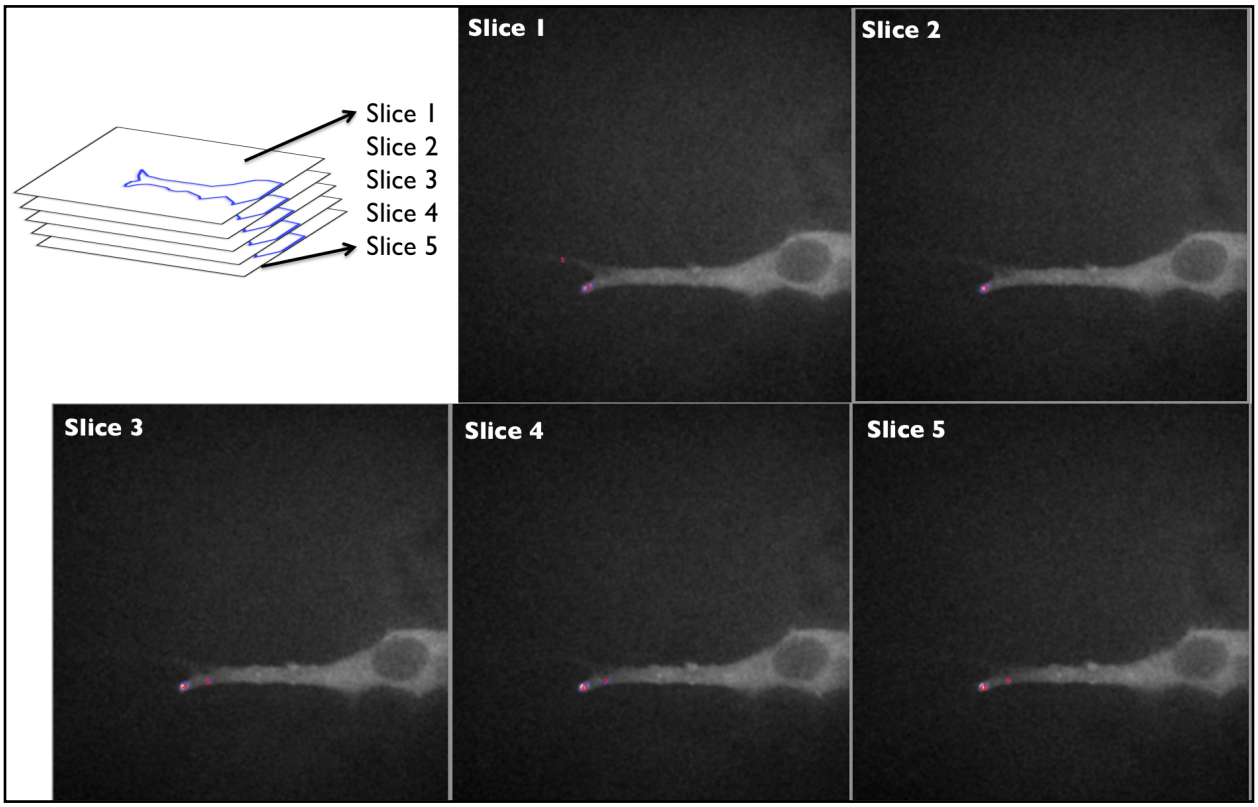


Figure 31. Focal Adhesion output images. The images are the slices of the 3D image stack with the identified focal adhesions which were identified using the semi-automated method previously described. Raw image provided by the Donna Webb Laboratory at Vanderbilt University.

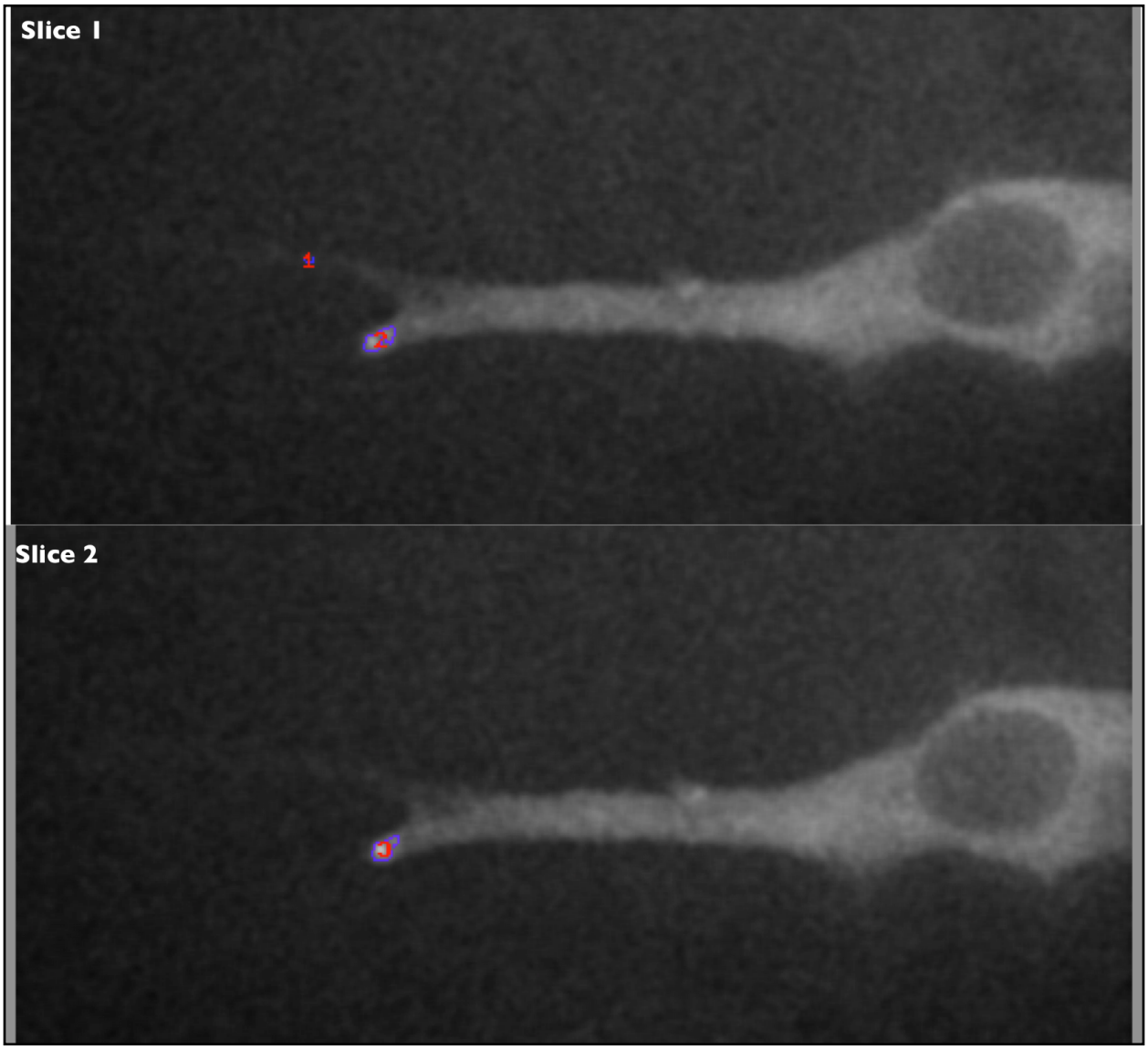


Figure 32. Different Focal Adhesions. The examination of the identified focal adhesions illustrates that there are different focal adhesions identified for the same image yet different slices. This is attributed to the selection based on focal adhesion intensity and size. Object 1 from slice 1 is not shown in slice 2 since the intensity for that object is lost likely due to light attenuation in the image capturing process.

The data presented in table 1 corresponds to the image presented in figure 31. This data illustrates the previously mentioned problem with FA identification in between slices. The FA labeled 1 has an area of only 16 pixels (compare to the other FA with an area 5 times larger), on the cusp of the lower limit set at 15 pixels by user-designated settings. The user could easily lower the pixel limit in order to explore the existence of this FA in other slices at the risk, however, of generating additional "ghost" FAs. Clearly, user knowledge is key.

The center of each FA in the slice is additionally identified using the *x*- and *y*- coordinates in the image. The intensity for each FA is the raw sum of pixel intensity. The intensity is not integrated for the area of the FA rather the total of all pixels. The raw counts enable the user to integrate over the area of the FA to determine if the intensity strength corresponds with a desired experimental effects. Specifically is a high intensity reading the result of a thresholding error or is it a real focal adhesion complex in the cell.

The nearest neighbor tracking is currently rudimentary compared to the larger tracking scheme for cells, and uses only the MATLAB internal nearest neighbor algorithm which was previously highlighted. This nearest neighbor search is currently designed to only account for tracking within a specific image from slice to slice and not image to image. The tracking is currently limited for two reasons. One, the algorithm does not know how to handle the disappearance of a FA. Two, the algorithm does not know which slice and correct FAs should be used for the next image in the sequence. The Webb lab is currently working to compile corrected image data in order to create a training set. This training set can then be used to generate distribution of of area, distance traveled, and intensity in order to properly track intensity. The Webb lab is also attempting to determine which set of *x*- and *y*- locations for a FA should be used for image-to-image tracking. For example, are the FAs tracked by the slices in the 3D image stack which continue from one image to the next (slice 1, image 1 to slice 1, image 2) or is there one set of FAs generated for all the slices of an image to compare to the next image.

Conclusions

Though the results of the semi-automated focal adhesion algorithm are at an early stage, the potential for a fully automated algorithm is evident. The initial data indicate correct identification of FAs is being achieved. However, a few issues remain to be addressed. One issue being identifying a viable intensity threshold to process the images since the light attenuates in depth. This process is new subsequently the user of algorithm and the instrument need understand the capabilities and limitations of the both. By tracking light intensity in the instrument and how that effects thresholding intensities will enable users to accurately and identify focal adhesions in the first pass. Such understanding includes knowing the changes in the density of the cell media and when the light on the instrument is dying or has been replaced. These factors directly effect the light in the image and effect the image processing.

Counts	ImageNumber	Slice	CentroidX	CentroidY	Area	Intensity	IndexID	Distance
1	1	1	139.0625	327.6875	16	1136251	1	0
2	1	1	171.028037	363.82243	107	9353745	2	0
3	1	2	169.757576	364.919192	99	8590111	2	1.27046163
4	1	3	170.466667	365.085714	105	9148862	1	0.70909091
5	1	3	200.714286	358.25	28	2873135	1	41.7410816
6	1	4	170.132075	365.235849	106	9217851	1	0.33459119
7	1	4	198.9	358.9	20	2033674	2	1.81428571
8	1	5	169.075	366.05	80	6913636	1	1.05707547
9	1	5	201.65625	358.84375	32	3299035	2	2.75625

Table 1. Focal Adhesion data for the image presented in figure 25. The counts are the labels on the image in each slice of figure 25 and correspond to the subsequent columns which include image number, slice number, X- and Y- location of the adhesion center, area, intensity and the index ID and distance which are from the nearest neighbor search in the subsequent slice. The intensity counts are the number of pixels in the adhesion area. The intensity is not integrated over the area rather it is the raw count data.

Further experimentation and image processing will also enable provide a databased of properly identified focal adhesions. This database with its physical characteristics of focal adhesions can then be used to process future images both in terms of identification of focal adhesions and tracking. The low number of focal adhesions and the limited number of frames in the focal adhesions movies make the focal adhesion a potential candidate for integer programming tracking.

More results of the 3D focal adhesion semi-automated process is pending work by the Webb Laboratory and results are slated for publication in 2014.

[1] F. Frischkneckt, The history of biological warfare: Human experimentation, modern nightmares and lone madmen in the twentieth century, EMBO Reports special issue on Science and Society: 4:S47-S52, 2003.

[2] United Nations Office for Disarmament Affairs, "The Biological Weapons Convention," www.un.org/disarmament/WMD/Bio/

[3] There is a distinction between "offensive" and "defensive" biological warfare agents. Signatory countries are allowed to maintain secure stockpiles of biological agents in order to maintain vaccine and antidote stores.

[4] "Non-signatory States," http://www.opbw.org

[5] The Centers for Disease Control and Prevention defines bioterrorism as "the deliberate release of viruses, bacteria, or other germs (agents) used to cause illness or death in people, animals, or plants." Centers for Disease Control and Prevention website, "http://emergency.cdc.gov/bioterrorism/overview.asp"

[6] On February 19, 2010, the Justice Department, the FBI, and the U.S. Postal Inspection Service formally concluded the investigation into the 2001 anthrax attacks and issued an Investigation Summary. Dr. Ivins took his own life before charges could be filed against him. http://www.fbi.gov/about-us/history/famous-cases/anthrax-amerithrax

[7] The term "support" means both physical and financial support; specifically, access to laboratories such as research universities or pharmaceutical companies.

[8] Associated Press, "Trial of Mississippi man charged with sending ricin letters may be delayed," September 13, 2013.

[9] Shannon Richardson pleaded guilty on December 10, 2013 to sending the ricin-laced letters in an effort to frame her husband. http://www.usatoday.com/story/news/nation/2013/12/11/guilty-ricin-obama-bloomberg/3983865/

[10] Ricin is a small, toxic carbohydrate-binding protein found in castor oil beans. To be an effective BW agent, it must be extracted from the beans and purified to a concentration to deliver an aerosolized dose of 10-15 micrograms per kilogram. Consequently, for a person weighing 180 pounds at least 820 micrograms of ricin need to be present. (http://www.cdc.gov/biosafety/publications/bmbl5/bmbl5_sect_viii_g.pdf). In terms of sophistication, a ricin letter is a simple device and does not require a complex dispersion method since it is presumed that the person opening the letter is the intended target. Neither the exact concentration nor dispersal properties of the ricin in the letters have been made public; however, the concentration was high enough to set off detectors in the mail-processing facilities.

[11] Steve Coll and Susan B. Glasser, "Terrorists Turn to the Web as Base of Operations," Washington Post, August 7, 2005.

[12] A bioterrorist is one who simply employs biological weapon agents unmodified, while a terrorist biohacker is one who modifies a known toxin or biological agent with malicious intent.

[13] page I-7, "FM 3-11.9 Potential Military Chemical / Biological Agents and Compounds," United States Military, January 2005.

[14] page I-2, "FM 3-11.9 Potential Military Chemical / Biological Agents and Compounds," United States Military, January 2005.

[15] Ibid.

[16] Ibid.

[17] Ibid.

[18] The impact of BW agents on human health proceeds from organ failure and tissue destruction, but is ultimately defined by toxic effects on cellular functions, with the most severe being cell death. Consequently, before the effects are seen at the level of the organism, they occur on the molecular and cellular scale, and continue from the point of infection and even beyond the appearance of medical symptoms. Presymptomatic detection of early signatures of an infection could mitigate some threats.

[19] The study was conducted by JASON, an independent group of scientists operating through the MITRE Corporation, who advise the United States government on issues related to science and technology.

[20] The Joint Biological Point Detection System (JBPDS), a continuous environmental aerosol monitor, is currently available for point biosurveillance. These devices collect samples over a four-hour interval and then the sample is transported to a "central laboratory" for analysis. Detection of a biological agent in a city, for example, would require a large area of systems and technicians not only to collect samples but also to test them. The cost of extending the JBPDS beyond the few existing, strategic locations would be overwhelming. Advances in technology will undoubtedly produce compact, lower-cost automated detection systems that could be much more widely disseminated, but this then presents an increased risk for accidental or intentional false alarms and hence requires a rapid and highly accurate second-level validation.

[21] The value of 288 million is based on the census data available during the 2003 study by JASON.

[22] JASON, "Biodetection Architectures," February 2003.

[23] The National Strategy for Biosurveillance signed in July 2012 specifically says, "Where efforts since the tragic terrorist attacks of September 11, 2001, have focused largely on threats associated with the deliberate use of CBRN weapons, this Strategy embraces the need to engage in surveillance for WMD threats and a broader range of human, animal, and plant health challenges, including emerging infectious diseases, pandemics, agricultural threats, and food-borne illnesses."

[24] This classification of biohacker strategies was developed by John Wikswo, David Cliffel, and John McLean at Vanderbilt University and presented in December 2012 at the Johns Hopkins University Applied Physics Laboratory's Cellular Sensing Systems Workshop.

[25] Epitopes are specific amino acid sequences on the surface of a cell, or certain BW agents such as anthrax, that invoke a specific immune response. The unique amino acid sequences are identifiable traits of certain BW agents and are viewed as biomarkers. The concept of epitope can be extended to include any amino acid sequence that can be detected though a molecular affinity assay, such as aptamers (see, for example, Gold *et al.,* Aptamer-Based Multiplexed Proteomic Technology for Biomarker Discovery, PloS One, 5 (12): Article e15004, 2010).

[26] Gene expression dynamic inspection (GEDI) studies (S. Huang *et al*., Cell fates as high-dimensional attractor states of a complex gene regulatory network, Phys. Rev. Lett. 94:128701, 2005) demonstrate that HL-60 under different environmental conditions will present different genes throughout their transformation process to neutrophils, 168 hours later. So identification through gene expression at a given time point would identify two different agents. The concept of gene expression phase space and epigenetic attractors is treated in more detail in S. Huang and D. E. Ingber, A non-genetic basis for cancer progression and metastasis: Self-organizing attractors in cell regulatory networks, Breast Dis. 26:27-54, 2007.

[27] The Aum Shinrikyo cult in 1993 twice dispersed large amounts of anthrax around Tokyo using a variety of methods. The anthrax strain acquired by the cult was designed as a vaccine for cattle, however, and did not have any effect on humans. http://abcnews.go.com/Technology/story?id=98249

[28] Herfst S. et al, "Airborne transmission of influenza A/H5N1virus between ferrets," Science, 2012, 1534 - 1541.

[29] Jenifer Mackby, "Strategic Study on Bioterrorism," 2006.

[30] John Wikswo, "A top-down approach to cellular sensing: Platforms and mircofluidics," presented December 3-4, 2012.

[31] Lawrence Livermore National Laboratories, "Quickly Identifying Viable Pathogens from the Environment," S&TR, September 2010.

[32] Jason Committee, "Biodetection Architectures," February 2003.

[33] Jason Committee, "Biodetection Architectures," February 2003.

[34] S. Huang, G. Eichler, Y. Bar-Yam, and D. Ingber, "Cell fates as high-dimensional attractor states of a complex gene regulatory network," Physical Review Letter, 2005, Volume 94.

[35] G. Eichler, S. Huang, and D. Ingber, "Gene Expression Dynamics Inspector (GEDI): for integrative analysis of expression profiles," Bioinformatics, 2003, Volume 19, Issue 17, p2321- 2322.

[36] Sven Eklund, Roy Thompson, Rachel Snider, Clare Carney, David Wright, John Wikswo, and David Cliffel, "Metabolic Discrimination of Select List Agents by Monitoring Cellular Responses in a Multianalyte Microphysiometer," Sensors, 2009, 9, 2117-2133.

[37] Alexander Anderson and Vito Quaranta, "Integrative mathematical oncology," Nature Reviews: Cancer, Volume 8, March 2008, 227 - 234.

[38] Boyd Scott, "Fibercell Systems' Hollow Fiber Cell Culture System," Life Science Articles, August 2005.

[39] Walter Georgescu et al, "CellAnimation: an open source MATLAB framework for microscopy assays," Bioinformatics, 2011.

[40] Al-Kofahi, O., Radke, R., Goderie, S., Shen, Q., Temple, S., and Roysam B., "Automated Cell Lineage Construction: A Rapid Method to Analyze Clonal Development Established with Murine Neural Progenitor Cells," Cell Cycle, February 2006, 327-335.

[41] Geoergescu, W., Wikswo, J., Quaranta, V., "CellAnimation: an open source MATLAB framework for microscopy assays," Bioinformatics, November 2011,

[42] "What is R?" www.r-project.org

[43] Bernard Rosner, Fundamentals of Biostatistics, Brooks/Cole Cengage Learning, Seventh Edition 2011

[44] "Classification Using Nearest Neighbors," MATLAB 2013b Documentation.

[45] "Binary Integer Programming" MATLAB 2013b Documentation.

[46] Ibid.

[47] Tyson, DR., Garbett SP., Frick PL., and Quaranta V., "Fractional Proliferation: a method to deconvolve cell population dynamics from single cell data," Nature Methods, Volume 9, pages 923-928, September 2012.

[48] ImageJ Documentation, "www.rsbweb.nih.gov/ij/index.html," NIH, Bethesda, Maryland

[49] Dong-Hwee Kim and Denis Wirtz, "Focal Adhesion size uniquely predicts cell migration," The FASEB Journal - Research Communication, April 2013, Volume 27, p 1351- 1361.

[50] Nagasaki, A., Kanada, M., and Uyeda, T., "Cell adhesion molecules regulate contractile ring-independent cytokinesis in Dictyostelium discoideum," Cell Research, 19, 236-246, 2009.

[51] Webb, D.J., Parsons, J.T., and Horwitz, A. F., "Adhesion assembly, disassembly and turnover in migrating cells: over and over and over again," Nature Cell Biology, 4, E97-100, 2002.

[52] Pelham, R.J., and Wang, Y., "Cell locomotion and focal adhesions are regulated by substrate flexibility," Proceeding of the National Academy of Science, 94, 13661-13665, 1997.