Office of Graduate Studies

Dissertation / Thesis Approval Form

This form is for use by all doctoral and master's students with a dissertation/thesis requirement. Please print clearly as the library will bind a copy of this form with each copy of the dissertation/thesis. All doctoral dissertations must conform to university format requirements, which is the responsibility of the student and supervising professor. Students should obtain a copy of the Thesis Manual located on the library website.

Dissertation/Thesis Title: Automatically Improving Cell Segmentation in Time-Lapse Microscopy Images Using Temporal Context From Tracking and Lineaging

Author: Mark Winter

This dissertation/thesis is hereby accepted and approved.

Signatures:

Examining Committee

Chair ______________________

Members ______________________

____

____

____

____

Academic Advisor ______________________

Department Head ______________________

Automatically Improving Cell Segmentation in Time-Lapse Microscopy Images Using Temporal Context From Tracking and Lineaging

A Thesis

Submitted to the Faculty

of

Drexel University

by

Mark Winter

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

December 2016

Acknowledgments

The application of computational analysis to biology and microscopy is inherently a collaborative endeavor. I have been incredibly fortunate in working with so many talented biologists and computer engineers during my studies, and there can be no doubt that my advisor, my peers, and our collaborators improved this work immensely.

I must begin by thanking my advisor Andrew Cohen, who has been a mentor to me since I was in high school. He taught me to code, long before I ever considered graduate school, and he has been a source of great inspiration and *mostly* sound advice ever since I joined his lab. Our discussions and arguments have refined my understanding and shaped my ability to communicate the work within and beyond the lab. I am also so grateful to all my coworkers and friends in the lab. Eric, in particular, has been a constant sounding board for my ideas, no matter how silly they seemed, and his work on the original LEVER software set the stage for this work.

Biological systems are fascinating in their complexity, and it takes a true expert to communicate effectively with those of us who are not biologists. I am indebted to Sally Temple and her group at the Neural Stem Cell Institute for their patience and their desire to push the boundaries of both biological and engineering research. I very much enjoyed working with Maria Apostolopolou, Mo Liu, Susan Goderie, and Chris Bjornsson, thank you all for the countless hours you spent supporting us, we would be unable to do this work without your tireless efforts and guidance.

The path of research is often a meandering one, full of emotional highs and lows. I am so grateful to my family and friends who supported me when the path lead me downhill and were still there to celebrate with me when I came back up. Thank you for your unconditional love and support Mom and Dad. Sam, you have always been there for me throughout this process, I hope that I can repay your kindness as you pursue your own research. To my friends, Adam, Beth and Aubrey, thank you for being so supportive over the last few years, and to all my friends thank you for everything, I'm so glad to be a part of your lives, and that you are a part of mine.

Table of Contents

List of Figures

Abstract

Automatically Improving Cell Segmentation in Time-Lapse Microscopy Images Using Temporal Context From Tracking and Lineaging
Mark Winter
Andrew R. Cohen, Ph.D.

Over the past decade biologists and microscopists have produced truly amazing movies, showing in wonderful detail the dynamics of living cells and subcellular structures. Access to this degree of detail in living cells is a key aspect of current biological research. This wealth of data and potential discovery is constrained by a lack of software tools. The standard approach to biological image analysis begins with segmentation to identify individual cells, tracking to maintain cellular identities over time, and lineaging to identify parent-daughter relationships. This thesis presents new algorithms for improving the segmentation, tracking and lineaging of live cell time-lapse microscopy images. A new "segmentation from lineage" algorithm feeds lineage or other high-level behavioral information back into segmentation algorithms along with temporal context provided by the multitemporal association tracker to create a powerful iterative learning algorithm that significantly improves segmentation and tracking results. A tree inference algorithm is used to improve automated lineage generation by integrating known cellular behavior constraints as well as fluorescent signals if available. The "learn from edits" technique uses tracking information to propagate user corrections to automatically correct further tracking mistakes. Finally, the new pixel replication algorithm is used for accurately partitioning touching cells using elliptical shape models. These algorithms are integrated into the LEVER lineage editing and validation software, providing user interfaces for automated segmentation, tracking and lineaging, as well as the ability to easily correct the automated results. These algorithms, integrated into LEVER, have identified key behavioral differences in embryonic and adult neural stem cells. Edit-based and functional validation techniques are used to evaluate and compare the new algorithms with current state of the art segmentation and tracking approaches. All the software as well as the image data and analysis results are released under a new open source/open data model built around Gitlab and the new CloneView interactive web tool.

Chapter 1: Introduction

Recent advances in microscopy have made it possible to maintain viable cellular environments inside a microscope for weeks at a time. This enables long-term in vitro study of many important biological phenomena. Understanding the internal and external mechanisms that control stem cell fate is fundamentally important in developmental biology. Tissue growth and replacement therapies are also contingent on thorough comprehension of these mechanisms. Aging and cancer research benefit from accurately defining the factors that determine cell differentiation or continued proliferation. Live-cell microscopy is the only way to quantify the dynamics of cells as they develop and differentiate.

Live-cell microscopy can utilize transmitted light techniques such as phase contrast to keep cells alive *in vitro* for days or weeks, allowing the direct observation and recording of long-term development of cellular colonies. Live-cell fluorescent labeling may also be used alone or it can be used in concert with transmitted light microscopy to provide additional information. For example fluorescent markers could be used to periodically identify all cell nuclei or other subcellular structures of interest. Fluorescent labeling can also be used to provide information on cellular states, illuminating relative gene expression levels, or *e.g.* fluorescent ubiquitination cell cycle indicator (FUCCI) can be used to show the current phase of a cell in its life cycle[1]. Each of these techniques generates a significant amount of data in the form of time-lapse image sequences. It can quickly become infeasible to analyze data by hand from even a small experiment, as even a few proliferating cells can become dozens or hundreds of cells within a few days. This has prompted a collaborative effort between computational engineers, mathematicians, and biologists to build mathematical models and software tools to automatically analyze and quantify cellular behavior in time-lapse image sequences.

Biologists often analyze cellular proliferation using lineage trees. A cellular lineage tree or cellular family tree shows the parent-daughter relationships of all cells descended from a common ancestor, often referred to as a clone. In many cases, lineage trees are acquired by pencil-and-paper sketches or by manual software tools, which require identifying cells in each frame of a microscope video.

However since current microscope technology can allow for simultaneous capture of hundreds to thousands of cells using phase contrast imaging this presents a significant throughput problem for analysis. Completely manual lineage tree generation is also very time-consuming and error-prone. Importantly, without additional software support, a manual lineage tree does not allow for observation and quantification of individual cell behaviors. There are often subtle cellular behavioral differences indicative of changes in cell fate[2]. The discovery and quantification of these individual behavioral changes can pave the way for biological understanding of the underlying processes determining cell fate changes. It is paramount to develop software for automated identification, tracking and lineaging of individual proliferating cells from image sequence data. Equally important, these tools must support validation by expert users since even minor errors in the automated algorithms may drastically affect downstream clonal analyses.

The first step in automated lineage tree creation is segmentation. Segmentation is the process of delineating each individual cell in every frame of an image sequence. Pixels are identified as being background or interior to a specific cell. By their nature, segmentation algorithms are very sensitive to imaging conditions and to cell type. Robust segmentation is often very difficult and algorithms must be designed for a limited range of cell types and imaging paradigms. A key focus of this work is the segmentation from lineage algorithms, which incorporate behavioral information and temporal context to improve upon underlying segmentation algorithms. Once cells have been identified in each frame a tracking algorithm links cell detections over time to produce cell tracks.

Automated trackers connect cell detections in one or more sequence frames. There are many approaches to the multi-target tracking problem, in general however, the best connections are determined by optimizing a cost or probability function over a set of feasible tracking solutions. Accurate cellular motion models, encapsulated by the cost function, allow for robust tracking even in the presence of significant detection errors. This work utilizes a modified version of the robust tracking algorithm of Winter *et al.* to provide temporal context to refine segmentation and lineage results[3;4]. Lineage trees are created from individual cellular tracks by optimizing the probability or cost of a mitosis event for any tracks that begin after the first frame. Cellular lineage trees, acquired manually or automatically, are often used to quantify stem cell development. Lineage trees are powerful structural summaries of cellular development, providing quick display of cell cycle times and ancestry. Immunofluorescent staining indicating terminal cell fates at the end of an experiment can also be concisely displayed on the lineage tree (*e.g.* figure 4.3). Lineage trees also provide important high-level insight into organ and tissue development. Combining lineage information with individual cellular segmentation and tracking can provide biologists with a detailed understanding of the cellular dynamics that affect proliferation and differentiation. Software tools for acquiring and validating these cellular lineage trees are invaluable for quantifying these dynamics.

This research develops a new set of algorithms for automated correction of segmentation, tracking and lineaging. These algorithms, collectively referred to as "segmentation from lineage," provide the ability to identify, track and lineage hundreds of movies. The new algorithms iteratively improve segmentation and tracking using lineage information gathered from the user. This provides a rich data set containing cell delineation and identification in every frame, allowing for extensive analysis of live-cell dynamics from individual cells to clones, or across experimental conditions. Other information such as fluorescent markers indicating cell state, *e.g.* FUCCI, can be integrated to assist in automated processing as well as to gain insight into changes in cell dynamics[1]. These algorithms are also integrated into the LEVER lineage editing and validation software, providing an interface for fast, robust user validation and export to tools such a CloneView for open data visualization and collaboration[5;6].

1.1 Contributions

The contributions of the present work include:

- **New algorithms for cell segmentation** incorporating temporal information from tracking and lineaging together with human corrections to significantly reduce error rates compared to existing approaches. These algorithms include: Segmentation from lineage, tree inference, learn from edits and pixel replication[6;7].

- **The LEVER lineage editing and validation software** is a free and open source tool for automated cell segmentation, tracking and lineaging. LEVER provides an interface for users to correct automated results[4;5].

- **Biological applications and data analysis**, the software tools and algorithms described here have been used for identifying intrinsic behavioral differences in embryonic neural progenitor cells isolated from the anterior and posterior mouse cerebral cortex[6]. The same tools were used for the study of age-related behavior changes in mouse sub-ventricular neural stem cells[8;9]. The program has been applied to 2-D and 3-D fluorescent images[10], phase contrast images of hematopoeitic stem cells[11], embryonic and adult neural stem cells[4;6;9].

- **An open source/open data** approach that makes all of the results and analysis, as well as the original data available for interactive browsing online. All algorithms described in this work are available free and open source hosted at the Drexel bioimaging lab's Gitlab repository: `https://git-bioimage.coe.drexel.edu`. The analysis results and biological data are stored online and hosted through the interactive web application CloneView: `http://bioimage.coe.drexel.edu/CloneView`.

1.1.1 Segmentation from lineage - improving cell segmentation using the cellular lineage tree

The segmentation from lineage algorithms described in chapter 4 use lineage trees as structural knowledge for improving segmentation and tracking. The framework is versatile and can incorporate other sources of structural knowledge as discussed in section 5.1.2 and in Winter *et al.*[5;9;11]. Segmentation from lineage works by iteratively reevaluating segmentation results on a frame-by-frame basis. In each frame new segmentations are chosen so as to best fit the evidence from the cellular motion model and the constraints of cellular lineage trees or user corrections. The segmentation from lineage framework using lineage information and interactive user corrections, along with a biological application to embryonic neural progenitor cell development is presented in Winter *et al.*[6].

1.1.2 Tree inference - using cellular behavior constraints to improve automated lineaging

Tree inference is integrated into the segmentation from lineage framework described in chapter 4. This approach is used to improve initial automated lineage tree construction using cellular behavior constraints, for example cells take some time to pass through their life cycle and cannot divide too quickly. Similarly, cells being tracked cannot appear or disappear without cause. Using these constraints significantly improves initial lineage construction as well as fully automated segmentation from lineage accuracy.

1.1.3 Learn from edits - automated propagation of user corrections to improve cell tracking

A key aspect of the LEVER software is the correction interface, allowing users to edit automated segmentation and tracking results. The learn from edits algorithm is integrated into LEVER and the segmentation from lineage interface to propagate user corrections using temporal context provided by the tracking algorithm. The integrated algorithm is detailed as part of chapter 4, describing the segmentation from lineage framework. An early version of the learn from edits algorithm is also discussed as part of Winter *et al.*[5].

1.1.4 Multitemporal association tracking - providing temporal context to improve segmentation accuracy

Automated cell tracking is a key component of any cellular behavior analysis tools. Trackers maintain cellular identity over time by associating the most-likely cell detections across frames. The multitemporal association tracker has been shown to be very effective for axonal transport analysis and using a different motion model has been used to track neural progenitor cells[3;4]. A recent simulated particle tracking challenge also showed multitemporal association tracker was effective and competitive with other state of the art particle trackers[12]. The multitemporal association tracker is at the heart of the segmentation from lineage, tree inference and learn from edits algorithms, informing the most likely segmentation ensembles in each frame according to the motion model and constraints. As such we also quantify the accuracy of multitemporal association tracking on validated neural progenitor cell movies. The integration of the tracking algorithm within the segmentation from lineage, as well as evaluations of its accuracy across multiple object types represent an important component of this work.

1.1.5 Pixel replication - partitioning touching elliptical cells

In order to track and analyze cells over time, each cell must be segmented, *i.e.* delineated in every frame. Segmentation is one of the most difficult image processing tasks and affects all downstream processing such as tracking and lineaging. There are myriad approaches to the task of identifying cells in each image frame, however nearly all algorithms share the need for some type of noise removal, image thresholding and object partitioning. This work utilizes the segmentation from lineage framework to minimize errors in segmentation, in particular it can alleviate errors in noise removal and image thresholding by choosing the most probable foreground regions from a set of thresholds. The segmentation from lineage algorithm also assists in the cell partitioning task by providing a specific number K of cells that are likely to lie in a foreground component. An accurate cell partitioning algorithm, however is a key component of the segmentation from lineage framework, since the refinement framework must be able to partition initial segmentation results in the case that a cluster of cells was mislabeled as a single component. This work presents a new technique for identifying cells in foreground regions, referred to as pixel replication. Pixel replication is compared with standard watershed techniques for separating cells and is shown to be very effective for partitioning clusters of elliptical cells, significantly increasing segmentation accuracy relative to the watershed algorithm. Pixel replication was also shown to be more accurate than the spatial Gaussian mixture models used in Winter *et al.*[6].

1.1.6 LEVER lineage editing and validation software

The LEVER software tool provides a user interface for automated lineage editing as well as providing an interface for correcting automated results. The segmentation from lineage interface, as well as tree inference and learn from edits, have been integrated into the LEVER software tools to minimize the need for human correction of results. LEVER can also export to CloneView, a web-based application for interactively visualizing microscope data along with the LEVER analysis results. Together LEVER and CloneView support open data sharing, as well as research collaboration. A description of the use of LEVER interfaces, as well as segmentation from lineage interface is published in Winter *et al.*[5].

1.1.7 Analysis of embryonic and adult neural stem cells

The LEVER analysis tools have been applied to the validation and correction of hundreds of data sets across multiple experiments These experiments have been comprised of very different cell types including hematopoietic stem cells, embryonic neural progenitors, adult neural progenitors, cancer cells, etc.[6;9;11] . In this work the results of two important biological applications are presented. First, we show an analysis of intrinsic behavioral differences between embryonic neural progenitor cells (NPCs) cultured from the anterior and posterior of the mouse cortex. Detailed discussion and results along with a description of the refinement framework are published in Winter *et al.*[6]. We also present an analysis of morphological and phenotypic differences among mouse neural stem cells cultured from the sub-ventricular zone of the mouse cortex across four age groups. Interestingly, mouse type C neural stem cells appear to develop very differently as the mouse ages, but rather than a linear decrease in proliferation rate there is a valley in proliferation at 18 months (middle age). The results and analysis of aging-related behavior changes in aging neural stem cells are in review[8]. Additional results using a secondary process segmentation to identify changes in cell process dynamics are published in De La Hoz *et al.*[9]. Both these applications demonstrate the power and versatility of the new algorithms.

1.2 Organization of the thesis

Chapter 3 is a brief introduction to the use of the LEVER software and segmentation from lineage. This work is typeset here and was recently published as an application note in Winter *et al.*[5].

Chapter 4 provides an introduction to the segmentation from lineage, tree inference and learn from edits algorithms. We also present two applications of the segmentation from lineage framework to biological questions of interest. The first biological application utilizes the framework to identify intrinsic neural progenitor cell (NPC) differences in cells cultured from the anterior and posterior of the mouse cortex. This application along with a discussion of the segmentation from lineage framework is fully detailed in Winter *et al.*[6]. The second biological application identifies important morphological and proliferation behaviors among mouse neural stem cells derived from the subventricular zone across the life-time of mice. These results are a step toward a much more detailed understanding of the aging brain and are further discussed in Apostolopoulou *et al.*[8]. Validation strategies for evaluating our segmentation results are also discussed in the methods section.

In chapter 5 we introduce the multitemporal association tracker and provide a technical description of its use and integration in the segmentation from lineage algorithms. Multitemporal association tracking (MAT) has been used for fully automated tracking in several disparate applications. MAT was originally developed for tracking brain-derived neurotrophic factor transport in the axons of nerve cells[3]. Using a modified cost function MAT effectively tracked proliferating neural progenitor cells[4]. MAT was also a top performing entry in an international particle tracking challenge, and was the lowest error rate entry for 3 of 4 signal-to-noise levels in the microtubule transport category[12]. Chapter 5 focuses on application of MAT to cell tracking and detailing the integration of MAT into the segmentation from lineage framework.

Chapter 6 introduces the pixel replication algorithm for partitioning touching cells. Pixel replication is an improvement to spatial Gaussian mixture models that better partitions clusters of cells into individual components. The pixel replication algorithm uses a Euclidean distance transform to better approximate a mixture of Gaussian distributions from a threshold image. This chapter details the algorithm and provides some insight into generalized elliptical probability distributions and elliptical shape model fitting. Pixel replication is compared to standard watershed partitioning techniques as well as the spatial Gaussian mixtures discussed in Winter *et al.*[6] and chapter 4. A detailed discussion of the pixel replication algorithm and its applications to cell partitioning can be found in Winter *et al.*[7].

Additional biological materials and methods for each of the applications discussed in chapter 4 can be found in the appendix.

Chapter 2: Background

2.1 Live-cell microscopy

Modern microscopes use myriad techniques to visualize and interrogate the dynamics of living cells[13]. New techniques are being developed to improve microscope physical and temporal resolution even beyond the usual diffraction limits of visible light[14]. This work focuses on the analysis of 2-D phase contrast microscopy image sequences as well as some 2-D and 3-D live-cell fluorescent image sequences[13].

A standard bright-field microscope images cells by measuring light intensity (generally from an attached illumination source) that is transmitted through the cell and surrounding medium. The transmitted light can be focused onto an eyepiece for human use or onto a light-sensitive array for image capture. However, living cells generally have an index of refraction that is very near the surrounding medium (*e.g.* water). Cells therefore appear nearly clear under a bright-field microscope and it is difficult for either humans or computational algorithms to distinguish individual cells in image sequences. In the 1930s Frits Zernike created the phase contrast microscope[15;16]. Phase contrast microscopes use a specialized set of lenses that phase-shift the background illumination (passing through the medium but not the samples) by about 90°. The phase shifted background light interferes with light that passed through the cell samples, making the image intensity at each pixel proportional to the relative phase difference between light passing through the medium and light passing through the cells. This generally creates much greater image contrast because cells are somewhat thicker and have a slightly higher index of refraction than the surrounding medium, creating a large phase differential. Living cells also exhibit a “halo” effect, which creates high intensity pixels surrounding cells due to refraction of light near the boundary of the cell. The halo is an artifact of phase contrast imaging but as discussed in more detail in chapter 4 this can be used by segmentation algorithms to improve cell separation.

A second method for imaging living cells or subcellular structures is live-cell fluorescence labeling.

Originally studied in the 1960s and 1970s the green fluorescent protein (GFP) was identified in the *Aequorea victoria* jellyfish[17]. The protein fluoresces green when exposed to ultraviolet light. Attaching a GFP DNA sequence to the gene sequence of a protein of interest will fluorescently tag the proteins, indicating their level of expression within the cell. This allows the imaging of subcellular structures in living cells by periodically exciting the fluorescence and capturing the results[13]. Multiple fluorescent protein sequences can be attached to different proteins for multi-channel imaging. Additional fluorescent signals such as fluorescent ubiquitination cell cycle indicator (FUCCI) can also be used for live-cell imaging[1]. Confocal microscopy, which uses an optical pinhole to remove background light that does not originate at the correct focal distance, can be used along with fluorescent excitation to create 3-D volumetric imaging of cells[13].

Fluorescent imaging provides a powerful method for interrogating cellular state in both 2-D and 3-D. However, periodic laser excitation creates excess heat in cells and can cause phototoxicity and death if cells are imaged too often. This is the case for the embryonic and aging neural stem cell experiments discussed in chapter 4. These cells were imaged using only phase contrast microscopy, however after the completion of the experiment the cells can fixed and stained using immunohistochemical markers to indicate the differentiation of cells. Final frame staining is very important for identifying behavioral differences among fate-committed cells in the aging neural stem cell data, discussed further in chapter 4. Another method for long term cellular imaging is to combine transmitted light techniques, such a phase contrast, with occasional imaging of fluorescent signals to provide cellular state information and to assist in identifying cells. Some imaging examples shown in chapter 6 use this combination of transmitted light and fluorescent imaging.

2.2 Probability of events and conditional probability

Probability theory is at the heart of machine learning and pattern recognition. This section will introduce a few concepts of probability theory used throughout the work. Additional details can be found in Bain & Englehardt as well as Papoulis[18;18]. Probabilities can be defined in several contexts, this section will focus on the probability of events defined as regions in a sample space. For example, when rolling two dice, we could define an “event” that the sum of the dice is equal to

7, and the sample space would be all possible outcomes of rolling two dice.

Figure 2.1 (top) shows a Venn diagram of two events of interest A and B in the sample space S. Sampling uniformly from S we define the *probability* of event A occurring as $P(\mathrm{A}) = \frac{|\mathrm{A}|}{|\mathrm{S}|}$ where |A| and |S| indicate the area of region A and the sample space S, respectively. Similarly, the probability of event B occurring is $P(\mathrm{B}) = \frac{|\mathrm{B}|}{|\mathrm{S}|}$. We define the joint probability of events A and B as probability that both A and B occur simultaneously, *i.e.* the region of intersection between events A and B. The joint probability of A and B is written $P(\mathrm{A}, \mathrm{B}) = \frac{|\mathrm{A} \wedge \mathrm{B}|}{|\mathrm{S}|}$, where $\mathrm{A} \wedge \mathrm{B}$ represents the region of intersection, as shown in figure 2.1 (top).

One of the most important concepts in probabilistic inference and learning is "conditional probability." We define the conditional probability of event A given B as the probability that event A will occur if event B is guaranteed to happen. Conditioning on B means that rather than sampling from anywhere in S, B must occur, so the sample space must be the region B. Thus Bayes' famous formula defines conditional probability of A given B as $P(\mathrm{A}|\mathrm{B}) = \frac{|\mathrm{A} \wedge \mathrm{B}|}{\mathrm{B}}$, the area where events A and B occur, divided by the smaller sample space where B occurs. Figure 2.1 (bottom) shows this conditional region, the dotted line around region A indicates that only the intersection region is possible since event B is guaranteed. The relationship between joint and conditional probability is used throughout machine learning to infer models and events given observation data[19]. This work makes heavy use of these equations in chapter 5 for defining the probabilistic interpretation of segmentation from lineage.

2.3 Gaussian mixture models

Gaussian mixture models are used for partitioning clusters of cells as discussed in chapter 4 and they are the basis of the pixel replication technique discussed in chapter 6. This section provides a brief introduction to the Gaussian distribution and to mixtures of Gaussian distributions. Additional details about Gaussian distributions and the family of general elliptical distributions are discussed in chapter 6 as well as in Gomez *et al.*[20]. Further details on mixture models and clustering can be found in Theodoridis & Koutroumbas, and in Bishop[19;21].

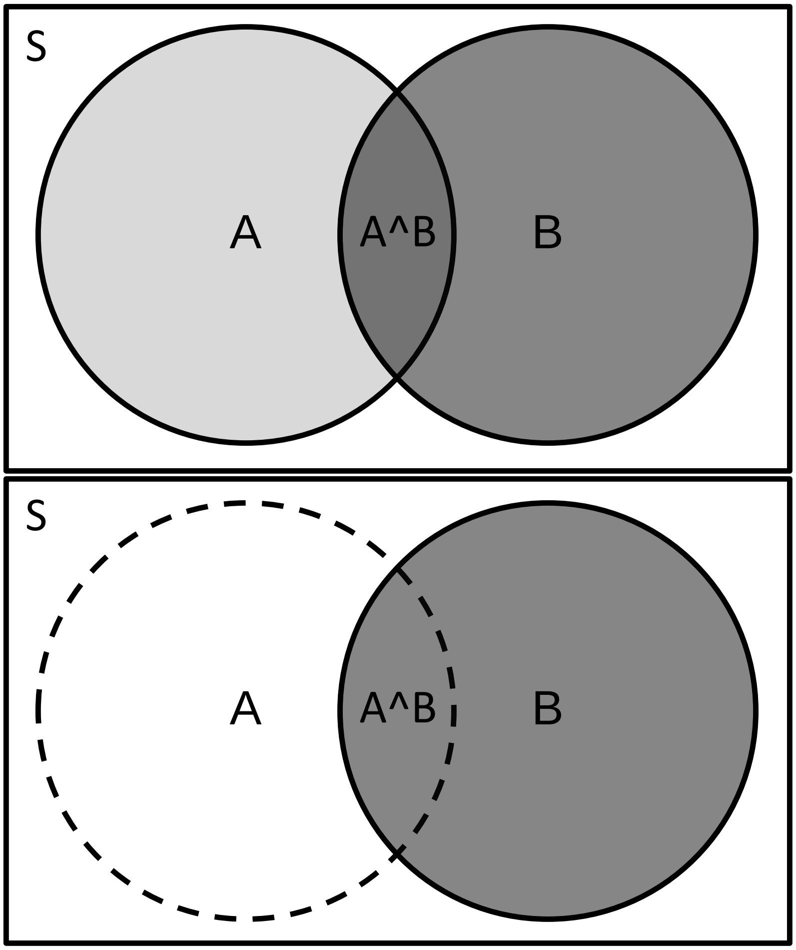


Figure 2.1: Probability and conditional probability of events. Uniform sample space S contains two sub-regions that define events A and B (top). The region $A \wedge B$ is the intersection between regions A and B where both events can occur. In the case that event B is given, only the subregion in B is feasible (bottom). The area of the intersection region $A \wedge B$ then defines the conditional probability of A given B has been observed.

2.3.1 Gaussian distributions

The Gaussian distribution is one of the most powerful, and certainly the most commonly applied probability distributions throughout all of science. Largely this is a consequence of the law of large numbers and the central limit theorem, which identify a large class of statistical sampling experiments that converge toward the Gaussian distribution when repeated many times. The one dimensional Gaussian probability density function is defined as,

$$\mathcal{N}(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

The standard normal distributions has mean $\mu = 0$ and variance $\sigma = 1$. This produces the well-known bell curve with the peak at the mean.

The d-dimensional Gaussian density function is similarly defined as,

$$\mathcal{N}_d(\mathbf{X};\mu,\Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left\{-\frac{1}{2}(\mathbf{X}-\mu)^T\Sigma^{-1}(\mathbf{X}-\mu)\right\}.$$

Geometrically, Gaussian distributions have elliptical isocontours defined by the d-dimensional ellipse equation $(\mathbf{X}-\mu)^T\Sigma^{-1}(\mathbf{X}-\mu) = C^2$[19]. Where C^2 defines the particular isocontour and the eigenvalues of the covariance matrix Σ define the distribution shape. The elliptical shape of the Gaussian makes it an excellent shape model for cells. In chapter 6 we also consider other elliptical probability distributions and derive relationships between the covariance of each distribution and its associated isocontours.

2.3.2 Probabilistic mixture models

Probabilistic mixture models are a set of powerful techniques used throughout statistics and machine learning for data clustering[19;21]. Mixture models can also be used to empirically identify distributions based on observed data or to generate new simulations from a given model. Gaussian mixture models are particularly common and expectation maximization algorithms exist for fitting a Gaussian mixture model to observations in almost all statistical software packages.

A probabilistic mixture model is a composite or linear combination of component distributions. We denote the i-th component distribution probability density $g(\mathbf{X}; \Theta_i)$ for any vector $\mathbf{X} \in \mathbb{R}^d$ where Θ_i are the parameters defining the i-th distribution. For example a Gaussian mixture model has $\Theta_i = (\mu_i, \Sigma_i)$, where μ_i and Σ_i are the mean and covariance of the i-th Gaussian distribution. There is no requirement for mixture components to come from the same family of distributions, however there is no fitting algorithm for mixtures of generalized distributions, so a non-linear optimization approach would need to be used. A K component mixture model density is defined in terms of its component distributions as,

$$f(\mathbf{X}) = \sum_{i=1}^{K} p_i g(\mathbf{X}; \Theta_i).$$

Where p_i is referred to as the mixing proportion or mixing probability of the i-th distribution, and $\sum p_i = 1$. In the case of a Gaussian mixture model, Gaussian densities make up the components of the mixture distribution with individual mean and covariance per component,

$$f(\mathbf{X}) = \sum_{i=1}^{K} p_i \mathcal{N}(\mathbf{X}; \mu_i, \Sigma_i).$$

Chapters 4 and 6 detail the use of Gaussian mixture models to represent cell shape.

2.4 Related work

2.4.1 Lineage analysis

Lineage analysis is regularly used to identify population statistics of proliferating cells, *e.g.* histograms over terminal cell fates. Gomes *et al.* manually extracted over a thousand rat retinal progenitor lineages to argue for a stochastic model of cell fates[22]. Similarly, manually analyzing thousands of B-cells over single generations, Duffy *et al.* proposed a stochastic race-to-fate model for activated B-cell fate determination[23]. In the work of Giebel and Punzel, manual lineage analysis revealed that differentiated blood cells are derived from hematopoietic stem cells[24]. These population level features however, cannot capture individual cellular behaviors. Cohen *et al.* showed that an important aspect for the analysis of cellular fates is observing the dynamic behaviors (motion and morphology) of individual cells[2;25]. By applying algorithmic information theoretic tools to individual dynamic behaviors before mitosis, they were able to predict the mitotic fate of retinal progenitor cells. Individual cellular behavior tracking is also exceedingly important for understanding highly plastic cells such as macrophages[26].

Software tools have been created for manual cellular tracking and lineaging. Schroeder *et al.* introduced "Timm's Tracking Tool" which requires a user to click on every cell in every frame to establish cell tracks and parent-daughter relationships[27;28]. These tools manually capture individual motion for all cells in a clone, however no size or morphological information is available because cells are not segmented either automatically or manually. Manual tracking and lineaging is also very time-consuming on even moderately sized clones since each cell must be manually identified in every frame. Our goal is to provide tools for full user validation of the automated cell segmentation, tracking and lineaging, without requiring manual adjustment of every cell.

Several automated tracking and lineaging applications have also been written. Li *et al.* created a software package to automatically track and lineage proliferating cells from phase contrast image sequences[29]. The automated tracking was very accurate with error rates below 5%, however the algorithm was only run on five movies and the model was very complex requiring over 1 million lines of code. A graphical model for tracking and lineaging in 3-D fluorescence sequences was introduced by Kausler *et al.*[30]. This tool constrained the model by minimum cell cycle times to increase the robustness of automated lineage creation. Another lineaging system was developed by Fernandez *et al.* using a single frame bipartite max-flow tracker to find lineages of plant cells in 3-D image sequences[31]. This algorithm is specific to the imaging setup and has been used only for lineaging plant cells. None of these software tools feature an editing or validation interface for improving upon automated results. The first automated cell lineage construction tool was created by Al Kofahi *et al.*[32]. This tool utilized a graph-based bipartite assignment tracker to automatically generate cell lineages from phase contrast sequences. Though it did not contain a correction interface, their decision to display lineage and cell tracking results side-by-side inspired the LEVER lineage editing and validation interface discussed in this work.

Recently plugins for the Icy bioinformatics framework have added support for active contour algorithms for cell segmentation and tracking[33]. CellProfiler also contains modules for tracking proliferating cells automatically[34]. Both Icy and CellProfiler focus on fluorescent image data and do not support correction of lineaging or tracking results. Supersegger from Stylianidou *et al.* has been used for automatic lineaging of prokaryotic bacterial cells in 2-D phase contrast[35]. The suppersegger software is fully automated, using the watershed transform to identify and partition overlapping cells. The software developed by Amat *et al.* for lineaging 3-D fluorescent data has been used to analyze early Drosophila development. Amat's tool also supports user editing of the automated lineaging results, however the lineaging tool and the segmentation is specific to 3-D fluorescence images. The lineage mapper software from Chalfoun *et al.* is an automated tracking tool that uses feedback from the tracking to possibly split incorrectly segmented cells[36]. This approach is similar to the segmentation from lineage technique discussed in this work, however, lineage mapper does not support a validation interface for human correction. Also, because lineage mapper relies on external segmentation routines, it can only split detections and cannot attempt to add new detections in the case that no cell was initially identified in a region.

2.4.2 Tracking

Multitarget tracking is a widely studied problem in many fields including ballistics, video surveillance and biological image sequence analysis[37]. Tracking techniques can generally be classified as either track-before-detection or track-after-detection methods. Track-before-detection methods, sometimes called model-based methods, generally segment and track objects simultaneously, *e.g.* by spatio-temporal active contour segmentation or by color histogram-based region tracking[38–40]. These methods can be very accurate and may simplify noise handling because segmentation and tracking noise are dealt with in the same manner. However, by their nature, track-before-detection methods are very dependent on the sensor or imaging modality and are less robust in high object density situations. In contrast, track-after-detection approaches take a set of detections or segmented objects as input and find the association of detections into object tracks that is optimal in some sense. Track-after-detection methods are more independent of the underlying imaging modality and more robust to high object densities. This is the approach that we have adopted.

Modern multitarget tracking techniques evolved from research in ballistics tracking[41]. In these applications, the set of detections in each frame, *e.g.* radar detections, are the implicit output from the sensors being used. Multitarget tracking research focused, then, on the problem of aggregating sensor detections in distinct frames into representative object tracks. This early research culminated in Reid's presentation of a multiple hypothesis tracking (MHT) framework for indefinitely maintaining and updating the complete set of track hypotheses[42]. The optimal MHT algorithm is in the class of NP-hard problems however, requiring exponentially more memory and computational effort for each additional frame of detections considered. Multitarget tracking algorithms must therefore implement some approximation to optimal MHT, which fall into two categories. First, single frame assignment approaches use either cost or likelihood functions to assign observations to hypothesized paths. The assignment of observations to tracks is done using only data from a single time instance. This approach is particularly common in biological analysis.

Recently, considerable research has been devoted to algorithms which solve the data association problem over multiple frames simultaneously. Known as multidimensional assignment (MDA), these algorithms calculate likelihoods for assigning detections to tracks over N frames, instead of just a single frame, extending the bipartite matching formulation of the data association problem[43]. Since multidimensional assignment is known to be NP-hard for $N > 2$, solutions require approximation techniques. Poore *et al.* approximate the multidimensional solution using Lagrangian relaxation techniques to recursively reduce the problem to a 2-dimensional case[44]. Other approximation techniques such as GRASP have also been applied to the MDA problem[45]. Qian and Medioni apply Markov Chain Monte Carlo (MCMC) random search techniques to solve the multitarget tracking problem[46]. MDA approaches generally outperform single frame assignment algorithms especially for high object density situations. However multidimensional assignment is much more computationally demanding than single frame assignment and care must be taken in implementation to ensure that solutions accurately approximate the NP-hard MDA solution. The difficulty in implementing these techniques has hampered the widespread adoption of multidimensional assignment. Both bipartite and multidimensional assignment perform poorly in the presence of significant amounts of detection noise. This motivated the development of multitemporal association tracking for biological tracking applications, discussed in chapter 5.

Several recent trackers have been specifically developed for tracking cells and subcellular structures in biological imaging. Trackmate is a plugin for the popular FIJI and ImageJ image analysis tools[47;48]. Trackmate is an extension of the linear assignment tracker written by Jaqaman *et al.*, originally designed for single particle tracking[49]. CellProfiler also uses a linear assignment optimization for tracking cells over time[34]. The multitemporal association tracker detailed in chapter 5 is also compared with a linear assignment tracker similar to Trackmate. For both simulated axon transport data and validated stem cell tracking the multitemporal association tracker achieves significantly lower error rates, the full comparison is discussed in chapter 5.

2.4.3 Cell Partitioning

Cell segmentation is the process of detecting each cell in every image frame. This process can be broken into several steps, denoising, thresholding and identification of cells in thresholded regions. Though often overlooked or combined with other steps, identifying the number of cells in a thresholded region is a difficult and error-prone task. Segmentation from lineage also requires an accurate technique for splitting foreground components into a given number of cells.

The most common approaches to object partitioning are techniques based on watershed catchment basins[50]. These first transform the threshold image, creating a distance image where each foreground pixel value represents the distance to the nearest background pixel. The boundaries separating basins containing regional maxima in the distance image are used as cell boundaries. watershed techniques are generally effective in the absence of noise in the threshold image used to create the distance transform, and when objects are relatively circular. However, watershed techniques are very susceptible to over-splitting in the case of elliptical cells or when the threshold boundaries are perturbed by significant noise.

Several edge fitting approaches have also been used for cell partitioning[51]. Edge or boundary geometry algorithms such as active contours and concave point detection require reliable edge and gradient information to accurately partition cells. Particularly in 2-D phase and bright-field images there is minimal gradient between overlapping cells, making these approaches inaccurate. Lin *et al.* integrated a gradient magnitude information with a standard watershed transform to provide better separation when gradient information was available[52]. Similarly, edge information as well as fluorescent information can be used to weight the distance transform used in pixel replication to assist in partitioning cells. These data fusion techniques have the advantage of remaining robust even when minimal gradient information is available.

Recently, density estimation techniques have been proposed for clustering and cell separation[53;54]. These techniques use assumptions about the concentration of fluorescent proteins inside each cell to assign voxels to individual cells. Density-based approaches require that fluorophore concentration increases towards the center of cells this occurs *e.g.* when using nuclear fluorescent markers. These approaches can be very effective in 3-D environments where fluorescent protein density is concentrated in the nucleus and the cells are spherical. However, density-based approaches cannot be used for clustering in 2-D images or 3-D fluorescence images where the fluorescent protein concentrations are variable throughout and between cells.

Another approach is to fit a weighted sum of Gaussian distributions to the spatial locations of each threshold pixel[6;55;56]. These Gaussian mixtures (GMs) have been used extensively in fitting color or intensity histograms; less common is their application directly to the spatial coordinates of the pixels. There are two advantages to using GM compared to watershed and edge-based techniques. First, the GM fit is optimized over every pixel in the foreground, while the watershed considers only regional maxima of the distance image, and edge-based techniques utilize only boundary information. This makes the GM more robust for noisy images. Second, Gaussians have a natural elliptical boundary and ellipsoids are a good model for the morphology of a wide variety of cell types[19]. However, the mass distribution of a Gaussian is very different from a “flat” threshold image. As discussed further in chapter 6, the pixel replication technique takes advantage of the Gaussian elliptical model along with the distance transform to improve on this mass distribution and better partition foreground regions into elliptical components.

Chapter 3: An introduction to the LEVER lineage editing and validation tools

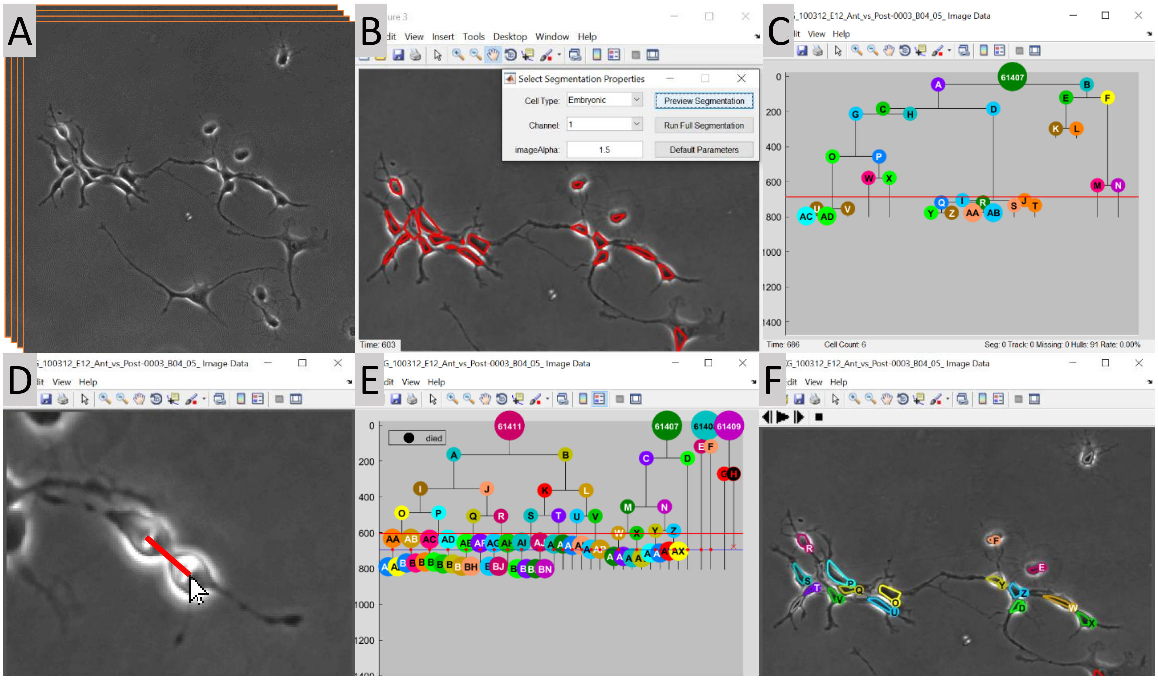
New and rapid development of microscope imaging technologies such as live-cell phase and fluorescence microscopes have enabled biological researchers to focus on the analysis of living cell behaviors. Manual analysis of the data is difficult and time consuming. Behavioral changes in cells may also be subtle and not easily identified by a human observer. These challenges necessitate the development of computational tools.

LEVER combines lineage analysis with the segmentation and tracking of individual cells. This allows the quantification of individual cellular properties and their changes across generations or between cells of different fate. In a twist on the usual processing flow, LEVER uses population information from the lineage tree to refine the segmentation results, informing the number of cells in each connected component of foreground pixels and greatly reducing error rates [6]. LEVER is designed so that errors in the automated results can be easily corrected. Extensions to LEVER supporting 2-D and 3-D multichannel fluorescence time-lapse images are also under active development [10;11].

LEVER images can be imported using the integrated OME bioformats library [57], or read directly (figure 3.1A). The segmentation preview window allows the user to compare results from several cell detection algorithms (figure 3.1B). Next, segmentation results are associated between frames using multitemporal association tracking (MAT). MAT has been effectively applied in a variety of applications all using the same implementation [3;4;6;11;12] . Tracking results are next used to form the lineage tree (figure 3.1C), or the tree can be specified by manually identifying mitosis events (figure 3.1D). Once the lineage tree is complete, the initial segmentation and tracking can be refined. In practice, we recommend this task be done in conjunction with validation, so any errors can be corrected online. Segmentation from lineage processes each frame in sequence making sure that every cell on the active lineage trees has a corresponding segmentation. As LEVER automatically corrects segmentation, tracking results are also updated automatically. Figure 3.1E,F shows a set of four interacting trees corrected using the segmentation from lineage. Error rates are estimated from the number of edits required to fully correct the analysis.

Once a tree has been fully validated, descriptive statistics such as phenotype tags, cycle times, average and standard deviations of size and speed for each cell can be exported to any spreadsheet. All of the LEVER data structures are described in `https://git-bioimage.coe.drexel.edu/opensource/lever`.

LEVER requires a 64-bit Windows environment to run. LEVER takes advantages of multi-core CPUs if available. Automated segmentation generally takes between 1-2 hours for large image data sets. Automated tracking and lineaging usually takes only a few minutes. The LEVER automated processing can be run in bulk on a collection of movies to avoid a processing bottleneck. Manual lineage identification generally takes only 15-30 minutes even for highly proliferative cells. Detailed documentation for using and extending LEVER is available at `https://git-bioimage.coe.drexel.edu/opensource/lever`.



Cell Type: Embryonic
Channel: 1
imageAlpha: 1.5

Figure 3.1: Overview of the LEVER processing pipeline. Cells in the input image sequence (A) are automatically identified. Cell detection algorithms and parameters are tested on single image frames using the segmentation previewer (B). LEVER automatically creates an initial lineage tree based on cell tracking (C). The user quickly identifies all mitotic events to specify a corrected lineage tree in minutes (D). Using four interacting lineages (E) LEVER segmentation from lineage can automatically correct most errors. Remaining errors are corrected manually during the refinement process (F).

Chapter 4: Computational analysis using segmentation from lineage

Cellular lineage analysis, understanding the factors that determine how quickly proliferating cells reproduce or differentiate, is a fundamental aspect of developmental biological research. Individual behavioral data of cells combined into lineage trees provides valuable insight into both intrinsic and extrinsic factors that are involved in tissue development. Measuring individual behavioral differences requires automated algorithms to assist with identifying, tracking and lineaging cells. Phase contrast is among the most challenging live imaging modalities for automatic analysis due to large variability in appearance within and between individual cells. In general, humans are better able to correctly identify and track cells than the best available software, but manual tracking is prohibitively slow. In order to efficiently analyze time-lapse phase image sequences of proliferating cells, we combine human visual capabilities with automated image analysis algorithms. However, human validation is essential in order to guarantee correct lineages since even a single tracking mistake can often greatly impact lineaging algorithms.

This chapter introduces the segmentation from lineage technique, which uses high-level information, such as a corrected cellular lineage, to improve the automated segmentation and associated tracking results. The process is most effective when run in a semi-automated fashion, allowing a human observer to correct any mistakes during refinement. This combination of automated processing and human validation has resulted in a significant reduction in error rates over fully automated approaches, as well as a significant reduction in human effort. The refinement algorithm runs across each frame using the lineage and track information to identify the most likely cell segmentations while also preserving the validated lineage structure. Segmentation from lineage has been integrated into the LEVER lineage editing and validation tool[5;6]. We have used the integrated refinement process to analyze two separate biological experiments, comprising hundreds of cell clones.

First, using segmentation from lineage we processed 160 clones comparing embryonic neural progenitor cells cultured from anterior and posterior regions of the mouse cortex. These regions of the mouse brain are structurally distinctive, the anterior region of the mouse brain is smooth while the posterior region is folded. Cells from both regions of the cortex were cultured identically (see Appendix A.1). Both the individual and lineage behaviors of anterior and posterior neural progenitor cells (NPC) were observed to be markedly distinctive. This suggests that intrinsic programming of the cells is different for the anterior and posterior regions of the mouse brain. Identification of these intrinsic differences could lead to better understanding the formation of complex neural structures in the brain[58].

In the second application of the framework, we processed 167 clones of mouse neural stem cells cultured from the sub-ventricular zone (SVZ) of mice aged 2, 6, 18 and 22 months. The sub-ventricular niche is one of the only regions in adult humans where neural progenitors cells can be found[59]. Previous work has shown that there are significant differences in the neural progenitor cells (NPC) in the sub-ventricular zones of mice and humans as they age[59]. Using the segmentation from lineage interface, we discovered significant behavioral differences in type A and C neural stem cells (NSC) of all ages. There is also a surprising and distinctive U shaped decrease in proliferation of type C cells with a minimum around 18 months, returning to nearly the same proliferation rate at 22 months old as at 2 months old. This coincides with variations in nuclear migration and with cell soma size changes. Recent advances have identified a similar trend in cell proliferation with a distinctive decrease around middle age, identifying factors controlling this process may be the first step towards reinvigorating the aging sub-ventricular niche. For all validated clones an additional process segmentation was applied, assigning cell process pixels to the nearest cell body, this additional segmentation process is discussed in De La Hoz *et al.*[9]. Type A and C process sizes also showed significant differences across all ages and type C process size followed a U shape across age similar to soma size and cell proliferation. For both experiments our analysis revealed morphological and behavioral differences among cell populations that would not have been identifiable without the accurate segmentation and tracking of each cell provided by these new algorithms[6].

A key component of our analysis of for the embryonic and aging experiments is the ability to provide a segmentation for every cell in every frame. It is difficult to validate segmentation results pixel-by-pixel, even in small data sets, and the two experiments discussed here contain millions of segmentation results across tens of thousands of cell tracks and hundreds of lineages. In order to validate our cell segmentations we use two functional approaches on the embryonic experimental data. First, we compare the tracking error rates using our complete segmentation results with the same tracking algorithm applied to only cell center of mass. Improvements in the tracking error rate using full segmentation indicates that we are capturing information relevant to expected cellular motion modeled by the tracker. We also compare the size ratio between initially plated anterior and posterior cells using our segmentation algorithm, and the same size ratio measured using flow cytometry (FACS). While flow cytometry and image segmentation sizes are not directly comparable a similar ratio of sizes across cell types indicates that we are capturing a similar feature. The validation techniques provide functional evidence for the accuracy of the segmentation approaches discussed here.

Both the embryonic and adult neural progenitor experiments imaged cell cultures *in vitro* using 2-D phase contrast microscopy. The same analysis pipeline was used for both experiments, with a slightly simplified segmentation algorithm used in the case of the aging NSCs. The process is shown in figure 4.1. An initial segmentation, tracking and lineaging is run first, these lineages are often inaccurate due to segmentation errors. The initial lineages are then manually corrected. Finally, segmentation from lineage is run across each movie guided by the validated cellular lineages.

4.1 Method for improving segmentation and tracking using the cellular lineage tree

This chapter focuses mainly on the computational analysis of biological image sequences. However, this work is by its nature highly collaborative and depends upon valid biological experimental design. A discussion of biological materials and experimental methods used by collaborating biological labs to generate the image sequences is included in Appendix A.1 and A.2.

4.1.1 Initial cell segmentation, tracking and lineaging in time-lapse phase contrast images

The initial segmentation was originally developed for retinal progenitor cells and was applied previously to segmenting neural progenitor cells[4;25]. Only a single parameter, the approximate cell radius (here 2.5 µm) is required. The algorithm begins with an adaptive Otsu thresholding of the unprocessed phase contrast images[60]. The thresholding identifies the pixels belonging to the phase contrast "halo" artifact. Mathematical morphology is then used to construct a cell mask[61]. This cell mask construction requires an approximate cell radius parameter specific to the cell type being imaged. The algorithm uses two separate models, one for bright cells and one for cells with a dark interior. The thresholded morphological gradient image is used to separate touching cells[61]. Finally, a post-processing step eliminates false detections using four empirically determined feature thresholds. Because they are formed as a ratio, these features automatically adjust to different cell sizes. The fourth feature is an area feature, computed from the approximate cell radius parameter.

After the initial segmentation, we apply the multitemporal association tracking (MAT) algorithm to associate all segmentations to cell tracks over time. The MAT algorithm is detailed in chapter 5 and its applications are further discussed in Chenouard *et al.*, Mankowski *et al.* and Winter *et al.*[3;4;11;12]. Our use of MAT for tracking stem cells requires two parameters, the approximate maximum velocity per frame and the same approximate cell radius that was used by the initial segmentation. For all of the adult and embryonic neural progenitor cells we have analyzed in this work and previously, imaged at a 5 min per frame time resolution, the maximum velocity was set to 40 pixels per frame (5 µm per min). MAT is a windowed, graph-based approach that determines the cost, given a cell motion model, of associating a given segmentation with all of the current cell tracks that are within the maximum velocity threshold. Together, these costs form the tracking graph.

Following the tracking step, an estimate of the lineage tree is formed from the tracking graph. Proceeding forward frame-by-frame, possible parent cells are identified for any newly appeared cells. The initial lineaging algorithm chooses the most likely parent, subject to a minimum cell cycle time constraint[4]. The lineage tree structure can be automatically improved by using a tree inference algorithm that also incorporates evidence from the tracking graph[62]. This inference approach uses Dijkstra's algorithm to iteratively extend each leaf node of the lineage tree so that it reaches the end of the image sequence, leaves the frame or dies[63]. This step is repeated until no further changes occur in the lineage. Extending tracks in this manner enforces the assumption that cells should not disappear from the image sequence without cause. An example of the results of the tree inference algorithm is shown in figure 4.5.

4.1.2 Segmentation from lineage algorithm

The segmentation from lineage runs on each image frame sequentially using a lineage tree as input to determine the tracks that need to preserve or acquire segmentation results. The algorithm improves the segmentation result associated with each cell on the lineage tree in every image frame, subject to the tracking motion model and the image pixel intensities. The MAT tracking algorithm (see chapter 5) identifies the set of segmentations in each frame that most conform to the motion model, *i.e.* minimize the total cost of each tracking assignment for all cells on the lineage tree. These also include the set of segmentations that exhibit the most evidence in terms of pixel-based image intensities, since segmentation from lineage incorporates a more aggressive version of the initial thresholding, as well as other localized segmentation techniques, in evaluating the need to add new detections. The iterative refinement of segmentations can be much more aggressive in searching for segmentation results because the lineage information in conjunction with the tracking localizes the search space to only the most probable regions.

During segmentation from lineaging, each cell on the lineage tree is processed to ensure that it has an associated segmentation result in each image frame. If a cell is missing its segmentation result in the given frame, the refinement algorithm explores an ensemble of possible segmentations for the cell by either adding new segmentations, splitting existing segmentations into multiple cells, or both. Once the refinement has generated new segmentation results, the tracking algorithm selects the best set of segmentations simultaneously for all cells on the lineage in the given image frame. The segmentation from lineage algorithm attempts to add a segmentation when there is no existing segmentation within an adjustable overlap with the previous segmentation for the cell that is being processed. To add a new segmentation for a given cell, we take a region surrounding the expected location of the cell and re-run the initial segmentation reducing the threshold level used to separate foreground and background pixels. This process can fail, with no additional segmentation being returned. In that case, if possible we try to split an existing segmentation.

Segmentation from lineage splits existing segmentations by using mixture of Gaussians clustering on the spatial coordinates of the foreground pixels of the segmentation to be split[19;21]. The Gaussian mixture models effectively partition elliptical shapes, rather than the round cells favored by k-means[11]. The watershed transform can also be used with a basin-merging strategy to obtain a given number of cells[64]. Gaussian mixture models provide effective spatial separation between a given number of objects with elliptical shape including retinal, neural and hematopoietic progenitor cells. Details on an improved partitioning technique “Pixel Replicated Gaussian Mixtures” are discussed in chapter 6, as well as a comparison of partitioning techniques on adult neural progenitors and synthetic data. The refinement framework maintains a record of all corrections made both automatically and manually in order to measure segmentation and tracking error rates.

4.1.3 Statistical comparisons of anterior and posterior cells

The features incorporated into the comparison of anterior and posterior cerebral cortex neural progenitors include cell cycle time, cell velocity and cell area. These features are only computed for cells that have been observed through an entire cell cycle, from birth through subsequent division. Cell cycle time is the duration in minutes between birth and the division event that creates two daughter cells from the given cell. Cell velocity is the mean displacement of the center of the cell in each frame divided by the time between frames. Cell area is the number of pixels defined as interior to the cell times the area of a pixel. We use the convex hull bounding the foreground pixels of each segmentation results as a proxy for cell area.

Significance of results was determined by using the non-parametric Wilcoxon rank sum method to test the equality of two distributions[18]. We also use a robust graphical estimate of confidence intervals from distribution quartiles[65]. These graphical confidence intervals are shown as error bars on figure 4.9, plots of motion, velocity and cell cycle features by individual cell, averaged across clones and also by generation for the different populations. This method has been shown to be a good visual approximation of a 95% confidence interval for non-parametric data[65]. These error bars are a visual representation of a statistical significance interval, and are intended to complement the Wilcoxon rank-sum test used to determine statistical differences in median feature values between progenitor cell populations.

In order to quantify behavioral differences between the anterior and posterior cell populations, we applied Wilcoxon rank-sum test to individual versus clone-averaged cell size and cell velocity. No significant difference was observed comparing cell features averaged across each clone with individual cell features, in the posterior population ($p > 0.08$). However, in the case of the anterior cell population the rank-sum test indicated a significant difference between individual and clone-averaged values of cell size and cell velocity ($p < 0.01$), implying a more heterogeneous structure in the anterior cell population.

The anterior population was partitioned into two groups using k-means clustering on clone-averaged cell cycle time. We applied the same rank sum test as above to the "fast-dividing" and "slow-dividing" anterior populations to verify that this partitioning reduced heterogeneity of the cell populations. We found no significant difference between individual and clone-averaged values for the fast-dividing anterior population ($p > 0.42$) and similarly for the slow-dividing anterior population ($p > 0.72$). This provides considerable evidence for treating the fast and slow-dividing anterior cell populations separately in all further analyses.

4.1.4 Functional validation of segmentation algorithms

Validating segmentations pixel-by-pixel was infeasible on datasets of this magnitude containing millions of individual segmentations each with hundreds of interior pixels. We adopt a functional validation approach, first we compare the error rate difference between using the multitemporal association tracker. The tracker is run on every segmentation in every frame, however instead of passing the full segmentation information, we treat each cell detection as a single point at its centroid. Any assignment errors are accounted for during the tracking and are immediately corrected to avoid propagation of single assignment errors.

The second validation approach compares average embryonic anterior-posterior size ratios derived from our segmentation results to an average size ratio derived from fluorescent cytometry forward scatter photon counts. Flow-cytometry based fluorescent activated cell sorting (FACS) is a common tool for measuring cell size[66]. FACS forward scatter integrated photon counts are an uncalibrated (unitless) measure, making a direct comparison to cell sizes acquired from our segmentation algorithms difficult. To enable comparison of our segmentation results with FACS measured sizes, we compare the ratio of cell sizes between embryonic NPC anterior and posterior populations measured using both approaches. While absolute measures are not comparable, ratio values provide a consistent mechanism for comparing the two approaches. Using the ratio of measured sizes between anterior and posterior populations also reduces the potential for FACS size results to be influenced by factors not related to cell size[66]. FACS data was obtained for cells plated 12.5 weeks post-fertilization (E12.5). The flow cytometry data for anterior and posterior cerebral cortex progenitor cells was compared to the cell size averaged across the 160 initially plated progenitors (82 anterior and 78 posterior) that were analyzed with our segmentation algorithms.

We model the distribution of cell sizes using a log-normal random variable. The log-normal distribution is commonly used for physical parameters such as cell size because unlike the normal distribution it cannot take values less than zero. There is also evidence that quantities such as cell size and cell cycle time can be modeled as a product of independent identically distributed variables, which will produce a log-normal distribution in the limit[67;68]. We use a maximum likelihood estimate to fit log-normal distributions to anterior and posterior populations of FACS data. The goal of the comparison is to show that the ratio of segmentation sizes obtained by our algorithms between anterior and posterior populations is consistent with these ratio distributions obtained from the FACS cell size data.

A Monte Carlo simulation is used to test the hypothesis that the average cell size ratios obtained from our segmentation algorithms comes from independent random samples from the same log-normal distribution that generated the FACS ratio data. The null hypothesis asserts that our segmentation results and the FACS data are different methods for measuring the same underlying cellular property. The p-value for this test is found by simulating random values from the FACS distributions and calculating the average ratio. Repeating this process many times creates an empirical cumulative distribution function (CDF). We simulate 82 size ratios sampled from the FACS distribution, representing the maximum available anterior and posterior cells to compute a ratio for our experiments. This sampling process is repeated one million times to produce the CDF of sample mean ratios. From the CDF, the probability of observing a sample mean ratio farther from the FACS mean value (0.9963) than the segmentation mean value (0.9988) is greater than 92% ($p = 0.9214$). This indicates that there is no statistical difference between the ratio of sample means acquired from our segmentation algorithms compared to the ratio of cell sizes between the two populations obtained from FACS. We cannot reject the null hypothesis with any confidence since p is much greater than the standard 5%. This provides strong statistical evidence of a consistent relationship being captured by the size data from the segmentation algorithm and the FACS size measurements.

4.2 Biological analysis results from segmentation from lineage

Image data from two separate experiments comprising hundreds of cell lineages and tens of thousands of cells was initially segmented, tracked and lineaged, according to the process outlined in figure 4.1. These initial segmentation and tracking algorithms have been applied in a number of recent applications[2–4;11;12;25]. We developed an iterative algorithm that uses lineage information to automatically improve segmentation and tracking accuracy in a step referred to as *segmentation from lineaging.* The lineage trees encapsulating parent-daughter relationships between cells and used by segmentation from lineaging was manually corrected for each dataset in both experiments. In future applications, automated mitosis detection approaches can be used to provide lineage information for refining segmentation results, allowing for both fully and semi-automated use of the segmentation from lineage algorithm[69]. The segmentation and tracking results were automatically improved from the corrected lineage information with human observers correcting any remaining segmentation and tracking errors interactively.

Figure 4.2 shows a montage of all 160 embryonic anterior and posterior NPC lineage trees, a total of 10,644 cells and 1,585,104 segmentations. Once validated, we extract features such as cell lifespan, location and size, enabling quantification of the cell cycle time, motion and morphology for individual cells, across clones and broken down by generation. The analysis of this feature data reveals significant differences in the patterns of development between anterior and posterior cerebral cortical NPCs.

Figure 4.3 shows a montage of all 167 adult mouse NPC lineage trees isolated from 2, 6, 18 and 22 month old mice. Each lineage is colored according to the cell-fate commitment (type A, B, C or oligodendrocyte) of its subtrees. By combining the lineage and fate information with individual cell segmentation and tracking, we are able to compare the behavior of NPCs with different fate commitment. In particular we see that the patterns of motion and morphology of type A and type C committed neural stem cells are significantly different regardless of age. We also observe significant changes in the behavior of type C cells as the mice age, whereas the behaviors of type A NSCs remain relatively unchanged across ages.

The web-based visualization program CloneView additionally provides an interactive way to explore the data and results. Figure 4.4 shows a screen shot of the CloneView program with a summary listing the clones in one window and one image frame with segmentation and tracking results overlaid in the other window. All of the image data, together with all segmentation and tracking results, are available through our tool, CloneView. CloneView runs on any computer that supports a modern web browser with no software to download. CloneView is available at `http://bioimage.coe.drexel.edu/CloneView`.

4.2.1 The lineage tree is used to refine the underlying segmentation

Of all the tasks required for this analysis, segmentation (the delineation of individual cells in each image frame) is the most challenging and error-prone. Even human observers often find it difficult to establish the correct number of cells in clusters of adherent cells using only a single image frame. When the number of cells has been correctly established, clustering models that incorporate morphological characteristics of the cells, together with temporal information from the tracking, reliably separate the foreground pixels into individual cells.

We begin with an initial segmentation algorithm originally developed for phase contrast images of retinal stem cells, and applied previously to neural stem cells[4;25]. Modified versions of this segmentation algorithm have been applied to oligodendrocyte precursors and hematopoietic stem cells[11]. Following the initial segmentation, we apply a Multitemporal Association Tracking (MAT) algorithm (see chapter 5). MAT was originally developed for tracking organelle transport[3;12], and was found to be effective for tracking stem cells[4], reducing the error rates compared to previous approaches by 86%[25;32]. Inference approaches automatically improve the lineage tree, using the tracking graph together with constraints that cells do not appear or disappear between frames except across the imaging border, unless there is a mitosis or cell death[62;63]. A key advantage of the segmentation from lineage method is that once the number of cells in the frame is established, as described above, other important characteristics such as cell morphology can be incorporated to further improve accuracy.

For all 160 clones the average initial segmentation error rate of 8.1% represents all the cell identification errors including both the automatic corrections generated by the post-lineage refinement (6.4%) and the user-provided manual corrections (1.7%). This represents a 79% reduction in segmentation error rate compared to the initial segmentation. This initial segmentation incorporates our previous development of stem cell segmentation algorithms[4;10;11]. The tracking error rate was 1%. The total error rate, calculated from the number of edit operations required to fully correct the segmentation, tracking and lineaging errors, was 1.3%.

We integrated human validation with the automated processing tasks because for the biological applications discussed here, the tracking and lineaging results must be 100% correct. A fully automated segmentation from lineage approach is also possible. We applied this automated approach to the example clone of figure 4.5. Figure 4.5A shows the original automated lineage tree, figure 4.5B shows the lineage tree improved using tree inference, and figure 4.5C shows the user corrected lineage tree. The fully automated algorithms still significantly reduces error rates, for applications that can accept some errors. For clone shown in figure 4.5, using the inference improved lineage to refine the initial segmentation produces an error rate of 1.5%. Using the human-corrected lineage reduces the error rate to 1.1%. Including the lineage and tracking errors and results, the total error rate for fully correcting the clone was 0.9%. This clone can be explored using CloneView, `http://n2t.net/ark:/87918/d91591?1`.

Figures 4.6 and 4.7 show the results of refining an existing segmentation for two different scenarios. Figure 4.6A shows three cells segmented as one by the initial algorithm. Figure 4.6B shows the segmentation results from a watershed transform[61]; note the over-segmentation that is a hallmark of the watershed approach. Figure 4.6C shows a clustering of the foreground pixels of the cell split using a k-means clustering algorithm. This algorithm does not take into account the elliptical morphology of the neural progenitor cells and fails to find the correct separation between the touching cells. Finally, figure 4.6D shows the results obtained using a mixture of Gaussians clustering algorithm on the spatial locations of the foreground pixels. This approach accommodates the elliptical morphology of the cells and finds the correct separation among the cells. A second example showing the partitioning of two touching cells is given in figure 4.7A-D. Further improvements to this partitioning approach using pixel replicated Gaussian mixture models are discussed in chapter 6.

Time-lapse image sequences of proliferating cells contain inherent ambiguities that can be difficult to resolve from even a long sequence of image frames, as illustrated in figure 4.8. The gray segmentations marked with arrows strongly resemble cells, but are actually cellular processes. Neither the segmentation nor the tracking alone resolves these as false detections. In the top panel of figure 4.8, the 21 segmentations belonging to actual cells are shown with colored outlines, while the segmentation results that are not cells are shown with gray outlines. By using the population information contained in the lineage tree, the software can automatically identify the 21 correct segmentation results in this frame.

4.2.2 Edit-based and functional validation of the segmentation, tracking and lineaging

When analyzing time-lapse image sequences of proliferating cells, any errors in tracking or lineaging will almost certainly corrupt the subsequent statistical analysis[2]. Given that the segmentation results presented contain over 200 million cell pixels, validation of segmentation accuracy at the individual pixel level is non-trivial. The LEVER validation does not enforce a pixel-accurate correctness, only that the segmentation has captured the correct number of cells. To validate the performance of the segmentation algorithms in assigning pixels to each cell, we used two functional approaches, based on the algorithms and analyses that use the segmentation results as input.

We compared the accuracy of the tracking algorithm with and without the full segmentation information. Instead of the complete set of pixels that constitute the cell segmentation, we provided the tracking algorithm with just the centroid for every segmented cell. Tracking errors that occurred were counted and then corrected so that errors would not propagate. This was repeated for every segmentation in every image frame for all 160 embryonic clones. The number of errors was measured as the number of edits required to correct any tracking mistakes. Using the full segmentation information resulted in an average per clone error rate of 1%. When only the centroids were used, this error rate increased to 3.5%. The 71% reduction in the number of tracking errors functionally validates the pixel-level segmentation accuracy in terms of tracking performance.

The second method used to validate the pixel accuracy of the segmentation algorithm was by comparison to forward light scatter used to measure cell size in flow cytometry fluorescent-activated cell sorting (FACS)[66]. FACS measurements are not directly comparable to our segmentation area results, however, given two populations of cells, the ratio of areas is independent of the measurement technique. In this case, we used a Monte Carlo approach to compare the FACS anterior to posterior average cell size ratio to the mean ratio observed in our image data. No significant difference was found ($p > 0.92$), providing a second functional validation to our segmentation at the pixel level, this time in the context of a biologically significant measurement.

4.2.3 Behavioral differences between embryonic anterior and posterior cerebral cortex progenitors

An advantage of the segmentation from lineage methodology is that it constructs a rich data set, allowing us to ask numerous questions about aspects of the cells and clones that have been imaged. We analyzed anterior and posterior mouse cortical NPCs, comparing them individually, across clones, and by generation, for size, motion and cell cycle time. We found anterior and posterior cells differ significantly across all three measurements, using both the Wilcoxon ranksum method and the two sample Kolmogorov-Smirnov test[18]. Posterior cortical cells were found to be faster-cycling ($p < 10^{-100}$), bigger ($p < 10^{-24}$), more rapidly moving ($p < 10^{-9}$), and to generate larger clones ($p < 10^{-8}$).

Figure 4.9 (left) shows the distributions of size and motion for anterior and posterior cells. Interestingly, the differences in motion and cell size between the anterior cells when considered individually versus averaging per clone were statistically significant, while for posterior cells the difference was not significant. In order to compensate for this effect, we clustered the anterior clones into slow and fast-dividing groups using clone average cell cycle time (figure 4.9). For these slow-dividing and fast-dividing anterior cell groups, there were no significant differences in cell sizes or velocity when looking at features of individual cells compared to averages per clone. Fast-dividing anterior NPCs behave more similarly to posterior NPCs than slow-dividing anterior NPCs. Cell size and velocity per clone were not significantly different between the posterior and fast-dividing anterior cells ($p > 0.13$), but were significantly different between slow-dividing anterior, fast-dividing anterior, and posterior NPCs ($p < 0.01$).

An outcome of the differences in cell cycle time between the posterior, slow and fast-dividing anterior cells is the number of generations produced. Figure 4.9 (right) shows the cell size, velocity, and cycle time change broken out by generation for each population. Generation zero is the first plated cell, plated 12.5 days after fertilization (E12.5), generation one cells result from the first cell division, etc. The differences in the observed features of motion, cell size and cell cycle time become greater in later generations, especially for the slow-dividing anterior cells compared to posterior and fast-dividing anterior cells. That embryonic cortical progenitors derived from different cortical regions are so clearly different is surprising and demonstrates the value of this approach to quantify dynamic aspects of cell behavior.

Figure 4.10 shows the additional ability to label NPCs retrospectively by fate-commitment from immunohistochemistry. The embryonic cells are fixed and then stained for β-Tubulin (neurons) and Nestin (NPCs). The staining results are overlaid on the final time-lapse image, and the lineage tree can be colored according to the generation when each NPC commits to progeny of the same fate. This type of fate-commitment analysis was used extensively in the second application of the segmentation from lineaging framework discussed below.

4.2.4 Age-related differences in sub-ventricular neural stem cells

Similar to the embryonic NPC data, the aging NSCs were compared individually, and across clones and ages. Additionally, since cell fate information was available for all datasets, we compared aging NSC fate-committed subtrees across cell types. In order to gather accurate cell statistics such as cell cycle time, size and process information, only cells with fully observed cell cycles (birth to mitosis) were analyzed. Very few type B and oligodendrocyte cells were fully observed and most were observed in the 2 month data set, so further analysis was not carried out on these cells.

Comparison of the type A and C fate-committed subtrees reveals significant differences in type A committed NSC behavior compared with type C committed cells for all mouse ages. The type A neural stem cells have significantly longer cycle times than type C cells across all ages ($p < 10^{-23}$). Type A cells also have smaller cell soma as well as smaller processes than Type C cells overall ($p < 10^{-48}$, $p < 10^{-68}$). Figure 4.11 shows a comparison of type A and type C cell cycle times, soma area, and process area.

Additionally, we compare type A and C committed cells across age. Figure 4.12 shows median cell cycle time (top) and cell area (bottom) for type C and type A committed cells across mice aged 2, 6, 18 and 22 months. Type C cells show very distinctive behavioral changes in mice of different ages. Interestingly, type C median cycle time increases significantly at 18 months ($p < 10^{-3}$), however the median cycle time at 22 months returns to nearly the same as a 2 month old mouse. An inverted phenomenon is observed in type C soma area, as the mouse ages the soma area grows smaller until 18 months, by 22 months the type C median soma area returns to near that of a 2-6 month old mouse. As observed in the overall comparison type A committed NSCs are smaller and less proliferative than the type C cells for each age group. Type A committed cells also appear to vary less with mouse age.

4.3 The segmentation from lineage algorithm provides key biological insight

Time-lapse phase contrast movies provide a wealth of information about dynamic cell behavior. By culturing cells in defined conditions, the impact of environmental factors, such as drug treatments, on dynamic events, such as morphological changes, migration, process outgrowth, cell division and cell death, can be captured using minimally invasive phase contrast imaging. A bottleneck encountered is effective analysis of the vast amount of captured video data. Automated segmentation and tracking algorithms are increasingly showing their value in providing rapid, objective image quantification. However, no program is error-free, and the challenge has become minimizing the time required for human validation. These results demonstrate the value of applying segmentation from lineage to reduce the manual effort involved in analyzing large data sets, while still producing fully corrected results. Utilizing high-level knowledge such as lineage information during, segmentation and tracking can reduce automated segmentation and tracking errors dramatically, improving program throughput and enabling the analysis of larger quantities of data.

We found significant differences at the cellular and clonal level from the anterior and posterior cortical NPCs as they progress from approximately 12.5 weeks post-fertilization to the equivalent of 16.5 weeks (E12.5 - E16.5). Posterior NPCs are larger, move faster, and divide more quickly, producing larger clones compared to anterior cells. The anterior population is more diverse, containing a mixture of slow- and fast-dividing clones. The differences in cell cycle time, size and motion become more pronounced with increasing generations. These differences were apparent when the anterior and posterior cell populations derived from the same animals were cultured concurrently in identical *in vitro* conditions, indicating that they are cell-intrinsic differences rather than based on environmental instructive factors. Given that the posterior cortex is larger than the anterior cortex, it is possible that the embryonic posterior NPCs are more proliferative to accommodate greater cell production and growth.

In a second series of experiments we analyzed mouse cortical neural stem cells across 2, 6, 18 and 22 month ages. These ages correspond to young child, teen, middle age and geriatric mice, respectively. Immunostaining of cells at the end of each experiment also provided fate-commitment information for each tracked cell. We found significant differences in the behavior of type A and type C NPCs across all ages. Type A cells are smaller and divide more slowly than type C cells. Type C NPCs divide quickly at young ages and this division speed slows with age up to 18 months. Intriguingly, type C cell division rate increases back to nearly that of a young mouse by 22 months. Type C size also decreased, following a U pattern with the inflection point at 18 months. A similar behavior was recently seen in human sub-ventricular zones across age, indicating that middle-age may mark an important change in the function and proliferation of neural progenitor cells. Further study of the changes in type C behavior across age may lead to an understanding of the mechanisms of aging and perhaps the ability to reinvigorate the aging sub-ventricular zone.

It would be possible to identify differences in clone size such as those observed in both the above experiments using only a static terminal image. However, by segmenting, tracking and lineaging each cell we obtain a rich dataset containing information about individual cellular behaviors and clonal dynamics. The behavioral differences in committed type A and C NSCs and the heterogeneous behaviors of the anterior cell population could not be observed with static analyses.

A key advance is the use of lineage information to refine the segmentation and tracking algorithms. The segmentation provides a unique identifier for every cell in every image frame and is the basis for the subsequent tracking algorithm and for extracting motion and morphology features. The segmentation results also enable robust validation and collaborative visualization of the tracking and lineaging results by allowing human observers to uniquely identify a particular cell in every image frame. A limitation of any approach to quantifying this type of image data is that once a human observer can no longer determine the correct segmentation, tracking and lineaging, validation becomes impossible. To some degree, this problem can be reduced by imaging at a higher temporal resolution or incorporating fluorescence channels. There is also the possibility for functional validation, as used here for measuring the pixel-level accuracy of the segmentation algorithms.

Our results include fully validated and corrected segmentation, tracking and lineaging results for 327 large and complex clones of proliferating neural stem cells. This data consists of over eleven thousand individual cells containing millions of segmentations. Only 1% of the results required human correction, with 99% of the work being done by the automated image analysis algorithms.

Segmentation from lineage provides a framework for integrating lineage information to improve the automated analysis, while also allowing human validation of the results. Application of these tools to the analysis of embryonic and adult neural stem cells has yielded biologically significant information about the nature of early brain development, as well as the change in proliferative activity as the brain ages. While some of these results could be obtained without the need for full segmentation and tracking of cultured cells, many of the behavioral changes, such as cell size and motion variations across both age and cell type could not be effectively identified using other experimental methods. These results highlight the power of iterative frameworks to continuously improve automated processing while maintaining support for human validation.

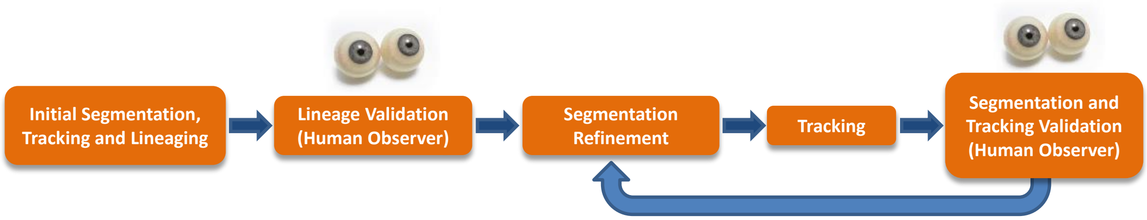
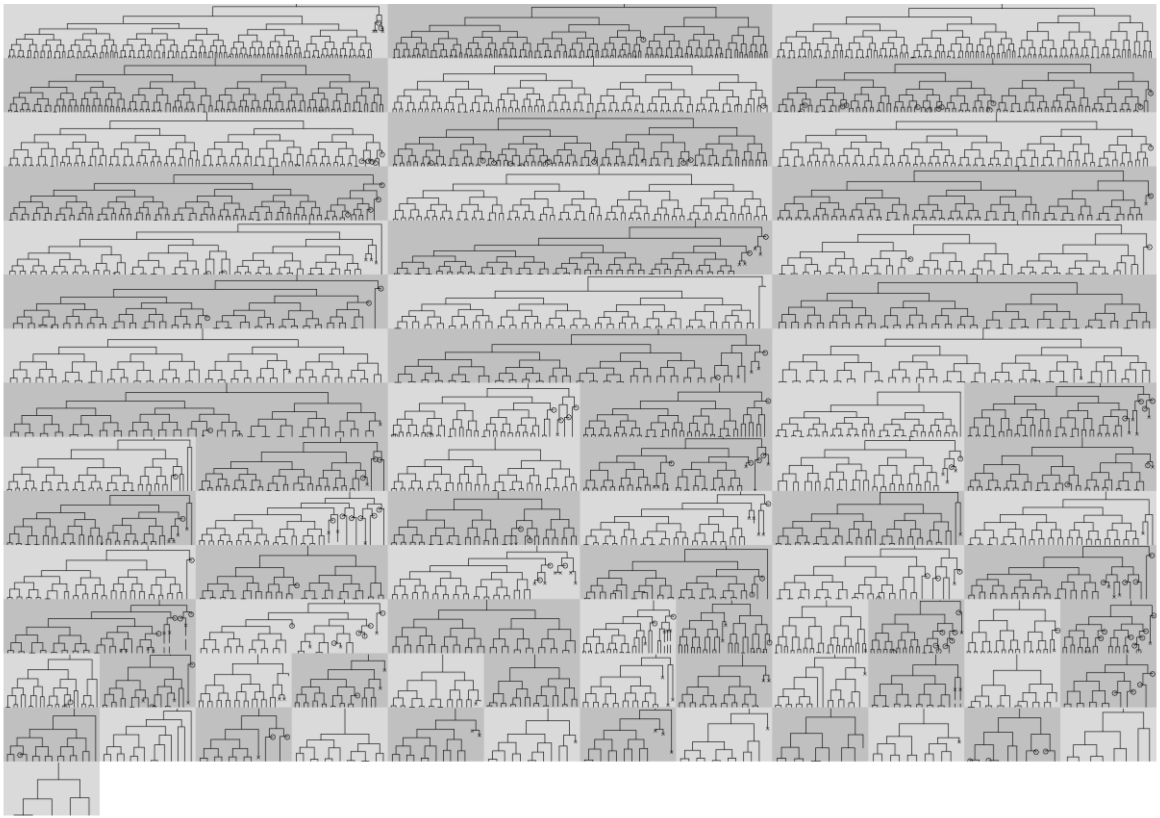
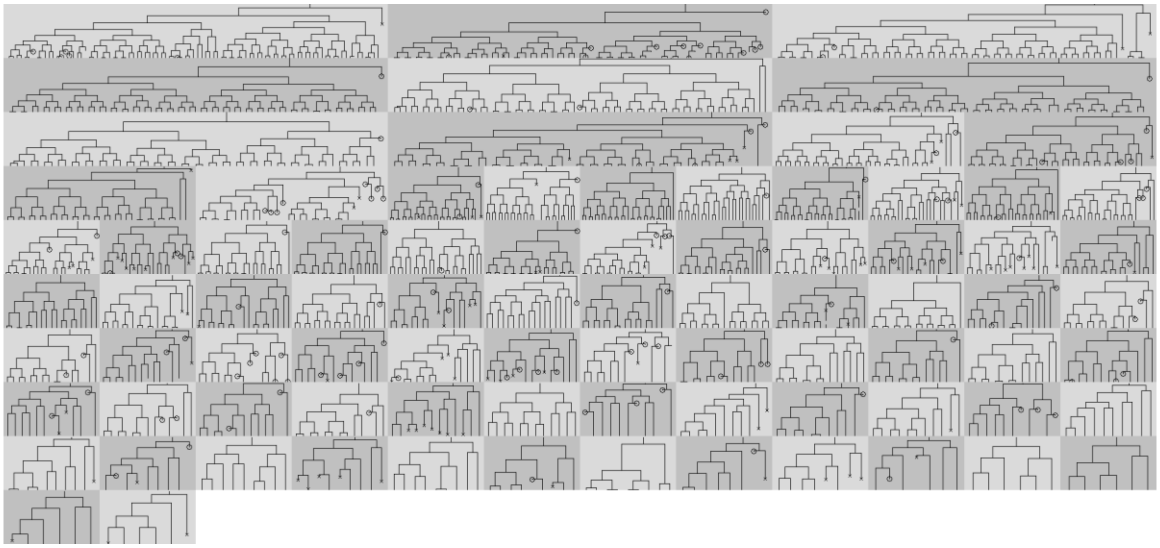


Figure 4.1: Overview of the segmentation from lineage approach Starting with an initial segmentation, cells are tracked through the image data and a lineage is obtained. The parent-daughter relationships in the lineage are validated by the human observer. The validated lineage is then used to refine the segmentation and tracking under supervision. This refinement and validation process is repeated for each image, achieving a significant reduction in the segmentation error rate.



Posterior clones



Anterior clones

Figure 4.2: Montage of 160 mouse embryonic neural progenitor cell (NPC) lineage trees created and validated using the segmentation from lineage interface. Shown are 78 clones cultured from the posterior region of the mouse cortex (top) and 82 clones cultured from the anterior region of the cortex (bottom). The differences observed in lineage structure and cycle-time between cells cultured in different regions is due to the intrinsic programming as all cells were grown in identical cultures after isolation from the cortex.

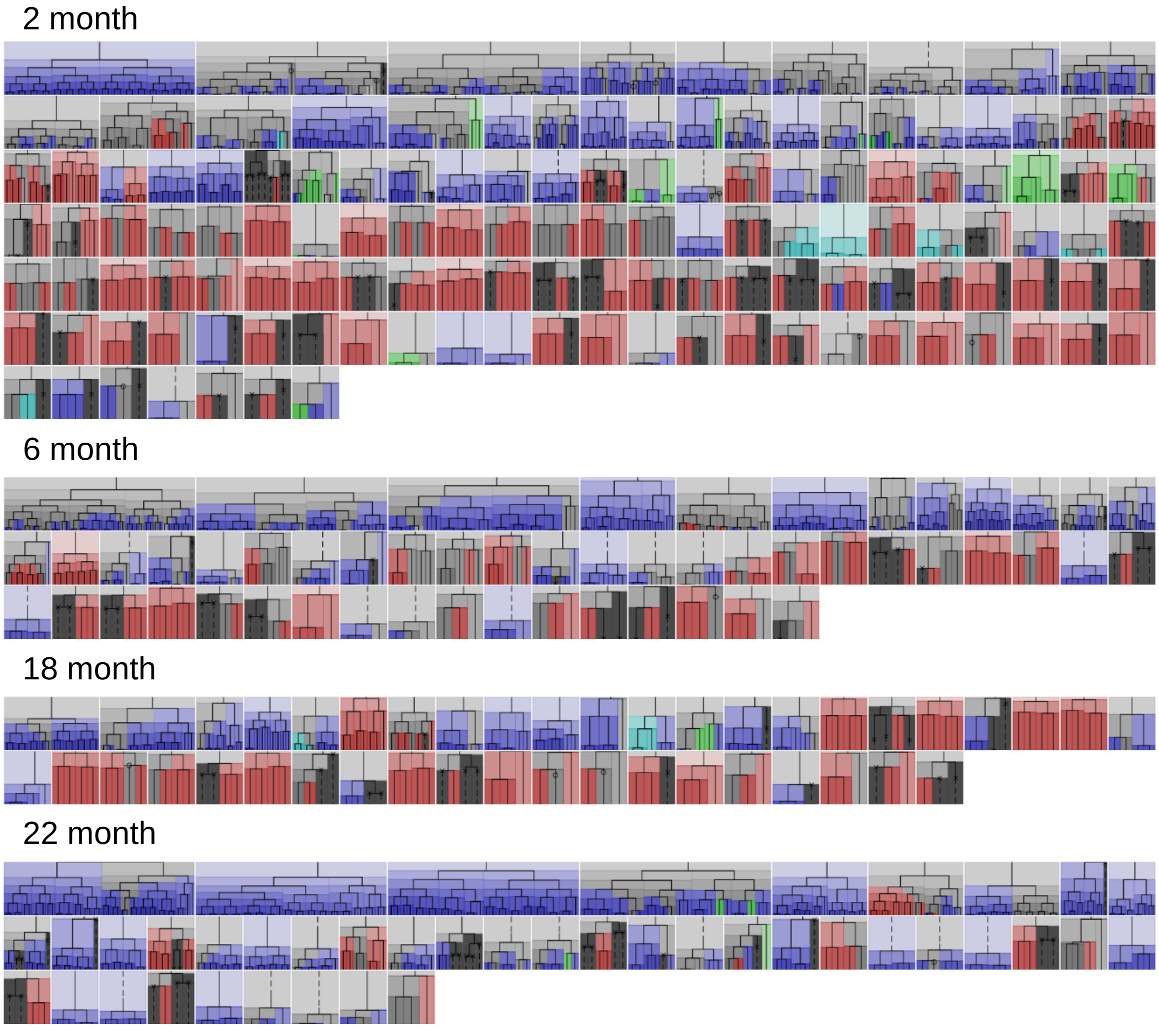
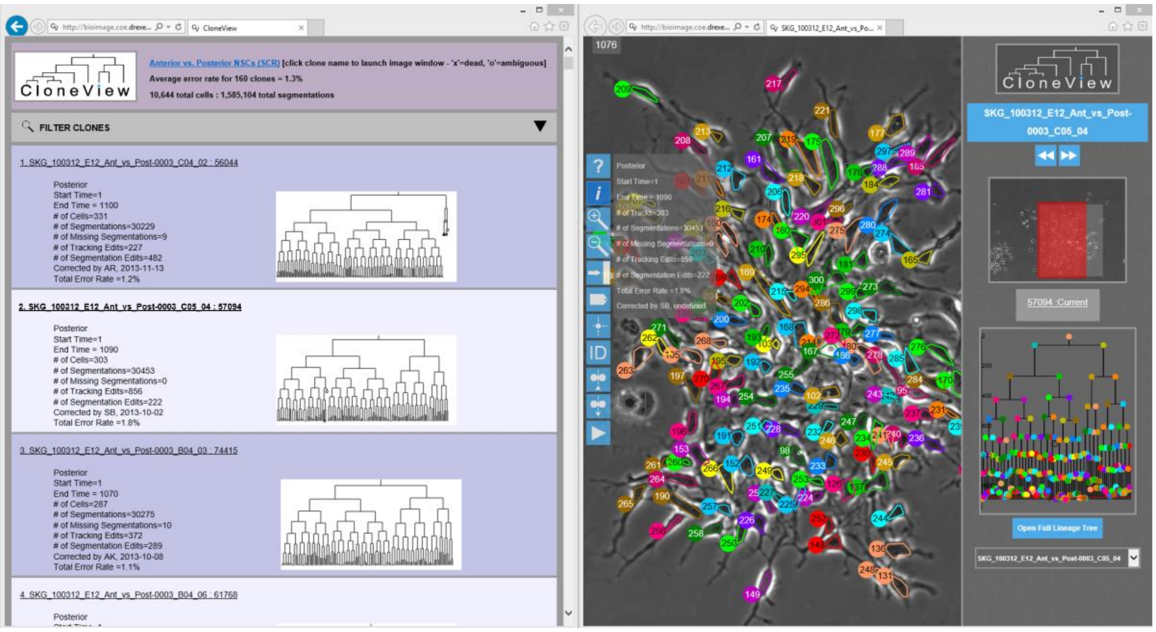


Figure 4.3: Montage of 167 mouse neural progenitor cell (NPC) lineages from age 2, 6, 18 and 22 month mice. Cell-type fate commitment on subtrees of the lineage is shown by gradiated background coloring. Subtree fate commitment to type A, B and C neural stem cells are indicated by red, green and blue backgrounds respectively. Oligodendrocyte fate commitment is indicated in cyan, death is indicated in dark gray, and a light gray indicates no stains were detected. All linages were created and validated using the segmentation from lineage interface.



Anterior vs. Posterior NSCs (SCR) [click clone name to launch image window - 'x'=dead, 'o'=ambiguous]
Average error rate for 160 clones = 1.3%
10,644 total cells : 1,585,104 total segmentations

1. SKG_100312_E12_Ant_vs_Post-0003_C04_02 : 56044

Posterior
Start Time=1
End Time = 1100
# of Cells=331
# of Segmentations=30229
# of Missing Segmentations=9
# of Tracking Edits=227
# of Segmentation Edits=482
Corrected by AR, 2013-11-13
Total Error Rate =1.2%

2. SKG_100312_E12_Ant_vs_Post-0003_C05_04 : 57094

Posterior
Start Time=1
End Time = 1090
# of Cells=303
# of Segmentations=30453
# of Missing Segmentations=0
# of Tracking Edits=856
# of Segmentation Edits=222
Corrected by SB, 2013-10-02
Total Error Rate =1.8%

3. SKG_100312_E12_Ant_vs_Post-0003_B04_03 : 74415

Posterior
Start Time=1
End Time = 1070
# of Cells=287
# of Segmentations=30275
# of Missing Segmentations=10
# of Tracking Edits=372
# of Segmentation Edits=289
Corrected by AK, 2013-10-08
Total Error Rate =1.1%

4. SKG_100312_E12_Ant_vs_Post-0003_B04_06 : 61768

Posterior

Posterior
Start Time=1
End Time = 1090
# of Tracks=303
# of Segmentations=30453
# of Missing Segmentations=0
# of Tracking Edits=856
# of Segmentation Edits=222
Total Error Rate =1.8%
Corrected by SB, undefined

Figure 4.4: CloneView web application. All images together with the results can be browsed from the CloneView web application. The main CloneView page for an experiment displays lineage thumbnails and summary information for each clone (left). Selecting a clone opens a new browser window that allows the images to be explored, with segmentation and tracking optionally overlaid (right). Both the embryonic and aging experiment data discussed here is available for interactive browsing from the cloneview webpage, CloneView: `http://bioimage.coe.drexel.edu/CloneView`.

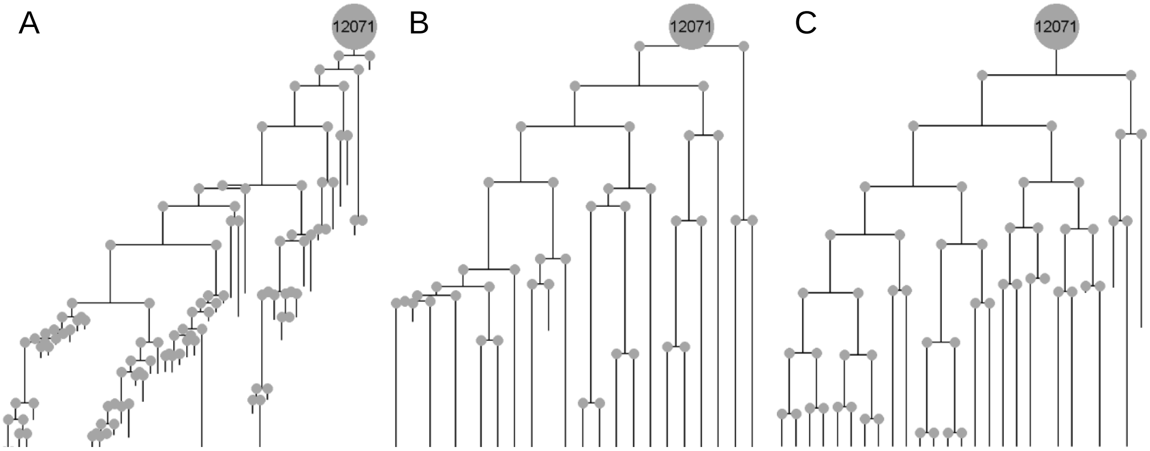


Figure 4.5: Automatic tree inference example. Initial *a priori* lineage tree generated by segmentation and tracking algorithms (A). Inference automatically improves lineaging results (B). Automated segmentation from lineage results in a segmentation error rate of 1.5%. User-corrected lineage tree with all mitotic events manually identified (C). Segmentation from lineage with user feedback applied to the corrected lineage tree results in a segmentation error rate of 1.1% and a total error rate of 0.9% for segmentation and tracking, providing fully validated reproducible segmentation, tracking and lineaging. CloneView: `http://n2t.net/ark:/87918/d91591?1`.

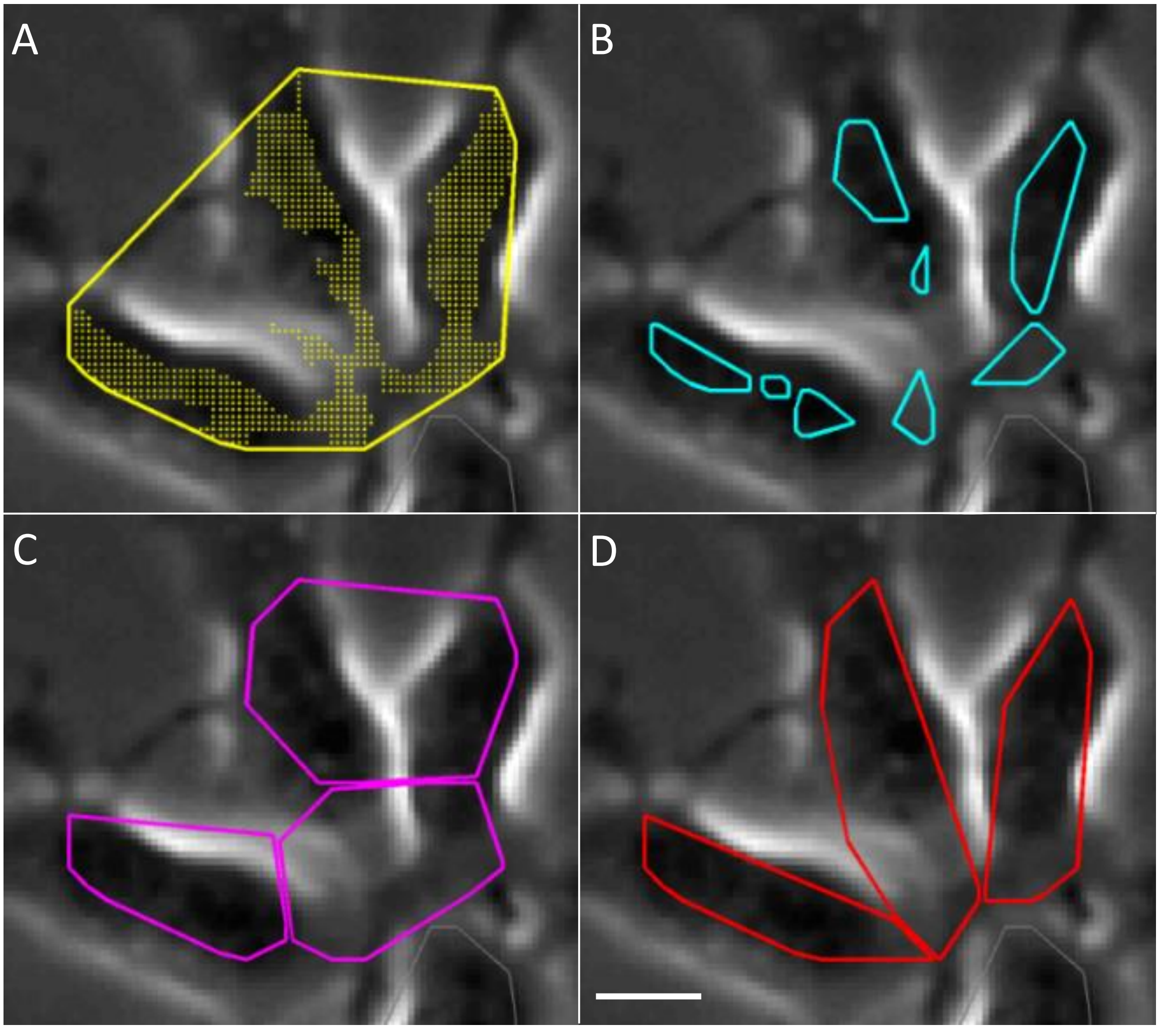


Figure 4.6: Segmentation from lineage comparison of different partitioning algorithms for a cluster of three cells. Segmentation example starting from initial segmentation results that incorrectly identify the three cells as a single component (A). We use the lineage tree to correctly establish the number of cells, improving over traditional methods such as the watershed transform (B). Partitioning of the pixels into three cells (C) using a k-means clustering is improved by using a Gaussian mixture model that incorporates elliptical shape (D). Further improvements to elliptical cell partitioning using pixel replication are discussed in chapter 6. The scale bar represents 10 µm. CloneView: `http://n2t.net/ark:/87918/d91591?3`.

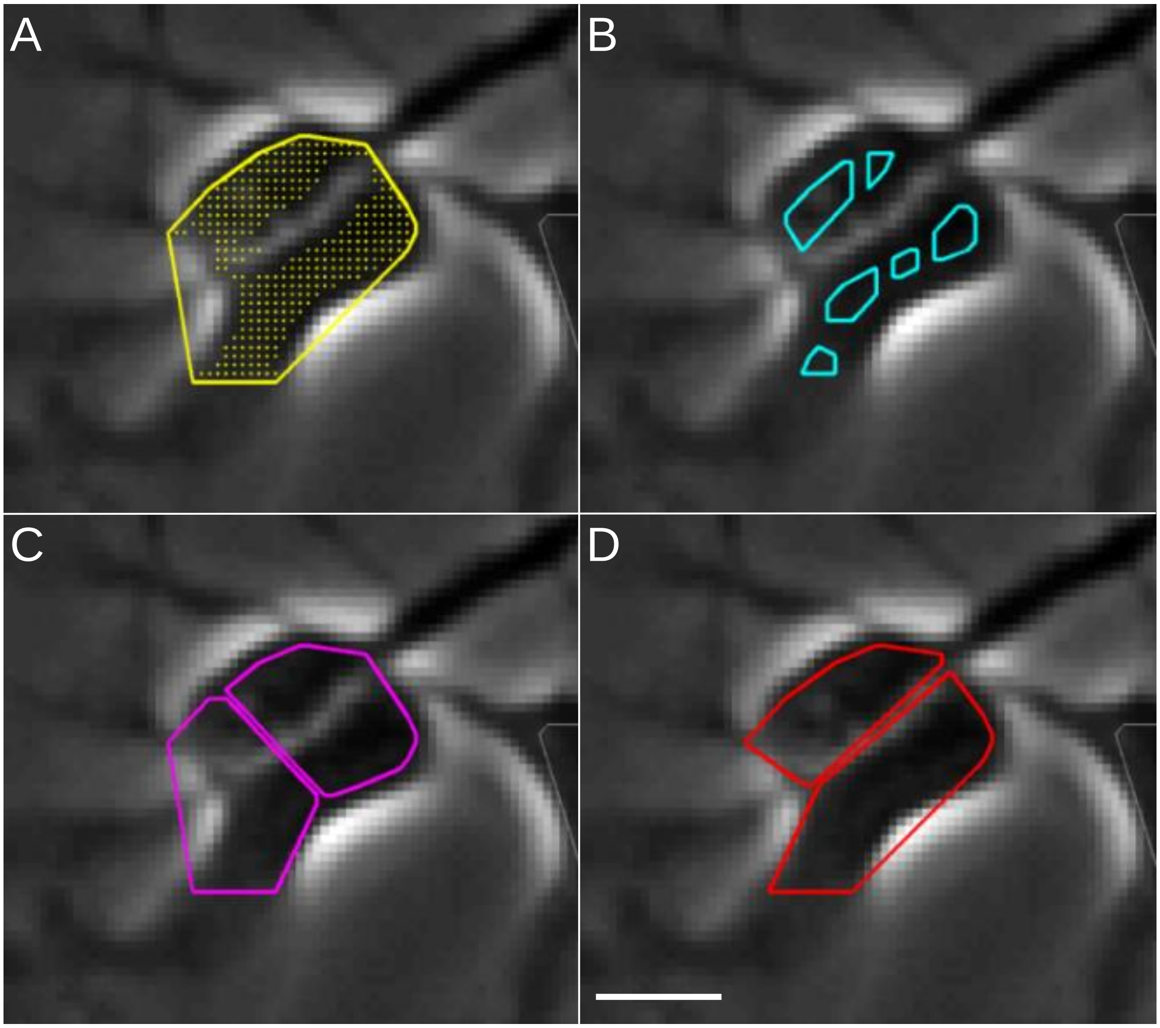


Figure 4.7: Segmentation from lineage comparison of different partitioning algorithms for a cluster of three cells. An additional example starting from initial segmentation results that incorrectly identify the two cells as a single component (A). The lineage tree correctly establishes the number of cells, improving over traditional methods such as the watershed transform, which overestimates the number of components (B). Partitioning of the pixels into two cells (C) using a k-means clustering is improved by using a Gaussian mixture model that incorporates elliptical shape (D). The scale bar represents 10 µm. CloneView: `http://n2t.net/ark:/87918/d91591?4`.

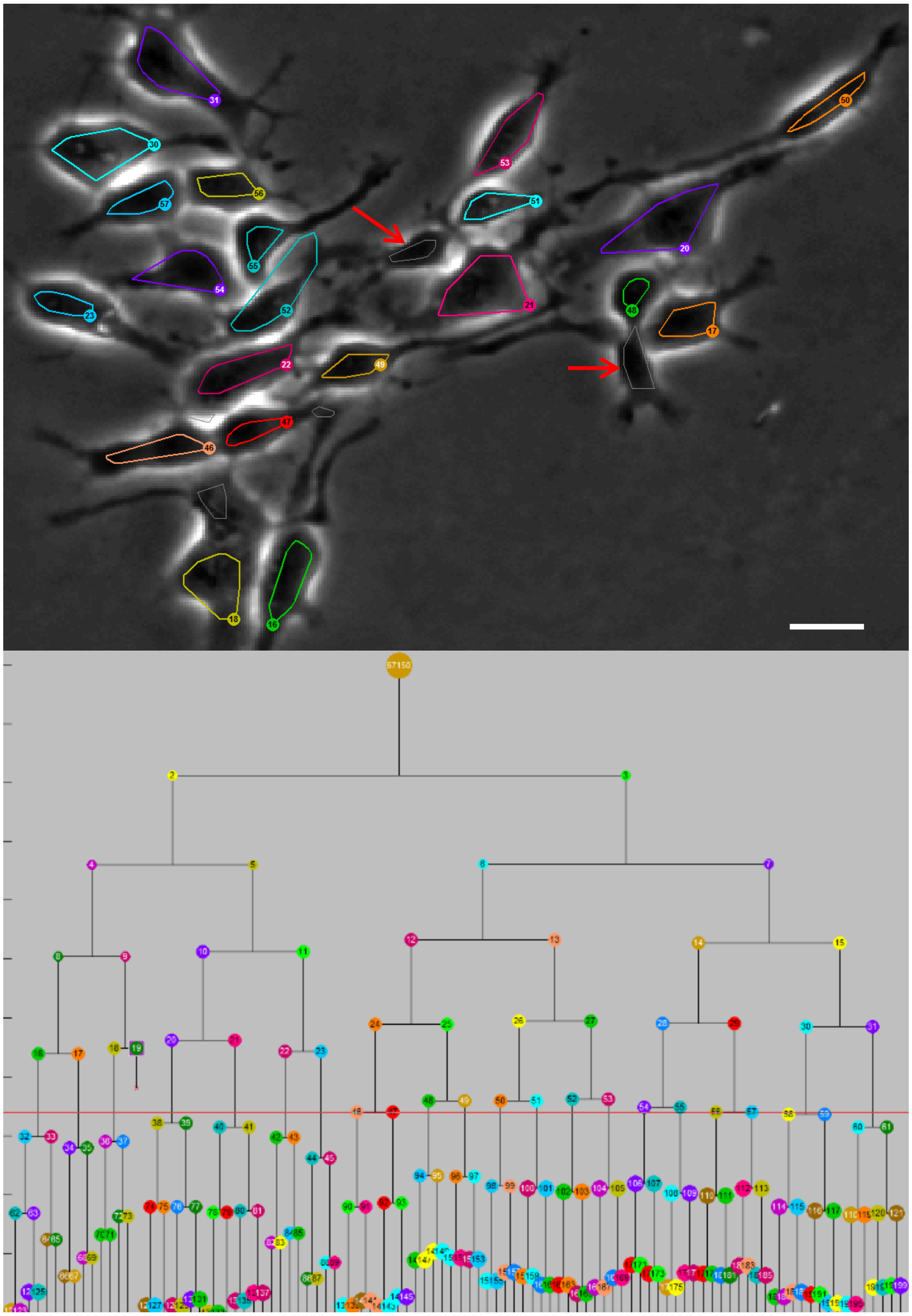


Figure 4.8: Lineage information resolves visual ambiguities. The segmentations marked with red arrows (gray outlines) are cell processes, not cells. These structures persist for over 20 frames and are indistinguishable from actual segmentations in isolated frames. The lineage tree (bottom) shows that there are 21 cells (colored outlines) in the current frame indicated by the red line. This allows the correct segmentations to be automatically identified. The scale bars represent 20 µm, CloneView: `http://n2t.net/ark:/87918/d91591?5`.

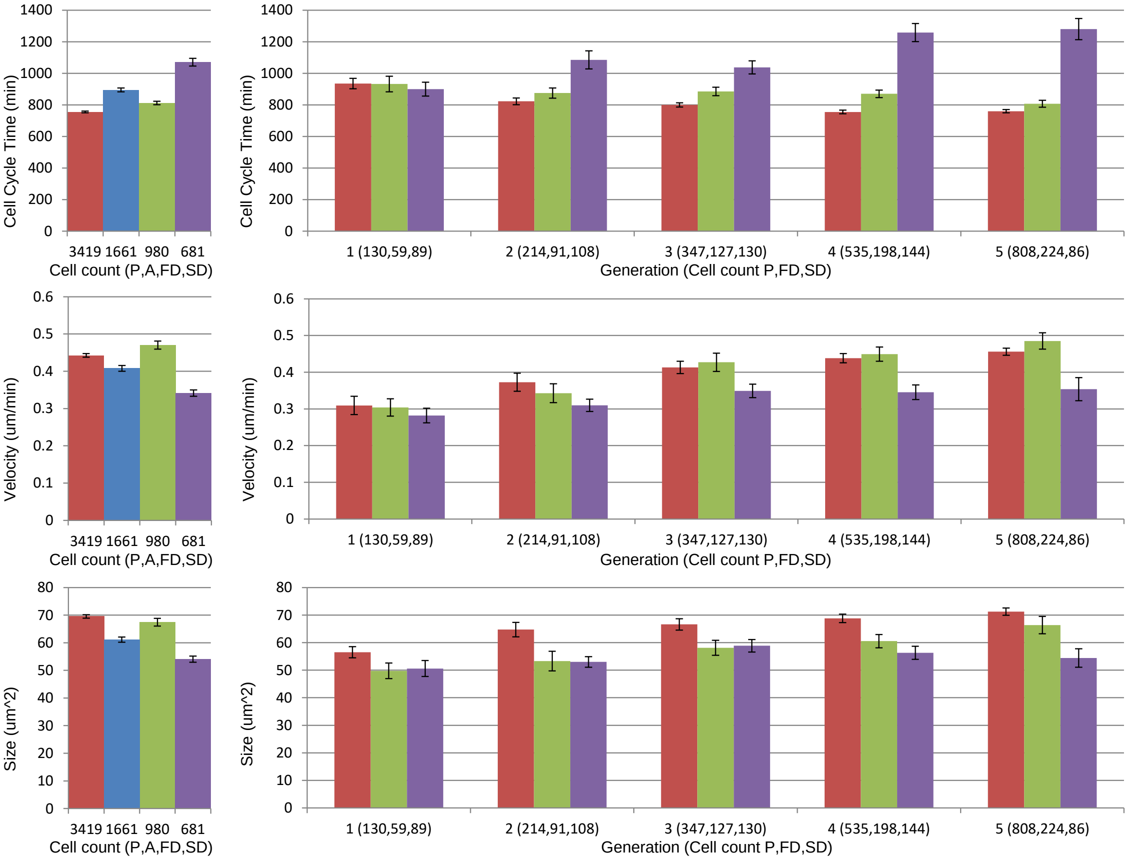


Figure 4.9: Differences in characteristic behavior of anterior and posterior cerebral cortex progenitor cells. A comparison of all cells (left) and by generation (right) is shown for posterior cells (P) in red, anterior cells (A) in blue, the fast-dividing subset of anterior cells (FD) in green, and the slow-dividing subset of anterior cells (SD) in purple. The whiskers represent 95% confidence intervals.

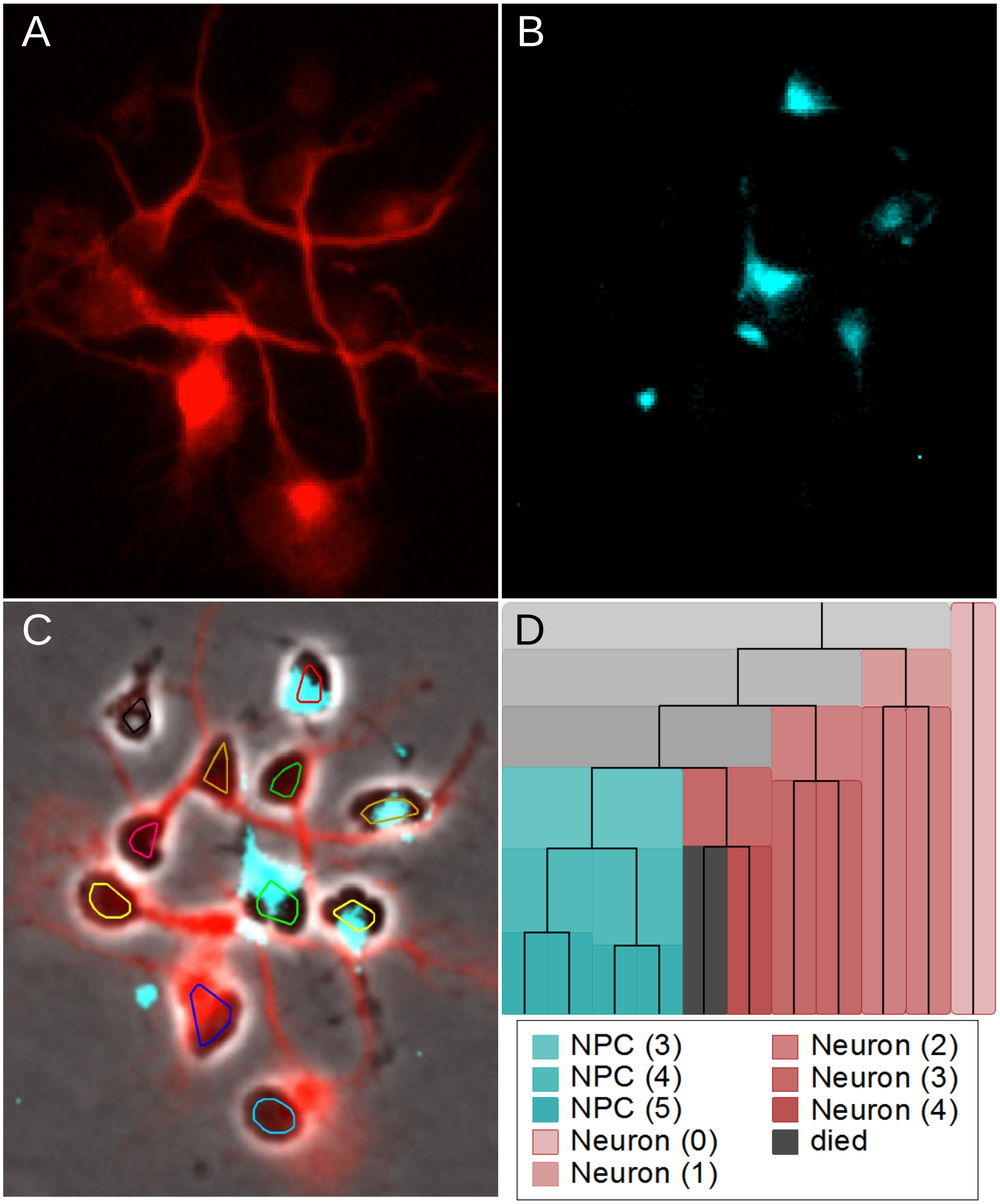


Figure 4.10: Immunohistochemistry can be used to display fate commitment by generation on the lineage tree. Stain images for β-Tubulin (red, neuron, A) and Nestin (cyan, neural progenitor cell, NPC, B). The final frame of live-cell time-lapse sequence (C) with segmentation and tracking overlaid and stain results blended. The cell fate commitment is shown on the lineage tree colored by the generation (shown in parentheses) when all offspring take on the same fate.

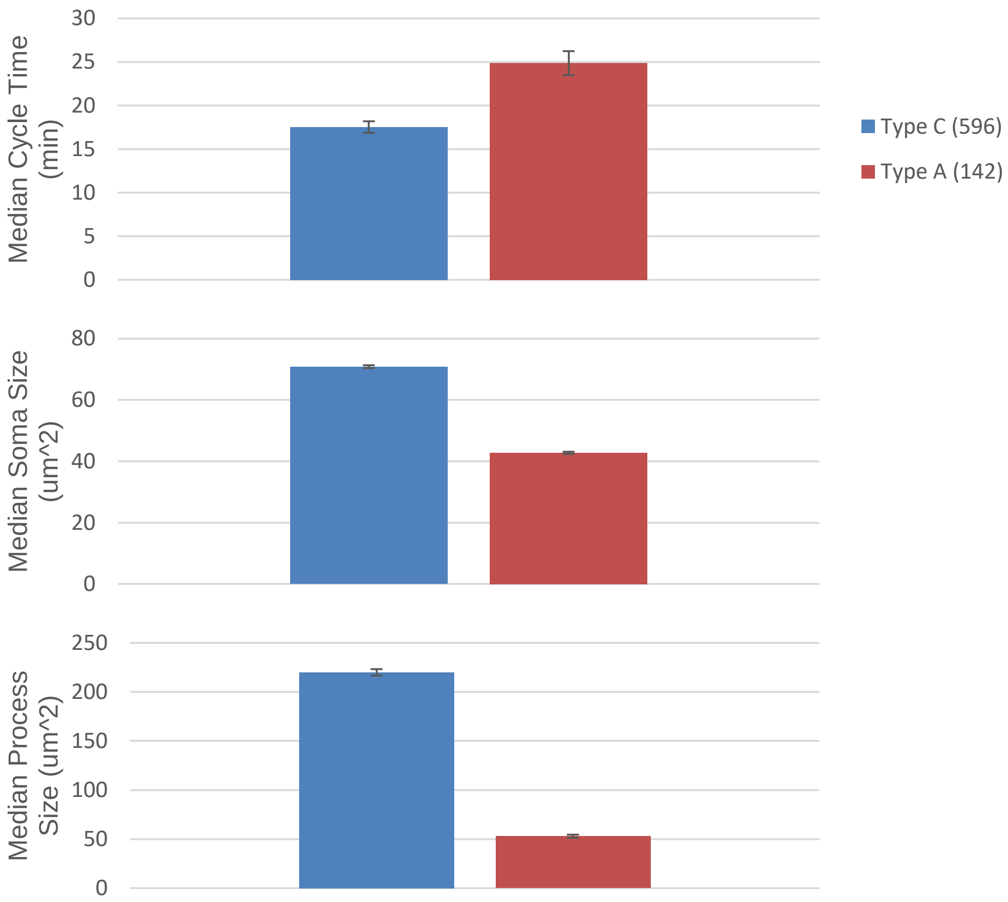


Figure 4.11: Differences in behavior and morphology of type C and type A neural stem cells (NSC)). A comparison of behaviors for type C (blue) and type A (red) NSCs across all mouse ages. Median cell cycle time (top), median cell soma area (middle) and median cell process area (bottom) are shown. The type A neural stem cells are smaller, have significantly less process area, and exhibit much slower cycle times. Cell counts are indicated in parentheses and the whiskers represent 95% confidence intervals.

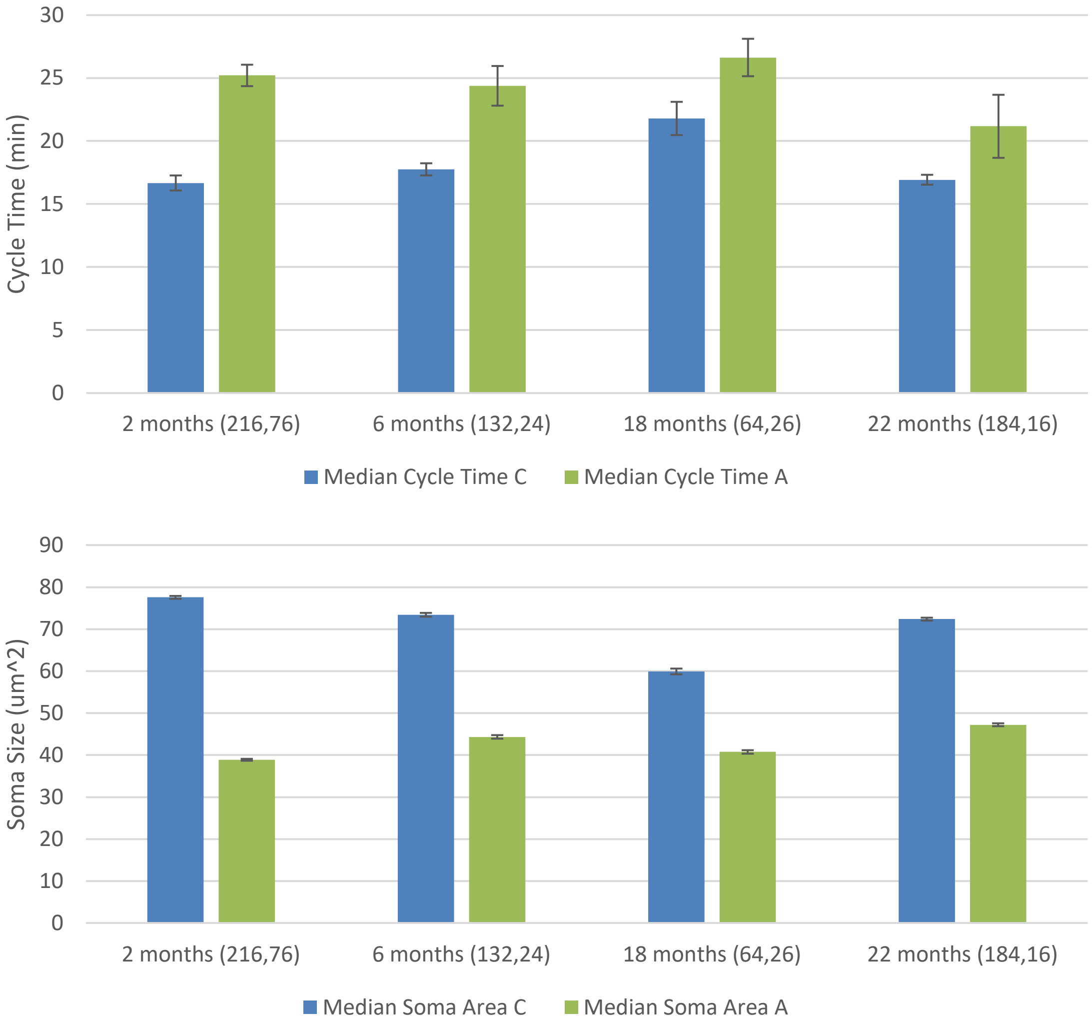


Figure 4.12: Age comparison of type C and type A neural stem cells (NSC)). A comparison of behaviors for type C (blue) and type A (green) NSCs for mice aged 2, 6, 18 and 22 months. These age groups correspond to young, teen, middle aged and geriatric mice. Median cell cycle time (top), median cell soma area (bottom) are shown. The distinctive inflection point cell cycle time and cell size in type C cells at 18 months may indicate a change in the development and function of type C NSCs[8]. Type A neural stem cells vary less across age but remain overall smaller and less proliferative than the type C NSCs. Number of fully observed cells are indicated in parentheses. Whiskers represent 95% confidence intervals.

Chapter 5: Multitemporal association tracking integrated into segmentation from lineage

Given a time sequence of object detections extracted from image sequence data by segmentation algorithms or by user input, the multitarget tracking problem is to associate these detected objects into a set of tracks such that each track contains exclusively detections corresponding to a single object. This establishes the long-term identity of objects, *e.g.* cells, across the entire movie. We have developed the multitemporal association tracking (MAT) algorithm an approach to multitarget tracking of biological objects such as cells or subcellular structures. The multitemporal association tracking algorithm was developed to robustly solve the tracking problem by considering a collection of frames simultaneously, looking forward from the current frame to establish typical object paths. MAT is robust to differing imaging conditions and segmentation errors. MAT does not require labor-intensive tuning of parameters related to the imaging conditions or culture density, instead establishing a tracking cost function encapsulating typical cell or object behaviors just once for the type of object being tracked. Association of objects to tracks is done at each frame and uses a minimum spanning tree optimization, iteratively assigning the lowest cost, or equivalently highest likelihood, associations between tracks and feasible object paths[63]. By minimizing a cost cost function, MAT approximates the Bayesian *a posteriori* probability estimate for the data association problem[70]. A key difference between MAT and other approaches to solving the multitarget tracking problem is in the relaxation of the requirement that a single track be assigned to at most one detection, or future path in the case of windowed tracking solutions. Allowing multiple tracks to share a common path forward matches the nature of the biological problem where clusters of cells will often occlude each other. This also greatly reduces the complexity of the tracking solution. The MAT algorithm is detailed below and in Chenourd *et al.* and Winter *et al.*[3;4;12].

Biological image sequences are characterized by very low signal-to-noise ratio, sudden variations in illumination and appearance, frequent occlusion and high object density. As a result, segmentation errors including under segmentation (identifying two objects as one), over segmentation (splitting one object into multiple objects), missed detections and false detections are prevalent. Dating back to Reid's seminal paper on multiple hypothesis tracking, it was apparent that simply assigning a detected object to its most likely track was not an effective strategy[42]. Instead, optimization approaches were developed that allow the assignment of detected objects to tracks to be optimized jointly across the set of detected objects and tracks such that some total cost for all assignments is minimized. These approaches are formulated generally as bipartite matching problems, and they are at the core of many current multitarget tracking approaches. Bipartite matching finds the optimal solution for rectangular assignment problems described by a cost matrix capturing the likelihood of each detection belonging to each track. Bipartite matching is simple and computationally efficient to implement, and is the best solution when determining the association between a single time instance of detections and the current set of tracks. Some non-biological applications have also applied tracking solutions based on multidimensional assignment (see chapter 2), solving the matching problem across multiple time instances simultaneously.

There is a fundamental limitation in the bipartite matching and multidimensional assignment approaches to tracking in biological image sequences: these techniques can only control the assignment of segmentation noise by the use of application specific thresholds in computing pairwise costs between segmented objects. Thus even when the cost function accurately captures the object behavioral patterns for objects, the bipartite or multidimensional assignment algorithms may choose to assign an incorrect segmentation to a track in order to optimize globally across all segmentation results.

The multitemporal association tracker was originally written for analyzing organelle transport along neuronal axons. MAT has been used to identify differences in transport of brain-derived neurotrophic factor protein in Huntington's versus wild-type mice. By modifying only the cost function the same MAT tracker was shown to be effective in tracking neural progenitor cells. MAT with the same cost function has now been used to track thousands of cells across hundreds of datasets and highly varied imaging conditions. For example, MAT has been used for tracking adult and embryonic neural progenitor cells as well as hematopoietic stem cells.

Using the corrected data from the biological experiments discussed in chapter 4, MAT was compared with a linear assignment tracker similar to that used in TrackMate[48]. MAT has also been evaluated on simulated axon transport data and was compared with the particle tracker of Jaqaman *et al.*[49]. MAT along with a particle detection segmentation algorithm was also submitted for the simulated microtubule transport category of the 2012 International Symposium on Biomedical Imaging (ISBI) particle tracking challenge.

MAT implicitly maintains a dynamic tracking graph that is ideal for iteratively evaluating the most likely among a set of feasible segmentation ensembles. The segmentation from lineage framework makes heavy use of the integrated MAT to choose appropriate segmentation results and to maintain corrected lineage structure. The MAT graph concisely represents constraints from user edits and high-level knowledge, such as lineage information or automated mitosis detection. These constraints are then used in MAT to identify the most likely segmentations frame-by-frame.

5.1 The multitemporal association tracking algorithm

Segmentation from lineage (chapter 4) turns the standard automated processing pipeline of segmentation, tracking then lineaging back on itself, using lineage parent-daughter relationships and other high-level information to explore the most probable among an ensemble of segmentation results. The tracking graph is recomputed for these segmentation ensembles and the segmentations that best adhere to the tracking motion model are chosen. This approach assumes that the tracking motion model (cost function) accurately encapsulates the cellular behaviors, and relies on the multitemporal association tracker to provide the tracking graph framework that integrates high-level lineage constraints.

Given a set of all object detections in a sequence returned by a segmentation routine, we write the set of all detections in frame t as S_t and denote the i-th detection in frame $t-1$ as s_i and the j-th detection in frame t as s_j. Graph-based approaches such as MAT construct a tracking graph G with $S = \{S_t\}_{\forall t}$ as its vertices then all edges $e_{ij} = (s_i, s_j)$ of G represent feasible forward-associations of detections from frame $t-1$ to t. Any path τ through the graph G then represents a possible object track. Tracking is posed as a combinatorial optimization problem by assigning to each association a cost, representing the likelihood that an object assumes a given sequence of states, and choosing the set of tracks, T, so as to optimize some objective function over the edge costs. New tracks are initialized with previously untracked detection results. Tracking associations that constitute physically impossible behaviors of the objects are assigned an infinite cost. Tracks are extended so as to minimize the track costs for each track $\tau \in T$, using a minimum spanning tree approach[63].

We denote a partially constructed object track terminating at vertex s_i in frame $t-1$ as τ_i. Tracking considers all feasible extensions of τ_i passing through detections $s_j, s_k, ..., s_u$ in frames $t, t+1, ..., t+W$ respectively. The set of all such feasible extensions starting from s_j in frame t is written ρ_j. The cost of extending from s_i to s_j is defined as

$$c_{ij} = \min_{\rho_j}\{C(\tau_i, \rho_j)\}. \tag{5.1}$$

Where $C(\tau_i, \rho_j)$ is a cost function encapsulating typical object motion. A matching edge cost in frame $t-1$, c^*_{ij} exists if

$$c^*_{ij} \leq c_{in} \text{ and } c^*_{ij} \leq c_{mj}, \tag{5.2}$$

for all feasible edge extensions $e_{in} = (s_i, s_n)$ and $e_{mj} = (s_m, s_j)$. The set of all assigned edges from $t-1$ to t is $\{e^*_{ij}\}$, and tracks are extended by

$$\tau_j = (\tau_i, e^*_{ij}) \forall e^*_{ij}. \tag{5.3}$$

This optimization approach guarantees a minimum cost for the extended object tracks τ_j within the window size W. This follows directly from (5.2) since any edge with cost $c_{ip} < c^*_{ij}$ would imply c^*_{ij} does not satisfy (5.2) and thus that e_{ij} was not a matching edge. Similar to a minimum spanning tree approach we always choose the minimum cost edge to extend tracks[63]. One important consideration when using graph-theoretic approaches to analyze multi-target tracking problems is that whenever we extend a track, it changes the costs of other connected edges in the graph, due to the nature of the tracking problem. The algorithm must therefore be run iteratively extending tracks from previously assigned edges and subsequently computing new costs to future track extensions. This approach is similar to belief propagation approaches to the data association problem described in Chen *et al.*[71]. In contrast to Chen *et al.*, we relax the constraint that detections must be assigned to only one track allowing a softer assignment of tracks to detections.

This general algorithm may be applied to any tracking problem where window size (W) frames of detections are all available beforehand and a cost function encapsulating expected object behaviors can be defined. For post-processing within the tracker and downstream processing *e.g.* cell lineaging (see chapter 4), we have found it convenient to maintain the graph structure G during tracking. The tracking graph G is continually as the costs are calculated for each initial tracked frame, and as costs are recomputed during segmentation from lineaging.

Practical biological imaging scenarios are generally crowded and noisy. This is challenging for a tracker as it creates a large number of false detections, occlusions and under-segmentation errors. The minimum spanning tree implicitly handles false detection noise. Occlusions, under-segmentation and missed detection errors are handled initially by allowing longer l-frame track extensions for $l > 1$. For example, tracks that are not initially extended from $t-1$ to t can also be considered for two-frame extensions when computing matching edges in frame $t+1$. If these l-frame extension costs satisfy (5.1)-(5.3) then they will be assigned in the current frame allowing previously terminated paths to share or skip intermediate detections. If the cost is calculated using intermediate detections, we consider this an occlusion extension, costs calculated without considering intermediate detections correspond to missed detection errors. Allowing l-frame track extensions provides flexibility to the tracker to handle short-term segmentation errors during initial tracking. The segmentation from lineage framework is used to more effectively identify and correct even long-term segmentation errors to improve overall accuracy of the processing pipeline.

5.1.1 A cost function for stem cell tracking

Multitemporal association tracking integrates cost functions that accurately models the expected behaviors of the biological objects to be tracked. The MAT cost function for tracking proliferating cells is an enhancement to the stem cell tracker of Al Kofahi *et al.*[32]. The fundamental assumption of the tracking is that cells are imaged with sufficient temporal resolution so that they do not move significantly within a single frame. For adult and embryonic neural progenitor cells such as those discussed in chapter 4, 12 frames per hour, *i.e.* 5 minutes per frame, was an effective capture speed. In other applications, such as hematopoietic stem cells, a faster capture speed, *e.g.* 2 minutes per frame, was required[11].

The cost function uses the connected-component distance between detections in adjacent frames to model the assumption of slow relative cell motion. Connected-component distance between two segmentations s_i and s_j is the smallest pairwise euclidean distance between any pixel located in s_i and any pixel in s_j.

$$d_{CC}(s_i, s_j) = \min_{\mathbf{x}_i \in s_i, \mathbf{x}_j \in s_j} \{\|\mathbf{x}_i - \mathbf{x}_j\|_2\}. \tag{5.4}$$

In general when cells are moving slowly relative to the capture speed of the microscope, segmentations corresponding to the same cell will overlap in adjacent frames, causing the connected-component distance to be zero. In these cases, the connected-component distance cannot provide useful tracking relationships, instead a pixel overlap ratio is used.

$$d_{OR}(s_i u, s_j) = \frac{|s_i, s_j|}{\min\{|s_i|, |s_j|\}}. \tag{5.5}$$

The combined overlap and connected-component distance is the main component of the cell tracking cost. The cost function also incorporates direct euclidean distance between segmentation centers of mass, as well as segmentation size differences. These additional cost components have very low weights and only affect tracking when there are several feasible segmentations with similar connected-component distance.

The cost function is well-defined for tracking both 2-D and 3-D cells. A maximum cell motion threshold is used to limit the number of feasible tracks that MAT explores. The motion threshold is the only parameter used by the MAT algorithm for cell tracking.

5.1.2 Probabilistic interpretation of segmentation from lineage

Segmentation from lineage iterates through each frame t in order to identify the most likely segmentation results. For each frame we identify the set of segmentation labels that maximize the joint probability of segmentation and tracking in frame t given the previous segmentation and tracking and any constraints from higher-level knowledge. In particular, the segmentation from lineage technique focuses on the parent-daughter relationships of one or more cellular lineage trees. This section defines a general probabilistic inference interpretation of the segmentation from lineage approach.

We write the random variable representing segmentation labels in frame t as S_t. The segmentation labeling is a per-pixel labeling identifying connected components in frame t with unique positive unordered integer labels. A pixel label of 0 always represents background. The random variable representing the tracking edges terminating in frame t is denoted T_t. The frame t segmentation and tracking history, comprising all segmentation and tracking assignments before frame t, *i.e.* from $1 \dots t-1$ is denoted H_t. The set of all object or tracking constraints derived from lineages or other high-level information are denoted L. The segmentation from lineage focuses on constraining the number of cells in each frame, as well as the set of tracks that should be maintained, based on the parent-daughter relationships in L. Given the history $H_t = (S_{t-1}, T_{t-1}) \dots (S_1, T_1)$ and L, we can use Bayes' rule to write the joint frame t segmentation and tracking probability, conditioned on the history and lineage information as

$$P(S_t, T_t | H_t, L) = P(T_t | S_t, H_t, L) P(S_t | H_t, L). \tag{5.6}$$

Assuming a fixed set of detections from t to $t+W$ and given a fixed history H_t and lineage constraints L, the probability of tracking assignments terminating at frame t are independent in MAT. Thus $P(T_t | S_t, H_t, L) = \prod P(e_{ij} | S_t, H_t, L)$ where e_{ij} is a particular tracking edge assignment from detection $s_i \in S_{t-1}$ to detection $s_j \in S_t$. We encapsulate any temporal context in the tracking likelihood T_t, therefore the segmentation is independent of history, *i.e.* $P(S_t | H_t, L) = P(S_t | L)$.

Substituting into (5.6) gives

$$P(S_t, T_t|H_t, L) = \prod P(e_{ij}|S_t, H_t, L)P(S_t|L). \tag{5.7}$$

Note that we only concern ourselves with tracking edges that are part of the lineages in L, this is one of the important constraints of segmentation from lineage. We write the conditional tracking probability for edge e_{ij} in terms of the MAT edge assignment cost as $P(e_{ij}|S_t, H_t, L) = 2^{-c_{ij}}$. Then the negative log-likelihood of the segmentation and tracking in frame t is

$$-\log_2(P(S_t, T_t|H_{t-1}, L)) = \sum c_{ij} - \log_2(P(S_t|L)). \tag{5.8}$$

This equation is convenient as it represents the joint probability of segmentation and tracking in terms of the MAT tracking costs and a lineage-informed segmentation prior. Periodic fluorescent signals indicating *e.g.* cell nuclei or cell shape model information (see chapter 6) can be integrated into L or the segmentation prior to effect the refinement algorithm. As discussed below edge constraints such as lineage information are integrated into the cell tracking costs to provide important structural information to segmentation from lineage algorithms.

Equation 5.8 is the basis of the refinement framework, identifying the most probable segmentation ensembles frame-by-frame. Segmentation from lineaging incorporates a set of parent-daughter relationships from one or more cellular lineages into L, established either by automated image-based mitosis detection or by user identified mitotic events. For any frame t the set of starting nodes $s_i \in S_{t-1}$ is constrained by L to only the segmentations assigned to the lineage trees in L in frame $t-1$. So for any particular segmentation labeling $s_j \in S_t$ under evaluation, (5.8) can be optimized using bipartite assignment[63]. This allows the algorithm to evaluate each ensemble of segmentation results to find the segmentations that maximize the joint segmentation and tracking probability for frame t. We write that the subset of valid lineage assigned objects in frame $t-1$ is $s_i \in L_{t-1} \subset S_{t-1}$. For all valid labeled objects $s_i \in L_{t-1}$ we compute the costs to all feasible segmentations S_t in frame t. It is infeasible, however to exhaustively search the space of all segmentation label images S_t. The segmentation from lineage algorithm examines two possible scenarios for modifying initial segmentation results.

First, there may be a need to add new segmentations if the cost of all current track edges indicates they are very unlikely. We require a stringent cost threshold on the addition of new cell segmentations because the initial algorithms are more likely to produce false detections than missing detections in the case that the microscope is well focused. This threshold is currently set to one order of magnitude less than the maximum cell motion allowed.

Secondly, it may be necessary to partition the j-th segmentation into K subcomponents if K objects have lowest costs associated with the j-th detection in t. A partitioning of the cells is computed using a mixture of Gaussians model as discussed in chapter 4 or a pixel replicated mixture model as discussed in 6. The pixel replication model is a more robust model model for elliptical cells[7].

Once the new segmentations and partitionings have been computed the segmentation model that minimizes (5.8) is chosen. Unlike the initial multitemporal association optimization that uses a minimum spanning tree approach, equation (5.8) for a fixed history is optimized using a bipartite matching.

5.2 Comparison of multitemporal association tracking to standard linear assignment tracking

Multitemporal association tracking provides the basis for segmentation from lineage by facilitating the evaluation of segmentation ensembles in each frame. As such it is important to quantify the accuracy of the MAT algorithm. MAT has been shown to be effective to be effective in tracking both organelles in simulated microtubule and axonal transport data[3;12]. The results are summarized here. Additionally MAT is applied to corrected neural progenitor cell data and compared to a linear assignment tracker similar to that used in Trackmate[47;48].

5.2.1 Simulated microtubule and axonal transport data

It is of paramount importance to quantify and understand the accuracy and behavior of a tracking algorithm such as MAT. However, given the high variability of biological data and the difficulty of validation even for humans, it is often useful to evaluate tracking using simulated data with a defined noise model. In this section we discuss the results of application of MAT to two simulated organelle transport data sets. First we compare MAT to the linear assignment tracker of Jaqaman *et al.* as well as to a linear assignment optimization tracker on simulated axon transport data[49]. Next we summarize the results of the ISBI 2012 particle tracking challenge, microtubule transport category[12].

MAT was originally developed to quantify behavioral changes in protein transport along neuronal axons. Our simulated data was designed to mimic these axon transport problems. The motion model for axonal and microtubule transport is a constant speed, with occasional stops, following along an axon or microtubule curve. Rather than introduce an object detector, the axonal transport model directly simulates additional false detections distributed uniformly across the active area around the axon. The number of active vesicles transported, as well as the percentage of false detections were varied to show overall tracking behaviors. Each tracker was applied to 20 random replicates for each arrangement of active vesicles and percentage of false detections. Error rates were computed as the total invalid edge assignments over total active vesicle edges. Figure 5.1 summarizes the tracking comparison results. MAT performs best with an average error rate of around 1%-2% in highly crowded and noisy conditions. The linear assignment tracker using the same multitemporal cost function is nearly 10 times less accurate with error rates around 12% in the most difficult case. The single particle tracker of Jaqaman *et al.* has a much higher error rate in the worst cases, however it should be noted that it performs effectively in the low-noise, low-crowding scenarios and that the cost function was not tuned to this tracking problem[49]. These results are detailed further in Winter *et al.*[3].

The multitemporal association tracker was submitted to the microtubule transport category of an international particle tracking challenge in 2012. This challenge presented several example data sets of simulated microtubule transport for algorithm design, followed by independent evaluation on the official test data sets. The data was generated directly as images, requiring the incorporation of an object detection algorithm as well as the tracker under evaluation. Algorithms were evaluated both on detection and tracking accuracy at different object densities as well as different signal to noise ratios (SNR). Figure 5.2 shows some example simulated image frames at two different SNR levels. Overall the MAT tracker (algorithm 5) performed very well across multiple scoring criteria. MAT suffered most in the lowest SNR case, however improvement of the underlying detection algorithm would likely do more to improve this case than modification of the tracking algorithm or cost function. The full results of the challenge and discussion of advantages and disadvantages of the techniques are published in Chenourd *et al.*[12].

5.2.2 Tracking comparison of corrected neural progenitor cells data

The simulated results shown in the previous section highlight the efficacy of MAT for tracking organelle transport along microtubule-like structures. While the segmentation refinement algorithms can be applied to organelle tracking problems, most current applications are focused on cellular behavior analysis. In this section we present a tracking accuracy comparison of MAT applied to tracking the corrected neural progenitor data discussed in chapter 4.

We compare MAT to a linear assignment tracker using the same multitemporal assignment cost function as MAT. This is similar to the linear assignment tracker used by Trackmate, which is also based on Jaqaman *et al.*'s single particle tracker[48;49]. In order to avoid conflating detection and tracking errors, both trackers are run on fully validated segmentation data as discussed in chapter 4 and in Winter *et al.*[6]. Accuracy rates are aggregated across all tracks in all data sets. The accuracy is measured as in the previous section, the total incorrect edges divided by all validated tracking edges. Any edges that were not on validated clones in the embryonic NPC data are ignored.

Figure 5.3 summarizes the results of the comparison. The MAT algorithm improves upon linear assignment by approximately 80% reducing total tracking errors from 7% to 1.2%. These results are consistent with the simulated axon data of the previous section, as well as the results in Winter *et al.*[4].

5.3 Multitemporal association tracking provides accurate object motion models for segmentation from lineaging

Multitarget tracking is an increasingly important aspect of live-cell biological analysis. Current approaches to multitarget tracking are dependent on a host of *a priori* probabilities that require extensive tuning for different imaging conditions and cell culture densities. By minimizing the association cost between current tracks and future possible paths over the assigned edge extensions rather than overall detections, MAT is more robust to segmentation errors and noise. MAT is also at the core of the segmentation from lineage algorithms, providing temporal context and integrating lineage information to infer the best segmentation ensemble given the evidence.

MAT has been applied to phantom data, fluorescently labeled brain-derived neurotrophic factor (BDNF) organelle transport along the axon, and Golgi-derived vesicle transport along the axon, and phase contrast images showing clonal development of embryonic and adult neural stem cells. In the stem cell application, MAT was able to track entire clones through multiple levels of cell division. In other studies we have used MAT along with the segmentation from lineage framework to track clonal development of hematopoietic stem cells, endosome transport in the developing retina, adult neural stem cells and 3-D fluorescence image data showing neural stem cells in their vascular niche[9–11].

Overall, a single implementation of MAT with different cost functions for organelle transport and stem cells has been applied to hundreds of image sequences and has consistently outperformed other tracking approaches. In crowded and noisy simulated organelle data, MAT achieved an error rate of 1.6% compared to an error rate of 12% for a linear assignment tracker using the same multitemporal cost function, and 72% for the single particle tracking solution developed by Jaqaman *et al.*[49]. MAT was a competitive entry in the international tracking challenge, outperforming the other entries in the microtubule transport category for signal-to-noise ratios above 1[12]. The simulated axon transport results suggest that improving the spot detection algorithm used in the challenge would allow for high-accuracy tracking even at lower SNR conditions. For stem cell tracking, MAT achieves an average error rate of 1.2%, compared to 8.6% for the stem cell specific tracker developed in[32].

Segmentation from lineage uses MAT to evaluate ensembles of segmentations to find the most likely segmentation give temporal and lineage evidence. The most accurate approach to refinement computes costs for each ensemble of segmentation partitions from $1 \dots k$ components for each cluster of objects that may require repartitioning, these ensemble hypotheses also need to be computed out to the window size $t+W$. The optimal segmentation ensemble is then chosen to minimize (5.8). This can be time consuming, in practice, we recompute the costs on only the K-subcomponent partition in frame t. Tracking assignments are then performed on this partitioning. These approximations have been effective in all applications of MAT and segmentation from lineaging to date, however the overall performance of the algorithm would likely improve by computing all ensemble hypotheses.

Multitemporal association tracking has been used to effectively track thousands of cells in vastly different imaging conditions[3;4;11]. MAT does require that a cost function be tailored to accurately capture the typical behavior of cells. However, stem cell tracking cost function is very accurate in tracking a wide range of cell types as long as the temporal resolution assumptions remain valid. The integration of MAT into the segmentation from lineage approach has been used to reduce both segmentation and tracking error rates making it feasible for users to correct and validate cell segmentation and tracking results. As discussed in chapter 4 key biological insights have been discovered through the use of the segmentation from lineage framework.

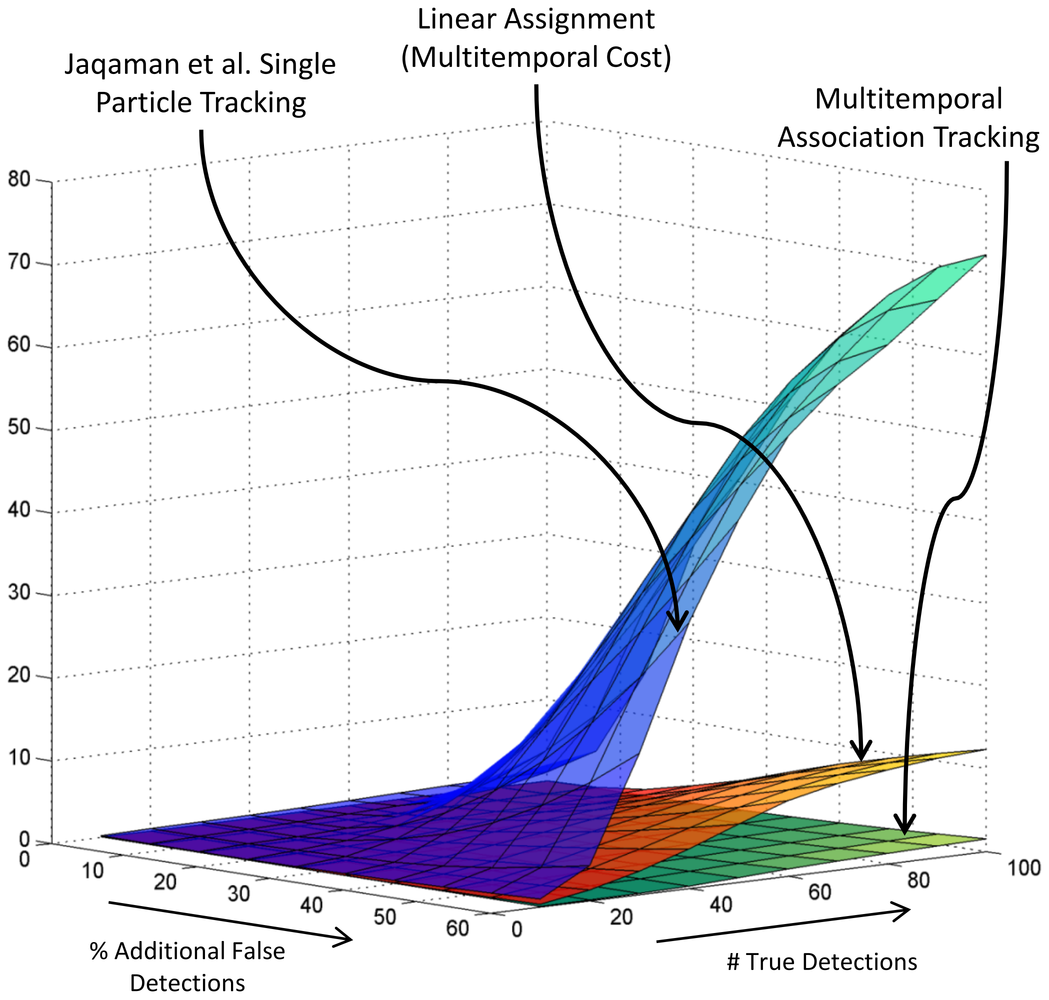


Figure 5.1: A comparison of tracking methods on simulated axonal transport data. Comparison of error rate for MAT (green surface), a linear assignment tracker using the same axonal transport cost function (orange surface) and a conventional single particle tracking solution using source code provided by Jaqaman *et al.*[49] (blue surface). MAT is significantly more accurate in the presence of high noise and target density. In the highest noise and target density simulations, MAT achieves an average error rate of 1.6% compared to 12% for the lineage assignment tracker which uses the same cost function.

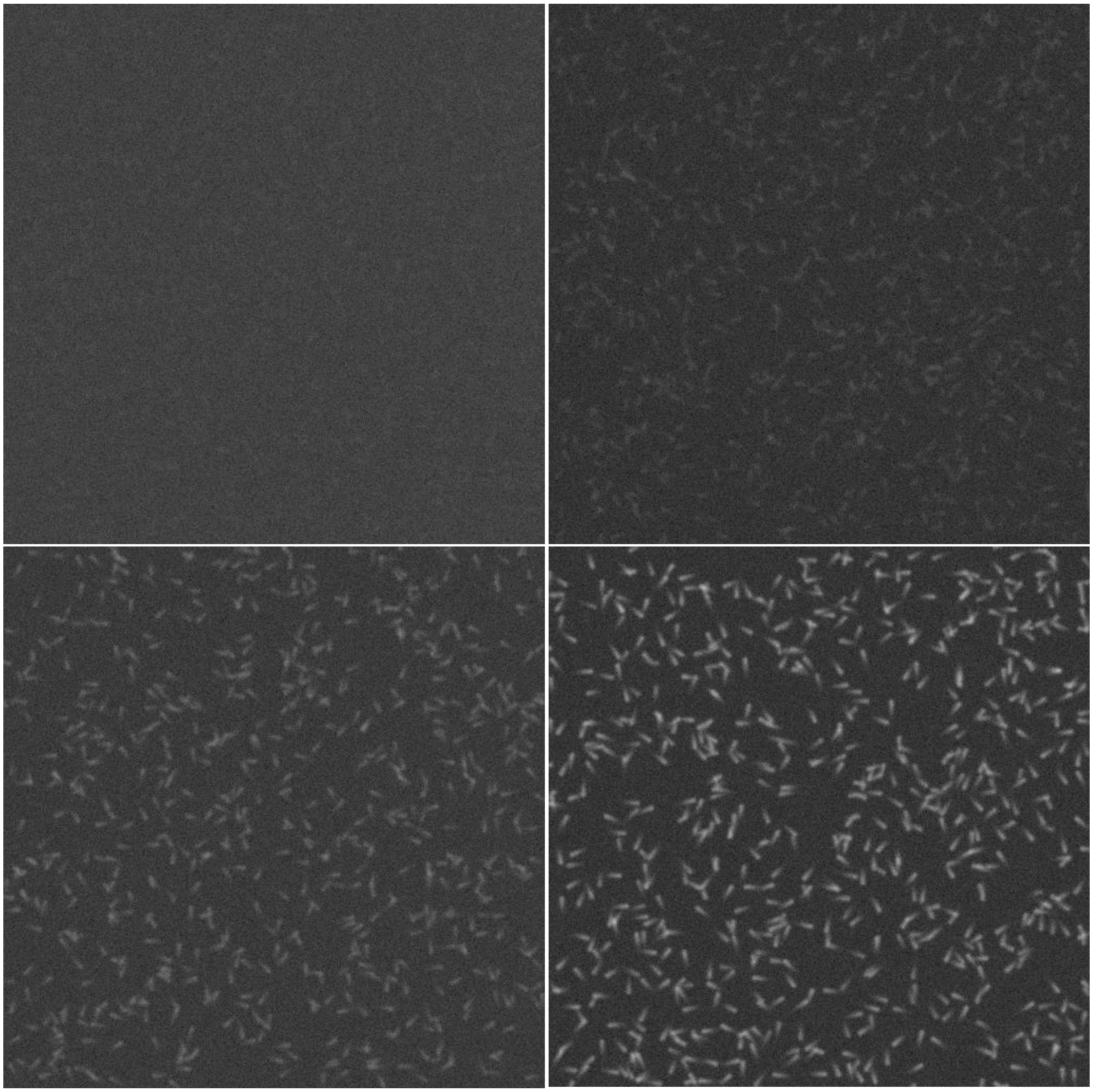


Figure 5.2: Example simulated microtubule transport image frames from 2012 ISBI particle tracking challenge[12]. The example image frames shown have progressively higher signal-to-noise ratios (SNR), $SNR = 1$ (top left), $SNR = 2$ (top right), $SNR = 4$ (bottom left) and $SNR = 7$ (bottom right). The multitemporal association tracker (algorithm 5) was one of the most effective trackers for $SNR > 1$. Improving the point detection algorithm used for the challenge would likely make MAT effective even in the lowest SNR case.

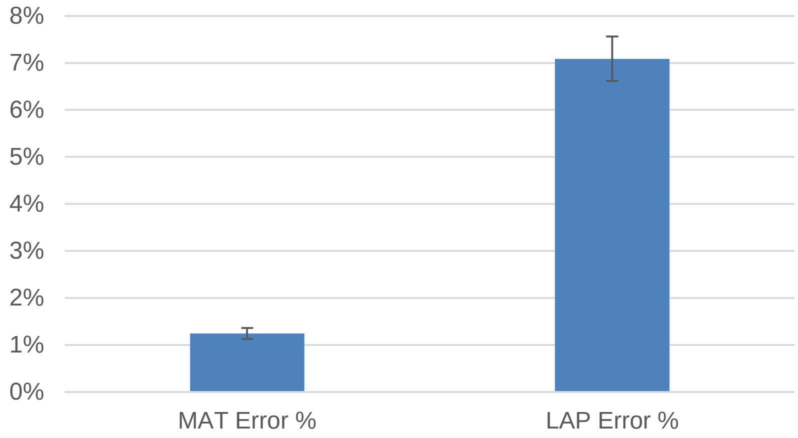


Figure 5.3: Tracking comparison on validated data of embryonic neural progenitor cells. Overall tracking error rates for multitemporal association tracking (left) and linear assignment optimization using the same multitemporal cost (right). Multitemporal association tracking (MAT) improves upon linear assignment (LAP) by 80%. This is consistent with the simulated axonal transport results shown in figure 5.1. Whiskers indicate 95% confidence interval.

Chapter 6: Partitioning touching elliptical cells using pixel replication

In biological microscopy, cell segmentation delineates individual cells within each image. This is one of the most important, complex and error-prone tasks in image analysis. While there are seemingly endless combinations of algorithms for cell segmentation, there are two broad categories of tasks common to many segmentation algorithms[72;73]. The first task is thresholding, converting the image to regions of foreground and background pixels (or voxels in 3-D). The second task is object partitioning, identifying which of the foreground pixels belong to each individual cell. This object partitioning step is a key challenge and is the focus of the present work. Partitioning foreground regions is also an important component of the segmentation from lineage framework, enabling the evaluation of feasible partitioning of previously identified segmentations in the case that a cell cluster may have been misidentified. Given a set of pixels that touch each other, called "connected components," the goal is to identify which pixels belong to which objects. Here we describe a new approach for the object partitioning task called pixel replication (PR).

Pixel replication transforms data from the threshold image to better approximate data generated by a Gaussian mixture (GM). Pixel r proceeds as follows. First, the distance image is computed, with each foreground pixel storing the distance to the nearest background pixel. Next, the spatial location of each foreground pixel is entered repeatedly in a list the "replication" step with the number of repeats based on the distance image value. Segmentation then proceeds as with fitting a GM to the spatial locations of foreground pixels in the threshold image, but instead it is fit to the pixel replication list. Pixel replication significantly improves on approaches based on watershed algorithms, and on using GMs fit to the threshold image. The data generated by pixel replication is more accurately described as a non-linear mixture of elliptical triangular distributions, but fitting a Gaussian mixture to the pixel replication data still improves upon a spatial Gaussian mixture model fitting in practice[20].

An important component of cell partitioning is finding the number of cells in a foreground region.

The parameter describing the number of cells in each component is often referred to as 'K'. Finding the number K that best fits a set of data is a difficult structural task. The watershed algorithm estimates K directly by using the number of basins found in the foreground region. This is not a robust estimator for K since watershed basins vary greatly with the eccentricity of the data and the level of noise in the threshold image. watershed will often significantly overestimate K. An imaging and object dependent smoothing must generally be applied as part of the watershed algorithm to alleviate this problem.

Clustering algorithms such as pixel replication and Gaussian mixture models can also estimate K from the data. This is difficult as goodness of fit measures for clustering algorithms tend to improve monotonically as the number of clusters K grows, encouraging over-segmentation. This has led to the development of a number of criteria that can be used for identifying the most appropriate number of clusters in a region. Popular methods that can directly estimate K from a foreground region include the Aikake information criterion (AIC), Bayesian information criterion (BIC), and the gap statistic[19].

Alternatively, temporal information from the tracking algorithm can be incorporated into the segmentation[25;56;74]. If available, population information from cell lineaging directly provides K to the underlying segmentation; the improved segmentation results then improve the tracking results[4–6;10]. This technique has proven very effective in several applications and was integral to acquiring the validated neural progenitor cell (NPC) data used here for tracking-based comparison of the watershed, pixel replication, and mixture of Gaussians algorithms.

The following sections detail the direct implementation of the pixel replication algorithm. A distribution sampling approach is also discussed to reduce the memory and time required in the case of large connected components. We also utilize the theory of general elliptical distributions to model threshold and distance transformed images[20]. In particular, covariance relationships between Gaussian, uniform elliptical and triangular elliptical distributions are derived. This enables the computation of isocontours representing elliptical boundaries using Gaussian mixtures or pixel replication parameter fit. Finally, the new pixel replication technique is compared to both watershed and Gaussian mixtures using simulated overlapping elliptical data in 2-D and 3-D. We also use tracking error rates to compare the segmentation techniques on real validated neural stem cell (NSC) data. Pixel replication improves error rates over watershed and GM partitioning for both simulated and real data.

6.1 The pixel replication algorithm

The implementation of pixel replication (PR) takes as input a threshold image and the number of cells K in each connected component of foreground pixels. The threshold image contains ones at each foreground pixel location and zeros at each background pixel location. We use the term pixel here to refer to both pixels and voxels and below omit the explicit z, but the application of PR is independent of spatial dimension and is well-suited for 3-D data.

PR starts with a threshold image T. The list of (x, y) coordinates of the foreground pixels of T is denoted as $T_{xy} = \{(x, y)_1 ... (x, y)_N\}$, with N being the total number of foreground pixels in T. The distance transform of image T is taken, and $d(x, y)$ denotes the distance from pixel location (x, y) to the nearest background pixel. This value is rounded to the nearest integer and scaled if desired (see below). Next, a list $D_{xy} = \{(x, y)_1^{d(x,y)_1} ... (x, y)_N^{d(x,y)_N}\}$ is generated, where the notation $(x, y)^{d(x,y)}$ means to repeat location (x, y) in the list $d(x, y)$ times. Finally, a Gaussian mixture (GM) with K components is fit to D_{xy} and the segmentation result is obtained by clustering the locations T_{xy} using that GM. A summary and example of the PR for two overlapping ellipses is shown in figure 6.1.

Combining different imaging modalities with pixel replication, as with the T cell FUCCI dataset from figure 6.7, is accomplished by thresholding each channel separately and combining the distance transforms from the various channels using addition. Combining edge, or gradient, information with PR can be done in two ways. One method is to threshold the gradient image and combine that logically with the thresholding of the intensity image. This approach has been used previously with the morphological gradient [4;6]. An alternative approach is to combine the gradient image with the distance transform to produce an edge-weighted distance transform [52]. That is the approach used for the human embryo images in figure 6.8 panels C-D.

6.1.1 Pixel replication with distribution sampling

The memory and time requirements for pixel replication, given a d-dimensional connected component of maximum radius r, is $O(r^{d+1})$. A d-dimensional sphere has volume proportional to r^d and the additional r factor is due to the linear increase of the distance transform up to r at the center of the d-sphere. There are two approaches to make pixel replication more efficient for large connected components. The first alternative would be to reduce the distance transform image by a constant scale factor before replicating. In practice we have found this to produce good results, but it may produce structured quantization errors. The second alternative is to treat the distance transform image as an empirical probability distribution and sample a specified number of points from this distribution. The method described here is a variant of the well-known rejection technique, modified to be more efficient for distributions with small support[75].

First, randomly choose a point (x, y) on the connected component. Next, generate a uniformly distributed acceptance probability P on $[0, 1]$. The point (x, y) is accepted if $P \leq f(x, y)/cu(x, y)$ where function $f(x, y)$ is the probability distribution estimated from the distance transform image and $u(x, y)$ is the constant uniform probability on the nonzero region of f. The value c is the marginal expected number of rejections per point computed as $\max_{x,y}\{f(x, y)/u(x, y)\}$. When the generated points are not pixel aligned we approximate f using local linear interpolation. This method is repeated until a specified number of points is generated. The accepted points from this process will follow distribution f.

6.1.2 Generation and evaluation of simulated overlapping ellipse data

The performance of pixel replication was evaluated first using the accuracy of the segmentation at each pixel on simulated overlapping elliptical regions. Randomly generated ellipses were created with principal axis radii uniformly distributed on $[10, 30]$. The ellipses were then rotated by a uniformly distributed angle on $[0, 2\pi]$. This was done one thousand times each for two, three, four and five ellipses. To combine ellipses for an overlap image, points on each ellipse are picked at random, and the centroids of the ellipses are translated by the vector connecting the two randomly chosen points. This allows us to evaluate the accuracy of different partitioning approaches for every pixel. The same process was repeated for a 3-D dataset.

Each of the four thousand simulated ellipse images was segmented using (1) pixel replication, (2) Gaussian mixtures fit directly to the spatial locations of the threshold image, (3) a waterfall algorithm using a region merging step with the watershed approach to partition to K regions, and finally by (4) a seeded watershed that uses the centroids from the pixel replication segmentation result as seeds together with a geodesic distance measure. Since partitioning is invariant to permutations of the cluster numbering, all label permutations for each technique were evaluated and the lowest error rate labeling was chosen.

A balanced error measure was used to compare the techniques. The balanced error rate is computed as the average across each ellipse of the number of misclassified pixels divided by the total area in each ellipse that could potentially be misclassified. Regions of the image that are overlapped by all ellipses cannot be misclassified and are excluded from the error rate calculation. The balanced error rate penalizes percentage of ellipse regions misclassified equally across all ellipses. This is important when partitioning overlapping ellipses of very different size. By contrast, direct misclassification rates encourage completely misclassifying a small ellipse in favor of correctly classifying all pixels of a large neighboring ellipse.

6.1.3 Comparing tracking results with different segmentation algorithms

Algorithms for object partitioning were next evaluated based on how many tracking errors the different partitioning approaches caused. The tracking errors were evaluated on time-lapse microscopy image sequences of a mixture of several kinds of proliferating cells derived from the adult mouse forebrain neural stem cell zone, here referred to collectively as neural stem cells (NSC). The NSC data, captured in eight different imaging experiments from 232 movies, contained 379 clones or family trees, 1,130,496 total segmentations and 108,178 segmentations that required partitioning. This data was manually validated using the LEVER program with any tracking errors corrected[5]. The fully validated dataset, including image data and segmentation, tracking and lineaging results, can be explored using the interactive CloneView tool, `http://n2t.net/ark:/87918/d9wc73/nscProcesses/index.html?Adult`.

The performance evaluation for tracking accuracy starts by segmenting each connected component in the entire dataset that required partitioning using each of the four different algorithms. Each algorithm is provided with the correct value of K from the ground truth data. The re-partitioned cells are matched with cells in the manually corrected ground truth data by maximal overlap between ground truth segmentation and the partitioned cell. The tracking for all cells in each image frame is then recomputed, using the same multi-frame tracking algorithm (MAT) used by the LEVER program[3;12]. All tracking connections, on validated trees, that differ from the ground truth tracking are counted, and then reset to the correct value so that errors do not accumulate. This process is repeated for every image frame in the sequence. A similar method was used in previous work for comparing tracking performance with the full segmentation results to tracking performance using only the segmentation centroids[6].

6.1.4 Modeling the threshold and distance transform images with elliptical distributions

The Gaussian distribution belongs to a family of distributions known as elliptical distributions. The Gaussian distribution has a number of properties that make it highly effective for partitioning cells from regions of connected foreground pixels. First, the isocontours of the Gaussian distribution are elliptical. Second, techniques to fit mixtures, or linear combinations of Gaussians based on expectation maximization, are widely available and practically effective[19;21]. The idea behind pixel replication (PR) is to make the data obtained from the threshold image better suited to fitting with a Gaussian mixture. This section describes the use of the uniform elliptical distribution as a model for the threshold image, and the triangular elliptical distribution as a model for the distance transform of the threshold image. Equivalence relationships among the covariance and shape matrices for the Gaussian, the uniform and triangular elliptical distributions are derived. This allows us to determine the equivalent cell boundaries directly from the covariance matrix obtained from the Gaussian fit. This also gives a theoretical basis for the improved performance of pixel replication.

The uniform elliptical distribution is a constant-valued probability density function that is zero everywhere outside of an elliptical boundary. This accurately models the threshold image for cells with elliptical shape. The triangular elliptical distribution is also zero everywhere outside an elliptical boundary, but has a probability value that increases linearly towards the center of the ellipse. The triangular elliptical distribution is an accurate model of the distance transform of a threshold image for circular cells. For non-circular shapes the triangular elliptical distribution provides an effective analytical approximation.

Gomez *et al.* provide a comprehensive survey of the family of elliptical distributions, and describe a technique for calculating their moments[20]. Following their approach, we derive here the relationship between the covariance matrices for the Gaussian, uniform and triangular elliptical distributions. The approach starts with a shape matrix Σ whose eigenvalues and eigenvectors define the shape of the elliptical isocontours of the distribution. A d-dimensional elliptically distributed random variable $X \sim E_d(\mu, \Sigma, g)$ has centroid μ, shape matrix Σ and $g(t), t \in [0, \infty)$ a non-negative function describing the relative distribution of mass along the squared radius of the ellipse. For example, $g(t) = e^{-t/2}$ corresponds to the family of Gaussian distributions parameterized by μ and Σ where the covariance matrix and the shape matrix are equivalent. Constraining the domain of g to $[0, 1]$ for the uniform and triangular elliptical distribution allows the boundaries to be completely defined by Σ. A valid elliptical distribution must also have finite radial mass[20],

$$M_d = \int_0^\infty t^{\frac{d}{2}-1} g(t) dt < \infty.$$

We construct an elliptically distributed random variable $X \sim E_d(\mu, \Sigma, g)$ stochastically by sampling from the compound random variable $X = \mu + A^T R U^{(d)}$ where $\Sigma = A^T A$, $U^{(d)}$ is a random variable distributed uniformly on the unit sphere in $\mathbb{R}^d$ and R is a random variable with probability density function

$$h(r) = \frac{2}{M_d} r^{(d-1)} g(r^2), r \in [0, \infty).$$

This representation is convenient for simulating from any elliptical family and it also provides a form for relating the statistical properties of X to the moments of R. In particular, if $\mathbf{E}[R]$and $\mathbf{E}[R^2]$

exist then $\mathbf{E}[X] = \mu$ and

$$\mathbf{E}[X] = \mu, \tag{6.1}$$

and the covariance of the distribution is

$$\mathbf{Cov}[X] = \frac{1}{d}\mathbf{E}[R^2]\Sigma. \tag{6.2}$$

To compute the Gaussian, uniform and triangular elliptical distributions with the equivalent covariances then requires only the second moments of R. For the uniform elliptical distribution in d-dimensions let $g(t) = 1$for $t \in [0, 1]$ and zero otherwise. The radial mass is then written as

$$\begin{aligned} M_d &= \int_0^\infty t^{d/2-1} g(t) dt \\ &= \int_0^1 t^{d/2-1} dt \\ &= \frac{2}{d}. \end{aligned}$$

Since M_d is finite this is a valid elliptical distribution, and the second moment $\mathbf{E}[R^2]$ is

$$\begin{aligned} \mathbf{E}[R^2] &= \int_0^\infty r^2 h(r) dr \\ &= \frac{2}{M_d} \int_0^1 r^{d+1} dr \\ &= \frac{d}{d+2}. \end{aligned} \tag{6.3}$$

Substituting (6.3) into (6.2), $\mathbf{Cov}[X] = \frac{1}{d+2}\Sigma$ for the uniform elliptical distribution in d-dimensions. Using a Gaussian fit to uniform data a multiplier of $\sqrt{d+2}$ on the principal radii obtained from the eigenvalues of the covariance matrix would represent the boundary of the uniform data. Figure 6.2 shows a standard 1-D uniform elliptical distribution with zero mean and unit variance. Note that the radius of the distribution (boundary distance on the x-axis) is $\sqrt{3}$.

Similar to above we can derive the scale factor for a triangular elliptical distribution, setting

$g(t) = 2(1 - t^{1/2})$ for $t \in [0, 1]$ and zero otherwise,

$$\begin{aligned} M_d &= \int_0^\infty t^{d/2-1} g(t) dt \\ &= 2 \int_0^1 t^{d/2-1} - t^{(d+1)/2-1} dt \\ &= 2 \left(\frac{2}{d} - \frac{2}{d+1} \right) \\ &= \frac{4}{d(d+1)}. \end{aligned}$$

Again this is a valid elliptical distribution with second moment $\mathbf{E}[R^2]$,

$$\begin{aligned} \mathbf{E}[R^2] &= \int_0^\infty r^2 h(r) dr \\ &= \frac{4}{M_d} \int_0^1 r^{d+1} - r^{d+2} dr \\ &= d(d+1) \left(\frac{1}{d+2} - \frac{1}{d+3} \right) \\ &= \frac{d(d+1)}{(d+2)(d+3)}. \end{aligned} \tag{6.4}$$

Substituting (6.4) into (6.2), the covariance for the triangular elliptical distribution is related to the shape matrix Σ of the elliptical boundary by $\mathbf{Cov}[X] = \frac{d+1}{(d+2)(d+3)} \Sigma$. Scaling the principal radii obtained from the eigenvalues of the covariance matrix by a factor of $\sqrt{\frac{(d+2)(d+3)}{d+1}}$ will give the boundary radii for a triangular elliptical distribution. In the case of a 2-D elliptical triangular distribution this scale factor is $\sqrt{20/3}$. These scale factors provide a method for visualizing ellipse fits based on Gaussian mixture models or pixel replication. Figure 6.2 shows the standard triangular elliptical distribution with zero mean and unit variance. Note that the radius of the distribution is given by $\sqrt{6}$.

6.1.5 Identifying elliptical decision boundaries from pixel replicated data

As described above, the elliptical triangular distribution is an effective model for the pixel replication data for a single cell. Accurately modeling multiple overlapping cells requires a non-linear mixture of elliptical triangular distributions. The parameters of this mixture model can be found using non-linear optimization techniques, but in practice we have found that fitting Gaussian mixtures has better convergence properties. Gaussian mixtures have infinite support and are smooth, facilitating the search for optimal solutions.

Gaussian mixtures are generally fit using an expectation maximization (EM) algorithm[19;21]. Implementations of this are available in most image analysis platforms, including MATLAB, OpenCV, Java and Python. For our purposes, we repeat the EM algorithm five times with different randomly chosen initial values. The replicate with the maximum overall log-likelihood is selected as the best partitioning. A maximum of 100 iterations is allowed for each replicate, and the algorithm is considered to converge if the overall log-likelihood changes by less than 10^{-6}.

6.2 Accuracy of pixel replication compared with watershed approaches

The accuracy of the pixel replication approach was compared on both simulated overlapping elliptical data and real adult and embryonic stem cells imaged using phase contrast microscopy. For both the simulated and real data pixel replication was compared against a spatial Gaussian mixture fitting as well as two watershed transform methods. The first watershed approach, often referred to as the Waterfall algorithm iteratively merges regions until only K regions remain[64]. Waterfall uses a simple merge strategy that nevertheless produces good results in many cases. The basin minimum to edge height is computed for all regions and the edge with the smallest difference is merged iteratively. An analogy to this merge approach is to raise the "water level," beginning at each basin minimum, at the same rate. The first basin region that overflows will be merged into the receiving neighbor and the process is repeated iteratively until the specified number of regions are left. Pixel replication was also compared to a seeded watershed approach[76]. The seeded watershed algorithm uses centroids found by pixel replication as the seeds. These seed points are treated as basin minima and the standard watershed transform is applied to cluster the foreground to the nearest points.

Segmentation pixel accuracy was first evaluated directly using simulated overlapping ellipse data. Random configurations were clustered using all four algorithms. Data containing 2, 3, 4, and 5 elliptical components was generated for both 2-D and 3-D data. A balanced error rate was used for evaluation of accuracy. Figure 6.3 shows a sample of each clustering method applied to simulated

2-D ellipse data containing 3 overlapping ellipses, balanced error rates on this example is noted below each technique in parentheses. Similarly, figure 6.4 gives an example of partitioning 3 overlapping ellipsoids in 3-D. Each algorithm was given the correct number of clusters K for each configuration. Pixel replication significantly outperformed both watershed techniques and the direct application of GM to the threshold pixel locations (p-values using paired Wilcoxon signed-rank test[18]). Figure 6.5A summarizes the results of the simulated data comparison. PR was consistently better as measured by segmentation accuracy on the synthetic data ($p < 10^{-23}$).

Second, tracking accuracy was compared against manually validated tracking results from the aging neural stem cell (NSC) image dataset (discussed in chapter 4) consisting of 8 experiments with 232 movies and a total of 1,130,496 tracking connections. Tracking errors make an excellent functional performance metric for a segmentation algorithm since single tracking errors can impact a large portion of the data being analyzed. Tracking errors can be corrected quickly whereas pixel accurate segmentation correction is time-consuming and difficult. Segmentation correction is also highly user dependent making it difficult to establish a trusted ground truth. Many applications are also robust to segmentation errors but require that tracking errors be manually corrected[2]. Each cluster of segmentations in the adult NSC datasets was reevaluated using all four partitioning methods then tracked frame-by-frame, counting tracking errors for each method. An example segmentation from this dataset is shown in figure 6.6. All of the methods were given the correct number of cells, K, for each segmentation. Pixel replication again outperformed direct spatial Gaussian mixture fitting as well as the watershed techniques, reducing the number of overall tracking errors in the NSC data ($p < 10^{-5}$). Figure 6.5B summarizes the tracking comparison results. For the NSC dataset, pixel replication improved the tracking accuracy compared to Gaussian mixtures by what seems a small margin 0.06%. However, for the over 1 million tracking assignments in just one of the datasets considered here this is a reduction of 7,308 tracking errors that would otherwise need to be corrected manually. For real NSC data and 3-D simulated data there is less overlap compared to 2-D simulated data and the cells show more eccentricity. Increasing the eccentricity of the 2-D simulated data increases the accuracy of the GM and PR algorithms relative to the watershed approaches because the Gaussian boundaries are elliptical while the watershed uses Euclidean or geodesic distance. Pixel replication cuts error rates by more than a factor of two compared to the waterfall algorithm for both real and synthetic data.

Pixel replication has been successfully applied to a wide variety of cell types and imaging modalities. Several examples of pixel replication segmentation results are shown in figures 6.6 and 6.7. These examples were chosen because they are moderately difficult yet the correct partition is visually apparent. Figure 6.6 shows mouse adult neural stem cells. There are two cells in the image. Despite the lack of gradient information between the two cells pixel repliation still separates the cells effectively using only the morphology implied by the distance transform, overlaid as a gray surface in figure 6.6B. The individual Gaussian components fit for each adult neural stem cell are also shown as magenta and yellow surfaces. Figure 6.7 shows mouse T lymphocytes imaged using both bright-field and fluorescence ubiquitination cell cycle indicator (FUCCI)[1]. The FUCCI signal is not present in every cell. Pixel replication is able to naturally fuse information from the fluorescence and bright-field threshold images by combining the depth transform images from each channel. Figure 6.8 panels A-B show human embryonic stem cells labeled with a fluorescent nuclear marker (Histone-2B). PR was able to accurately recover the segmentation based only on the morphology from the threshold image, without requiring gradient information. Figure 6.8 panels C-D show pixel replication applied to segmenting cells within a human embryo during its progression to blastocyst. Here, edge gradient information is incorporated to better separate the highly overlapping cells; a weighted linear combination of the Hough accumulation array with the distance transform image is used as input to pixel replication algorithm[52]. Finally, in figure 6.8 panels E-F, 3-D image data of mouse neural stem cells labeled with GFAP-GFP are shown. In 3-D, pixel replication is used in exactly the same manner as 2-D. The key point is that the same implementation, with no additional logic, achieves excellent results on all of these examples.

6.3 Pixel replication accurately partitions elliptical cells

Segmentation algorithms are often the first step in automated analysis of biological image data, and many downstream results are dependent on the accuracy of the segmentation. As such, segmentation algorithms are one of the most important and often error-prone components of an image processing pipeline. Often a simple thresholding approach will effectively separate foreground regions of cells from the background, however these regions will generally contain clusters of cells.

Pixel replication is a new segmentation technique for identifying objects, such as cells in a thresholded connected component. The key idea of pixel replication is that foreground regions often represent a cluster of objects and must be partitioned into multiple components. This is an alternative to the watershed approach for splitting foreground regions into individual cell objects. Pixel replication uses the distance transform of foreground regions to improve the fit of a spatial Gaussian mixture model to threshold pixel coordinates. PR is robust to noise in the threshold boundary and effectively fits objects with elliptical or approximately elliptical shape. Unlike the watershed transform, pixel replication does not estimate the number of cells in each region K. The knowledge of K is an important parameter for pixel replication. Several techniques, *e.g.* AIC, BIC, gap statistic, etc. can be used to estimate K directly from the fit data. Temporal context has also been used to accurately estimate the number of cells in each foreground cluster[6].

The pixel replication algorithm significantly outperforms watershed techniques as well as spatial Gaussian mixtures for separating the pixels of simulated overlapping elliptical data. PR also significantly improved tracking accuracy on validated adult neural stem cell data. The data was comprised of 8 experiments and 232 data sets. Using either watershed or Gaussian mixture fitting produced thousands more tracking errors that would require correction before further statistical analysis. This is a key improvement because many algorithms are robust to segmentation errors but require accurate tracking to effectively identify behavioral differences among cells[2].

Pixel replication assumes that objects have elliptical boundaries. This is reasonable for many cell types and morphologies, particularly stem cells. However, this is not a practical technique for fitting *e.g.* tubular structures such as blood vessels. PR also requires solid objects suitable for computing a distance transform. In cases where objects contain large holes or extreme internal irregularities that make the distance transform unreliable, segmentation may be more accurate using a non-replicated GM. A compelling area for future research would investigate how effectively the distance transform mass distribution is fit by the GM model. This could lead to a hybrid approach for partitioning elliptical objects with varying morphology.

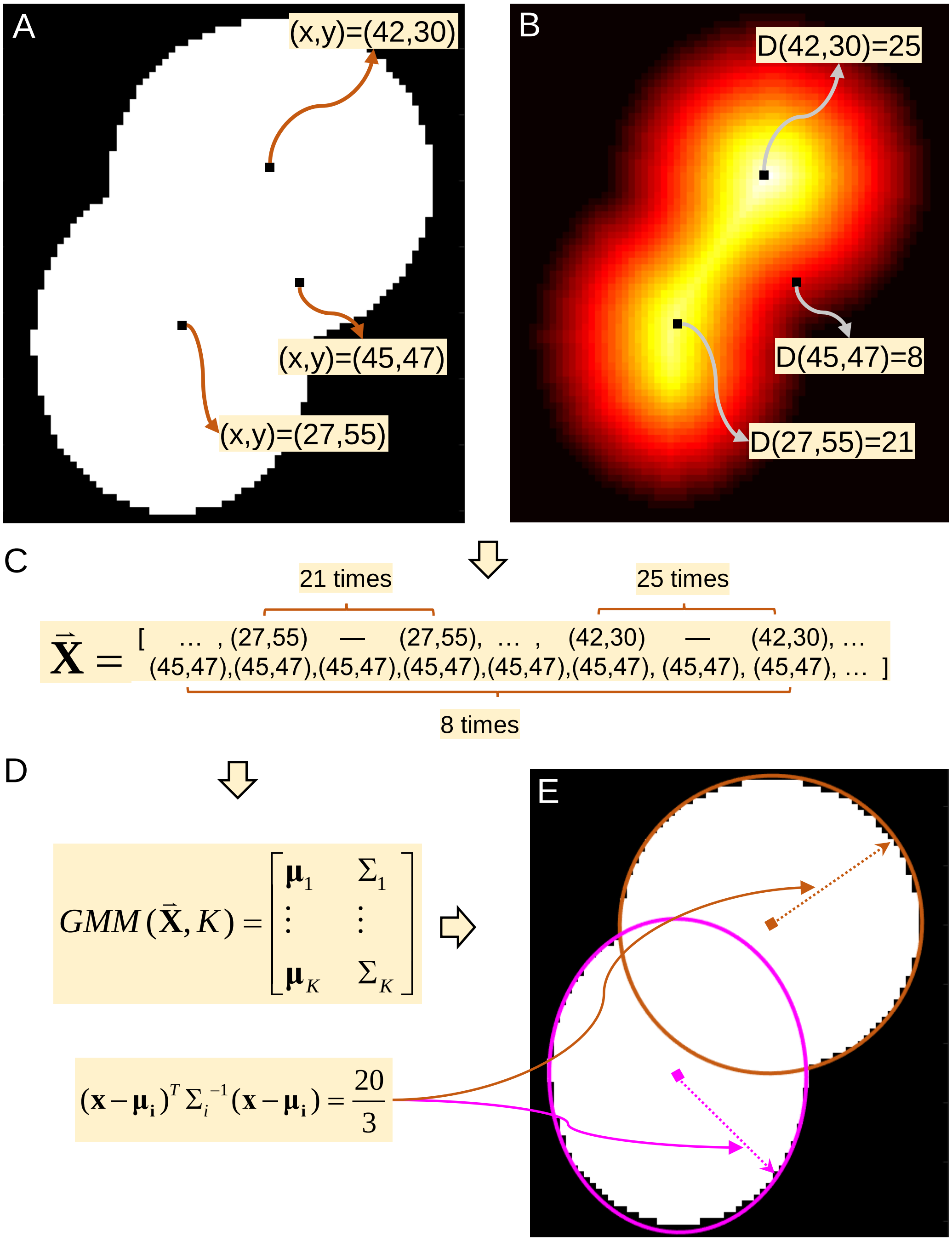


Figure 6.1: Overview of pixel replication (PR) algorithm. PR takes a threshold image (A) and the number of cells K in each connected component as input. The Euclidean distance transform, representing the distance from each foreground pixel to the nearest background pixel is computed (B). A list $\vec{\mathbf{X}}$ is generated by entering the spatial coordinates of each foreground pixel repeatedly. Each coordinate is repeated a number of times equal to the distance value at that pixel (C). The replicated coordinates are fit to a Gaussian mixture model (D). Each Gaussian component represents a single cell. Pixels are separated into individual cells using a maximum likelihood classifier. Elliptical cell boundaries are recovered using the covariance relationship between the Gaussian and elliptical triangular distributions derived in the present work (E), for 2-D the cellular boundary corresponds to the isocontour at a Mahalanobis distance of $\sqrt{20/3}$[19].

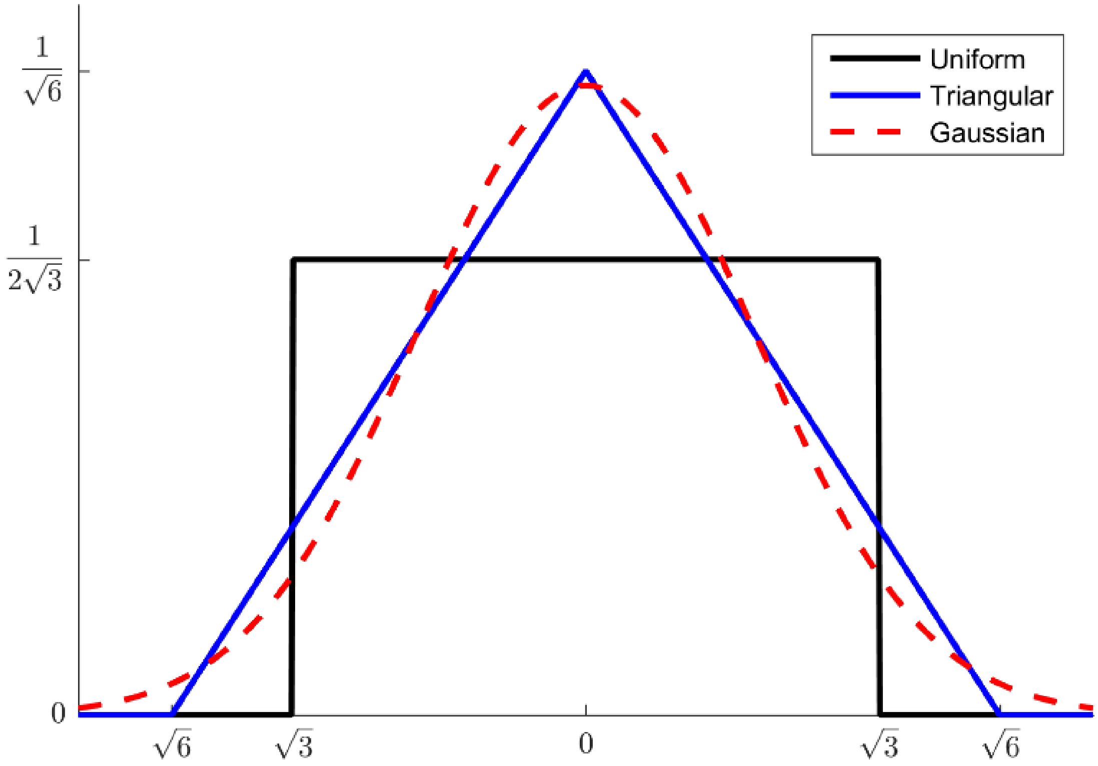


Figure 6.2: A comparison of 1-D Gaussian, uniform and triangular elliptical distributions with zero mean and unit variance. The uniform elliptical distribution models a threshold image. The triangular elliptical distribution models the distance transform image and has a mass distribution that is much more like a Gaussian distribution. Pixel replication generates data from the distance transform of the threshold image following more closely the triangular distribution and improving the fit by Gaussian mixtures.

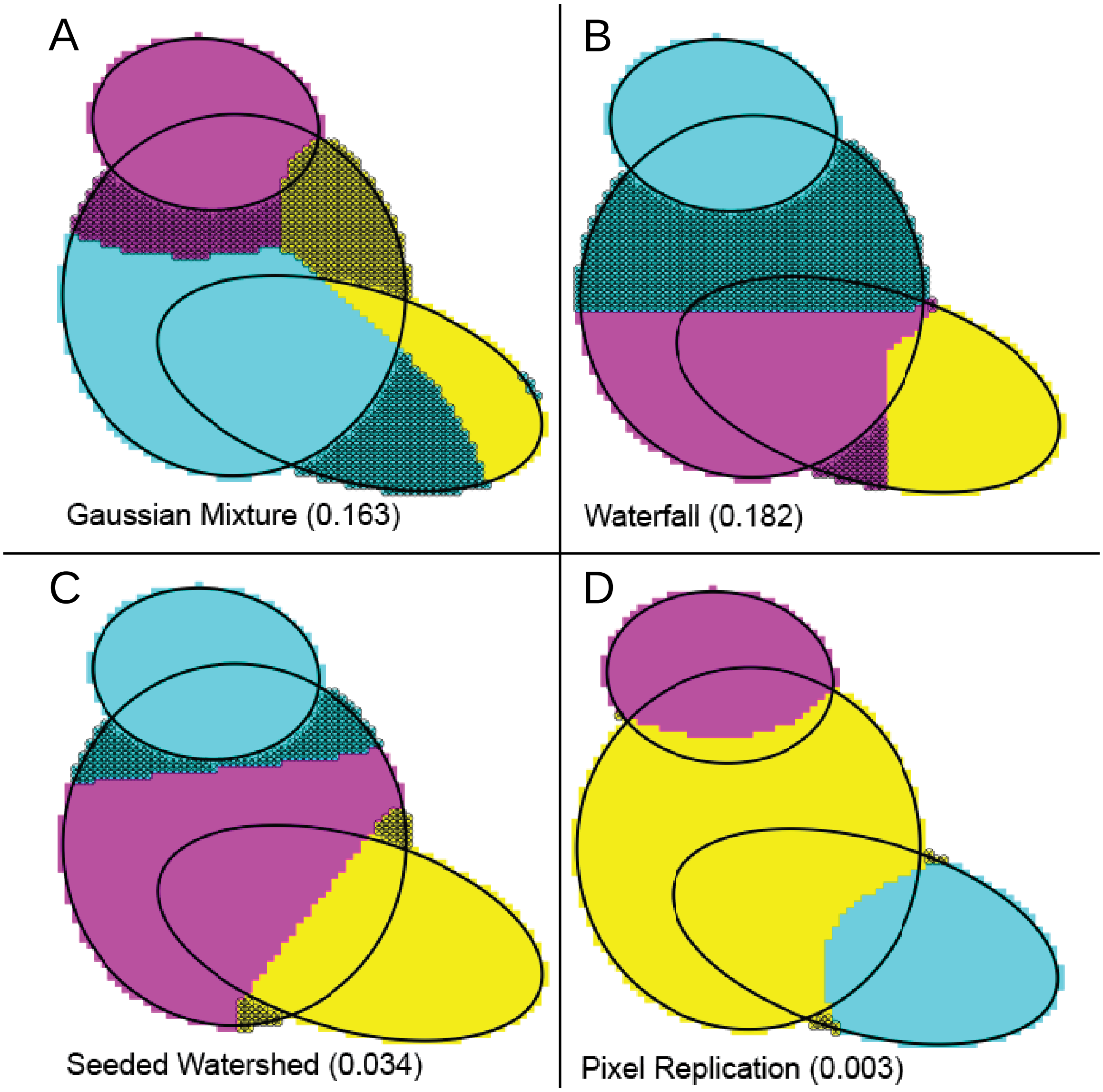
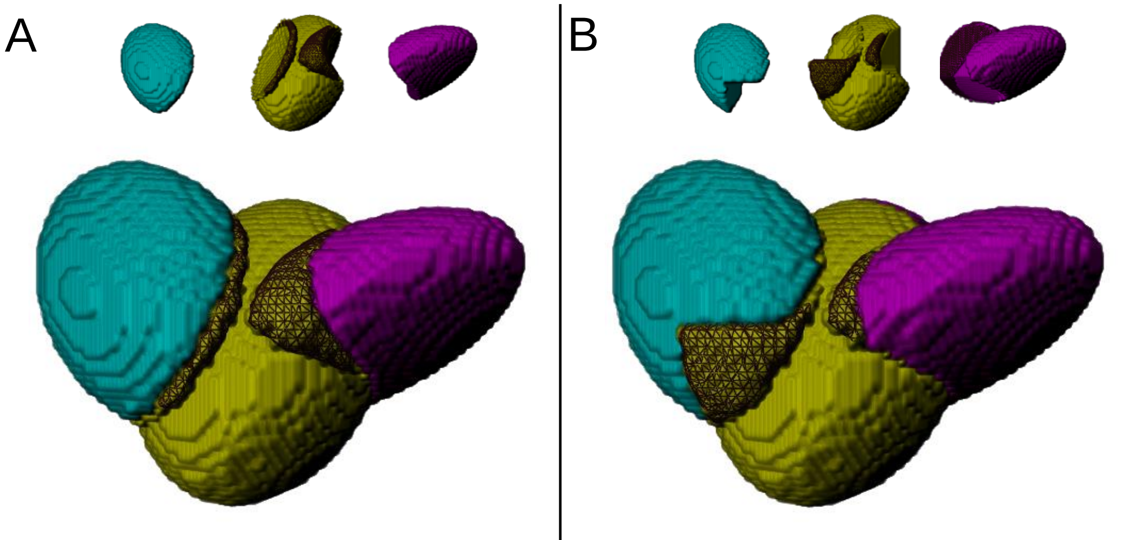
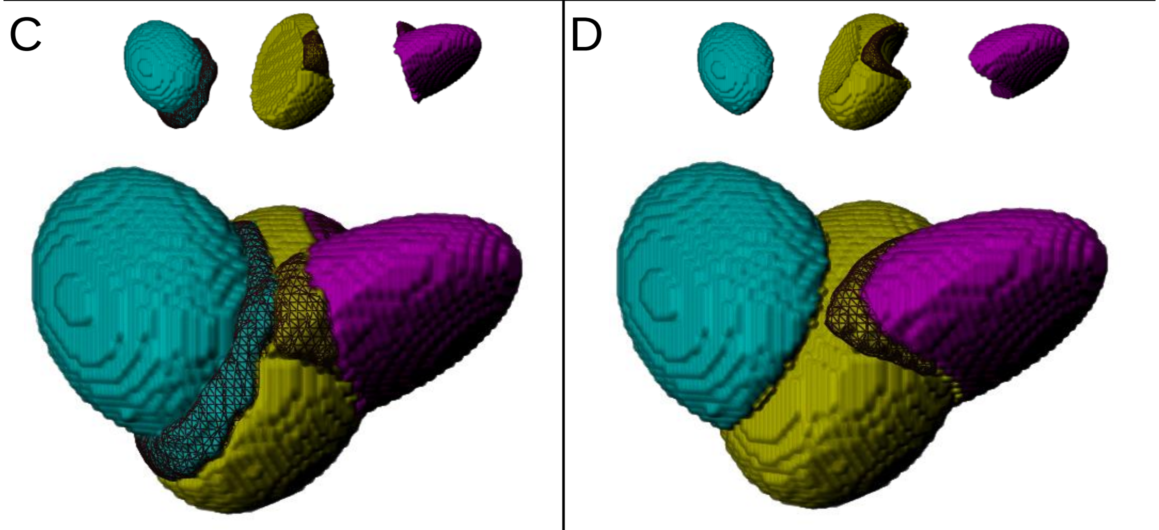


Figure 6.3: Visualizing the performance of methods for segmenting 2-D touching cells. Random configuration of three ellipses in 2-D partitioned using spatial Gaussian mixture models (A), waterfall (B), seeded watershed (C) and pixel replication (D). Partitioning assignment for each method is indicated by pixel color (magenta, cyan, yellow). Ground truth is indicated using black elliptical outlines. Incorrectly labeled pixels are marked with black crosshatching and balanced error rates for each technique are shown in parenthesis.



Gaussian Mixture (0.012)

Waterfall (0.079)



Seeded Watershed (0.032)

Pixel Replication (0.004)

Figure 6.4: Visualizing the performance of methods for segmenting 3-D touching cells. Random configuration of three ellipsoids in 3-D partitioned using spatial Gaussian mixture models (A), waterfall (B), seeded watershed (C) and pixel replication (D). Partitioning assignment for each method is indicated by voxel color (magenta, cyan, yellow). Incorrectly labeled pixels are marked with black crosshatching and balanced error rates for each technique are shown in parenthesis. Pixel replication outperforms all other methods for both 2-D and 3-D simulated data.

Figure 6.5: Simulated and edit-based error rates of algorithms for segmenting touching cells. Error rates for 1,000 each of [2, 3, 4, 5] randomly overlapping ellipses (A). Pixel replication was compared to a waterfall algorithm (the watershed transform followed by a merge strategy down to K basins), to a seeded watershed initialized with the centroids found by pixel replication, and to a Gaussian mixture fit to the spatial locations of foreground pixels in the threshold image. Pixel replication significantly outperformed other methods for pixel-level accuracy on simulated elliptical data ($p < 10^{-23}$). A neural stem cell (NSC) dataset containing over one hundred thousand touching cells and one million total segmentations was processed with each algorithm (B). Tracking errors for each algorithm were identified using manually established ground truth. Pixel replication also significantly outperformed other methods in the number of tracking errors ($p < 10^{-5}$). For both 3-D simulated and real neural stem cell data there is less overlap compared to 2-D simulated data, preserving the eccentricity of the underlying objects and improving the performance of both pixel replication and Gaussian mixture models compared to the watershed approaches. The difference in performance between pixel replication and Gaussian mixtures for the NSC dataset resulted in a reduction of 7,308 tracking errors. Whiskers represent standard error.

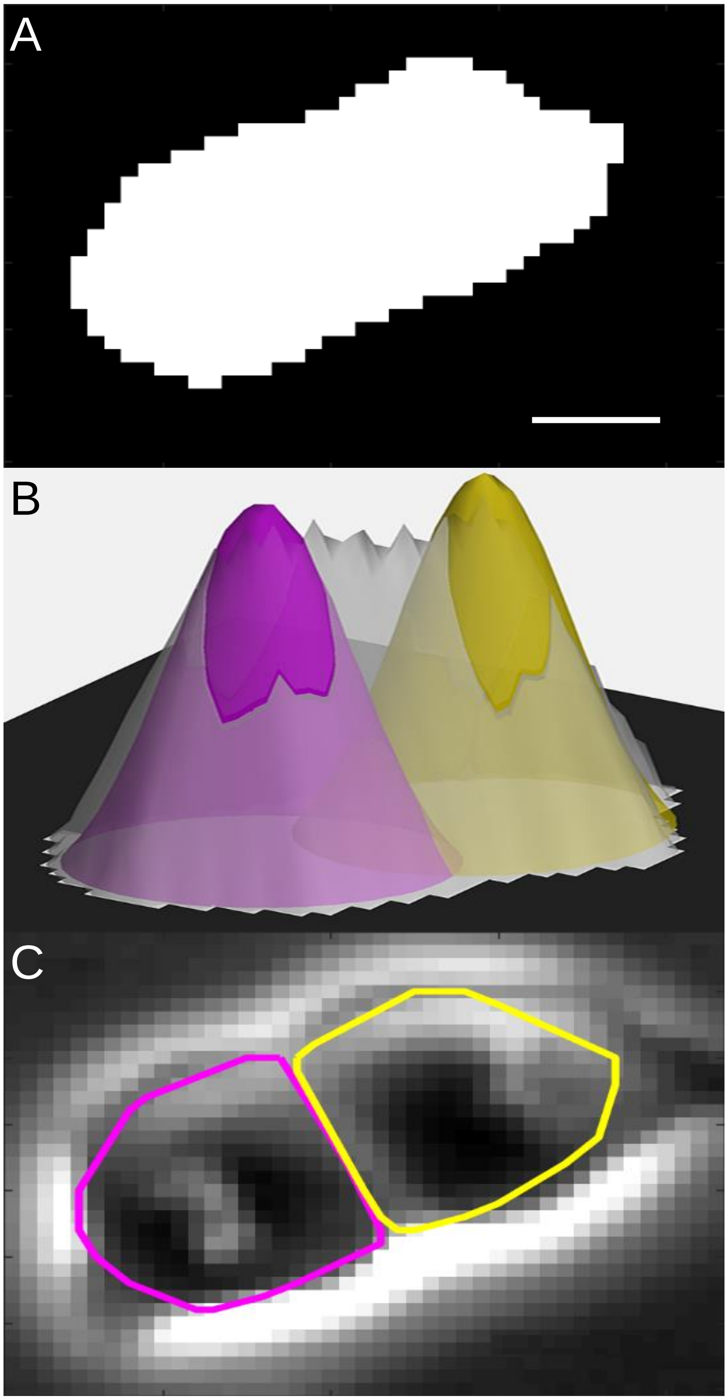


Figure 6.6: Example phase-contrast segmentation using pixel replication. Phase contrast image of mouse adult neural stem cells undergoing division; threshold image used as input to pixel replication (A), distance transform of threshold image rendered in 3-D gray overlaid with Gaussian mixture components obtained from pixel replication colored in magenta and yellow (B). Original image with two cells and segmentation overlaid in magenta and yellow (C). The scale bar represents 5 µm.

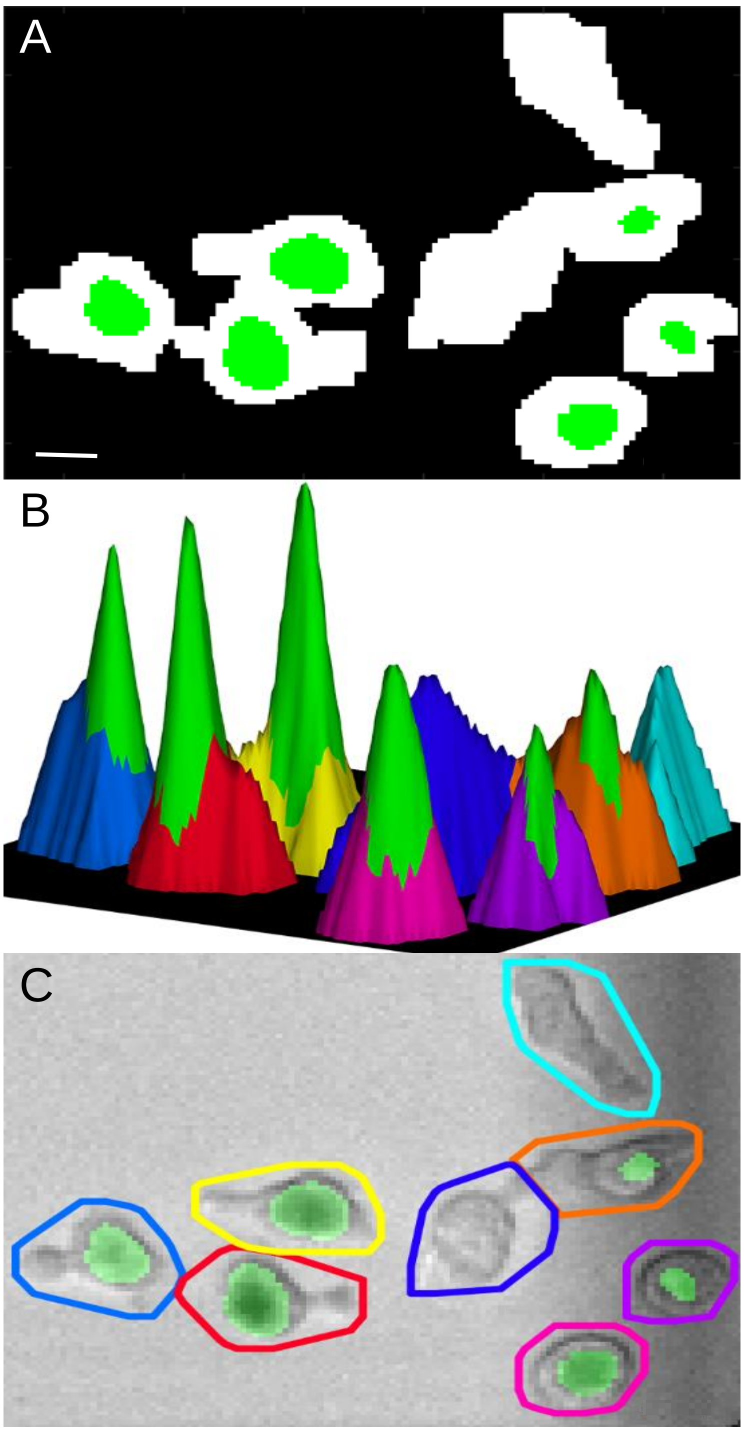


Figure 6.7: Example of combined phase and fluorescent segmentation using pixel replication. Combined fluorescence ubiquitination cell cycle indicator (FUCCI) and bright-field mouse T lymphocytes; threshold image showing bright-field threshold image (white) combined with FUCCI threshold (green) (A); distance transform in 3-D of threshold image with green (FUCCI) transform superimposed on bright-field transform colored by segmentation (B), original bright-field and FUCCI combined image with segmentation overlaid (C). The scale bar represents 5 µm.

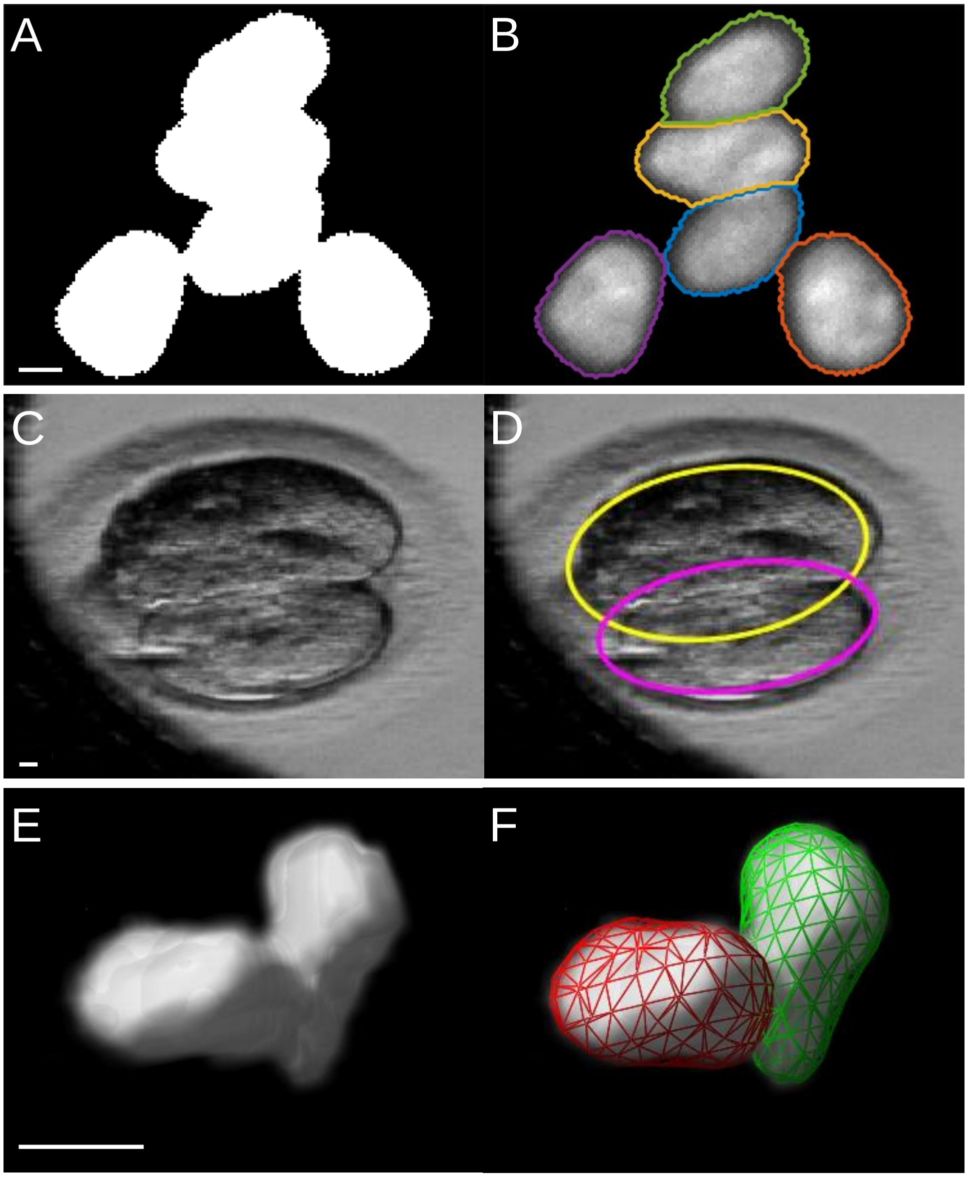


Figure 6.8: Additional example segmentation applications of pixel replication. Fluorescent human embryonic stem cells using Histone-2B nuclear markers (A-B); threshold image (A), segmentation overlaid on original image (B). Note that the segmentation was recovered purely from the morphological information in the threshold image. Human embryo images (C-D); original image (C), isocontours of Gaussian obtained from PR colored in magenta and yellow (D). A 3-D mouse embryonic neural stem cell image (E-F). Cells are labeled with a GFAP-GFP marker (E), and segmentation overlaid (F). The same pixel replication implementation was used for each cell type and imaging modality shown in this figure as well as figures 6.6 and 6.7. The scale bars represent 5 µm.

Chapter 7: Conclusion

Computational tools are becoming extremely important in biological analysis. Live-cell microscopy is the key tool for studying the dynamics of living cells. The wealth of data being produced through live-cell microscopy, particularly the combination of fluorescent and transmitted light techniques, has the potential to illuminate the fundamental elements that control cellular development. It is becoming increasingly necessary to combine state-of-the-art research in biology, imaging, and computational analysis in order to build a systematic understanding of the cell. The ability of software tools to increase the speed of biological research, as well as to provide reproducible results, will make collaboration continually more important in the coming years.

Segmentation from lineaging is a new set of computational algorithms for improving cell segmentation and tracking. Segmentation from lineage takes advantage of temporal context as well as additional sources of information about cells, such as lineage information or periodic fluorescent signals to iteratively update segmentations in every frame. User corrections are also integrated into the segmentation from lineage algorithms to allow for fast semi-automated edit-based validation of segmentation and tracking on real cell data.

The multitemporal association tracker (MAT) is a graph-based multitarget tracking algorithm for biological objects such as cells or organelles. MAT uses an optimization that focuses on the most likely tracks using a cost function that encapsulates the expected behaviors of the objects to be tracked. The tracking algorithm has been used in many biological applications including microtubule protein transport analysis, adult and embryonic neural stem cell motion analysis in 2-D and 3-D, hematopoietic stem cell tracking and cancer cell analysis. In simulated tests the multitemporal association tracker provided an improvement of 80%-90% over linear assignment trackers and it was the top entry in the microtubule transport category of the 2012 ISBI particle tracking challenge. When tested against a linear assignment tracker similar to that used in Trackmate, MAT improved edge tracking accuracy on validated neural stem cell data by about 80% across 160 cellular lineages containing more than 1 million segmentations. MAT provides temporal context for the segmentation from lineage framework, evaluating ensembles of segmentations to identify the most likely segmentation set for each frame.

Pixel replication is an alternative to the watershed transform method for partitioning clusters of touching cells. Pixel replication improves upon a spatial Gaussian mixture fit by integrating a Euclidean distance transform to produce a mass distribution from a threshold image that is similar to a Gaussian mixture model. Simulated results show that pixel replication is more accurate than two common watershed techniques as well as direct Gaussian mixtures. Pixel replication has also been shown to improve tracking accuracy compared to watershed and Gaussian mixture techniques, significantly reducing the number of tracking corrections that would be required to fully validate neural stem cell data. Segmentation from lineaging uses pixel replication along with multitemporal association tracking to evaluate and automatically partition touching cells when necessary.

The iterative nature of segmentation from lineage, which turns the usual segmentation, tracking and lineaging analysis pipeline back on itself, produces significant improvements in segmentation and tracking error rates. Segmentation from lineaging builds on the temporal information provided by multitemporal association tracking to evaluate multiple segmentation ensembles in each frame. The underlying segmentation algorithms are used to search for missing cell detections and pixel replication is used to partition cell clusters that may have been initially under-segmented. Together these tools form a powerful framework for understanding proliferating cell dynamics.

The segmentation from lineage algorithms are integrated into the extensible, open source LEVER lineage editing and validation software[4;5]. LEVER provides a simple user interface for automated segmentation, tracking and lineaging, as well as user editing and validation of the results. Segmentation from lineage facilitates the extraction of fully validated and corrected segmentation, tracking and lineaging results from the image data with a minimum of human effort. The segmentation from lineaging algorithms have proved to be a versatile set of tools revealing previously unknown intrinsic differences in the patterns of clonal development and cellular behavior between anterior and posterior cerebral cortical neural progenitor cells. These algorithms have helped identify intriguing variations in cellular motion and morphology, as well as clonal development across age in mouse neural stem cells. An additional cell process segmentation, guided by the refinement algorithm, also provides insight into the behavioral differences between type A and type C neural stem cells. These results are the first steps in understanding the the changes that occur in brain development as we age, and may hold clues to reinvigorating the sub-ventricular niche.

Research on segmentation, tracking and lineaging is ongoing. Segmentation from lineage significantly improves the accuracy of automated analysis, however it currently requires a corrected lineage tree. We are actively researching image-based automated mitosis and cell death detection algorithms for both fluorescent and phase contrast imaging similar to Huh *et al.*[69]. Automated mitosis and death detection could provide a solution for fully automating the segmentation from lineaging interface. Additional information from fluorescent markers such a FUCCI can also be integrated into the segmentation from lineaging framework to increase accuracy while reducing the need for human interaction.

The elliptical triangular distribution model underlying pixel replication is exact only for circular objects. In the case of eccentric ellipses, the elliptical triangular distribution only approximates the mass distribution created by the euclidean distance transform. This causes an overestimation of the principal radii in highly eccentric ellipses. Additional investigation is required to accurately predict the modeling deviation.

Finally, segmentation from lineage achieves best results when a user is able to identify and correct any remaining tracking errors as the refinement iteration is run. Integrating new evaluation criteria for cell segmentation and tracking will allow the refinement algorithm to better warn a user of possible errors and draw their attention to the associated region. Identifying regions of uncertainty for users to correct later will also improve algorithmic throughput, making it possible to analyze larger experimental data sets. This information could also be fed into machine learning tools to improve automated analysis on similar data in the future.

Bibliography

[1] Asako Sakaue-Sawano, Hiroshi Kurokawa, Toshifumi Morimura, Aki Hanyu, Hiroshi Hama, Hatsuki Osawa, Saori Kashiwagi, Kiyoko Fukami, Takaki Miyata, Hiroyuki Miyoshi, Takeshi Imamura, Masaharu Ogawa, Hisao Masai, and Atsushi Miyawaki. Visualizing spatiotemporal dynamics of multicellular cell-cycle progression. *Cell*, 132(3):487–498, 2008. ISSN 0092-8674.

[2] A. R. Cohen, C. S. Bjornsson, S. Temple, G. Banker, and B. Roysam. Automatic summarization of changes in biological image sequences using algorithmic information theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1386–1403, 2009. ISSN 0162-8828.

[3] Mark R. Winter, Cheng Fang, Gary Banker, Badrinath Roysam, and Andrew R. Cohen. Axonal transport analysis using multitemporal association tracking. *International Journal of Computational Biology and Drug Design*, 5(1):35–48, 2012.

[4] Mark Winter, Eric Wait, Badrinath Roysam, Susan K. Goderie, Rania Ahmed Naguib Ali, Erzsebet Kokovay, Sally Temple, and Andrew R. Cohen. Vertebrate neural stem cell segmentation, tracking and lineaging with validation and editing. *Nat. Protocols*, 6(12):1942–1952, 2011. ISSN 1754-2189.

[5] M. Winter, W. Mankowski, E. Wait, S. Temple, and A. R. Cohen. Lever: software tools for segmentation, tracking and lineaging of proliferating cells. *Bioinformatics*, 2016.

[6] M. R. Winter, M. Liu, D. Monteleone, J. Melunis, U. Hershberg, S. K. Goderie, S. Temple, and A. R. Cohen. Computational image analysis reveals intrinsic multigenerational differences between anterior and posterior cerebral cortex neural progenitor cells. *Stem Cell Reports*, 5(4): 609–20, 2015.

[7] Mark Winter, Walt Mankowski, Eric Wait, Edgar Cardenas De La Hoz, Angeline Aguinaldo, Andrey Kan, Susanne Heinzel, Philip D. Hodgkin, Simone C. Oostindie, Francisco Gomes, Faraz Janan, Rafael E. Carazo Salas, Sally Temple, and Andrew R. Cohen. Separating touching cells using pixel replicated elliptical shape models. *IEEE Transactions on Image Processing*, In Review.

[8] M. Apostolopoulou, T. R. Kiehl, M. Winter, E.C. De La Hoz, S. K. Goderie, Y. Wang, Cohen A. R., and S. Temple. Occurrence of peaks and valleys in cell behavior and gene expression during aging of the mouse subventricular zone neural stem cell niche. *Neuron*, In Revision, 2016.

[9] Edgar Cardenas De La Hoz, Mark R. Winter, Maria Apostolopoulou, Sally Temple, and Andrew R. Cohen. Measuring process dynamics and nuclear migration for clones of neural progenitor cells. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part I*, pages 291–305, Cham, 2016. Springer International Publishing.

[10] Eric Wait, Mark Winter, Chris Bjornsson, Erzsebet Kokovay, Yue Wang, Susan Goderie, Sally Temple, and Andrew Cohen. Visualization and correction of automated segmentation, tracking and lineaging from 5-d stem cell image sequences. *BMC Bioinformatics*, 15(1):328, 2014.

[11] W. C. Mankowski, M. R. Winter, E. Wait, M. Lodder, T. Schumacher, S. H. Naik, and A. R. Cohen. Segmentation of occluded hematopoietic stem cells from tracking. *Conf Proc IEEE Eng Med Biol Soc*, 2014:5510–3, 2014.

[12] N. Chenouard, I. Smal, F. de Chaumont, M. Maska, I. F. Sbalzarini, Y. Gong, J. Cardinale, C. Carthel, S. Coraluppi, M. Winter, A. R. Cohen, W. J. Godinez, K. Rohr, Y. Kalaidzidis, L. Liang, J. Duncan, H. Shen, Y. Xu, K. E. Magnusson, J. Jalden, H. M. Blau, P. Paul-Gilloteaux, P. Roudot, C. Kervrann, F. Waharte, J. Y. Tinevez, S. L. Shorte, J. Willemse, K. Celler, G. P. van Wezel, H. W. Dan, Y. S. Tsai, C. O. de Solorzano, J. C. Olivo-Marin, and E. Meijering. Objective comparison of particle tracking methods. *Nat Methods*, 2014.

[13] Douglas B. Murphy and Michael W. Davidson. *Fundamentals of Light Microscopy and Electronic Imaging.* Wiley-Liss, New York, N.Y., 2012.

[14] B. C. Chen, W. R. Legant, K. Wang, L. Shao, D. E. Milkie, M. W. Davidson, C. Janetopoulos, X. S. Wu, 3rd Hammer, J. A., Z. Liu, B. P. English, Y. Mimori-Kiyosue, D. P. Romero, A. T. Ritter, J. Lippincott-Schwartz, L. Fritz-Laylin, R. D. Mullins, D. M. Mitchell, J. N. Bembenek, A. C. Reymann, R. Bohme, S. W. Grill, J. T. Wang, G. Seydoux, U. S. Tulu, D. P. Kiehart, and E. Betzig. Lattice light-sheet microscopy: imaging molecules to embryos at high spatiotemporal resolution. *Science*, 346(6208):1257998, 2014.

[15] F. Zernike. Phase contrast, a new method for the microscopic observation of transparent objects. *Physica*, 9(7):686 – 698, 1942.

[16] F. Zernike. Phase contrast, a new method for the microscopic observation of transparent objects part ii. *Physica*, 9(10):974 – 986, 1942.

[17] R. Y. Tsien. The green fluorescent protein. *Annu Rev Biochem*, 67:509–44, 1998.

[18] Lee J. Bain and Max Engelhardt. *Introduction to probability and mathematical statistics.* The Duxbury advanced series in statistics and decision sciences. PWS-KENT Pub., Boston, 2nd edition, 1992.

[19] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern recognition.* Academic Press, San Diego, CA, 4th edition, 2009.

[20] Eusebio Gomez, Miguel A. Gomez-Villegas, and J. Miguel Marn. A survey on continuous elliptical vector distributions. *Revista Matematica Complutense*, 16(1):345–361, 2003.

[21] Christopher M. Bishop. *Pattern recognition and machine learning.* Information science and statistics. Springer, New York, 2006.

[22] F. L. Gomes, G. Zhang, F. Carbonell, J. A. Correa, W. A. Harris, B. D. Simons, and M. Cayouette. Reconstruction of rat retinal progenitor cell lineages in vitro reveals a surprising degree of stochasticity in cell fate decisions. *Development*, 138(2):227–35, 2011.

[23] K. R. Duffy, C. J. Wellard, J. F. Markham, J. H. Zhou, R. Holmberg, E. D. Hawkins, J. Hasbold, M. R. Dowling, and P. D. Hodgkin. Activation-induced b cell fates are selected by intracellular stochastic competition. *Science*, 335(6066):338–41, 2012.

[24] B. Giebel and M. Punzel. Lineage development of hematopoietic stem and progenitor cells. *Biol Chem*, 389(7):813–24, 2008.

[25] A. R. Cohen, F. L. Gomes, B. Roysam, and M. Cayouette. Computational prediction of neural progenitor cell fates. *Nat Methods*, 7(3):213–8, 2010.

[26] P. J. Murray and T. A. Wynn. Protective and pathogenic functions of macrophage subsets. *Nat Rev Immunol*, 11(11):723–37, 2011.

[27] H. M. Eilken, S. Nishikawa, and T. Schroeder. Continuous single-cell imaging of blood generation from haemogenic endothelium. *Nature*, 457(7231):896–900, 2009.

[28] T. Schroeder. Long-term single-cell imaging of mammalian stem cells. *Nat Methods*, 8(4 Suppl): S30–5, 2011.

[29] Kang Li, Eric D. Miller, Mei Chen, Takeo Kanade, Lee E. Weiss, and Phil G. Campbell. Cell population tracking and lineage construction with spatiotemporal context. *Medical Image Analysis*, 12(008):546–566, 2008.

[30] Bernhard X. Kausler, Martin Schiegg, Bjoern Andres, Martin Lindner, Ullrich Koethe, Heike Leitte, Jochen Wittbrodt, Lars Hufnagel, and Fred A. Hamprecht. A discrete chain graph model for 3d+t cell tracking with high misdetection robustness. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia" Schmid, editors, *Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III*, pages 144–157, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[31] Romain Fernandez, Pradeep Das, Vincent Mirabet, Eric Moscardi, Jan Traas, Jean-Luc Verdeil, Grgoire Malandain, and Christophe Godin. Imaging plant growth in 4d: robust tissue reconstruction and lineaging at cell resolution. *Nature Methods*, 7(7):547 – 553, 2010.

[32] Omar Al-Kofahi, Richard J. Radke, Susan K. Goderie, Qin Shen, Sally Temple, and Badrinath Roysam. Automated cell lineage tracing: a high-throughput method to analyze cell proliferative behavior developed using mouse neural stem cells. *Cell Cycle*, 5(3):327 – 335, 2006.

[33] Fabrice de Chaumont, Stephane Dallongeville, Nicolas Chenouard, Nicolas Herve, Sorin Pop, Thomas Provoost, Vannary Meas-Yedid, Praveen Pankajakshan, Timothee Lecomte, Yoann Le Montagner, Thibault Lagache, Alexandre Dufour, and Jean-Christophe Olivo-Marin. Icy: an open bioimage informatics platform for extended reproducible research. *Nat Meth*, 9(7): 690–696, Jul 2012.

[34] Mark-Anthony Bray and Anne E. Carpenter. Cellprofiler tracer: exploring and validating high-throughput, time-lapse microscopy image data. *BMC Bioinformatics*, 16(1):369, 2015.

[35] Stella Stylianidou, Connor Brennan, Silas B. Nissen, Nathan J. Kuwada, and Paul A. Wiggins. Supersegger: robust image segmentation, analysis and lineage tracking of bacterial cells. *Molecular Microbiology*, 102(4):690–700, 2016.

[36] Joe Chalfoun, Michael Majurski, Alden Dima, Michael Halter, Kiran Bhadriraju, and Mary Brady. Lineage mapper: A versatile cell and particle tracker. *Scientific Reports*, 6, Nov 2016.

[37] E. Meijering, I. Smal, and G. Danuser. Tracking in molecular bioimaging. *Ieee Signal Processing Magazine*, 23(3):46–53, 2006.

[38] Dirk Padfield, Jens Rittscher, and Badrinath Roysam. Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. *Medical Image Analysis*, 15(4), 2011.

[39] D. Padfield, J. Rittscher, N. Thomas, and B. Roysam. Spatio-temporal cell cycle phase analysis using level sets and fast marching methods. *Medical Image Analysis*, 13(1):143–155, 2009.

[40] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142–149 vol.2, 2000.

[41] Samuel S. Blackman and Robert Popoli. *Design and analysis of modern tracking systems*. Artech House radar library. Artech House, Boston, 1999.

[42] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.

[43] A. M. Frieze and J. Yadegar. An algorithm for solving 3-dimensional assignment problems with application to scheduling a teaching practice. *Journal of the Operational Research Society*, 32 (11):989–995, 1981.

[44] A. B. Poore and S. Gadaleta. Some assignment problems arising from multiple target tracking. *Mathematical and Computer Modelling*, 43(9-10):1074–1091, 2006.

[45] A. J. Robertson. A set of greedy randomized adaptive local search procedure (grasp) implementations for the multidimensional assignment problem. *Computational Optimization and Applications*, 19(2):145–164, 2001.

[46] Q. Yu and G. Medioni. Multiple-target tracking by spatiotemporal monte carlo markov chain data association. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12): 2196–210, 2009.

[47] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, and Benjamin Schmid. Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7):676–682, 2012.

[48] J. Schindelin, C. T. Rueden, M. C. Hiner, and K. W. Eliceiri. The imagej ecosystem: An open platform for biomedical image analysis. *Mol Reprod Dev*, 82(7-8):518–29, 2015.

[49] K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S. L. Schmid, and G. Danuser. Robust single-particle tracking in live-cell time-lapse sequences. *Nat Methods*, 5(8):695–702, 2008.

[50] L. Vincent and P. Soille. Watersheds in digital spaces - an efficient algorithm based on immersion simulations. *Ieee Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.

[51] J. C. Neves, H. Castro, A. Tomas, M. Coimbra, and H. Proenca. Detection and separation of overlapping cells based on contour concavity for leishmania images. *Cytometry A*, 85(6): 491–500, 2014.

[52] G. Lin, U. Adiga, K. Olson, J. F. Guzowski, C. A. Barnes, and B. Roysam. A hybrid 3d watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks. *Cytometry A*, 56(1):23–36, 2003.

[53] Shenghua Cheng, Tingwei Quan, Xiaomao Liu, and Shaoqun Zeng. Large-scale localization of touching somas from 3d images using density-peak clustering. *BMC Bioinformatics*, 17(1): 1–12, 2016.

[54] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492, 2014.

[55] Y. B. Li, F. Rose, F. Di Pietro, X. Morin, and A. Genovesio. Detection and tracking of overlapping cell nuclei for large scale mitosis analyses. *Bmc Bioinformatics*, 17, 2016.

[56] Fernando Amat, William Lemon, Daniel P. Mossing, Katie McDole, Yinan Wan, Kristin Branson, Eugene W. Myers, and Philipp J. Keller. Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data. *Nat Meth*, 11(9):951–958, 2014.

[57] Melissa Linkert, Curtis T. Rueden, Chris Allan, Jean-Marie Burel, Will Moore, Andrew Patterson, Brian Loranger, Josh Moore, Carlos Neves, Donald MacDonald, Aleksandra Tarkowska, Caitlin Sticco, Emma Hill, Mike Rossner, Kevin W. Eliceiri, and Jason R. Swedlow. Metadata matters: access to image data in the real world. *The Journal of Cell Biology*, 189(5):777–782, 2010.

[58] D. D. O'Leary, S. J. Chou, and S. Sahara. Area patterning of the mammalian cortex. *Neuron*, 56(2):252–69, 2007.

[59] Sally Temple. The development of neural stem cells. *Nature*, 414(6859):112–117, Nov 2001.

[60] N. Otsu. A threshold selection method from gray-level histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(1):62–66, 1979.

[61] Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins. *Digital Image Processing Using MATLAB.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003.

[62] Judea Pearl. *Probabilistic reasoning in intelligent systems : networks of plausible inference.* The Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann Publishers, San Mateo, Calif., 1988.

[63] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization : algorithms and complexity.* Dover Publications, Mineola, N.Y., 1998.

[64] Serge Beucher. *Watershed, Hierarchical Segmentation and Waterfall Algorithm*, volume 2 of *Computational Imaging and Vision*, book section 10, pages 69–76. Springer Netherlands, 1994.

[65] Robert McGill, John W. Tukey, and Wayne A. Larsen. Variations of box plots. *The American Statistician*, 32(1):12–16, 1978.

[66] Howard M. Shapiro. *Practical flow cytometry.* Wiley-Liss, New York, 4th edition, 2003.

[67] A. L. Koch. The logarithm in biology. 1. mechanisms generating the log-normal distribution exactly. *J Theor Biol*, 12(2):276–90, 1966.

[68] A. L. Koch. The logarithm in biology. ii. distributions simulating the log-normal. *J Theor Biol*, 23(2):251–68, 1969.

[69] S. Huh, D. F. E. Ker, R. Bise, M. Chen, and T. Kanade. Automated mitosis detection of stem cell populations in phase-contrast microscopy images. *IEEE Trans Med Imaging*, 30(3): 586–596, 2011.

[70] M. Saerens, P. Latinne, and C. Decaestecker. Any reasonable cost function can be used for a posteriori probability approximation. *IEEE Transactions on Neural Networks*, 13(5):1204–1210, 2002.

[71] L. Chen, M. J. Wainwright, M. Cetin, and A. S. Willsky. Data association based on optimization in graphical models with application to sensor networks. *Mathematical and Computer Modelling*, 43(9-10):1114–1135, 2006.

[72] Andrew R. Cohen. Extracting meaning from biological imaging data. *Molecular Biology of the Cell*, 25(22):3470–3473, 2014.

[73] P. Bajcsy, A. Cardone, J. Chalfoun, M. Halter, D. Juba, M. Kociolek, M. Majurski, A. Peskin, C. Simon, M. Simon, A. Vandecreme, and M. Brady. Survey statistics of automated segmentations applied to optical imaging of mammalian cells. *BMC Bioinformatics*, 16:330, 2015.

[74] M. Schiegg, P. Hanslovsky, C. Haubold, U. Koethe, L. Hufnagel, and F. A. Hamprecht. Graphical model for joint segmentation and tracking of multiple dividing cells. *Bioinformatics*, 31(6): 948–56, 2015.

[75] Sheldon M. Ross. *Introduction to probability models.* Academic Press, Amsterdam ; Boston, 10th edition, 2010.

[76] X. D. Yang, H. Q. Li, and X. B. Zhou. Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and kalman filter in time-lapse microscopy. *Ieee Transactions on Circuits and Systems I-Regular Papers*, 53(11):2405–2414, 2006.

[77] Erzsebet Kokovay, Yue Wang, Gretchen Kusek, Rachel Wurster, Patty Lederman, Natalia Lowry, Qin Shen, and Sally Temple. Vcam1 is essential to maintain the structure of the svz niche and acts as an environmental sensor to regulate svz lineage progression. *Cell Stem Cell*, 11(2):220–230, 2012.

[78] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, P. Golland, and D. M. Sabatini. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol*, 7(10): R100, 2006.

Appendix A: Biological methods

A.1 Embryonic murine anterior and posterior cell culture procedure

Anterior and posterior regions of E12.5 cerebral cortex are dissected, with the intervening mid-region of approximately 75-100 microns removed and discarded. The anterior and posterior tissue is dissociated enzymatically using 10 units/ml papain (Worthington), and then gently triturated to produce single cells. Each well of a 24 well plate (Corning/Costar) previously coated with poly-l-lysine, was seeded with 5,000 single embryonic cortical cells, in DMEM (Invitrogen), N2(Invitrogen), B27(Invitrogen) and 10ng/ml bFGF(Invitrogen). Immediately after seeding, plates are placed into the time-lapse system. The system is based on a Zeiss Axio-Observer Inverted Z1 microscope equipped with a motorized XYZ stage for imaging multiple points. Imaging 9 fields per well with 3 wells per condition typically gives 5-10 clones per field using a 10x neofluar objective. The microscope is fitted with a Pecon incubation chamber with controlled temperature at 37 °C, 98% humidity and 5% CO2. Images are captured every five minutes with a Hamatsua Orca high resolution black and white digital camera for up to five days.

A.2 Aging murine comparison cell culture procedure

A.2.1 Animals

Animals used (Charles River Laboratories, Wilmington, MA, USA) were male wild-type a/a C57BL/6 mice, 2, 6, 18 and 22 months old, weighing: 24 g+/-1g, 30 g+/-0g, 36 g+/-1g, and 36 g+/-1g respectively and were ordered as close as possible to the intended age of use.

All animals were kept at a Specific Pathogen Free animal facility at SUNY Albany. Mice were housed in Individually Ventilated Cages on Sani-Chip® bedding and had a maximum of 4 cage companions. Mice were kept in a 12 hour light/12 hour dark cycle at 72°F. Animals had free access to Prolab IsoPro RMH 3000 Irradiated food pellets and water and were not subject to any kind of environmental enrichment. Welfare assessments followed the recommendations of the Guide for the

Care and Use of Laboratory Animals, Eighth Edition.

Animals were sacrificed by isoflurane inhalation, and their SVZs dissected and dissociated to single cells as previously described[77].

A.2.2 In vivo cell cycle comparisons and immunostaining of sections

CldU+IdU+ double labeling: Three mice of each age category (2, 18 and 22 month old) were injected intra-peritoneally with 50 mg/kg CldU (MP Biomedicals, Santa Ana, CA, USA) three times, 2 hours apart. Nineteen hours after each injection, animals were injected with 50 mg/kg IdU (Sigma-Aldrich, St. Louis, MO, USA). 24 hrs after the last injection animals were transcardially perfused with a 4% formaldehyde solution (Toussimis, Rockville, MD, USA), while under the effects of 0.15cc of a 390 mg/mL solution of Beuthanasia-D.

Brains where then removed and immersed in a 30% sucrose (Sigma-Aldrich) solution then in O.C.T. compound (Fisher Scientific) and frozen. Coronal sections 40 µm thick of the SVZ were cut from approximately 1.94 mm to −0.1 mm in relation to bregma. Every second or third section from a single animal was used for each replicate (6-9 sections/replicate). Sections underwent antigen retrieval by being boiled with 10 mM citrate buffer (pH 6; reagents from Sigma-Aldrich) for 15 minutes, then incubated with 2 M HCl (Sigma-Aldrich) for 30 minutes at room temperature (RT), and subsequently washed twice with 0.1 M borate buffer (pH 8.5; reagents from J.T. Baker, Phillipsburg, NJ, USA) for 5 minutes at RT. Sections were permeabilized with 0.5% Triton-X (Sigma-Aldrich) for 1 hour at RT and then blocked with 10% donkey serum (Jackson Immunoresearch, West Grove, PA, USA) in permeabilization buffer for 1 hour at RT. Sections were stained sequentially with anti-CldU (1:50; overnight at 4oC; Novus Biologicals, Littleton, CO, USA), and then anti-IdU (1:250; 2 hours at RT; BD, Franklin Lakes, NJ, USA), and DAPI (1:1000; for 10 minutes at RT; Invitrogen, Carlsbad, CA, USA).

Secondary antibodies used were Alexa 568 and 488 (Invitrogen) and were applied at 1:300 overnight at 4 °C. Images of the entire SVZ were taken for both SVZs of each section with a Zeiss LSM 780 NLO multiphoton microscope. All nuclei surrounding the SVZ that were CldU+ alone were counted using Cell Profiler, while CldU+IdU+ were counted manually[78].

For other experiments, 3-6 mice of each age category were injected intra-peritoneally with 50 mg/kg EdU (Invitrogen) and sacrificed at 19 hours as described above. Coronal brain sections 18 or 40 µg thick were cut (5-9 sections/replicate) and blocked as already described and stained with anti-Dcx (1:300; overnight at 4 °C; Santa Cruz, Dallas, Texas, USA) or Mash1 (a generous gift from Dr. Jane Johnson), the EdU kit (Invitrogen) according to the manufacturers specifications, anti-Ki67 (1:200; over two nights at 4 °C; Thermo Fisher Scientific) and DAPI (as mentioned above). Images were taken as described above. Dcx+ area surrounding the entire SVZ (both SVZs of each section) was counted using Cell Profiler[78]. Mash1+ cells, and cells double- or triple-stained for Ki67, Dcx or Mash1, and EdU surrounding the entire SVZ (both SVZs of each section) were counted manually.

In all cases, counts were blinded and means were generated for each mouse brain first and then compared between the animals of each age group.

All sections used in the same experiment (regardless of age) were stained contemporaneously and imaged using the same microscope settings. RGB levels on presented microscopy images were adjusted as appropriate for visual clarity.

A.2.3 Cell Culture Methods

Growth medium was DMEM (Thermo Fisher Scientific), supplemented with 1 mM Na-pyruvate (Thermo Fisher Scientific), 2 mM L-glutamine (Thermo Fisher Scientific), 0.16 µg/mL N-Acetyl-L-cysteine (Sigma-Aldrich), N-2 supplement (Thermo Fisher Scientific), B-27 supplement (Thermo Fisher Scientific), 0.01 µg/mL FGF2 (Thermo Fisher Scientific), 0.01 µg/mL BDNF (R&D Systems, Minneapolis, MN, USA), 0.05 µg/mL NRG1 (R&D Systems), 0.1 mM QVD (Calbiochem, San Diego, CA, USA) and 0.01 mM LY (Stemgent, Cambridge, MA, USA). Cells were allowed to settle for 2 hours before media was changed to remove tissue debris. Cells were re-fed once at day 2 after isolation and experiments were ended at 4 DIV.

A.2.4 Cell culture immunostaining

Cells were fixed with 5% paraformaldehyde (Polysciences, Warrington, MA, USA) in PHEM buffer pH 6.9 (120 mM PIPES, 50 mM HEPES, 20 mM EGTA and 8 mM MgCl2.6H2O (Sigma-Aldrich) for 20 mins at RT. Cells were then stained sequentially for O4 (for 45 min at RT) and EGFR (overnight at 4 °C). Cells were then permeabilized with 0.1% Triton-X (Sigma-Aldrich) and sequentially stained for GFAP (for 45-60 mins at RT), after which the first set of images was taken. Cells were then stained for β-tubulin III (for 45-60 mins at RT) and DAPI (for 5 mins at RT; Invitrogen) and the final set of images was taken. Primary antibodies used were an O4 hybridoma produced in the Temple lab, anti-EGFR (at 1:200; Thermo Fisher Scientific), anti-GFAP (at 1:400; EMD Millipore, Billerica, MA, USA) and anti-β-tubulin III (at 1:500; Sigma-Aldrich). Secondary antibodies used were Alexa 647, 546 and 488 (Invitrogen). Cells were stained with secondary antibodies for 20-30 mins at RT. Images were taken using a Zeiss Axiovert D1 microscope. RGB levels on presented microscopy images were adjusted as appropriate for visual clarity.

A.2.5 Time-lapse microscopy of isolated SVZ cells

Plates were imaged using a Zeiss Axiovert Z1 microscope with an environmental chamber at 37 °C and 5% CO2. For each replicate, nine fields of vision with cells were chosen randomly from 4 wells/age group, totaling 36 fields/age group/replicate. Fields were imaged every 5 min then cells were fixed and immunostained as described above. 3-4 experimental replicates were performed for each age group.

Vita

Mark Winter is currently a Ph.D. candidate in Electrical Engineering at Drexel University. He received his Master's of Science in Electrical Engineering from the University of Wisconsin - Milwaukee in 2011. He received a Bachelor's of Science in Electrical & Computer Engineering and a Bachelor's of Science in Applied Mathematics from the University of Colorado in 2005. Mark has worked as a computer scientist for a decade, beginning as a 3-D game developer at America's Army. He transitioned to algorithms research, writing image segmentation and reconstruction algorithms for IMTEC, a manufacturer of industrial and dental computed tomography scanners, before returning to graduate school at UW-Milwaukee. Mark's work is currently focused on integrating human validation, expected cellular behaviors, fluorescent signals, etc. to improve automated cell identification and tracking algorithms at Drexel University.

Publications

E. C. De La Hoz, M. R. Winter, M. Apostolopoulou, S. Temple, and A. R. Cohen, "Measuring Process Dynamics and Nuclear Migration for Clones of Neural Progenitor Cells," *ECCV*, vol. 9913, 2016.

M. Winter, W. Mankowski, E. Wait, S. Temple, and A. R. Cohen, "LEVER: software tools for segmentation, tracking and lineaging of proliferating cells," *Bioinformatics*, 2016.

M. R. Winter, M. Liu, D. Monteleone, J. Melunis, U. Hershberg, S. K. Goderie, et al., "Computational Image Analysis Reveals Intrinsic Multigenerational Differences between Anterior and Posterior Cerebral Cortex Neural Progenitor Cells," *Stem Cell Reports*, vol. 5, 2015.

E. Wait, M. Winter, C. Bjornsson, E. Kokovay, Y. Wang, S. Goderie, et al., "Visualization and Correction of Automated Segmentation, Tracking and Lineaging from 5-D Stem Cell Image Sequences," *BMC Bioinformatics*, vol. 15, 2014.

W. C. Mankowski, M. R. Winter, E. Wait, M. Lodder, T. Schumacher, S. H. Naik, et al., "Segmentation of occluded hematopoietic stem cells from tracking," *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, 2014.

N. Chenouard, I. Smal, F. de Chaumont, M. Maska, I. F. Sbalzarini, Y. Gong, et al., "Objective comparison of particle tracking methods," *Nature Methods*, 2014.

M. R. Winter, C. Fang, G. Banker, B. Roysam, and A. R. Cohen, "Axonal transport analysis using Multitemporal Association Tracking," *International Journal of Computational Biology and Drug Design*, vol. 5, 2012.

M. Winter, E. Wait, B. Roysam, S. K. Goderie, R. A. N. Ali, E. Kokovay, et al., "Vertebrate neural stem cell segmentation, tracking and lineaging with validation and editing," *Nature Protocols*, vol. 6, 2011.