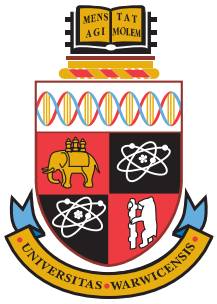


Extracting fluorescent reporter time courses of cell lineages from high-throughput microscopy at low temporal resolution

by

Mike Downey

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Supervisors: Dr Till Bretschneider, Dr Keith Vance and Dr Graham Ladds

MOAC

June 2012

THE UNIVERSITY OF WARWICK

Contents

List of Figures

List of Tables

Abstract

Live Cell Imaging and High Throughput Screening are rapidly evolving techniques and have found many applications in recent years. Modern microscopy enables the visualisation of internal changes in the cell through the use of fluorescently tagged proteins which can be targeted to specific cellular components.

A system is presented here which is designed to track cells at low temporal resolution within large populations, and to extract fluorescence data which allows relative expression rates of tagged proteins to be monitored.

Cell detection and tracking are performed as separate steps, and several methods are evaluated for suitability using timeseries images of Hoechst-stained C2C12 mouse mesenchymal stem cells. The use of Hoechst staining ensures cell nuclei are visible throughout a time-series. Dynamic features, including a characteristic change in Hoechst fluorescence intensity during chromosome condensation, are used to identify cell divisions and resulting daughter cells.

The ability to detect cell division is integrated into the tracking, aiding lineage construction. To establish the efficiency of the method, synthetic cell images have been produced and used to evaluate cell detection accuracy. A validation framework is created which allows the accuracy of the automatic segmentation and tracking systems to be measured and compared against existing state of the art software, such as CellProfiler. Basic tracking methods, including nearest-neighbour and cell-overlap, are provided as a baseline to evaluate the performance of more sophisticated methods.

The software is demonstrated on a number of biological systems, starting with a study of different control elements of the *Msx1* gene, which regulates differentiation of mesenchymal stem cells. Expression is followed through multiple lineages to identify asymmetric divisions which may be due to cell differentiation.

The lineage construction methods are applied to *Schizosaccharomyces pombe* time-series image data, allowing the extraction of generation lengths for individual cells. Finally a study is presented which examines correlations between the circadian and cell cycles. This makes use of the recently developed FUCCI cell cycle markers which, when used in conjunction with a circadian indicator such as Rev-erbα-Venus, allow simultaneous measurements of both cycles.

Acknowledgements

There are many people to thank, starting with my supervisors: Till Bretschneider, Keith Vance and Graham Ladds. Next there are the staff of MOAC and Systems Biology, both past and present, including Alison Rodger, Dorothea Mangels, Monica Lucena, Sarah Grilli-Shute, Anne Maynard and Brent Kiernan.

I would like to thank my fellow MOAC and Systems Biology students and the various members of Graham Ladds' and Georgy Koentge's labs. You are too numerous to mention individually. I would also like to thank Jonathan Millar, of the Medical School, for providing a strain of *Schizosaccharomyces pombe* which proved invaluable during the latter half of the PhD.

Other people who contributed in one way or another include my Advisory Panel: Sascha Ott, Georgy Koentges and Lorenzo Frigerio. There are also the people I collaborated with for the C5Sys analysis: David Rand, Peter Krusche, Kathy Tamai and Filippo Tamanini.

Finally I would like to thank the EPSRC for funding my study at Warwick.

Chapter 1

Introduction

If we want to follow a biological process in real time at the cellular level, one way of achieving this is by monitoring levels of gene expression or levels of protein within a cell. This can be done by using a modified version of a gene which consists of the regulatory region and promoter of the target gene which then drives expression of a fluorescent protein such as GFP or YPF (Green/Yellow Fluorescent Protein), which will be produced simultaneously with the gene of interest. The cells are imaged using a process such as fluorescent microscopy and the intensity of the image will be related to the levels of protein present.

Measuring the levels of a protein at a particular time will only give limited information about cell activity. It is much more useful to observe the levels of protein over time. This requires the cells to be observed multiple times and any cell motion will complicate the measurements since cells will not be in the same positions from image to image. If we are able to track the cells as they move in culture we can then follow inner processes over time such as circadian or cell cycles and gene expression. If we are able to follow gene expression in any daughter cells we can then see if expression is consistent across daughter cells or over multiple generations. The large numbers of cells involved means the process will benefit from automation. When we have high densities or rapidly moving cells then tracking can become a tricky problem.

The first part of the challenge is to actually detect the cells. Humans seem to have a natural ability to pick out objects, even in low contrast scenes or when objects are overlapping. This is less straightforward for computers and there are many different ways of tackling the problem of locating and isolating objects

within a scene. The next part is to take the detected objects and follow them as they move. Different methods are investigated for detecting and tracking the objects.

1.1 Aims and Objectives

The aim of project is to develop a set of tools which will be able to measure changes in fluorescence intensity over time and thus obtain the temporal change in expression of a fluorescently tagged gene. The tools were originally applied to measuring expression of *Msx1* in C2C12 mouse mesenchymal cells but were subsequently adapted to track yeast cells and then to obtain frequency signatures of circadian and cell-cycle oscillators.

In the *Msx1* study, it will be important to compare the transcriptional signature of daughter cells so there needs to be a method of constructing a cell lineage to allow comparison across daughter cells.

The tracking algorithm must be able to handle the low frame rates which are often a consequence of experimental setup, such as reducing the effect of using ultra-violet light where it is required to excite fluorescent dyes.

1.2 Outline of Thesis

This thesis describes the work performed and results obtained during the course of my PhD. It is divided into the following sections:

1. **Introduction** Overview of current state of single-cell analysis and the techniques used.

2. **Materials & Methods** Experimental techniques, image analysis and data analysis methods used.

3. **Method Development** Detailed description of the data analysis methods.

4. ***Msx1* Expression** Applying the analysis methods to track gene expression in C2C12 cells.

5. **Performance and Validation** Detailed description of the construction and use of the data sets created to both validate and optimise the data analysis.

6. ***Schizosaccharomyces pombe*** Constructing lineages for *Sz.pombe*.

7. **Cell Cycle Analysis** A case-study using the LineageTracker software to perform analysis of circadian rhythms and cell cycle.

8. **Discussion** Overview of the results

Appendices Including detailed description of the data formats used in the analysis.

1.3 Extracting time course data from live cells

The extraction of fluorescence time course data is a major bottleneck in high-throughput live-cell microscopy. Under the control of different regulatory promoters, live cell fluorescent reporter-based techniques reveal the dynamics of gene expression [1–4], in individual cells and over periods of several days. This allows relative quantification of protein levels within cells.

In a typical population, cells will be at different positions within the cell cycle and accordingly individual genes will show different levels of expression. Additionally, there may be instances where only a small proportion of cells exhibit a particular transcriptional behaviour. Measurements must therefore be made on single cells, rather than whole cell populations, to determine what is actually happening and prevent interesting cell behaviours being 'averaged out'.

In order to achieve this, a method is required which can locate cells within an image and keep track of individual cells in subsequent frames should they move from their initial positions. The cell tracking method may rely on measurements such as cell position and intensity to aid identification of the cell in future frames.

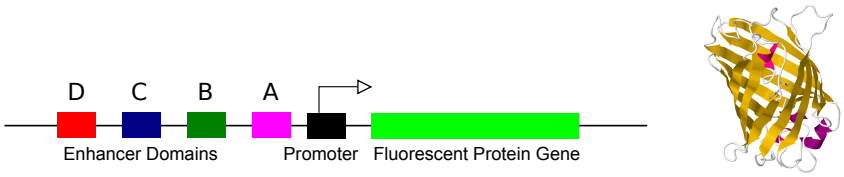
Image acquisition using fluorescent microscopy often requires high intensity light to sufficiently excite any fluorophores which may be present. In some situations this may also require the use of ultra-violet light, which is potentially damaging to cells. When measurements are required over long time periods, up to several days, it often becomes necessary to increase the sampling interval to reduce the damage caused to cells and to prevent photobleaching or phototoxicity. Large time intervals can equate to large cell displacements which may prove problematic when tracking cells.

The project was conceived as part of a study into expression of *Msx1* in C2C12 mouse mesenchymal stem cells. Mesenchymal stem cells are multipotent stem cells which are capable of differentiating into different types of cell including muscle, cartilage, fat or bone cells, depending on their environment.

C2C12 cells are highly motile which presents a challenge to a tracking algorithm, especially when confronted with high density cell populations. The typical doubling time of C2C12 cells is around 20 hours [5] which allows many fluorescence measurements to be obtained between divisions, while also allowing a significant number of divisions to occur during overnight or 24 hour time-series experiments. By measuring expression, tracking cells and following lineages, it will be possible to determine whether the expression of *Msx1* varies during a lineage or between daughter cells. Any heterogeneity between cells or within populations can thus be measured.

Automatic cell tracking requires cells to be visible at all times so a nuclear marker, such as Hoechst, can be utilised as a permanent stain. Any changes in nuclear shape or size will be measurable and can therefore be used to obtain information which can aid in detecting cell divisions. Hoechst is a DNA binding stain (which binds to the minor groove) so the chromosome condensation which occurs prior to division will be visible as an increase in the emitted fluorescence.

Earlier work had detected two regulatory regions for the *Msx1* gene [6]. Recent work at the University of Warwick by Sascha Ott and Keith Vance has identified 4 such regions (see Figure 1.1a). The PhD began with the creation of C2C12 cell lines expressing variants of the fluorescently tagged *Msx1* promoters (see Section 2.1.2). The methods developed during this time are described in Chapters 3–5.



a) b)

Figure 1.1: a) The *msx1-gfp* promoter construct. b) GFP structure

The second half of the PhD moved into imaging the fission yeast *Schizosaccharomyces pombe* in the laboratory of Graham Ladds. The aim was to investigate ways of visualising the cells to enable tracking and lineage construction, which will enable measurement of cell cycle times. The yeast cells are rod-shaped, and the length can vary under different growth conditions so an ability to measure the dimensions of the cells, not just the nuclei, will prove useful in future work. These requirements led to modifications of the segmentation and tracking software to deal with different appearance and motion of the cells. The yeast imaging and analysis is discussed in Chapter 6.

The software found an additional application courtesy of David Rand from the Systems Biology department. Time-course data was obtained from the C5Sys project* where cells exhibited three different fluorescent reporters associated with cell cycle and circadian rhythms. The aim was to extract the oscillating signals and investigate any connections between the periods and phases of the oscillators. Further details and data analysis are in Chapter 7.

1.4 Measuring Gene Expression in Single Cells using Fluorescent Proteins

One of the standard ways of measuring gene expression in individual cells is to take the regulatory region of a gene and replace the coding region with a sequence which encodes a fluorescent protein (as illustrated in Figure 1.1a). Measuring the fluorescence of the cells will then indicate whether the original protein was being expressed or not.

Green Fluorescent Protein (GFP) was discovered in 1960s by Osamu Shimomura at Princeton University, in the jellyfish *Aequorea victoria*. In the mid 1990s Martin Chalfie, working at Columbia University, succeeded in creating a transgenic *C. elegans* which produced GFP using a promoter from *β-tubulin* to drive expression of the protein. This enabled the researchers to see where the protein was being produced during development [7]. Different variants of GFP have been produced, following work pioneered by Roger Tsien. The original GFP required oxygen to fluoresce but newer versions are active under different conditions and emit a wide variety of different wavelengths. This work

*`http://www.erasysbio.net/index.php?index=272`

eventually earned Osamu Shimomura, Martin Chalfie and Roger Tsien the Nobel Prize for Chemistry in 2008 [8].

Cells expressing such fluorescent proteins, coupled with computerised cell detection, allow measurement of gene expression in individual cells [9]. The long half-life of these proteins (for example 14 hours for Yellow Fluorescent Protein, YFP [10]) will prevent measurement of rapid changes in expression since the continuing fluorescence of any existing protein will swamp small changes.

There are sequences of amino acids such as PEST, which contains high proportions of Proline (P), Glutamic Acid (E), Serine (S) & Threonine (T). The presence of such sequences acts as the signal for ubiquitin-tagged degradation of the protein by a proteasome [11, 12]. These are commonly used to decrease the half-life of the protein and thus improve temporal resolution.

Fluorescence microscopy provides relatively high contrast images with dark backgrounds, compared to Bright-Field microscopy (including Phase-Contrast or DIC) where the background intensity and the cells can exhibit overlapping intensity ranges. Such images were less straightforward to analyse and methods such as thresholding would only isolate part of the cell from the background.

Recently there have been advances in segmentation methods for bright-field images. One approach is based on a z-stack obtained from confocal microscopy [13] and takes advantage of the observation that the variation in intensity across the stack is lower for a point in the image background than for a point within a cell. Projections through the stack are constructed using statistical methods including standard deviation, which are used to calculate a segmented image. Another approach involves image deconvolution based on computational modelling of the light path taken through the sample [14].

Tracking cells during *in-vivo* imaging is a non-trivial task for most organisms due to the opaque nature of the tissue. Constructing lineages during development is often only possible during early stages of embryogenesis up to the formation of the blastoderm. The situation is improved, however, when the embryo is transparent as in the case of Sea Urchin [15, 16] or Zebrafish [17–19], or even *C.elegans* where the entire organism remains transparent throughout development [20, 21].

1.5 High Throughput Screening is a rapidly evolving field

In recent years, live cell imaging and High Throughput/High Content Screening (HTS/HCS) have become popular techniques used in research [22]. The combination of fluorescent proteins and high quality imaging systems has allowed rapid protein quantification in live cells [23], which allows the internal cellular processes to be studied at a higher resolution than previously possible.

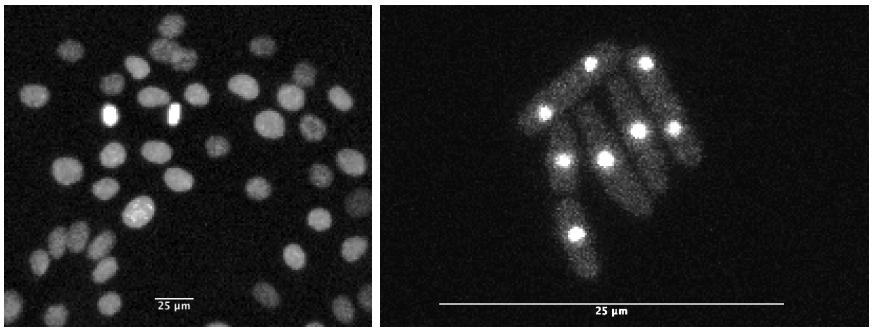
If all the cells being studied exhibit the same behaviour then a measurement of the mean fluorescence of an image containing multiple frames will provide sufficient information. All the cells in the culture will need to be synchronised [24] to ensure they are at the same phase of their circadian or cell cycle, depending on the feature being measured. Such fluorescence measurements also require all the cells to behave in a coherent manner with equal cycle lengths otherwise they will rapidly go out of phase (examples of this are given in appendix B.1).

Many studies have been published by the Alon group detailing single cell analysis [25], including cell cycle dependent nature of nuclear proteins [26] which used a version of CellProfiler. The CellProfiler software performs cell image analysis on fluorescent microscopy images and works by following a set of instructions (called a 'pipeline') which perform the analysis and measurement steps. One disadvantage with CellProfiler (which is shared by other cell analysis software) is an inability to allow recovery from errors, such as cells being incorrectly identified during measurement or tracking. Further details on CellProfiler are given in Section 1.6.2.

Results of large-scale proteomics studies have followed [27], where nearly 1,000 proteins were tagged and followed in cells, but the image analysis for this used custom-written software. The above techniques, in conjunction with measurements of subcellular features [28], have been used in large-scale cell phenotyping [29, 30] by using machine-learning to train classifiers to identify proteins.

HTS developed from video microscopy and benefited greatly by advances in computing power, enabling analysis to to keep up with the quantities of image data being generated. Instruments can acquire many thousands of images per

day, which requires suitable automatic data analysis. Manual image analysis is a time consuming process which will often be the rate limiting step of the acquisition-analysis pipeline.



a) C2C12 b) *Sz.pombe*

Figure 1.2: Examples of cells exhibiting different fluorescent markers: a) C2C12 cells with the Hoechst nuclear marker, b) *Sz.pombe* with a fluorescent-tagged nuclear protein.

Systems can be based on confocal [31] or wide-field microscopy [32, 33] but a common feature is a form of automatic sample-changer to allow different cells to be imaged without constant user intervention. Fully automatic 'off the shelf' systems such as *Cellomics KineticScan* and *GE IN Cell Analyzer* incorporate sample handling, microscopy, controlled environments and analysis software as a complete package. These systems commonly use 96- or 384-well plates to enable multiple cell lines or transfections to be imaged simultaneously. Many of the automated systems can also perform sample preparation on a large scale [34, 35] which increases the scope of an experiment and the repeatability of results.

1.6 Cell Tracking using multiple features

Recently software has become available for high resolution cell tracking and spatio-temporal analysis of protein dynamics in sub-cellular compartments (for example, QuimP [36], CellTracker [37]). These methods are designed to track cell boundaries and work best when the cells only move by small amounts. If the cells do not overlap then the task of matching a section of boundary with the equivalent section in the subsequent frame becomes much more complex.

Conventional tracking methods often require at least a minimum overlap

to link cell positions between consecutive frames, measured either in absolute pixel counts, or relative to object size. This is the approach used by CellID [38], CellTracer [39], and Overlap-Based Cell Tracker [40]. Tracking algorithms based on overlap alone will be unable to handle cells with high motion or those captured with a low frame rate, where it is possible for cells to move by amounts greater than their diameter leading to no measurable overlap. High density populations, such as in Figure 3.13, can also be problematic since there can be several potential cells in the vicinity to confuse identification. In the absence of guaranteed cell overlaps, the cell positions may be used where the proximity to the previous coordinates identifies the cell in subsequent frames. If cells exhibit persistent motion and cell collisions are infrequent, keyhole tracking algorithms can be applied which calculate the probability of finding matching cells in a particular direction [41].

A number of single particle tracking methods have been developed recently which are able to track multiple non-overlapping objects, and can in principle be applied to tracking cells [42]. These approaches are related to tracking methods for 'point-like' particles [43] such as in colloids [44] or subcellular features such as protein localisation [29, 45]. The process is performed in separate detection and linking steps to build up the trajectory for each individual particle.

Another approach is demonstrated in the particle filter methods which have been developed for tracking objects [46, 47]. These are probabilistic methods where future positions of objects are predicted using a motion model, and then matched with objects at the real positions. This usually involves solving a global linear assignment problem [48]. Both graph-based and hidden Markov model approaches can easily be extended to include additional object features such as shape, size, colour, or texture. However, for huge problems including time-series with thousands of cell positions, global optimisation approaches are computationally very costly. Furthermore, particle filters work best for small frame to frame displacements where motion across frames is highly correlated. As a result they have found application in microtubule analysis [47, 49] where direction of motion is largely unchanged. In time-series with low temporal resolution and considerable cell motion these approaches can perform poorly.

Where cell position and motion are insufficient to provide suitably accurate

identification it becomes necessary to place emphasis on measured features of the cell rather than simply relying on motion characteristics. The point-tracking methods are built on the expectation that the object is below visible resolution and appears as a blurred point-spread function which is often approximated as a Gaussian to locate the position with sub-pixel resolution. A different approach can be used for cells which exhibit measurable shape and size, where the similarity in such features across different frames can be used to aid identification. Fluorescence intensity and texture measurements may be used in a similar manner [50, 51].

1.6.1 Different approaches to constructing cell lineages

There have been some approaches to lineage construction based on appearance or behaviour of cells during mitosis [52]. Debeir [53] computes tracking in reverse from the final frame and divisions are detected by the merging of two daughter cells. As the cells approach mitosis, the size decreases and the cells approach closer to each other. When size and distance are below a given threshold, the reverse mitosis event has completed. Wang [51] calculates texture based features and uses feature reduction methods including PCA to reduce 145 features to between 15 and 20. Divisions are detected by treating each stage of the mitosis event as a hidden state in a Markov chain. A training set was used to calculate the probabilities for the chains. Similarly, Markov trees were used in [54] to map cell states to lineages.

Al-Kofahi *et al.* [50] construct lineages by calculating a significance score based on the observation that daughter cells have a similar size. The Ellenberg group has developed a powerful framework for automatic detection of cell divisions and chromosome phenotypes [55, 56]. Their approach, which is based on 3D time series with stacks captured at 5-7 minute intervals, makes use of region adaptive thresholding and a feature point tracking method. Probabilities for detecting mitosis events are based on size and distance of chromosome sets for which weights are determined empirically. Li *et al.* [46] and a more advanced version by Bise *et al.* [57] use phase contrast images for cell segmentation and detection of mitosis events which appear brighter in phase contrast. Cell trajectories are assembled into shorter fragments first, called tracklets, that

are stitched together by using a global optimisation problem *a posteriori*. Accuracies achieved are 87% for tracking (correctly identified cell-cell linkages between frames) and 68% for detecting divisions correctly.

Padfield [58] make use of a Hoechst label to segment nuclei, although imaging at higher frame rates of 6 or 15 minutes. They use a wavelet based method for cell segmentation. Subsequently a graph flow method is used for tracking cells and they report 99.2% of cells tracked with complete accuracy (with an average track length of 13 frames) and 97.8% correctly identified divisions, validated using 104,000 cell positions. The methods by Bise and Padfield are both advanced methods, however they result in markedly different detection rates and accuracies. It is difficult to pinpoint a single cause for this, most likely reasons being experimental differences in cell density, movement and clustering. For example, the net translocation of cells observed by Padfield is small (after correction for stage drift) and thus makes validation of large numbers of cells comparatively easy. Comparison of different methods is almost impossible since many of them are only available as part of an integrated commercial platform or not publicly available. Often, precision of different segmentation routines is not validated based on objective ground truth using synthetic data, but by human observers [58] and it is difficult to obtain a comprehensive list of all parameters being used. Since there is currently no standard for exchanging track-data for evaluating different methods, a new framework was developed, based around ImageJ, which allows comparison between different segmentation and tracking methods.

The method presented here incorporates the tracking of cell lineages in the statistical scoring framework for cell tracking. It makes use of dynamic feature changes, especially characteristic changes in Hoechst distribution and nucleus area during and after cell division, which are described in Section 3.4.11.

1.6.2 Freely available High Throughput Screening Software

In addition to the commercial systems mentioned in Section 1.5, there are two main freely-available systems: CellProfiler and ImageJ. Both have the advantage of being *Open Source*, which means the source code is made available for developers and end-users which makes it possible to modify and extend the

software.

CellProfiler is available for Windows, Linux and Apple. It supports the most common image file formats produced by high-throughput imaging systems including TIFF, PNG, AVI movies as well as some proprietary formats. The application was designed for detecting cells and subcellular features and measuring associated shape, size and intensity features and has been used to help identify different phenotypes in tissue imaging [59]. It is highly flexible and supplies a range of image processing and analysis methods. The analysis steps are built up using a pipeline process where each stage produces output which may be used by subsequent stages. A typical pipeline will consist of stages to load the image data followed by initial image processing such as background correction or contrast adjustment. The segmentation step is based on thresholding and provides a number of different methods for separating cells from the background. There is some support for coping with over or under-segmentation where larger objects can be split or smaller objects can be combined.

ImageJ [60] is primarily an image processing and analysis tool with emphasis on scientific imaging. It is an extensible platform with a plugin-based architecture where modules can be installed to add additional functionality. It accepts all common image formats and there are plugins such as BioFormats which give access to many more, including a wide range of proprietary microscopy formats.

Image segmentation can be performed using thresholding methods or by applying different image processing operations in sequence using the built-in scripting language which provides the ability perform sequences of operations (called 'macros') which can be repeated on multiple images. The Fiji Project* builds upon ImageJ and provides additional image analysis methods.

1.6.3 There is a need for new software

Existing systems were either fully automatic (such as CellProfiler) or fully manual (such as ImageJ with MTrackJ) [61]. No fully automatic system is capable of segmenting and tracking with complete accuracy. ImageJ was chosen as the platform since it is widely used in scientific image analysis with an extensive and active community which discusses ideas and applications. It is

*Fiji Is Just ImageJ, http://pacific.mpi-cbg.de/wiki/index.php/Fiji

readily extended using plugins which can be written in Java.

Existing cell tracking methods for ImageJ are currently very limited, however. The Particle Tracker plugin is an implementation of Feature Point Tracking [43] and provides both segmentation and tracking based on the intensity moment of the particle images. Mtrack2 performs tracking but requires the segmentation to be performed beforehand. Trajectories are assigned by selecting the nearest particle in the following frame. SpotTracker [62] is designed to follow bright spots in fluorescent microscopy images. Unlike many methods which perform segmentation and tracking as two separate steps, SpotTracker combines both into a single step. The algorithm can only track single spots which makes it unsuitable for tracking large cell populations.

None of the pre-existing systems described above provided a suitable balance of automatic and manual segmentation and tracking to suit the imaging conditions. The outputs of the fully automatic software all require a small amount of manual correction to segmentation or tracking. As a result, software was written to provide these facilities, which performed the initial image analysis and tracking but then allowed user-intervention to correct any mistakes made by the software.

A statistical scoring approach was formulated which is based on a similarity matrix where scores are calculated for possible target cells within a maximum distance that can be covered by a cell in a given time interval (see Section 3.3). Computational demand for this local optimisation problem simply scales linearly with the number of cells to be tracked. Relevant similarity features are selected from a larger list of possible features based on the temporal correlation of each feature. Additionally the system was made compatible with existing data formats (Cellomics KineticScan, CellProfiler, MTrackJ, Mtrack3) to allow initial analysis to be performed elsewhere then imported into the software for further analysis or reporting.

Chapter 2

Materials & Methods

2.1 C2C12 cells transfected with *msx1-gfp*

C2C12 mouse myoblast cells (ECACC, Catalogue No 91031101) were grown in DMEM (containing phenol red) supplemented with 10% Fœtal Bovine Serum (FBS) at 37 °C in an atmosphere of 5% CO2. At each passage step the media was removed and the cells were washed twice in Phosphate Buffered Saline (PBS). Trypsin solution (0.5%) was added and incubated for 30–60s, then removed. The cells were resuspended in fresh media, then transferred to a fresh culture vessel.

2.1.1 Transient Transfection

Full details of the experimental procedure is given in [63].

For transient transfection the cells were transferred to a 96-well plate at a density of 1.25×10^4 cells per well. Cell densities were determined visually using a Hæmocytometer. Media containing cells (30 μl) was applied to the hæmocytometer slide. The marked area on the slide holds 10^{-4} ml of media.

Hoechst 33342 (Invitrogen) 400 ng/ml in DMEM was added and incubated at 37 °C for 30 minutes. Cells were then washed twice with PBS and 200 μl DMEM (without phenol red) was added. Cells in each well were subsequently transiently transfected with 200 ng of reporter plasmid using lipofectamine 2000 (Invitrogen) according to the manufacturers instructions.

The endogenous *Msx1* and the modified fluorescent-expressing genes are expected to be expressed at the same time. Transfected cells may contain multiple copies of the plasmid and during division each copy of the fluorescent

reporter gene may only pass to one of the daughter cells, leading to a reduction in copy number and therefore fluorescence intensity in successive generations. If all daughter cells need to reliably express the new gene, then a 'stable transfection' is required, where the new gene is incorporated into the cell's genome.

2.1.2 Stable Transfection

An improved version of the Venus Fluorescent Protein (VFP) is available which is more suitable for time-dependent measurements. Original VFP [10] has a long half-life of around 14 hours, which can mask rapid changes in gene expression. The modified version of the reporter contained the PEST sequence from the Myc protein [64] which increases proteolysis and reduces the half-life of the Venus to around 30 minutes.

Cloning the Plasmid

The fluorescent reporter is created as plasmid consisting of the *Msx1* regulatory region expressing the modified version of the Venus-mycPEST. To increase the brightness of the protein, two copies of Venus were separated by a 2A peptide

The plasmid was inserted into 'Top 10' competent *E.coli* cells (Invitrogen). The cells were slowly defrosted on ice, 3 μl of plasmid were added to each tube and gently mixed. The cells were returned to ice for a further 30 minutes before being heated at 42 °C for 45 seconds. SOC medium (150 μl) was added and the cells were placed in a shaking incubator for 1 hour. 150 μl of the culture was pipetted onto LB-Amp plates and incubated overnight at 37 °C.

Colonies were picked from the LB plates and grown in 3ml LB-Amp media overnight at 37 °C. The standard miniprep (QIAgen) procedure was followed to extract plasmid DNA, using 1.5ml of cell culture, with the final elution using 40 μl of water.

A stable transfection was produced via the addition of a drug resistant gene cassette (pTk-Hyg) in addition to the fluorescent *Venus-Msx1*. A small number of cells will have incorporated the DNA into their genome in a suitable location, where it can be readily expressed and not bound up within chromatin. After several generations, only those cells which had suitably incorporated the DNA would survive.

Two solutions were prepared for transfection:

1. 1 μg *msx1*Venus
   100ng pTK-Hyg cassette
   200 μl Optimem.

2. 6.6 μl Lipofectamine
   200 μl Optimem

Cells were cultured in DMEM media with 10% FBS and the addition of hygromycin (400 μl/ml) to screen out cells which had not taken up the pTk-Hyg vector.

Selection of clones

The transfected cells were cultured in duplicate, along with a control which consisted of un-transfected cells.

Individual clones were isolated using cloning cylinders and cells were transferred to a 96-well plate by washing with PBS and treating with trypsin. Fresh media was used to transfer each clone to a separate well (200 μl of each PBS, Trypsin and Media used).

2.1.3 Image Acquisition using Cellomics KineticScan

Images were obtained using a Cellomics KineticScan KSR with a 10x 0.4NA objective, using a 1024x1024 CCD. Signal to noise was reduced by pixel binning (performed in hardware) where the intensities of 2x2 blocks of pixels were combined, which reduced the resolution to 512x512. Two image channels (Hoechst and vGFP) were obtained at intervals of between 30 & 60 minutes. An infra-red diode laser based autofocus system was used to focus on the inner surface of the bottom of the wells. Light is provided by a high-pressure mercury gas discharge lamp which emits light in the ultra-violet region of the spectrum. Filters were used to select the appropriate excitation wavelengths (XF136 for Hoechst, which is excited at 352 nm and emits at 461 nm and XF100 for Venus/GFP, which have an excitation peak around 500-510 nm and emit between 500-530 nm) before being collected by a CCD camera. Data was imported using custom software described in Section 2.4.2.

2.2 *Schizosaccharomyces pombe*

2.2.1 Cell Culturing

Stocks of *Sz.pombe* were maintained on Yeast Extract (YE) plates made according to Table 2.1 with the addition of 8% agar.

The minimum media (Table 2.2) was prepared by dissolving the chloride, phosphate and phthalate salts in $\frac{3}{4}$ of the final volume of water followed by autoclaving. After allowing to cool to 60 °C, the remainder of the ingredients are added. Stock solutions (Tables 2.3–2.5) are sterilised by filtration and stored at 4 °C.

Cultures for imaging were prepared by transferring cells from a plate to liquid media (10–25 ml) using a flame-sterilised loop. Cultures were grown for 24–48 hours in a shaking incubator at 30 °C.

Yeast Extract	2.5 g
Glucose	12.5 g
Adenine	250 mg
Leucine	250 mg
Uracil	250 mg

Table 2.1: Yeast Extract (YE) Growth Media (per 100ml)

Ammonium Chloride	5 g/l
Dibasic Sodium Phosphate	2.2 g/l
Potassium Hydrogen Phthalate	3 g/l
20% Glucose	100 ml
Stock Vitamins (Table 2.3)	1g
Stock Minerals (Table 2.4)	100 μl
Stock Salts (Table 2.5)	20 ml

Table 2.2: *Sz.pombe* Minimal Media (per litre)

Nicotinic acid	1 g
Inositol	1 g
Pantothenic acid	100 mg
Biotin	1 g

Table 2.3: 1000x Stock Vitamins (per 100ml)

Citric acid	2 g
Boric acid	1 g
$MnSO_4 \cdot H_2O$	1 g
$ZnSO_4 \cdot 7H_2O$	800 mg
$FeCl_3 \cdot 6H_2O$	400 mg
Molybdic acid	610 mg
Potassium Iodide	200 mg
$CuSO4 \cdot 5H_2O$	80mg

Table 2.4: 10,000x Stock Minerals (per 100ml)

$MgCl_2 \cdot 6H_2O$	26.25 g
$CaCl_2 \cdot 2H_2O$	367 mg
KCl	25 g
Na_2SO_4	1 g

Table 2.5: 50x Stock Salts (per litre)

2.2.2 Image Acquisition

A 2–3 μl aliquot of cell culture was deposited on YE-agar plugs fixed to a microscope slide and allowed to air-dry for 1–2 minutes. The slides were incubated for 30 minutes to 1 hour at 30 °C.

All images were obtained using a Leica SP5 confocal microscope fitted with a temperature controlled stage maintained at 30 °C. Image resolution was 1024x1024 using 8-bits per pixel. Oil immersion x63 or x100 objectives (both 1.4 NA) were used.

Time-series were obtained which comprised of a z-stack of 20 slices (total thickness of 20–30 μm) with frame intervals between 5 and 15 minutes. A maximum projection of this z-stack was subsequently used in image analysis.

2.3 Cell Cycle Analysis

Chapter 7 follows the cell cycle in Zebrafish PAC2 and NIH3T3 cells using the FUCCI markers which indicate the phase of the cycle based on the activity of two ubiquitin ligases, which degrade cell-cycle regulating proteins at appropriate phases [65, 66].

2.3.1 Zebrafish PAC2 embryonic cells

A time-series image sequence of Zebrafish PAC2 cells was supplied by Kathy Tamai and David Whitmore's lab in UCL [67]. The images were obtained from cells derived from 24-hour embryos which had been transfected with FUCCI constructs mKO2-zCdt1(1/190)/pT2KXIGΔin (for G1) and mAG-zGeminin(1/100)/pT2KXIGΔin (for S-G2-M), using plasmid pcDNA3.1/myc-His A (Invitrogen). The FUCCI constructs were provided by Professor Atsushi Miyawaki at the Riken Brain Science Institute [68].

Time-series images were obtained at 28 °C on an inverted Leica SPE confocal microscope using a x10 0.3 NA objective. Images were acquired at 15 minute intervals for a total of 65 hours.

2.3.2 NIH3T3 mouse fibroblasts

Images were provided by Filippo Tamanini which were obtained by Shoko Saito at the Erasmus University Medical Center in Rotterdam. NIH3T3 mouse embryonic fibroblasts were prepared which expressed Venus-NLS-PEST under control of the mouse Rev-erbα promoter [69]. Cells were cultured in 1:1 DMEM:F10 (Lonza) with 10% FBS and penicillin-streptomycin. The FUCCI cell cycle markers were slightly modified from the original versions, using CFP to visualise the S-G2-M phase instead of the original mAG.

Images were obtained at 15 or 30 minute intervals using a ZEISS LSM510 with a Plan-Neofluar x20 0.75 NA or a x40 1.3 NA oil-immersion objective. The CFP channel was visualised using a 458nm laser and a BP470-500 filter; Venus using a 488nm laser and a BP505-550 filter; and mKO2 used a 561nm laser and LP585 filter.

2.4 Computational Analysis

Image data were obtained in different formats depending on the data source. The Cellomics images were stored in a 'Device Independent Bitmap' (DIB) format which was incompatible with the Microsoft Windows 'DIB' files (see Appendix A.3.2 for further details). Images from confocal microscopy were converted into industry standard 'Tagged Image Format Files' (TIFF) within ImageJ.

2.4.1 Software Development Environment

The software was developed and tested on different computers, including: Java version 1.6.0_23 (32 bit) on Windows XP Service Pack 3, Java 1.6.0_26 (64 bit) on Mac OSX 10.6.8 , and Java 1.6.0_17 (64 bit) on SUSE Linux (kernel 2.6.16.60).

Image analysis was provided by ImageJ* version 1.43o or higher. Multi-channel visualisation required Image5D 1.2.0 or higher and image conversion used Bio-Formats Importer 4.3 (both available in Fiji†)

Additional java libraries from the *Apache Commons*‡ project were used: commons-lang 2.4 (general purpose additions to Java) commons-math 2.0 (mathematics and statistical functions). Some data plotting was performed using JFreeChart§ 1.0.13. Database import was provided by Jackcess¶ version 1.1.21. Java software was developed using the NetBeans 6.9.1 Integrated Development Environment.

Data analysis used Matlab; versions R2010a and R2011b on OSX, R2009b on Windows XP, R2008b on Linux with additional data analysis using Perl version 5.10.0. Curve fitting used Matlab libraries EzyFit Toolkit version 2.40‖, and the Gaussian Mixture fitting function**.

CellProfiler†† 2.0β build 10415 [59] and ParticleTracker‡‡ v1.5 were used in the segmentation and cell tracking performance comparisons.

2.4.2 Data Import

A custom import module was written (based on software by S. Ott and T. Bretschneider) to convert Cellomics data (version 1.35) from the proprietary Microsoft Access™ format into text files suitable for importing into other applications. Jackcess, a library for reading and writing Microsoft Access databases, was used for this purpose.

*http://rsb.info.nih.gov/ij/
†http://fiji.sc/
‡http://commons.apache.org/
§http://www.jfree.org/jfreechart/
¶http://jackcess.sourceforge.net
‖http://www.fast.u-psud.fr/ezyfit/
**http://www.mathworks.com/matlabcentral/fileexchange/4222-a-collection-of-fitting-functions
††http://www.cellprofiler.org/
‡‡http://weeman.inf.ethz.ch/particletracker/

Images were obtained from the imaging systems in proprietary formats which required converting to the general purpose TIFF format for processing and analysis. Cellomics images were converted from a proprietary DIB format to TIFF using custom methods. Image files from the Leica SP5 microscope were converted to TIFF using BioImaging plugin in ImageJ.

Chapter 3

Method Development

There are two requirements for reliable segmentation: the software must accurately recognise that cells are in existence at a particular location and must also accurately obtain the outline or extent of the cells in order to measure the degree of fluorescence within the cell. The segmentation method is based on Gaussian Maxima and Seeded Growth methods, where the cell seeds are detected by applying a Gaussian kernel to the image followed by the growth stage. The version presented here incorporates a novel extension to handle multi-channel images.

The tracking algorithm needs to be able to handle low temporal resolution where cells are likely to move by large amounts between frames. This is a consequence of the experimental setup where exposure to ultra-violet light (used to visualise the Hoechst stain) had to be reduced by increasing the intervals between images. Since long total durations of at least 2 days were required to increase the number of cell divisions observed, the sampling intervals were at least 30 minutes, which reduced the damage due to UV cytotoxicity [70].

Development of the cell tracking method begins with a study of the cell feature measurements, using correlation plots and principal component analysis to identify the features which will then be used in the tracking. This section also includes an analysis of the cell motion, comparing two ground truth datasets with simulated motion. The tracking algorithm is designed to handle cells imaged at low temporal resolution so one of the ground truth sets was created based on an experiment with 30 minute frame intervals. This is a larger interval than typically used elsewhere, which will provide a challenge to the tracking

algorithms.

3.1 Segmentation Methods

Different cell segmentation methods were evaluated, using ImageJ or CellProfiler. These are described below.

3.1.1 Thresholding based segmentation

Segmentation by thresholding distinguishes between background and foreground objects by partitioning the pixels into two or more categories according to the pixel intensity. Global threshold methods calculate a single intensity value for a whole frame where any pixels in the image with greater intensity are labelled as foreground, with all other pixels labelled as background. The resulting images typically use values of zero for background and maximum intensity (255 for 8-bit images) for foreground.

Both ImageJ and CellProfiler provide a wide range of threshold-based segmentation methods. The simplest of these are 'global thresholds' such as Li [71] or Otsu [72] which calculate a single threshold for the entire image and thus assign cells to foreground or background (see Figure B.2 in Appendix for an example and [73] for a summary of segmentation methods).

The Otsu method (see Figure 3.1b) is a common threshold algorithm which is a histogram-based method where the threshold is selected by maximising the variance between the foreground and background regions (or minimising the variance within each region). While this is commonly used to create a binary threshold, there are variants of the method which calculate multiple thresholds [74]. Similar results are obtained from clustering methods, such as K-Means [75] which assigns pixels to one of several clusters based on the difference between the pixel values and the mean intensities within each cluster (Figure 3.1c).

Adaptive thresholding uses a more flexible method where thresholds are calculated for sub-regions of the image instead of for the entire image. These methods can outperform global thresholding under certain circumstances, such as vignetting where image intensity decreases towards the edge of a frame, or uneven illumination of the field. An example of this is the Niblack method

[76] which calculates the threshold from a linear combination of the mean and standard deviation of the pixels within a small window around the pixel of interest (Figure 3.1d).

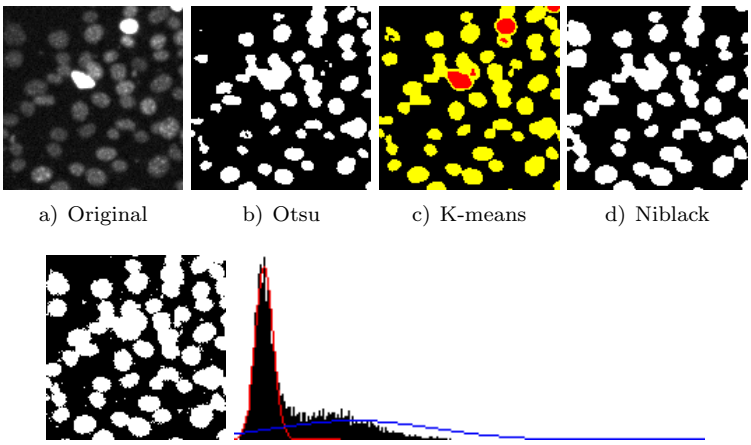
Histogram based methods attempt to choose the threshold based on the distribution of intensities in the image. This could involve analysing the shape and locating a dip between high and low intensities. A more complex example of this is the Mixture of Gaussians thresholding where the thresholds are determined by fitting gaussian components to the histogram (Figure 3.1e & f).

Threshold methods cannot handle objects which are touching or clustered together. Multiple touching objects will be detected as a single entity and a second processing step is required to separate them. The Watershed Transform [77] is used in these situations which can break the clusters into individual objects.

3.1.2 Scaling Index measures structural features

The concept behind the Scaling Index filter [78, 79] is related to the idea of 'fractal dimension'. This arose from the study of objects displaying scale invariant features (such as a coastline looking jagged at any scale) by Benoit Mandelbrot in the 1970s [80]. In classical geometry, objects are limited to integer dimensions such that a point has zero dimensions (no height, width or depth), a line is 1-dimensional and a plane is 2-dimensional. In fractal geometry, dimensions are not integers but are from the continuum of numbers. For example, a coastline takes up more 'surface area' than a straight line so has a higher dimension (> 1) but less area than a bounded plane (< 2). The concept can similarly be extended to surfaces which take up more volume than a flat surface but less than a solid cube. Point-like structures have a fractal dimension close to zero.

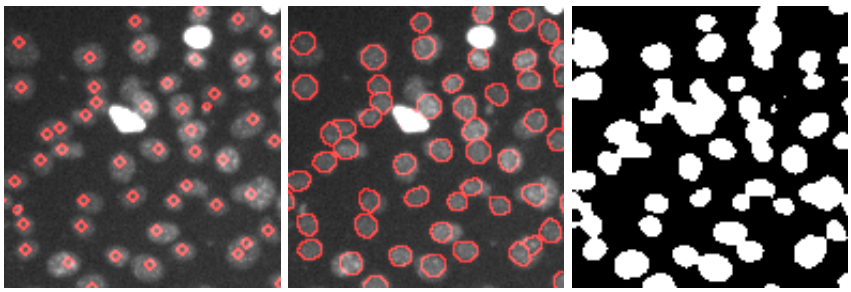
The scaling index calculates a property of the image related to this dimension, which can be used to extract features of a particular size and shape. Subsequent thresholding with upper and lower bounds will select objects within a range of dimensions. This can be used to locate cell nuclei (which are effectively large points) while ignoring linear features or large flat areas of



a) Original b) Otsu c) K-means d) Niblack

e) Gaussian Mixture f) Histogram with gaussian mixtures

Figure 3.1: Comparison of threshold segmentation methods. a) The original image used in the comparisons. b) Otsu global thresholding. c) K-means segmentation using 3 clusters. The image is compartmentalised into background, mid-level and bright pixels. d) Niblack local thresholding. e) Gaussian Mixture segmentation. f) Histogram with gaussian mixtures overlaid. The x-axis is pixel intensity and the y-axis is pixel frequency.



a) The start of a seeded growth segmentation. b) Part way through seeded growth segmentation. c) Image thresholded with the Niblack method for comparison.

Figure 3.2: The early stages of Seeded Growth segmentation showing the individual cells growing. A threshold based segmentation is included for comparison, showing how the seeded growth segmentation is more successful at separating touching cells.

illumination.

Scaling Index α is given by Equation 3.1:

$$\alpha = \frac{\log N(\mathbf{x}_i, r_2) - \log N(\mathbf{x}_i, r_1)}{\log r_2 - \log r_1} \tag{3.1}$$

Each point i in the image is represented by position $\mathbf{x}_i$. The number of points which lie within a radius r is given by $N(\mathbf{x}_i, r)$. Two radii, $r = r_1$ and $r = r_2$, are selected which are related to the size of the features within the image. Smaller values of r_1 allow the method to resolve small tightly clustered points whereas increasing values of r_2 improves the detection of large objects, at the expense of smaller objects.

The Scaling Index was originally devised for scanning probe microscopy where each point $\mathbf{x}_i$ is in 3-dimensional space so $\mathbf{x}_i = (x_i, y_i, z_i)$. For a 2-dimensional image where each pixel has intensity $I_i = I(x_i, y_i)$ the number of points within a radius can be replaced by the sum of intensities of pixels within a radius. The resulting image has similar properties to the 3-dimensional scaling index.

When the Scaling Index is applied to a microscopy image containing cells, the point-like cells will have a lower index than the image background. Additional segmentation methods can then be applied to the image to detect the cells. The Scaling Index method was written in Java following the above specification and implemented as an ImageJ plugin.

3.1.3 Seeded Growth is a flexible and robust segmentation method

The main segmentation method is a version of the seeded growth [81, 82] algorithm and consists of two parts:

1. Identifying the seeds which will be used to grow the objects.
2. Growing the cells from these seeds. This involves adding neighbouring pixels to a seed if they appear to belong to the cell rather than the background.

Both the thresholding and scaling index methods have shortcomings which are largely related to touching or overlapping cells. The seeded growth method

is an attempt to overcome these problems by locating cells based on the intensity profile, where the centre of each cell is a peak in intensity [83]. This ensures that, provided a cell is not completely obscured by another, there is an improved likelihood of detecting both cells.

The centres of the nuclei act as seeds and the intensity profile is used to guide the growth. Pixels in the surrounding ares are added to the seed if the intensity is above a threshold value.

If the image data consists of multiple image channels, the segmentation can be performed on a selected channel or the sum of all image channels. The cell seeds are obtained by convolving the image with a Gaussian kernel, where the standard deviation, σ, is proportional to the expected cell radius [84]. The cell centres are located by searching for local maxima based on the intensity difference between individual pixels and surrounding pixels and background (see below). Each seed pixel is then visited in turn and the neighbouring pixels are examined. If the pixel has a lower intensity than the original seed pixel and does not currently belong to another cell, then it is added to the current cell. The process continues until all of the pixels have been assigned.

This approach has an advantage over threshold based segmentation: since seeds are identified from maxima in the image, cells in close proximity will have their own maxima even if they appear to touch in the image. Threshold methods are unable to separate touching cells without additional processing steps. Unlike the Scaling Index method, which is very sensitive to the dimension threshold, the seeded growth method is more robust with a wider range of acceptable settings so small changes to parameter value will not normally have a dramatic effect on segmentation performance. It is possible to combine this approach with a threshold-based method by performing an initial threshold step to identify the image background. This step is most effective when there is good contrast between foreground and background and foreground cells all are of a similar intensity. The background is then disregarded during the seeded growth stage.

Locating maxima in an un-processed image will result in a large number of spurious peaks caused by image noise or any uneven distribution of DNA in the nuclei. The maxima detection must be preceded by a smoothing operation to remove this detail but retain the nuclei of interest. The ImageJ maximum finder

algorithm is used. This locates individual peaks or plateaux with intensity at least N greater than neighbouring pixels where N is the 'tolerance value'.

Two implementations of the growth algorithm were tested: the first was a simple neighbouring pixel method which examined the 8 pixels surrounding each seed, the second method was a radial growth method based on spokes emanating from the seed point (see Figure 3.3). Both versions take a threshold value and a maximum size limit for the growing cells and produce a cell mask image which is produced has the same dimensions as the original image, and starts off with every pixel set to zero apart from the seed positions which are plotted as single pixels where the pixel value is the cell ID number, assigned consecutively. The threshold is multiplied by the peak intensity for each maxima and growth continues until the intensity falls below this value. The resulting image is a mask where a pixel value of zero represents background and a non-zero number represents the cell ID.

A Radial Growth variant was developed which creates the same type of 'cell mask' image but prepares it in a different manner. Each seed is represented by a centre position and a set of vectors (angle and length) which point away from the centre. As the cells grow, the points on the outline will become separated by increasing distances. If the distance between adjacent points exceeds a set value (default = 5 pixels) a new interpolated point is inserted with angle and length calculated as the arithmetic mean of the neighbouring vectors. The algorithm is summarised in the steps below:

- Initialise the vectors with length $R = [1, 1, 1, 1]$ and angles Θ=[0°,90°,180°,270°]
- Initialise M = pixel mask & C = image of cells.
- Define function $I(Img, x, y, r, \theta)$ to return the pixel intensity of image Img at pixel coordinates $(x + r\cos(\theta), y + r\sin(\theta))$.
- Loop through each seed point (x,y):
  - For each $r = R_i$ & $\theta = \Theta_i$, Check the pixel value in the mask $I(M, x, y, r + 1, \theta)$
  - If the pixel value $I(M, x, y, r+1, \theta)$ is non-zero and the pixel intensity in the original image $I(C, x, y, r + 1, \theta) > I(C, x, y, r, \theta) \times$ threshold, increase the length of the vector by 1.

- If the distance between neighbouring vectors is greater than the limit, interpolate a new vector by taking the mean of adjacent vectors.
- Redraw the new seed on the mask image.

- Repeat until maximum size limit.

All cells are segmented simultaneously, with the outlines growing until they reach the threshold intensity or a neighbouring nucleus. This method is similar to the one described in [85] but was developed independently.

Segmentation Parameters

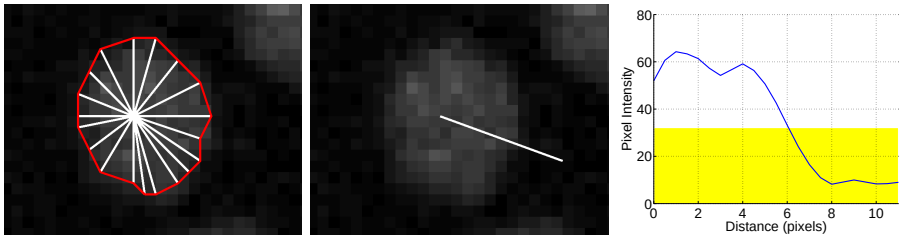
The most important segmentation parameters are: peak threshold (T_p), growth threshold (T_g) and number of iterations (N_{max}). The peak threshold is used in the maximum finder step and defines the difference in intensity between background and peak, or the depth of the 'valley' required between adjacent peaks. The growth threshold influences the rate of growth of the cells. This threshold is multiplied by the peak intensity and growth continues until that value is reached, so lower values lead to larger segmentation masks. The maximum number of iterations of the growth step is set as the expected size of the nuclei or cells (see Figure 3.4) and chosen such that it is greater than the longest axis of the biggest cell in the population.

Typical values of these parameters are $T_p = 17$, $T_g = 0.65$, $N_{max} = 30$. These values were obtained by visual examination of the segmentation output and were chosen to provide an acceptable balance between nucleus detection, false positive detections and outline shape. The outlines in Figure 3.5b were generated using the above parameters. The Seeded Growth is compared quantitatively with other methods in Chapter 5.

3.1.4 Extending the method to handle Multi-Channel images

The original Seeded Growth segmentation only takes a single image channel into consideration. An extension to the method is presented here where multiple channels are considered during the growth stage.

The new version of the method contains an additional constraint based on the 'colour difference' (Equation 3.3) between pixels which prevents two cells of



a) b) c)

Figure 3.3: Cells are grown by expanding spokes outwards until a threshold is reached. a) Cell with outline and spokes. b) One of the spokes highlighted. c) Intensity profile of the spoke with the intensity threshold highlighted in yellow.

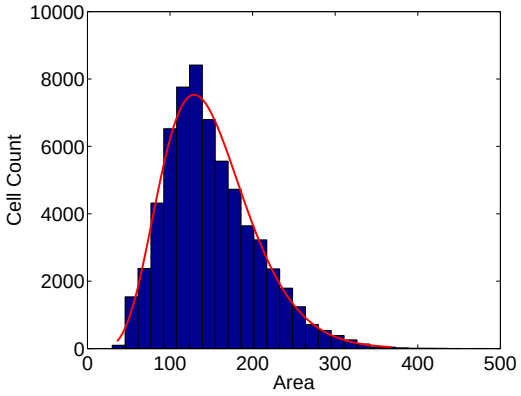
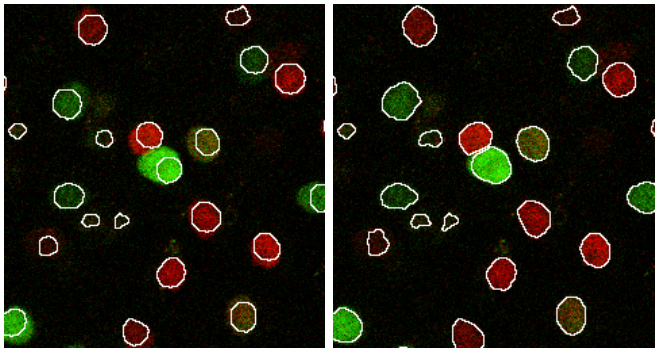


Figure 3.4: Distribution of nuclear sizes for C2C12 cells. The areas (measured in pixels where 1 pixel = 1.36 μm) follow a gamma distribution. Over 62,500 cells were measured. *Adapted from Downey et al. PlosOne (2011)*



a) Halfway through the seeded growth. b) Multichannel seeded growth completed.

Figure 3.5: Segmentation of Zebrafish PAC2 cells using the 'Multi-Channel Segmentation' method. The image consists of two channels (red and green) representing the two FUCCI markers. a) Seeded growth in progress. The red cell near the centre stops growing when it reaches the neighbouring green cell. b) After the final iteration of the seeded growth.

similar greyscale intensity but different colour from growing together. A pixel is added to the seed if both the colour and intensity criteria are satisfied. The intensity limit is dealt with as in Section 3.1.3.

Unit vectors holding the pixel intensities are calculated according to Equation 3.2, for the centre pixel and the pixel being considered. The colour difference is calculated using Equation 3.3 and pixels are rejected if the difference exceeds the threshold.

$$\hat{\mathbf{a}} = \{p_1, p_2, \ldots p_i\} \cdot \frac{1}{\sqrt{\Sigma(p_1^2 + p_2^2 + \ldots + p_i^2)}} \tag{3.2}$$

where p_i is the intensity of the pixel in channel i.

$$\text{Colour Difference} = \sqrt{\sum_i |\hat{\mathbf{a}}_i - \hat{\mathbf{b}}_i|^2} \tag{3.3}$$

i = channel number $[1, 2, \ldots]$ and $\hat{\mathbf{a}}$ & $\hat{\mathbf{b}}$ are normalised intensity vectors for the seed pixel and candidate pixels respectively.

Relationship between Seeded Growth and Watershed

The Watershed transform is a commonly used segmentation method which is related to the Seeded Growth in that it attempts to locate local maxima or minima. The method identifies catchment basins analogous to its namesake, the geological watershed which describes where water flows through a landscape [86]. To visualise the action of the watershed transform on an image, it helps to consider the image to be a 'height map' where the pixel intensities are related to the altitude of a landscape. If a Watershed transform is applied to such an image, the output will be the catchment basins which identify the locations of lakes or seas.

If an image containing fluorescent cells is to be segmented, brighter cells will appear as peaks rather than basins so the inverted image may be used. The watershed transform simulates flooding the landscape by pouring water into the deepest minima (which will form the catchment basins). As more 'water' is added, these basins will grow until two touch. At this point a dividing line (or dam) is drawn where they meet and more water is added to identify the remaining basins. A watershed implementation is available for ImageJ [87]

3.2 Tracking Algorithms

Several pre-existing tracking algorithms were applied to the cell position data as part of the method comparisons in Chapter 5. The methods are briefly described here.

3.2.1 CellProfiler tracking methods

CellProfiler provides four different tracking algorithms: An overlap-based tracker, 'nearest object' tracker, a feature-based tracker and a multi-object tracker. The operation of an overlap-based tracker is described in Section 3.4.2 and in [40].

The feature-based tracking method can utilise any cell or nucleus measurement such as object overlap, distance, intensity or morphology, and assigns the cell which has the lowest difference in the chosen feature value. A maximum range is considered so tracking will only select cells within that radius but otherwise will not take distance into account.

The multi-object tracker is based on the method by Jaqaman [88] which is based on finding a solution to the tracking by global optimisation of a cost matrix. This method also accounts for splitting and merging of particles.

3.2.2 ImageJ tracking plugins

The main tracking systems for ImageJ include ParticleTracker, MTrackJ and MTrack2. Particle Tracker is based on Feature Point Tracking [43] which includes segmentation based on locating local maxima which has been optimised for point-sized particles instead of extended cells. The segmentation and tracking methods can be run independently which enables the tracking to be performed on data obtained through other means, allowing the tracking to be run on the output of other segmentation methods.

The MTrackJ plugin allows rapid manual tracking to be performed and stores multiple tracks in a well-documented text format*. This provides a portable mechanism for storing and transferring individual cell trajectories.

The MTrack2 plugin (and the recent modification MTrack3) use 'nearest object' tracking and require a thresholded or pre-segmented image. As a result,

*`http://www.imagescience.org/meijering/software/mtrackj/fileformat.html`

there is no intensity information available to discriminate between neighbouring cells.

3.2.3 Matrix Minimisation by the Hungarian Algorithm

A common feature of many tracking algorithms is a matrix which holds cost values for potential cell-cell transitions. The optimum set of transitions is obtained following a matrix minimisation process such as the Hungarian Algorithm [89]. This is a solution to the matrix assignment problem where a minimum overall score is achieved by pairing each row with a column, analogous to assigning jobs to people where each entry in the matrix is the cost of each person performing each job [90–92].

The method assigns optimal matches by first 'reducing' the matrix by subtracting the lowest values from each row to determine whether each person has a unique best job to assign. If not, the matrix is further reduced by subtracting the minimum values from each row. A further iterative procedure modifies the table by taking the minimum value present in the matrix and subtracting it from all cells then adding it to the rows and columns which contain assigned elements.

An open-source implementation* of the Hungarian Algorithm was used in the cell-tracking code, which was adapted to handle the cost matrix as detailed in Section 3.4.1.

3.3 Development of the Detection and Tracking Software

The software was designed to handle data from the Cellomics KineticScan. The Cellomics data is stored as a database which holds the list of which wells were imaged in the 96-well plate and how many image fields were obtained for each well (described in Appendix A.3.1).

The Segmentation and Tracking viewer (see Figure A.2a) is used to import and segment the Cellomics data. After selecting the experiment, the layout of the 96-well plate is displayed where the desired wells can be imported. Data

*`http://sites.google.com/site/garybaker/hungarian-algorithm/assignment`

analysis routines were written to measure aspects of the data including cell motility, tracking performance and segmentation performance.

The software allows different segmentation, tracking and data analysis methods to be used and these are implemented using a 'plug-in' architecture similar to the one in ImageJ itself, where new methods are written to follow a particular specification and placed in the ImageJ directory. When several methods are available, the user can select between them and also adjust any settings or parameters (see Figure A.3c). Multiple image channels can be handled (Figure A.3b).

The Experiment Viewer (Figure A.4) allows not only viewing of the cell tracks but also editing the segmentation and tracking and exporting data and analysis results. The image window (Figure A.4c) displays the overlaid channels and cell outlines. The main part of the user interface (Figure A.4a) allows interaction with the data such as adding and removing cells, editing tracking and extracting fluorescent timecourse data. Segmentation and tracking information can be imported from other sources, including CellProfiler.

3.4 Identifying Features to use in Cell Tracking

During the segmentation step, several numerical features of the cells or nuclei are measured. These include shape and texture features similar to those utilised in feature-based cell-type classification methods [28, 45, 93] and methods for predicting cell fates of retinal progenitor cells [94].

The requirement for measuring features in addition to position becomes apparent when considering a simple example such as Figure 3.6. The scores are calculated using feature similarity taking a combination of position, size and intensity values, where a higher value indicates less change in a feature. The cell images show the original positions of the cells with the outlines indicating the positions of the cells in the following frame. The arrows connect cell positions in one frame with the positions of nearby cells in the following frame, where arrows of the same colour all originate from the same cell, pointing towards potential matches. Different colours of arrows, e.g. number 3 (red) and number 4 (blue) point to the same target cell in the centre of the image. Although connection 4 is the shortest, it turns out that connection 3 achieves the highest 'red' score and is

preferred over 4. This indicates that positional information alone is insufficient to discriminate which of the possible target cells is the correct one.

Texture-based features, such as standard deviation of pixel intensity, are measured on the primary image channel which is typically the permanent stain or Hoechst marker. The integrated intensity (sum of pixel intensities for a cell) is measured individually for all image channels. The tracking algorithm requires the most informative features to be identified, which are those where the values do not change too much from frame to frame but the values for the cells in an individual frame are distributed across a wide range. These values are used to compute probabilities for cell-cell transitions, which are calculated for all cells in consecutive frames, and are subsequently stored in a matrix (see Section 3.4.1 and Figure 3.15 for an example).

The tracking algorithm relies on features remaining similar from frame to frame, where it is expected that the measured features for a particular cell do not change greatly in the interval between frames. Correlation scatter plots were produced which compared the values of the features across successive frames (see Figure 3.9).

Figure 3.8 shows characteristic condensation of the Hoechst marker during cell division (at the 60 min timepoint), followed by segregation into daughter cells. This is an essential feature, which is used to identify cell divisions, as will be shown later on. The change in area and intensity of dividing cells is illustrated in Figure 3.9, where daughter cells immediately following division are plotted in red. It can be seen that the integrated intensity drops to approximately half in the daughter cells whereas, since the daughter cells are smaller than the parent cell, the mean Hoechst intensity remains similar to that in the previous frame.

When calculating correlation scores, the dividing and non-dividing cells were treated separately. Dynamic features were plotted where the change in feature value was calculated. Good features to use in tracking are ones where the values cover a wide range while the correlation between cells in adjacent frames is good. The features are listed in Table 3.1 along with the correlation coefficients in column 2. The third column holds the R^2 value multiplied by the dynamic range of the feature, calculated as the log to base 10 of the range of values.

The features were divided into 5 groups: features derived from Hoechst

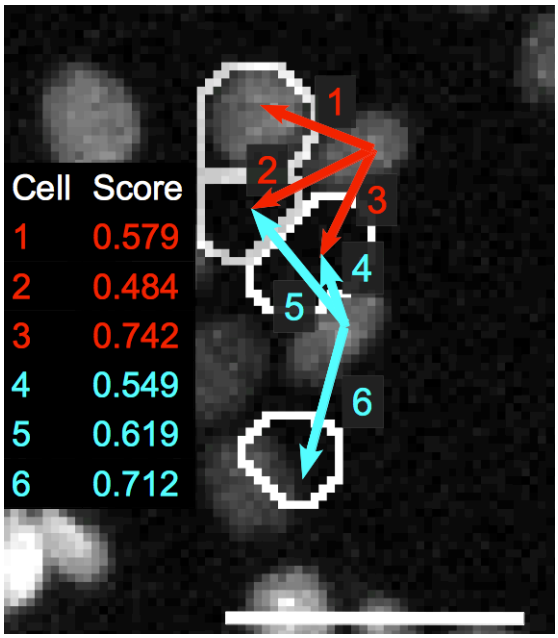


Figure 3.6: Potential ambiguity in linking cells in subsequent frames (white outlines). Arrows represent potential trajectory assignments with numbers representing the calculated score for each potential assignment. *Adapted from Downey et al. ©IEEE 2011*

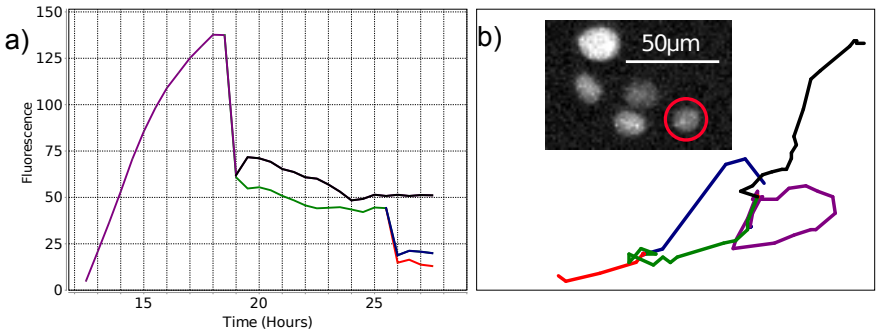


Figure 3.7: Change in fluorescence intensity during cell division. a) Drop in intensity at division. b) Motion of the cell with the sections colour-coded to match the intensity plot (a). *Adapted from Downey et al. PlosOne (2011)*

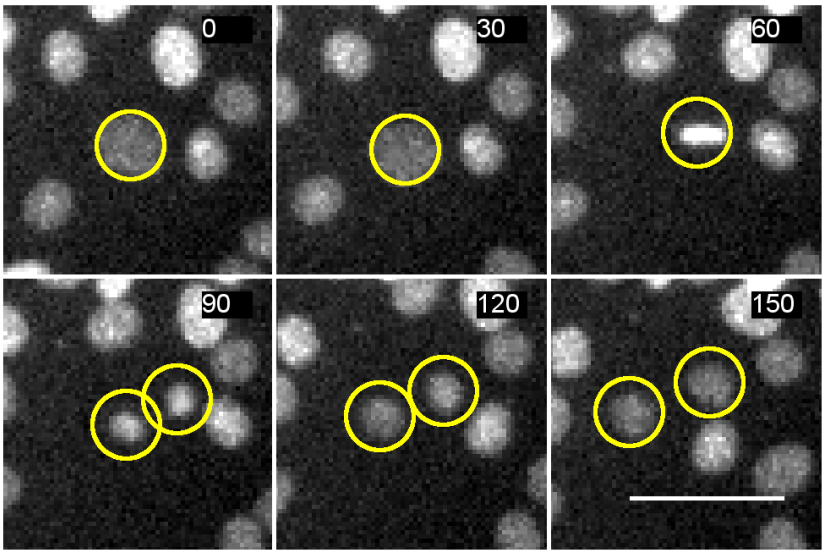


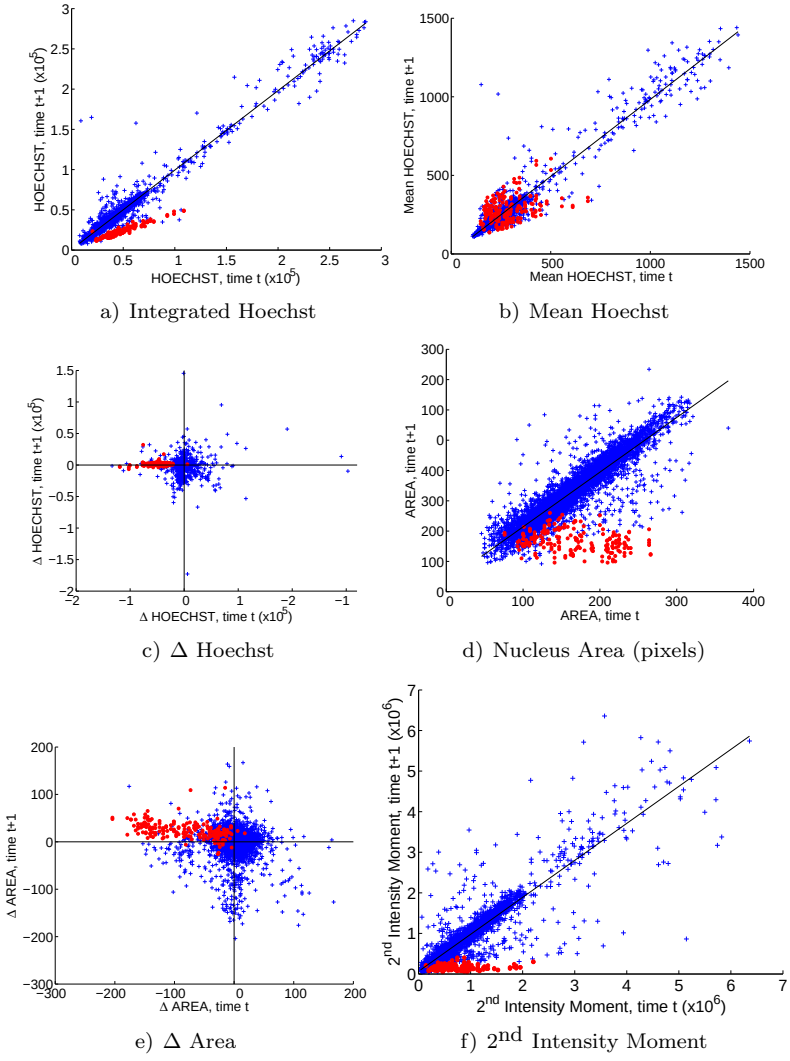
Figure 3.8: Sequence of a dividing C2C12 cell showing Hoechst enrichment prior to division. Time in minutes, scale bar: 50 micron. *Adapted from Downey et al. PlosOne (2011)*

intensity (integrated, mean and difference in intensity), GFP intensity, Shape (including area and major axis angle), Intensity Moment derived features and statistical features (including standard deviation and median). The table was sorted according to the third column and the highest scoring feature in each group was selected to be used in the tracking. The feature selection was confirmed by comparing tracking accuracies for different sets of features (see Section 5.5).

3.4.1 Developing a Multi-feature Score Based Tracking Scheme

Tracking is calculated by considering a pair of adjacent frames and attempting to match a cell in the frame t with the corresponding cell in the frame $t+1$. Where cells are present in high density, there will be a number of possible candidate cells within a small region surrounding the original cell's position in the second frame. A similarity score is calculated for each cell, indicating the likelihood of matching each nearby cell in the subsequent frame.

The similarity scores are assembled into a transition matrix, which holds the scores for each cell in a particular frame against all candidate cells in the following frame. The rows of the matrix represent cells in the current frame,



a) Integrated Hoechst b) Mean Hoechst

c) Δ Hoechst d) Nucleus Area (pixels)

e) Δ Area f) 2^{nd} Intensity Moment

Figure 3.9: Correlations of different features between consecutive frames. Tracked cells are plotted in blue. Cells that divided between consecutive frames are plotted as red circles. R^2 values are given only for very highly correlated values. a) Integrated Hoechst intensity. Non-dividing cells show a very high correlation in Hoechst between frames (blue R^2=0.97). Red cells show that Hoechst levels are halved during division (red R^2=0.90). b) Mean Hoechst intensity (blue R^2=0.94). c) Change in Integrated Hoechst. d) Nucleus area. (blue R^2=0.84). e) Change in nucleus area. f) 2^{nd} Order Intensity Moment. *Adapted from Downey et al. PlosOne (2011)*

Feature	R^2	$R^2 \times \log_{10}$(max-min)
Centre co-ordinates of nucleus	**1.00**	**2.71**
2nd Intensity Moment	**0.85**	**1.89**
Standard Deviation Hoechst int.	**0.92**	**1.64**
2nd Moment (Area*Int. normalised)†	0.80	1.60
Integrated Hoechst Intensity	**0.97**	**1.51**
Integrated GFP Intensity	**0.91**	**1.49**
2nd Moment (Area normalised)†	0.90	1.35
Mean Hoechst intensity†	0.94	1.06
2nd Moment (Intensity normalised)†	0.78	0.94
Median Hoechst Intensity	0.86	0.89
Nucleus Area	**0.84**	**0.84**
Major Axis Angle	0.20	0.71
Relative standard deviation†	0.50	0.50
Δ Integrated GFP Intensity	0.07	0.40
Axis Ratio	0.37	0.37
Δ 2nd Intensity Moment	0.04	0.30
Δ Circularity	0.18	0.18
Circularity	0.16	0.16
Δ Area	0.01	0.03
Δ Hoechst Standard Deviation	< 0.01	0.02
Δ Hoechst	< 0.01	< 0.01

Table 3.1: Measured and derived features, along with R^2 Correlation Coefficient and correlation multiplied by the order of magnitude change in the feature value. Features in bold are used in the tracking system. †Derived from other features. R^2 values are calculated using non-dividing cells only.

with columns represent cells in the following frame. Each element in the matrix holds a movement score representing the similarity in position and measured feature values between the cells. A value of 1 indicates that the position and feature values are unchanged between frames.

The individual cell-cell transition scores are calculated by measuring the features described above and calculating score contributions for each feature. Threshold values for each feature indicate how much change in feature value is allowable before the match become less favourable.

There are two separate steps toward calculating the movement scores, where individual similarity scores are calculated for each measured feature, followed by a combining step where the overall movement score for a cell-cell transition is calculated based on a weighted combination of each score.

3.4.2 Calculating Feature-Similarity Scores

Each feature score is calculated based on the difference in measured feature between candidate cells. The actual means of calculation varies depending on the feature being considered.

The movement score for an individual feature, $M(f)$, is given in Equation 3.4–3.5 where T_f is the threshold for feature f and f_t is the value of the feature for the candidate cell in frame t, α determines the steepness of the curve (the value will be obtained through optimisation, see Section 5.5.7). The sigmoid shape penalises large changes in feature value, greater than the threshold T (see Figure 3.11).

$$M(f) = 1 - (1 + e^{s(f)})^{-1} \tag{3.4}$$

where

$$s(f) = \alpha \left(\frac{T_f - [f_{t+1} - f_t]}{f_{t+1} - f_t} \right) \tag{3.5}$$

Cell Distance and Direction

The Euclidean distance between candidate cells is obtained for each pair of cells and the score contribution is calculated where the threshold value is the expected maximum distance a cell will move between frames. Similarly, where the change in angle is used as a feature, it is used directly to obtain the score from Equation 3.4, using the expected maximum change in direction as the threshold.

Area, Intensity and Texture Features

The difference in feature is calculated as a percentage change from the cell in the feature, compared to the value from the cell in frame 't'. The threshold is therefore the percentage change allowable before the score rapidly decreases.

Cell Overlap

Overlap-based tracking is used in many applications where it is not expected for the cells to move beyond the previous outline, either due to low motility or high temporal resolution [38, 40]. A version of overlap-based tracking

was implemented to allow comparison with the feature-based tracking being developed.

The movement score is based on the degree of overlap between cell or nuclei images between frames. The overlap is calculated from the intersection of the shapes describing the cell outlines (Figure 3.10).

The overlap score calculation does not take a threshold value: the score is simply the overlap-fraction measured according to Equation 3.6 or 3.7,

$$\text{overlap score} = \frac{|\text{c1} \cap \text{c2}|}{|\text{c1} \cup \text{c2}|} \tag{3.6}$$

$$\text{overlap score} = 2\left(\frac{|\text{c1} \cap \text{c2}|}{|\text{c1}| + |\text{c2}|}\right) \tag{3.7}$$

where c1 and c2 are the candidate cell outlines in successive frames. Equation 3.7 is the same formula as the Kappa Index (Equation 5.5), used in calculating the degree of congruence between segmentation output and ground truth pixel data.

The overlap score can be used as the overall movement score in the transition matrix or can be taken as another dynamic feature property of moving cells and considered when combining the individual scores below.

An overlap-based tracker is used, along with a simple 'Nearest Cell' tracker, as a benchmark to compare the multi-feature tracking method.

3.4.3 Combining the individual Feature Scores to obtain a single score for each cell

Each of the individual scores calculated above are combined into a single cell-similarity score (referred to as the 'Movement Score') for a potential cell-cell transition. Equation 3.8 takes the movement scores and weights for each feature and calculates a weighted product where the score contribution is unity if the weight is zero (i.e. the feature has no influence on the score), decreasing to the score itself when the weight $W(f) = 1$.

$$M_{ij} = \prod_{f} \Big(1 - W(f)(1 - M_{ij}(f))\Big) \tag{3.8}$$

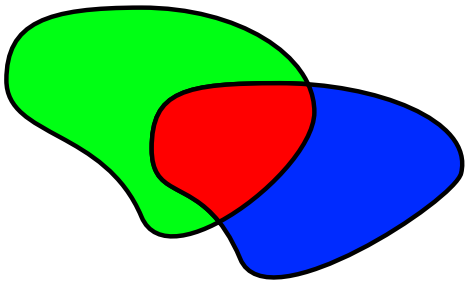


Figure 3.10: Calculating the movement score using cell overlap. Green: cell in frame 't' Blue: cell in frame 't+1' Red: Score is fraction overlap measured as the intersection of the two shapes.

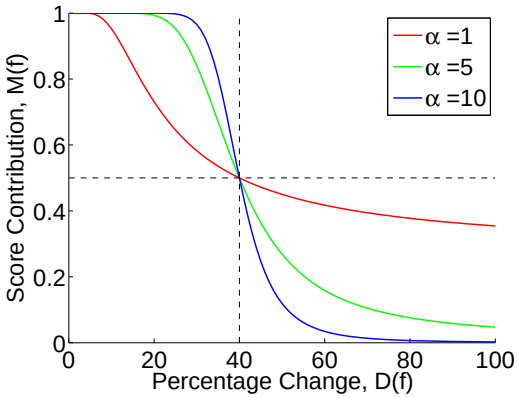


Figure 3.11: Change in feature score according to a sigmoid curve. The higher the α parameter, the sharper the decrease in score as the threshold is exceeded. At the threshold (in this example, T=40%), the score contribution drops to 0.5

3.4.4 Not all features have equal weighting in identifying cells

Each of the cell features described in Section 3.4 and table 3.1 has an associated *weight* and *threshold* which adjust the relative importance or influence of each feature when calculating the movement score.

Threshold values are obtained by performing an initial tracking followed by analysis of the change in features (see Figure 3.12 and Table 3.2). A threshold can be selected by choosing a high percentile (95th-99th) as a cut-off, which will give a value suitable for the majority of cells in the experiment. When the feature difference $D(f) = [f_{t+1} - f_t]$ is zero, the movement score contribution for the feature $M(f)$ is equal to 1, and drop to $\frac{1}{2}$ when the difference is equal to the threshold value.

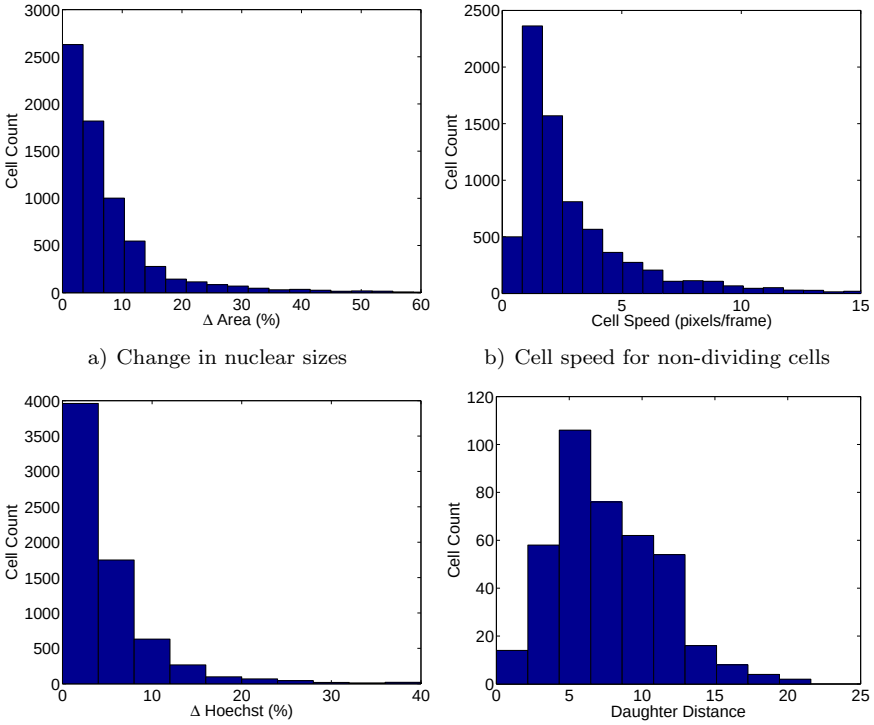
Each of the features has a weight, $W(f)$, which is proportional to the contribution towards the total movement score for the trajectory. These weights are used when the individual features are combined to calculate the overall movement score in Equation 3.8. This equation is formed such that a weight of zero means the feature has no effect on the score while a weight of 1 can bring the score close to zero for large differences.

Initial estimates of the weight values are obtained by determining the relative importance of each feature according to the strength of the correlation (see Figure 3.9, and R^2 values in Table 3.1). The features with the highest correlation values (coordinates and intensity) were assigned an initial weight of 0.9 with the other features assigned weights of 0.5.

3.4.5 Movement Scores are used to identify cells between frames

Figure 3.13 shows the Hoechst and GFP channels for a set of frames with a mean cell density of 1300 cells/mm^2 (densities typically reached 40 hours after transfection). The circled cell is moving through a region of relatively high cell density where tracking on the Hoechst channel alone would create ambiguities in identifying the cells. The accompanying GFP frames indicate that taking both channel intensities into account will aid cell identification.

The close up in Figure 3.6 gives an example of the scoring mechanism in



a) Change in nuclear sizes

b) Cell speed for non-dividing cells

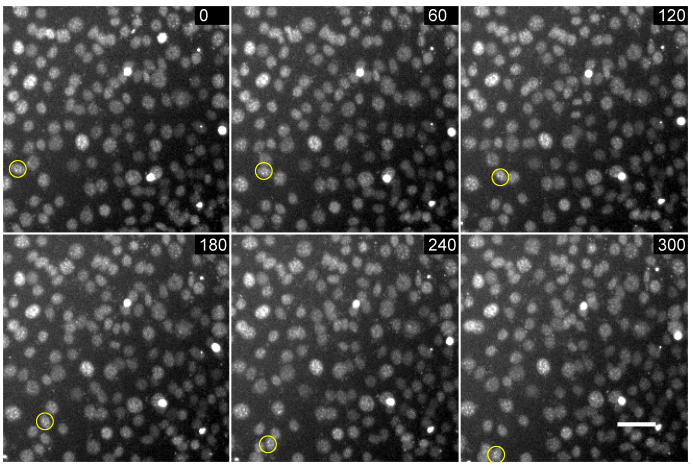
c) Change in Hoechst fluorescence

d) Daughter cell distances immediately following a division.

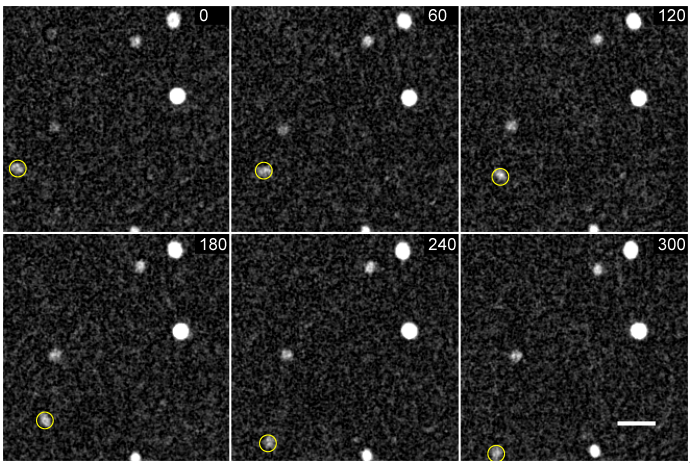
Figure 3.12: Measuring changes in features for cell-cell transitions during tracking. a) Change in nuclear areas (pixels) in adjacent frames. b) Distance moved by non-dividing cells in one frame. c) Percent change in Hoechst fluorescence for non-dividing cells. d) Distribution of daughter cell distances (in pixels) from parent cell in the frame immediately following a division. *Adapted from Downey et al. PlosOne (2011)*

Percentile	*Change in area*	*Displacement*	*Parent-daughter distance*
90	16.67	6.40	12.51
91	18.06	7.00	12.65
92	19.69	7.21	12.81
93	21.39	7.81	13.08
94	23.77	8.36	13.80
95	25.84	9.05	14.23
96	29.07	10.00	14.64
97	33.52	10.82	15.39
98	40.97	12.17	16.52
99	52.13	15.00	18.39

Table 3.2: 90–99th percentile values for change in area, frame to frame displacement during tracking, and parent-daughter distance following cell division. These values (measured in pixels) are used to select the threshold parameters used for tracking.



a) High density cell population



b) Small fraction of cells expressing GFP

Figure 3.13: Example of cell motion. The highlighted cell has been tracked through multiple frames. Scale bar is 50 microns. Numbers indicate time in minutes. a) Hoechst Channel b) GFP Channel. *Adapted from Downey et al. PlosOne (2011)*

practice. The nuclei are shown for the first frame with selected cell outlines for the following frame superimposed.

For the two selected cells, three arrows point to possible target cells in the subsequent frame, along with the calculated movement scores for each transition (6 in total). A cell receiving two inbound arrows from different cells (for example from Red 3 and Blue 4) indicate that distance alone is insufficient to accurately assign a cell-cell transition. Connection 4 is shorter than 3 but the associated movement scores are the other way round (0.549 and 0.742 respectively), indicating that Red 3 is the preferred transition from that particular cell. Blue 6, with a score of 0.712, is the highest (and therefore preferred) outbound transition from the 2^{nd} cell.

3.4.6 Converting scores into tracking is a Global optimisation problem

During tracking, the feature similarity scores are stored in a matrix where the the columns represent cells in the current frame and the rows represent cells in the following frame. Each entry in the matrix is the similarity score calculated between the 'row' and 'column' cells (see Figure 3.15 for an example).

Different methods may be used to assign the individual trajectory steps based on the constructed matrix. These methods either use the matrix representation or a graph-based approach to obtain a solution (either locally or globally optimum). The graph-based approaches rely on the representation of the potential tracking connections as illustrated in Figure 3.14.

3.4.7 A basic approach assigning highest scores is used as a baseline

A very simple approach, where the highest scoring transition for a cell is assigned as the next link, may be used as a benchmark to test the feature-based tracking. This method does not attempt to calculate an optimum solution to the tracking but is a crude 'greedy' algorithm where each cell picks the highest transition without consideration for whether the target cell would be more favourable if assigned elsewhere. The first cell (in the first frame) to 'choose' a particular target cell in the second frame will remove the target cell from consideration by

all other cells. If the target cell with the highest score has already been assigned then the cell with the next highest transition is selected.

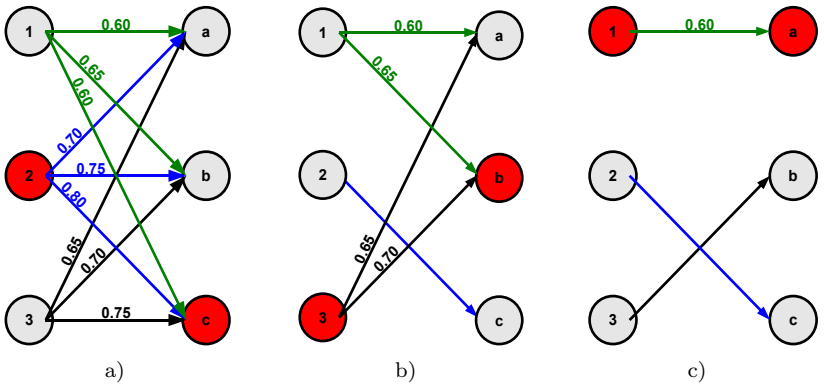


Figure 3.14: Demonstration of three iterations of the assignment step. 1, 2 & 3 represent three cells at time t, a, b & c are three cells at time $t+1$. Numbers on arrows indicate movement scores. a) The highest scoring link between 2→c is selected. b) Links to and from cells 2 & c are removed. The highest scoring link 3→b is selected. c) Links involving cells 3 & b are removed, leaving 1→a. *Adapted from Downey et al. PlosOne (2011)*

3.4.8 A global solution using the Hungarian Matrix Minimisation Algorithm

The method of assigning trajectories may be replaced with the Hungarian Algorithm [91,92], while retaining the initial matrix calculation. The Hungarian Algorithm is based on minimising the summed scores of assigning each row to a unique column (see Section 3.2.3). The scores are converted into the required form simply by subtracting from 1.

The algorithm requires a square matrix, which is only the case if the cell count is equal in consecutive frames; therefore an additional step is required to pad the matrix where there are different numbers of cells in adjacent frames. Any rows or columns required to square the matrix are filled with padding with cost = 1 to avoid the padding entries being assigned to existing cells unless all other lower values have already been assigned (see Figure 3.15 for an illustration of the process).

Cell ID	1	2	3	4
5	0.6	0.7	0.65	0.3
6	0.65	0.75	0.7	0.4
7	0.6	0.8	0.75	0.6

Cell ID	1	2	3	4
5	0.4	0.3	0.35	0.7
6	0.35	0.25	0.3	0.6
7	0.4	0.2	0.25	0.4
100	1	1	1	1

Figure 3.15: Construction of the matrix for the Hungarian Algorithm. The original matrix is on the left, with 4 cells in the first frame and 3 cells in the 2^{nd}. The modified matrix is on the right, with scores replaced by (1-score) and the extra padding indicated by shaded squares, where cell 100 is the extra 'dummy' cell added to ensure the matrix is square.

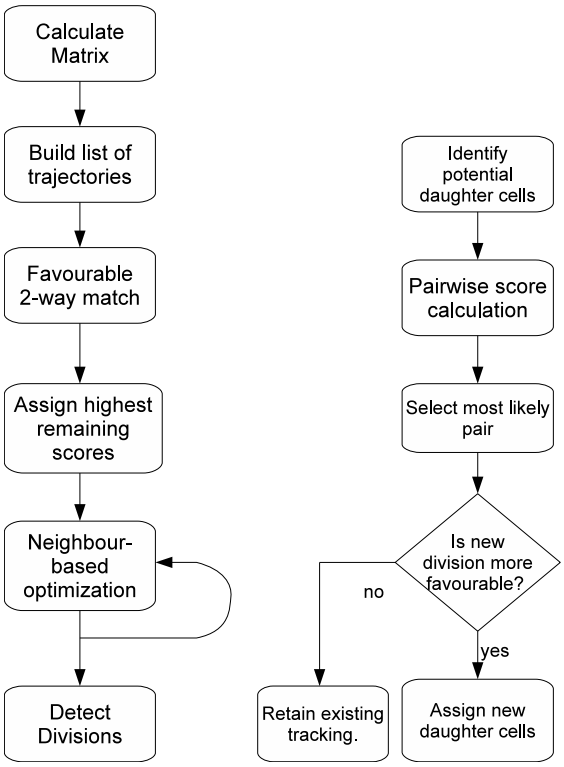
3.4.9 Co-Operative Greedy Algorithm is a rapid approximation to a global solution

This method is an extension of the Simple Highest Score tracking. Assigning movements is a four-stage process (see Figure 3.16a). The first step builds a list of potential target cells in the adjacent frames according to the movement scores in the transition matrix. Each cell holds a list of highest scoring cells in both the forward and backward directions ($t \rightarrow t+1$, i.e. those cells in the following frame which it may point to, and $t \rightarrow t-1$, i.e. those cells in the previous frame which point to it, respectively).

The second stage assigns a trajectory if the highest scoring forward transition agrees with the highest scoring inbound transition of the target cell at $t+1$ (see Figure 3.14). Step 2 is performed repeatedly until all such transitions have been assigned. The third step completes any remaining links by assigning the highest forward pointing transition.

The final step optimises the tracking by calculating the sum of transition scores for each frame. If two cells share potential targets (such as cells 1 & 2 each pointing to both a & b in Figure 3.14), a new pair of transition scores is calculated based on exchanging the trajectories. The new trajectories are retained if the exchange improves the total score.

While the Hungarian Algorithm is a well established solution for this problem, the new algorithm suggested here has a number of advantages, including execution speed. The graph-based implementation of the Co-Operative Greedy algorithm allows assignment of cell divisions to be performed by examining the cell neighbourhood, as described in Sections 3.4.10–3.4.11.



a) Tracking b) Detect Divisions

Figure 3.16: a) Tracking flow chart. The matrix is described in Section 3.4.6, the favourable 2-way match is illustrated in Figure 3.14, and the optimization is described in Section 3.4.9. b) Expanded flow chart for the Detect Divisions module. The individual steps are described in Section 3.4.11 *Adapted from Downey et al.* ©*IEEE 2011*

3.4.10 Constructing Lineages

So far the tracking has only considered simple links of cells. The next step is to add branches to the tracks to add cell division information.

The large frame intervals used in the experiments can lead to difficulties in identifying cell divisions. The M-phase (mitotic phase) of the cell cycle is relatively brief and can occur between frames, therefore the change in appearance of the nucleus during M-phase cannot be relied upon to detect divisions. Additionally, direction of travel of the daughter cells following division could not be used as there was a wide spread in angles, and regular changes in cell direction meant cell trajectories could not be traced backwards to associate daughter and parent cells.

3.4.11 Identifying Daughter Cells using a characteristic change in Hoechst fluorescence

The first step in identifying cells which may have divided involves comparing dynamic changes in measured features in particular characteristic changes in intensity and nucleus area (Figures 3.9c & 3.9e) which both decrease by at least 25% during cell division (Figure 3.7). In the frames immediately following division, the daughter cells will be close to the last known position of the mother cell (see Figure 3.12d).

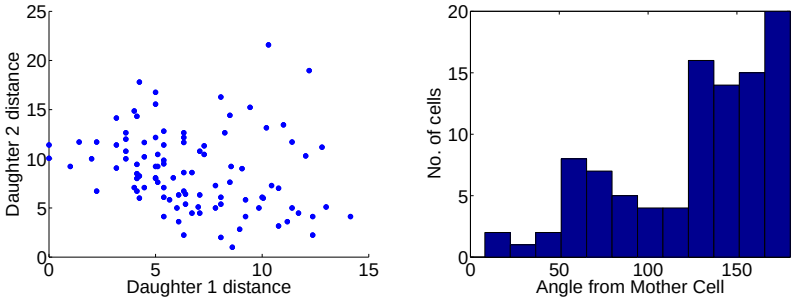
After a suspected division event has been detected, it becomes necessary to locate the daughter cells. In a sparse field of cells, daughter cells will be easy to identify since there is a high likelihood of them being the only cells in the region. However with high cell densities, separating new daughter cells from existing cells becomes more problematic.

Most daughter cells are found within 10–15 pixels of the mother cell but there is no correlation (R^2=0.05) between the distances of the two daughters (Figure 3.17a), so distance alone cannot be used to reliably identify daughter cells.

The angle of the daughter cells is summarised in Figure 3.17b. The angle is measured between the daughters and the last observed position of the mother cell, such that 180° refers to cells moving in the opposite direction to each other. A lot of the spread in angles is due to the movement of the cells both before

and after division: the low temporal resolution is insufficient to capture the immobile dividing cells.

While the bulk of the divisions lead to daughter cells moving roughly diametrically apart (52% of divisions produce daughter cells within 45° of a common axis, 74% are within 90°), there is still sufficient variability to be wary of using angles to locate cells, especially if there is a significant number of other cells within the 'search area'. The 'division angle' becomes more significant when applied to non-motile cells which are in possession of a major axis, such as *Sz.pombe* discussed later.



a) Distance of daughter cells from mother cell

b) Angle of daughter cells

Figure 3.17: Daughter cell position relative to mother cell, measured using Reference Standard 2. a) Distance of the two daughter cells (in pixels) from the last measured position of the mother cell. b) Angle of initial daughter cell motion, in relation to mother cell.

There is a high correlation ($R^2 = 0.95$) between the integrated intensity of the parent cell and the sum of the daughter cell intensities (sum of daughter intensities is 100±1.5% of parent intensity, where the error is the standard error of the mean for n=100 cell divisions). There is also a high correlation between the integrated intensities of the two daughter cells ($R^2 = 0.92$) where the mean difference between daughter cells is 6.0±0.5%.

Similarly, the areas of daughter cells are closely correlated ($R^2 = 0.95$) where the mean difference between daughter pairs is 12.6±1.0% and the mean cell area is 45.8 pixels. The average total sum of daughter cell areas is 110±4.3% of the parent cell area. There were some cases where a daughter cell was larger than the final measured area of the parent cell due to the long frame interval and chromatin condensation occurring during the previous frame.

The 'Locate Divisions' step of the tracking module identifies potential daughter cells by examining all cells within the distance threshold of the parent cell position. These cells are considered a pair at a time and a similarity score is calculated using Equation 3.4 based on intensity and size only. Since there was a larger variation in the area size, the area was weighted less than the intensity (weights=0.25 and 1 respectively).

The highest scoring daughter pair is then compared to the parent cell by re-evaluating Equation 3.8 using a 'composite cell' where the area and integrated intensities are the sums of the values for the potential daughter cells, again using weights of 1 and 0.25 for intensity and area. If the movement score calculated from this combined cell is greater than the movement score for the original tracked cell, the new daughter cells are assigned to the parent.

This approach relies on the tracking being able to be maintained in the interval leading up to the division. The increase in brightness due to chromatin condensation (Figure 3.8) does not interfere with tracking since it is the integrated intensity, not the mean intensity, which is used in the calculations. The cell area feature weight needs to be lower than the intensity weight to prevent the decrease in area from lowering the tracking score and causing the tracking to fail.

3.5 Conclusions

Several segmentation and tracking methods were developed which will be evaluated in Chapter 5. Cell motion was studied to determine features which may be utilised by the tracking algorithm and it was discovered that the cells within the two timecourse experiments could be divided into two populations: one represented by persistent cells, the other by cells exhibiting a more random motion.

In Chapter 5, the segmentation and tracking methods developed here are compared to third-party methods, including state of the art implementations. Several basic tracking methods are also proposed which use simplified methods and are intended to provide a baseline against which the more sophisticated methods are compared. Two different approaches are used in evaluating the segmentation methods, with both manually annotated cell images and

synthetically generated cell images being utilised.

The software has been adapted to work with *Schizosaccharomyces pombe* is described in Chapter 6 and a further application, utilising FUCCI cell cycle markers [95] and two additional cell types, is given in Chapter 7.

Chapter 4

Measuring *Msx1* expression

This chapter describes the data analysis performed on timecourse experiments performed on cells expressing GFP driven by different versions of the *Msx1* promoter. Five different constructs were prepared and transiently transfected into C2C12 cells as described in Chapter 2.

4.1 MSX1 is a Transcription Factor involved in stem cell differentiation

The MSX1 protein is a member of the homeobox family of transcription factors involved in vertebrate craniofacial and muscle development. Homeobox genes are important in positional control during development of the embryo [96]. There are several *Msx* and *Dlx* genes which are part of the same homeobox family. These contain a highly conserved sequence of amino acids known as the homeodomain, which has been demonstrated to bind to DNA [97, 98].

MSX1 is involved in regulating pluripotency of mesenchymal stem cells [99]. Expression of *Msx1* during embryogenesis maintains progenitor cells in their undifferentiated state (Figure 4.1) by upregulating cyclin D1 [100] which is one of several factors controlling progression through the cell cycle. For example over-expression of MSX1 has been observed to slow the cell cycle in ovarian cancer cell lines [101]. Mutations in the *Msx1* gene lead to cranial and dental defects [102] including cleft palate [103] and familial tooth agenesis [104].

In the absence of growth factors the cells will begin to differentiate. Normally this process is irreversible but certain species, such as members of the Urodele family which includes Salamanders, can reverse the process to regenerate lost

limbs [105]. The dedifferentiation process has been replicated in-vitro in mouse cells. Ectopic expression of MSX1 can result in muscle cells reverting to a form which is then capable of re-differentiating into different types of cells [106].

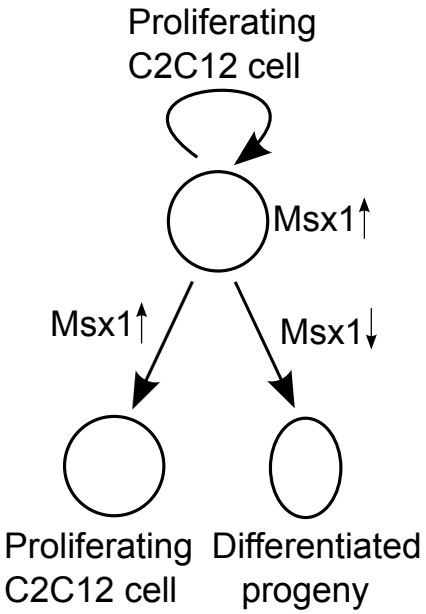
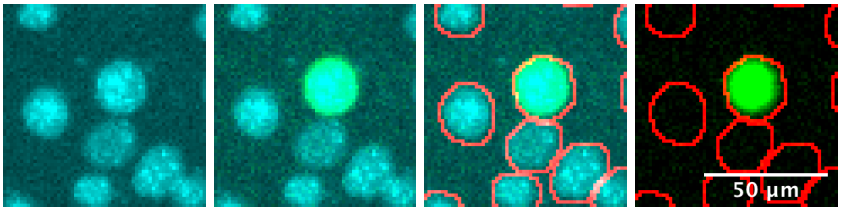


Figure 4.1: Differentiation or proliferation of C2C12 cells. Expression of *Msx1* maintains stem cells in the undifferentiated state.

4.2 MSX1 Fluorescent Imaging

The activity of reporters containing transcriptional control elements from the *Msx1* gene was measured. Several control elements have been identified [63], and a key objective for the development of our analysis method was to quantify the role these elements play upon transcription rates by using fluorescent reporters. Expression levels are proportional to the amount of reporter protein, provided the measured intensity is within the recommended range of the imaging system and saturation of the signal is avoided. Fluorescent reporters were modified by the addition of a Nuclear Localization Sequence (NLS) [107] which is a short sequence of amino acids which, when part of a protein, is a signal to a molecular chaperone and leads to post-translational relocation of the protein into the nucleus. A GFP molecule which contains an NLS will therefore appear colocalised with the nucleus and segmentation based on the Hoechst nuclear

stain can thus be used to measure reporter intensities in the nucleus (Figure 4.2).



a) b) c) d)

Figure 4.2: Segmentation of C2C12 cells on Hoechst channel. a) Hoechst only. b) Hoechst and GFP. c) as b with segmentation mask overlaid d) GFP with segmentation mask obtained from Hoechst channel. *Adapted from Downey et al. ©IEEE 2011*

4.3 Analysing the MSX1 Fluorescence Measurements

The software was used to study the expression of cis regulatory module (cRM) promoter driven GFP and to measure the partitioning of protein between daughter cells for dividing C2C12 cells. Transient transfections were performed with reporters containing four different *Msx1* transcriptional regulatory regions (A-D) upstream of the *Msx1* promoter and the promoter alone [63].

Since a transient transfection was used, not all of the cells will take up the plasmid. Only the GFP-expressing cells need to be followed, so a method is required to detect the onset of gene expression. While this could be done visually an automatic method would be preferable when a large number of cells are involved. Similarly an automated method would be preferable for locating cell divisions and constructing cell lineages.

The cells were seeded at relatively high densities of 4×10^5 cells per ml since the C1C12 cells will naturally differentiate once they reach confluence levels.

4.3.1 Identifying Dividing Cells

Candidate cells for analysis were identified by following the changes in fluorescence intensity and selecting the cells where a division occurred while GFP was actively being expressed.

The onset of gene expression was detected by smoothing the fluorescence intensity values using a moving average which removes small frame to frame variations in the imaging and measurement steps. A suitable window size for the moving average (20 timepoints) was determined empirically to give a smooth appearance to the curve. The actual onset of expression was taken to have occurred when there had been several consecutive increases in the smoothed fluorescent value.

The list of GFP-expressing cells was filtered further by removing cells which did not divide. Automatic division detection is described in more detail in Section 3.4.10 and the data structure is described in the Appendix (A.3.4). The cells are stored as a tree structure so the tree branches when divisions occur.

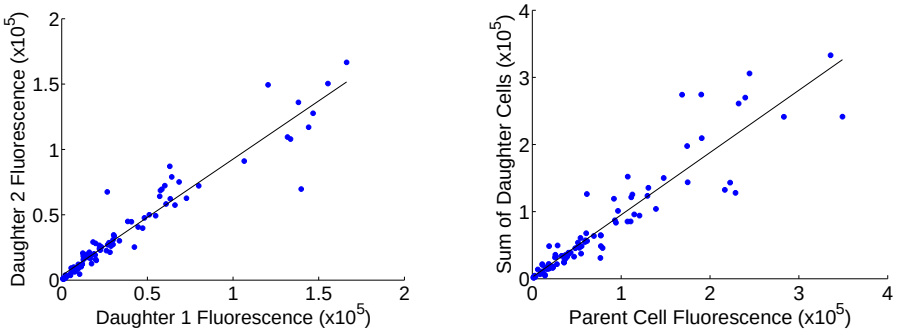
All of the cell divisions were manually checked to confirm that division occurred while the GFP was being expressed and to ensure that the correct daughter cells had been selected. Any deviations in the tracking or lineage construction were corrected at this stage.

4.3.2 Obtaining Fluorescent Timecourse Data using the LineageTracker Software

The software stores cells as individual objects with properties which include measured features and links to the next and previous cells if a cell belongs to a tracked lineage. The timecourse data for individual cells or lineages are exported from the LineageTracker software as text files suitable for use in spreadsheets or mathematical software such as Matlab. Each row in the text file holds the fluorescence intensities of all measured image channels. Additional rows hold the corresponding data for any daughter cells.

4.3.3 Comparing Protein Levels in Daughter Cells for different Msx1 Promoter Constructs

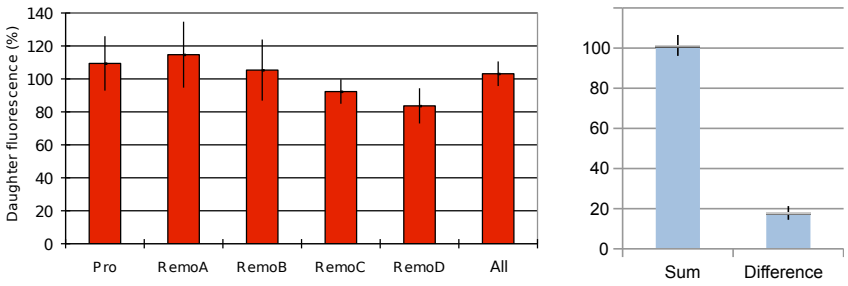
The images to be analysed were obtained from 7 different Cellomics experiments performed by Keith Vance during 2007. Five different constructs were tested where different *Msx1* regulatory modules were driven by the murine *Msx1* promotor (see Sections 1.3 and Figure 1.1). Frame intervals were 30 or 60 minutes. A total of 96 divisions were identified and the fluorescence activity of mother and daughter cells was measured in the frames either side of the division.



a) Fluorescence of daughter cells following division

b) Fluorescence of mother cell plotted against sum of daughter cells

Figure 4.3: Integrated fluorescence intensity. Parent cell was measured in the frame immediately prior to division and daughter cells were measured in the following frame. a) Daughter GFP fluorescence (R^2=0.92) taken from the 5 *Msx1* cRM constructs. b) Mother cell fluorescence and total daughter fluorescence (R^2=0.86). *Adapted from Downey et al. PlosOne (2011)*



a) Total daughter fluorescence for 5 different cRMs

b) Comparison of daughter cell fluorescence.

Figure 4.4: GFP Fluorescence of daughter cells, calculated as percentage of the mother cell intensity. Error bars are calculated at the 95% confidence level. a) Breakdown of sum of intensities for the 5 different *Msx1* cRM constructs. b) Mean sum of daughter fluorescence and difference between daughter fluorescence for all cRM constructs. *Adapted from Downey et al. PlosOne (2011)*

The partitioning of protein between daughters and the total fluorescence recovery are summarised in Figure 4.3. The correlation coefficients are R^2=0.92 & 0.86 respectively. The high correlation in the partitioning means that for all the different *Msx1* reporter constructs driving GFP expression we find that fluorescence is symmetrically distributed in the two daughter cells with a high degree of accuracy, ensuring that in most cases MSX1 levels are maintained during cell divisions to prevent differentiation.

4.3.4 Asymmetric Divisions are uncommon

In the sample of 96 divisions, there was only a single instance of asymmetric division where one daughter cell contained significantly more fluorescent protein that the other, shown in Figure 4.5.

An earlier study using *E.coli* [23] observed that the majority of divisions ($>$ 85%) produced daughter cells with a volume difference of less than 5%, and that differences in daughter cell size was the primary cause of asymmetric division. In the single case measured here the GFP was confined to the nucleus. The segmented areas of the daughter nuclei were almost equal (52 & 53 pixels respectively) as were the Hoechst intensities (4600 & 4400 as measured after background subtraction).

Another indicator of asymmetric division could be cell cycle timings, where differentiated cells might have different generation lengths. While the data used here has insufficient cells with multiple divisions, other studies [108] have failed to observe a significant difference in generation times, suggesting that asymmetric divisions are infrequent *in-vitro*.

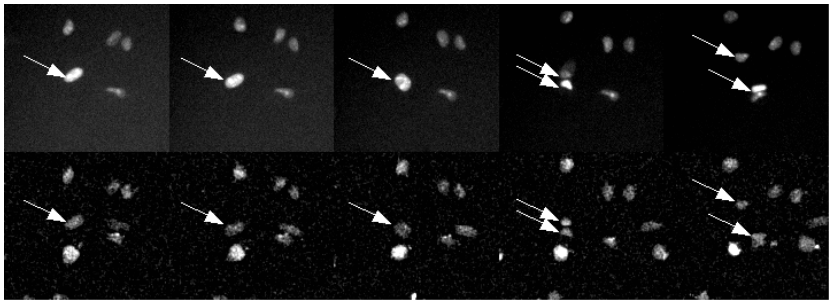


Figure 4.5: Example of an asymmetric division (arrowed). Top: GFP fluorescence, Bottom: Nuclei stained using Hoechst. The fluorescence was divided between the daughter cells in the ratio 28:72 for GFP and 51:49 for Hoechst.

4.4 Conclusions

An interactive framework was developed to aid analysis of cell data and this was used to extract fluorescence measurements for a number of cells which were subsequently used to study the partitioning of fluorescent protein during divisions. It was observed that in all but one instance, the protein was equally distributed between daughters and the fluorescence recovery following division was consistent.

Chapter 5

Performance and Validation of the LineageTracker Software

This chapter describes the accuracy measurement and performance validation on the segmentation and tracking methods. The tracking validation was performed using two manually tracked C2C12 data sets whereas the segmentation validation was based on a combination of manually annotated cell images and a set of synthetic images generated using third-party software.

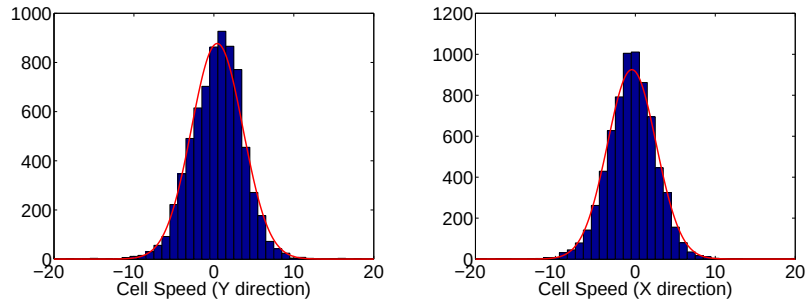
5.1 A Statistical Analysis of Cell Motility

The C2C12 mesenchymal cells are highly motile muscle precursor cells. During migration, cells release chemokines which in turn attract other cells and encourage motion in a particular direction [109]. Cells exhibit a range of behaviours ranging from random motion to migratory travel depending on their local environment.

An analysis of cell motion was performed (similar to [110]) to investigate whether any additional movement parameters could be used to improve the tracking. The manually validated tracked sets (see Section 5.5.1) were analysed.

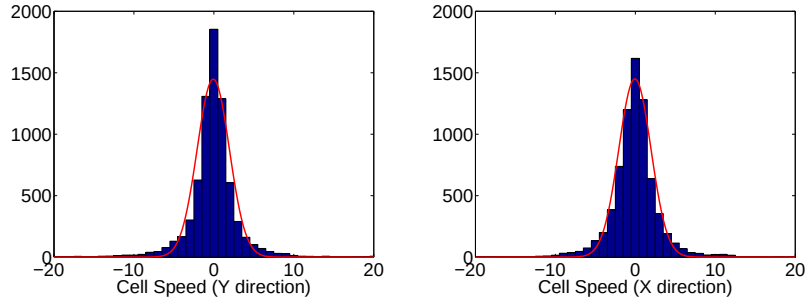
Taking cell motion in a single axis first, the speeds appear to follow a Gaussian distribution (see Figure 5.1). The mean was close to zero (± 0.45 for both x- and y-axes), suggesting there was little or no overall drift in any particular direction. The quality of fit ($R^2 > 0.99$) was good for all the speed

distributions.



a) Reference Standard 1: Cell speed in the y direction

b) Cell speed in the x direction



c) Reference Standard 2: Cell speed in the y direction

d) Cell speed in the x direction

Figure 5.1: a & b) Cell displacements from Reference Standard 1, with a Gaussian Distribution superimposed. c & d) Cell displacements for Reference Standard 2. Displacements are in pixels per frame, where 1 pixel = 1.36 μm.

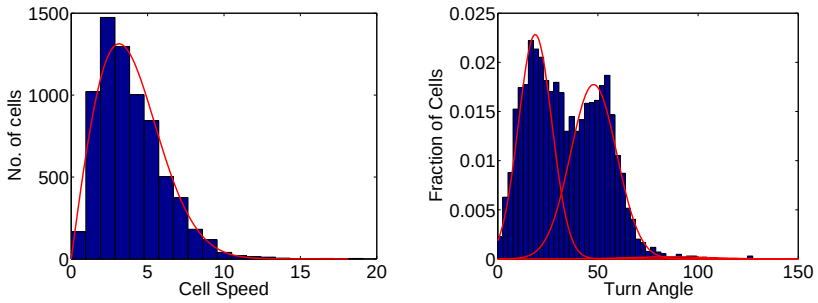
The peak in Figure 5.1a is offset slightly from the mean value so the Skewness values were calculated for the distributions, which measures the asymmetry around the mean where a normal distribution has skewness of zero. Taking Reference Standard 1, the skewness of the 'y' speeds is -0.20, which is of greater magnitude than the skewness of the 'x' direction which is < 0.01. Since cell motion is not truly random but is influenced by nearby cells, it is expected that the distributions may occasionally deviate from normal.

When a particle moves in two or more axes and the speed in each axis follows a Gaussian distribution, the particle velocities will be given by $v = \sqrt{\Delta x^2 + \Delta y^2}$ and will follow a Rayleigh distribution. This is shown in Figures 5.2a & c. The speed distributions were fitted to the Rayleigh distribution formula (given in Equation 5.1) and the quality of fit was calculated.

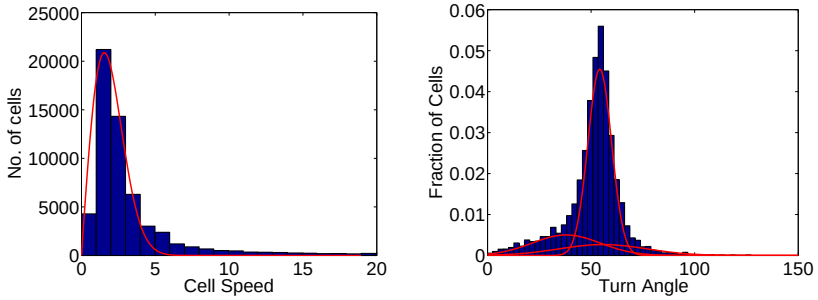
$$R(x) = \frac{x}{v^2} e^{\frac{-x^2}{2v}} \tag{5.1}$$

Reference Set 1 exhibits a slightly better fit ($R^2 = 0.96$) than Set 2 ($R^2 = 0.94$). Since a true Rayleigh distribution will only be obtained when the x & y components of the velocity are both normally distributed, any slight deviation from normal will be reflected in the quality of fit.

The majority of the cells ($> 95\%$) have velocities below 10 pixels per frame. This result could be used to determine the optimum 'Displacement Parameter' for the tracking algorithm.



a) Cell speeds from Reference Standard 1 b) Change in direction for Reference Standard 1



c) Cell speeds from Reference Standard 2 d) Change in direction for Reference Standard 2

Figure 5.2: Measurements from Reference Standard data sets. a) Cell speeds fitted with a Rayleigh distribution. b) Change in cell direction. c) Speeds from Reference Standard 2 fitted with a Rayleigh distribution. d) Change in cell direction for Reference Standard 2.

Changes in direction were calculated from 0 to 180° so a 90° angle could be either a left or right turn, as only the degree of turn was considered. The results are displayed in Figure 5.2b & d. The two distributions are quite different, with Set 1 having more cells with lower turn angles than Set 2. A visual

inspection of the two timecourse experiments shows large numbers of cells in Set 1 moving in roughly straight lines. A Gaussian mixture model was applied to the angle distributions and both datasets were found to be built from a mixture of distributions with similar mean values: $< 40°$, 50–54°and a much smaller component at $\geq 60°$, Set 1 had a greater contribution from the lower angle range.

The two angle distributions present in Set 1 suggest that there are two distinct subpopulations where half of the cells have persistent motion whereas the other half are exhibiting a random walk. The cell displacement histograms do not show a similar division, suggesting that cell speed is not dependent on whether the cell is in migration or random motion.

A brief analysis of the cell motion was performed, comparing the motion with random walks, summarised in Figure 5.3, where the mean total displacement squared is plotted against the number of steps.

Motile cells in a uniform environment with no chemoattractants, when viewed from above, will appear to be free to move in 2 dimensions. If the cell could change speed or direction without constraint, the motion would resemble a 'random walk' where the cells move in a random direction, with no correlation with previous motion. A traditional random walk with fixed step size and random direction would appear as a straight line on a plot of steps taken (or total distance covered) against distance squared (as shown in Figure 5.3a) whereas straight line motion appears as a quadratic increase as the distance will increase at a constant amount with each step.

The motion of the tracked sets is displayed in Figure 5.3b. The blue trace represents the first Reference Standard and this closely resembles the straight line trace in Figure 5.3a which is consistent with the low turn angles present in that set. The second set, shown in red, starts to deviate from the 'straight line' motion and the curve flattens out as cell motion either becomes more random or motion becomes constrained. Mean cell speed drops very slightly during the course of the experiment, a maximum of 3.2 pixels per frame near the start to 2.6 pixels per frame at the end but this would not be sufficient to cause such a drop in the curve. A simulation of a decreasing-step random walk is given in Figure 5.4a where the step size drops from 3.5 to 2.5 pixels per step but this

does not recreate the walk profile from the experiment.

A second simulation increases the angle throughout the run, and this produces a result much closer to the experimental results. One theory for this is that as density increases, persistence decreases since the cells are unable to move in straight lines (supporting data for this is given in Appendix B.5). Reference Standard 1 was of shorter total duration so may not have run for sufficient time to exhibit the same behaviour. An alternative visualisation of the random walk analysis is presented in Appendix B.6 using Distance-Pathlength heatmap plots.

5.2 Measuring Segmentation Performance

The segmentation step aims to separate images into different regions, which in the case of cell detection can be referred to as Cells (or Foreground) and Background. Each region has both position and area. A measure of accuracy should ideally take both of these into consideration. Sections 5.3 & 5.4 describe two different approaches which tackle the position and area measurements separately.

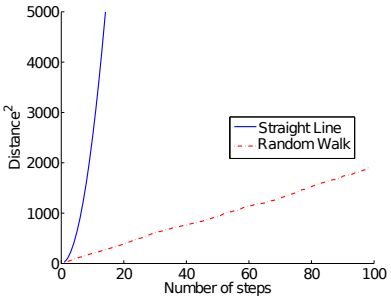
Many segmentation methods can be fine-tuned to work at a particular cell density but often struggle to work at different densities. Such conditions will be commonly encountered in time-series experiments of proliferating cells.

The segmentation of the Seeded Growth segmentation method (described in sections 3.1 and 3.1.3) is compared with a commercial solution, Cellomics, along with CellProfiler, which is another open source cell image analysis package.

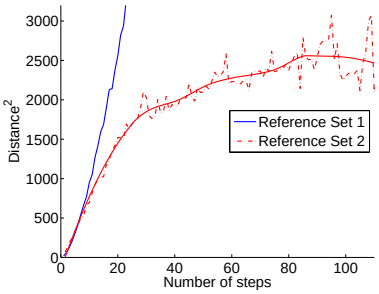
5.3 Quantifying Position Accuracy of different methods

The segmentation was first evaluated against a Reference Standard based on 4 frames selected over a 48 hour period of a single experiment (Figure 5.5). The frames were chosen to represent a range of cell densities, from 437–740 cells per image (equivalent to 902–1507 cells/mm^2). The upper limit is equivalent to 25–30% total area covered by nuclei, as measured using the Hoechst channel, which approximately corresponds to 90–100% cell confluency.

The positions of the nuclei were marked using the CellCounter plugin in

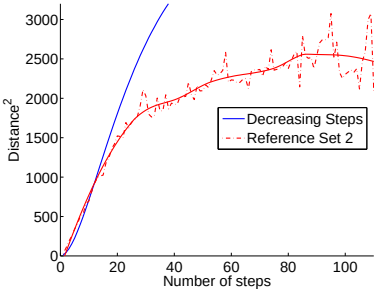


a) Random Walk and Straight Line motion

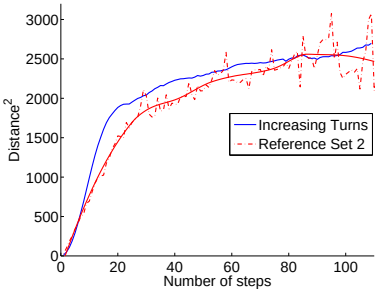


b) Reference Standard tracked sets

Figure 5.3: Distance plots for random walks and Reference Standard tracked sets. The x axis is the number of steps taken while the y axis is the square of the distance moved from the original location. a) Blue: fixed step size and constant direction. Red: random walk with fixed step size and random change of direction every step. b) Blue: Reference standard 1 (10 minute time intervals) Red: Reference standard 2 (30 minute intervals), dotted line is measured data, solid line is smoothed moving average.



a) Reference Standard 2 against Decreasing Walk



b) Reference Standard 2 against Increasing Turns

Figure 5.4: Random walk models simulating motion of Reference Standard 2. a) Length of step decreases from 3.5 to 2.5 during the run, while angle change remains constant. b) Step size remains constant at 3.5 but turn angle increases gradually during the run.

ImageJ, which allows the positions of cells to be marked and subsequently exported as a spreadsheet file for use elsewhere. Each cell can also be marked as being of a particular type and this was used to identify cells as belonging to one of four categories: 'genuine' cells, edge of frame, touching another cell, and noise or debris.

The marked cells belonging to the first two categories formed the ground truth data and were compared to the cell positions as obtained from the segmentation methods (full details of the algorithms are given in Appendix B.2.

5.3.1 Results of the Position Comparisons

To determine positional accuracy, we define a cell detection as true positive when the position is within 1 radius of a ground-truth cell. Cells which cannot be matched are classified as false positive. Cells in the ground truth data set which remain unassigned are classified as false negative.

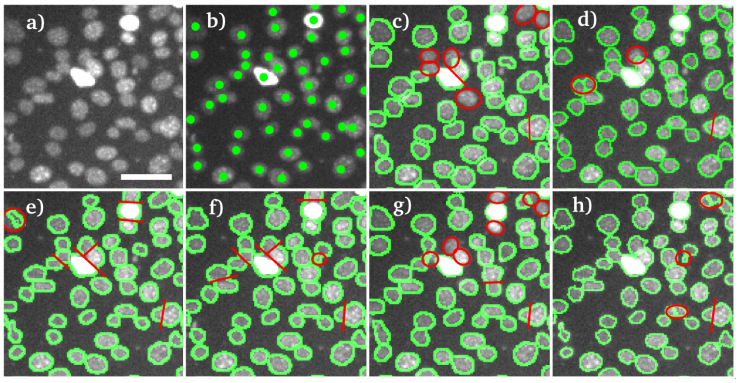


Figure 5.5: Segmentation of cell nuclei. a) Original nuclei (scale bar 50 micron) taken from the Reference Standard data set, cell density 1150 cells/mm2. b–h) Nuclei with segmentation examples overlaid. Ellipses indicate segmentation errors. Lines indicate unresolved clusters of cells. b) Manually marked cell position. c) Cellomics segmentation. d) Seeded Growth. e) Global Threshold. f) Local Threshold. g) Scaling Index. h) CellProfiler. *Adapted from Downey et al. PlosOne (2011)*

A selection of segmentation errors encountered during the evaluation, including over- and under-segmentation, are shown in Figure 5.5. None of the methods outperformed all others at all cell densities (see Figure 5.6, summarised in Table 5.1). The Seeded Growth, Scaling Index algorithms and CellProfiler perform slightly better regarding false negatives, which are consistently below

the Cellomics and threshold-based methods. However, these threshold based methods (Cellomics, Global and Auto Threshold) yield lower numbers of false positives (consistently below 1%), compared to the Scaling Index and the CellProfiler Background Adaptive method.

	Cells	%Found	% Not Found	% False Positive	% Missing (Edge of frame)
Reference Standard	2365	n/a	n/a	n/a	n/a
Seeded Growth	2179	92.14	7.45	0.71	0.41
Global threshold	1985	83.93	13.10	0.19	2.97
Scaling Index	2153	91.04	7.36	0.94	1.61
Auto Threshold	2009	84.95	12.46	0.26	2.60
CellProfiler	2161	91.37	6.65	2.19	1.98
Cellomics	1994	84.31	12.75	0.26	2.94

Table 5.1: Summary of Segmentation Results, listing percentage of cells detected, missed or incorrectly detected. Cells at the edge of the frame are ignored in many segmentation methods so weren't counted as failures and have been listed separately.

The increasing numbers of missed cells at high cell densities indicate there is currently no reliable method that can work in an accurate unsupervised manner when cultures become confluent. The Seeded Growth method was chosen for use as it provides a good balance between false positives and negatives for different cell densities. The Scaling Index method provided similar detection rates but required the radius and threshold parameter to be carefully chosen otherwise accuracy would decrease.

5.4 Quantifying Pixel Partitioning Accuracy using artificial Ground Truth Images

An accurate quantification of fluorescence will require an accurate identification of the outline or extent of the cell or nucleus, since the fluorescence intensity will be spread over this area. The pixel-accuracy of segmentation was measured using artificial ground truth images created by the Simcep software [111]. The pixel accuracy measurement was calculated based on the Precision & Recall values [112] which were combined into an F-Score to obtain a single value to represent the accuracy.

Artificial ground truth images created using Simcep were used to test the pixel-accuracy of the segmentation methods. The pixel ground truth images obtained from Simcep were quantitatively compared with the segmentation

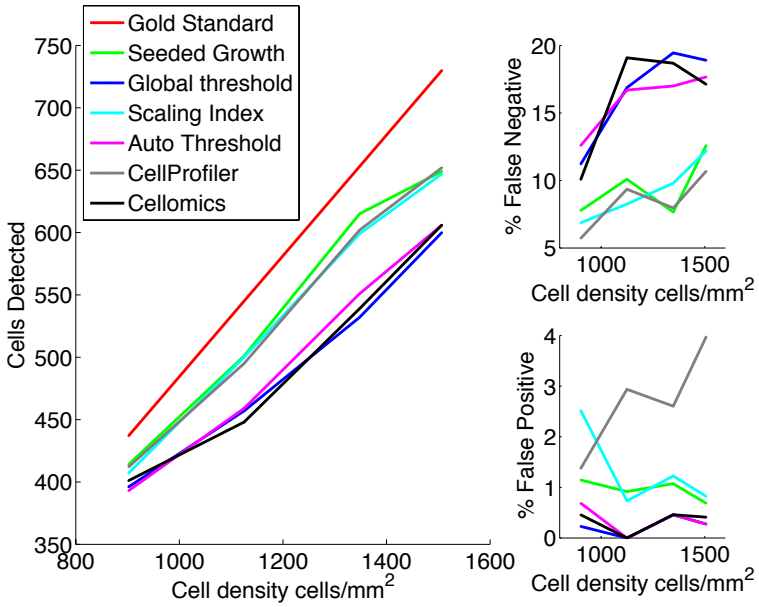


Figure 5.6: Cell detection accuracy measurements: Total cell count, false negatives and false positives comparing different segmentation methods to the Reference Standard (shown in red). *Adapted from Downey et al. PlosOne (2011)*

mask images. Five frames were created, at the same image size, simulating cell nuclei at densities between 425 and 703 cells per frame (2885 cells in total). The software parameters (given in Appendix B.4) were chosen to match experimentally observed nuclei sizes and distributions.

The F-score, calculated using Equations 5.2–5.4, indicates the overall accuracy of the segmentation according to this foreground/background partitioning. Unlike the position accuracy, this measurement does not detect situations where methods fail to separate clustered or touching cells and does not penalise methods under these circumstances. The precision value measures the proportion of detected pixels which genuinely belong to the ground truth, whereas the recall value is the proportion of ground truth pixels which were detected by the segmentation method. Collectively, these values indicate whether a segmentation method consistently over- or under-estimates the size of the detected objects. The method requires the numbers of True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) pixels.

$$\text{Precision (P)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5.2}$$

$$\text{Recall (R)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.3}$$

$$\text{F-Score} = \frac{(1+\beta^2)\text{P} \cdot \text{R}}{\beta^2(\text{P} + \text{R})} \tag{5.4}$$

A weighting factor β can be varied to give emphasis to either the precision or recall but the default value $\beta = 1$ was selected to give an equal weight to either metric since there is no reason to believe an alternative value would give better discrimination between methods. Tests performed using randomly generated images with different pixel coverage gave F-Score values which increased with increasing β when coverage was greater than the ground truth, and decreased when coverage was less. Changing β for the calculations using the output of the different segmentation methods had a similar effect with no method standing out for any single value of β.

The F-score performance of the different segmentation methods is illustrated in Figure 5.7. The Global Threshold (Li automatic threshold from ImageJ) resulted in the highest F-values ($\approx$0.95) for all cell densities, while the more sophisticated regional adaptive methods (including Seeded Growth and Scaling Index) performed comparatively poorly on the artificial data (0.85<F-score<0.91). This was contrary to the results obtained from the Cell Position results where the adaptive methods performed more successfully.

An alternative measurement, using the Kappa Index, measures the degree of overlap between two sets according to Equation 5.5:

$$\text{KI} = 2\left(\frac{|\text{A} \cap \text{B}|}{|\text{A}| + |\text{B}|}\right) \tag{5.5}$$

where A and B are ground truth and segmented pixel data, respectively.

Method	*Precision*	*Recall*	*F-Score*	*Kappa Index*
Seeded Growth	0.83	0.92	0.87	0.90
Scaling Index	0.93	0.89	0.91	0.88
CellProfiler	0.79	1.00	0.88	0.93
Global Threshold	0.96	0.95	0.95	0.95
Local Threshold	0.86	0.98	0.91	0.93

Table 5.2: F-Score and Kappa Index values comparing segmentation methods.

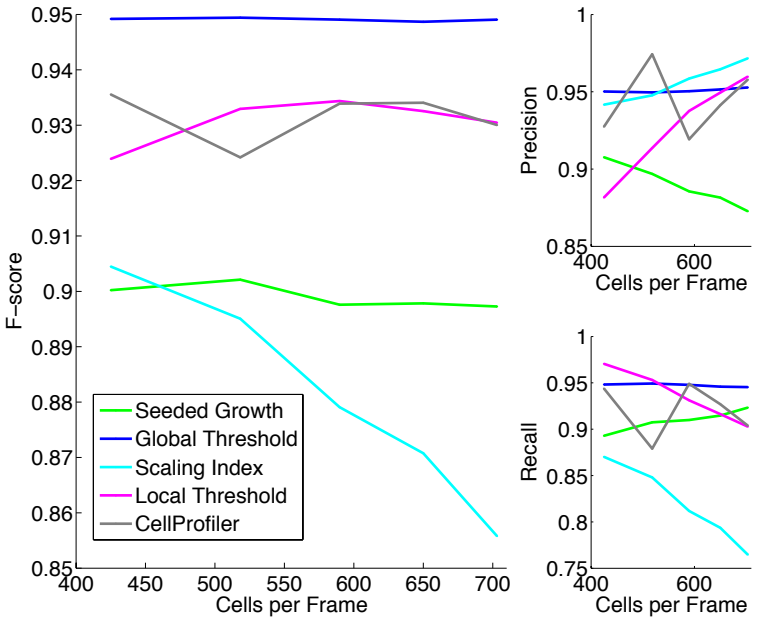


Figure 5.7: Segmentation accuracy on artificial cell images: Precision, Recall & F-Score for the SimCep images. *Adapted from Downey et al. PlosOne (2011)*

Using the Kappa Index to evaluate segmentation accuracy for the Simcep data, values, summarised in Table 5.2, are in the range KI = 0.88–0.95 (0.90 for the Seeded Growth algorithm) which compares well with the values of between 0.81 and 0.96 reported in [58].

5.5 Comparing Tracking Performance using Manually Tracked Ground Truth data

Accurate tracking is essential for obtaining useful fluorescent timecourse or lineage data from a set of measured cells. Where cells have been tracked, there will be an association between a cell image in one frame and the corresponding image in the following frame. Where these associations are available for many successive frames, it is possible to build up a full trajectory for a cell over an entire experiment.

Mistakes in the tracking can lead to a trajectory starting off following a particular cell but partway through the cell could be incorrectly associated with a different cell. Any successive correct trajectory assignments will only

be correct for the 'new' cell and not for the original cell, leaving the entire track to be considered suspect. Errors are thus cumulative and diverging and once an error has occurred, there is little chance of the mistake being reversed.

The accuracy of the tracking algorithm was evaluated by comparing the trajectories against two Reference Standard tracked sets, which were both based on Hoechst-stained C2C12 time-series images. These time-series experiments were provided by Keith Vance and Danuta Jeziorska.

5.5.1 Creation of the Reference Standard Tracked Sets

Both sets were created by allowing the LineageTracker software to perform automatic tracking then manually correcting any tracking links to create as many tracked cells or lineages as possible. Each cell had a 'Validated' true/false flag which was set to true once a lineage had been manually checked. These validated lineages were then exported to be used as the tracked sets.

Reference Standard 1

The first tracked set consisted of 24 frames with 10 minute frame intervals. Segmentation was performed using the Seeded Growth algorithm; segmentation and tracking were modified until over 50% of all cells belonged to a validated track. This set contains 7017 individual cell to cell linkages between frames, with 359 tracks ranging from 5 to 23 frames (average length = 19), but no cell divisions.

The average cell movement between frames was 3.9 pixels, with a maximum of 28 pixels (average nucleus diameter of 11 pixels). The cell density (1300 cells/mm2) was in the middle of the range of our 30 minute experiment described below.

Reference Standard 2

This tracked set consisted of 110 frames with 30 minute frame intervals. This was the same experiment used to create the Position Accuracy test set used in Section 5.3. The segmentation was based on processing from the Cellomics HCS Reader software. This enables the tracking performance to include the tracking results from the Cellomics software. The tracking was manually adjusted and lineages created until a total of 100 cell divisions had been marked as validated.

This dataset contained 157 cell trajectories containing a total of 7221 individual steps and 100 divisions. Tracks range from 5 to 110 frames (average length = 46). Average cell movement was 3.8 pixels per frame (maximum 29 pixels per frame, average cell diameter of 14 pixels)

5.5.2 Measuring the Tracking Accuracy based on Longest and Total Tracked Lengths

The tracking accuracy is measured by a stepwise comparison of the Reference Standard with the computer-generated tracking. All of the cells contain a flag which indicates whether they have been visited during the tracking evaluation. In this context, *cell* is taken to mean an instance of a cell in a particular frame, not a physical cell which exists in many frames. A tracked cell which appears in all 20 frames of an experiment will appear as 20 cells within the data structure, with links between the cells describing the tracking.

The tracking score was calculated by counting the number of individual links that were correctly identified using the automated methods and the longest continuously tracked section. Each lineage has two counters: one to hold the total number of correct links identified, the other holding the longest continuous correctly tracked chain. Each of the Reference Standard cells is visited in turn and a corresponding cell is located in the comparison set. The track length counters are both initialised to zero. If both cells have been tracked, the next cells in the tracks are obtained. If both of the tracked cells are the same (both positions are within the mean cell radius of each other), the track length counter is incremented. If the tracked cells do not match, the current track length is stored and the counter is reset to zero.

For each measured track i in the Reference Standard, there is a set of track lengths $\mathbf{L}_i$ which hold the lengths of the tracked sections where the reference standard and the method being evaluated are in agreement (illustrated in Figure 5.8). The longest of these lengths is designated L_i^{max}. The length of the track in the Reference Standard is given by L_i^{ref}. Two scores are calculated: Total track accounted for (Equation 5.6) and Longest continuous track (Equation 5.7), where n is the number of tracks present in the reference standard.

$$\text{Correct Steps} = \frac{100}{n} \sum_{i=1\ldots n} \frac{\Sigma \mathbf{L}_i}{L_i^{ref}} \tag{5.6}$$

$$\text{Longest Chain} = \frac{100}{n} \sum_{i=1\ldots n} \frac{L_i^{max}}{L_i^{ref}} \tag{5.7}$$

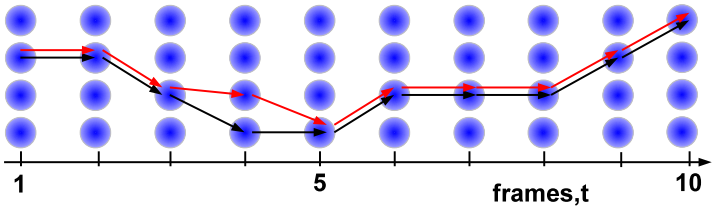


Figure 5.8: Measuring tracking accuracy. Horizontal axis shows time with the vertical axis representing cells in the frames. The red arrows indicate the manually tracked 'Reference Standard' route marked through the cells, and the black arrows show the calculated tracking. Tracking accuracy is measured by counting the total number of steps which match the Reference Standard and the longest continuous chain of correct steps. In the example here, 7 steps were correctly identified, with the longest chain of length = 5. *Adapted from Downey et al. PlosOne (2011)*

5.5.3 Comparing LineageTracker Performance with Third-Party Tracking Software

The custom tracking algorithm was compared to CellProfiler along with the Particle Tracker* and MTrack3† tracking systems available with ImageJ [113]. Each tracking method was run using both gold standards.

CellProfiler Tracking

There are four tracking algorithms available within CellProfiler. These are:

1. **Overlap Tracking** – Selects the cell pairs with the greatest overlap in adjacent frames.
2. **Distance** – Selects the cell in the following frame where distance, as measured from the perimeter, is smallest.
3. **Measurement** – Compares the values of the user-selected measured feature and selects cells where these values are closest.

*`https://weeman.inf.ethz.ch/ParticleTracker`
†`http://user.interface.org.nz/~gringer/hacking/mtrack3.html`

4. **Linear Assignment Problem (LAP)** – An implementation based on [88] which aims to handle high cell density and the possibility of temporary non-detection of cells.

The LAP tracking was chosen to represent CellProfiler in the tracking tests since it is a more sophisticated algorithm which is considered state of the art.

Particle Tracker (ImageJ)

The Particle Tracker plugin is an implementation of Feature Point Tracking [43]. There are two main parameters to vary: Link Range, which specifies how far ahead the algorithm looks, and Displacement, which is the search radius for cells. A displacement of 15 and link range of 2 were found to provide the highest tracking scores for the Reference Standard sets.

MTrack3 (ImageJ)

MTrack3 is based on the earlier MTrack2 plugin and was used in preference to the earlier version since it writes the tracking results to disk in a more easily managed format. The tracking is based on 'nearest cells' with optional velocity prediction. The main parameters are maximum velocity and prediction on/off. The algorithm does not make use of any intensity or feature values.

Different values of the parameters were investigated and a maximum velocity=10 was selected.

LineageTracker Algorithms

There are several different tracking algorithms available within LineageTracker. These are described in Section 3.4.1, along with the Hungarian Matrix Minimisation described in Section 3.4.8. Two additional methods were developed based on the tracking framework. The 'Simple Nearest Cell' tracking calculates the distance between cells in adjacent frames and assigns the nearest cell provided it is not currently part of a lineage. This is an intentionally crude implementation which is designed to provide a baseline to compare more sophisticated methods.

An Overlap-based tracker was developed to evaluate the utility of such a method when used with rapidly moving cells where there is a high probability of no overlap between successive cell images. The method calculates the overlap

fraction as the movement score and uses the Hungarian algorithm to assign the matches.

5.5.4 LineageTracker offers high quality tracking

All of the tracking methods manage to attain over 90% accuracy for individual cell-cell linkages (Table 5.3). This value is important if manual tracking adjustments are to be made prior to any fluorescence timecourse measurements since a better score indicates fewer incorrect linkages which will require manual correction.

The average successful tracked length (the second figure given in the table) is more relevant in traditional HTS where accurate tracking is required as part of an automatic analysis pipeline.

The custom methods developed here manage to achieve the highest cell-cell linkage accuracy, reaching nearly 99% accuracy for Reference Standard 1. The overlap-based tracking achieved a slightly lower score of just under 98% which is consistent with the mean cell motion (3.9 pixels) being within the typical cell radius (11 pixels). The mean overlap between successive cells in a trajectory was 57.4% with 1.6% of cell-cell links having no overlap between cells.

CellProfiler achieved the highest score of any third-party tracking system with a score of 95.9%

The accuracy levels were slightly lower for the second Reference Standard, with the custom tracking methods scoring 97–98% accuracy. The overlap-based tracking suffered a large decrease in accuracy score, dropping to 92%, although the mean overlap between cells had increased to 74.2%. The number of cells without overlap across frames increased slightly to 1.7%

CellProfiler was consistent in achieving the highest third-party accuracy, with a much higher accuracy than MTrack3, Particle Tracker and the Cellomics tracking methods.

5.5.5 The Effect of Velocity Prediction on Tracking Accuracy

The MTrack3 method features an optional velocity prediction component which uses the motion in addition to position to calculate the best match. The suitability of the velocity component should depend on the nature of the motion

Experiment:	**Reference Standard 1** *24 frames* (10 minute interval)	**Reference Standard 2** *110 frames* (30 minute interval)
Validated Positions	7321	7417
Validated Trajectories	359	157
Frame to Frame links	6886	7221
Mean track length	19	46
Tracking Scores for LineageTracker		
Co-operative Greedy	98.3/94.2	98.0/93.1
Hungarian Algorithm	99.0/97.1	97.2/89.9
Simple Nearest Cell	92.7/76.2	95.0/76.0
Overlap Tracking	97.9/90.6	92.4/82.6
*Tracking Scores for CellProfiler**		
Linear Assignment	95.9/88.3	96.1/85.4
Nearest Cell	87.8/78.5	87.7/76.7
Intensity Measurement	39.1/35.7	34.2/28.9
Overlap Tracking	90.3/80.8	89.6/78.4
Tracking Scores for other Third-Party Algorithms		
MTrack3 (ImageJ)[†]	93.1/84.5	85.4/76.3
Particle Tracker (ImageJ)	92.3/82.9	86.4/64.1
Cellomics KineticScan[‡]	n/a	85.9/55.9

Table 5.3: Results of Reference Standard tracked sets, excluding cell divisions. Two numbers are given for each measurement: total number of correctly tracked steps and longest continuously tracked section (as percentage of total steps). For the 10 minute interval experiment the Seeded growth segmentation was used, and cells were manually edited so that 50% of cells with positively validated segmentations were included in the tracking Reference Standard. The 30 minute interval experiment is based on the Cellomics segmentation as to allow comparison with the Cellomics tracking routines. The first 4 tracking scores are part of the LineageTracker software developed as part of this thesis. Overlap Tracking used Equation 3.7 to calculate the transition scores. *LAP = Linear Assignment Problem tracking. CellProfiler Intensity Measurement tracking uses the 'Feature Tracking' method and only considers a single feature, so does not consider cell distance. All CellProfiler tracking used distance limit=15, where applicable. [†]MTrack3 used velocity prediction with maximum velocity=10. [‡]The segmentation used for Reference Standard 1 contained manual corrections so could not be used to evaluate the Cellomics KineticScan software.

of the cells, where more random motion would benefit less. In Sections 5.1 & Appendix B.6, the cell motion is characterised and compared to random walk models. The motion of the Reference Standard 1 is more 'linear' than in Reference Standard 2, with smaller changes in angle (more of the cells lie along the diagonal of Figures B.6c & B.6d). The velocity prediction is a form of Keyhole Tracking, as discussed in Section 1.6.1. This would be expected to have a positive effect on the tracking where cell motion is more uniform or predictable. Table 5.4 summarises the effect of this on the tracking accuracy for MTrack3.

Tracking Scores	**Reference Standard 1**	**Reference Standard 2**
Velocity prediction On	93.1/84.5	85.4/76.3
Velocity prediction Off	92.6/83.0	85.8/77.0

Table 5.4: Effect of Velocity Prediction on the tracking accuracy for MTrack3.

For Reference Standard 1, there is an improvement when cell motion is taken into account, in line with expectations. The tracking results decrease for Reference Standard 2, where there is a greater change in cell direction (see Figure 5.2d).

5.5.6 Effect of Distance Threshold on Tracking Accuracy

Table 5.5 contains the tracking scores for different values of the Distance Threshold parameter. The highest scores are obtained for values close to 10, where 95% of cell displacements are below this threshold (Table 3.2). The time taken to perform tracking increases steadily as the displacement threshold increases. This is due to the increased radius leading to a larger number of cells being considered at each frame.

Displacement Threshold	*Score (links)*	*Score (track length)*	*Time (Seconds)*
5	91	72	5.1
10	97	89	5.5
15	97	88	6.1
20	96	85	6.7
25	95	79	7.6

Table 5.5: Changes in the tracking score as the displacement parameter is varied, using the 7,500 cells from the Reference Standard data set. This parameter specifies the expected maximum distance (in pixels) moved by a cell. The time required for tracking increases with the distance threshold since fewer candidate cells are rejected due to distance. The time column does not include the optimisation step.

5.5.7 Optimising Weights and Thresholds for Feature Calculations

The tracking parameters (weights and thresholds for distance, intensity, area and texture, along with the sigmoid shape parameter) may be optimised to obtain more accurate tracking for a particular cell type or motion model. Each parameter is taken in turn and the value is randomly perturbed by ±1% of the

parameter range. The tracking is re-evaluated and the new parameter values are retained if the scores improved.

Local minima may be encountered where the tracking score is sub-optimal but small changes in parameters are unable to improve the tracking score. The optimiser attempts to avoid these by increasing the scale of the perturbations if repeated iterations fail to improve the score.

Varying Number of Features affects Tracking Results

The tracking scores for the custom tracking algorithms given in Table 5.3 were based on all measured features (optimised feature values are given in Appendix B.3).

Features	*Tracking Score*
Distance Only	92.03/74.29
Distance+Intensity	97.52/91.57
Distance+Intensity+Area	98.24/94.23
All Features	98.25/94.23

Table 5.6: Effect of varying the features on the tracking score on Reference Standard 1.

Table 5.6 shows how the tracking score varies as more features are taken into consideration. The distance, intensity and area features are sufficient to allow accurate cell identifications and the addition of the texture features provides only marginal increase at the expense of additional computational cost.

5.5.8 80% of Divisions were Detected Accurately

None of the third party tracking systems supported detection of cell divisions so only the custom tracking method was considered.

The accuracy of cell divisions was determined using Reference Standard 2, which contained 100 manually annotated cell divisions, 80 of which were correctly detected by the software. There were 16 false positive divisions detected: two where a division was correctly identified, but the daughter cells were assigned incorrectly, and the remaining 14 where a division was detected and none occurred.

5.6 There is no 'One size fits all' solution

No single segmentation method was found to have overall superiority regarding cell detection and pixel accuracy. The Seeded Growth provided a good compromise between cell detection, false positives and false negatives.

The tracking method developed in Chapter 3 compared favourably with the third-party tracking systems which were available. The accurate matching tracks obtained from the methods (using both the Co-operative and Hungarian Assignments) were consistently longer than those obtained from the other methods.

Chapter 6

Constructing Lineages for *Schizosaccharomyces pombe*

Sz.pombe [114] is a fission yeast which is one of the main eukaryotic model organisms. It was isolated in the 19^{th} century from a millet beer brewed in east Africa ('pombe' is the Swahili word for beer) and the genome was sequenced in late 1990s. It is only distantly related to traditional bakers or brewers yeast (*Saccharomyces cerevisiae*). Of late 2011, 38% of genes have had their function experimentally confirmed with a further 43% having their function inferred from similarities with other genes [115]. The remaining 19% of genes are of unknown function and are divided between conserved genes which are also present in other organisms (11%), genes which are only present in fission yeast (7%) and entries which are not confirmed as protein encoding (1%).

Sz.pombe is used in the study of many cellular processes including cell signalling and mitosis. The short generation time of *Sz.pombe* (typically 2-4 hours) following gene expression over several hours will require following daughter cells. Dr Graham Ladds at the University of Warwick Medical School expressed an interest in adapting the methods developed in the previous chapters and applying them to time-course images of *Sz.pombe*.

This chapter begins with a discussion of a brief investigation of different staining protocols to visualise the cells in a manner suitable for the segmentation and tracking algorithms developed in Section 3.3. The segmentation and tracking methods were applied to images of *Sz.pombe* and the necessary

modifications are described in sections 6.2 & 6.3, including a novel segmentation method based on the image intensity gradient. Finally the results are presented where a lineage is constructed from a timeseries of proliferating cells.

6.1 Visualising *Sz.pombe* using Fluorescent Microscopy

Culturing and imaging conditions used in this chapter are described in Section 2.2. The segmentation methods were originally developed for bright fluorescent nuclei, based on Hoechst staining, so nuclear rather than full cell fluorescence would be desirable to enable the existing methods to be reused with little change.

6.1.1 Hoechst Staining over long durations was unsuccessful

A brief investigation into Hoechst staining was conducted. Cells were grown overnight in minimum media (Table 2.2) at 30 °C before being centrifuged for 3 minutes at 2000 rpm then resuspended. Different resuspension media were tried: water, PBS or minimal media (1 or 2 ml each). To each of these, 25 μl per ml of a 1:2000 Hoechst stock was added. Resuspended cells were subsequently incubated for between 4 and 14 hours.

The Hoechst-stained cells were imaged after initial incubation and again after 24 hours. None of the tests produced viable stained nuclei after 24 hours. There had been proliferation of cells but any remaining live cells were not stained. The only cells visible in the Hoechst channel were dead.

There are transport mechanisms present which can transport Hoechst out of lipid bilayers [116] and some cells, including Dictyostelium [117] have been reported to expel Hoechst so while they may use a completely different mechanism, *Sz.pombe* may also have the ability to expel Hoechst from cells.

6.1.2 gar2-GFP expression in *Sz.pombe* allows nuclei and cytoplasm to be detected simultaneously

Jonathan Millar, from the Medical School at the University of Warwick, kindly donated a selection of GFP-expressing strains of *Sz.pombe*. These were imaged to determine which displayed suitable nuclear fluorescence, which could then be

used as the basis for further experiments. The most suitable was one which expressed GAR2-GFP, which displayed a bright compact nucleus surrounded by a fainter cytoplasm, which would allow both nuclear and whole-cell measurements to be made. The GAR2 protein operates within the nucleolus [118, 119] and is involved in the production of ribosomal subunits. Since it is already largely nucleus-based, there is no need to append a nuclear localisation sequence.

6.2 Adapting the segmentation methods for Yeast

The bright nuclei of the GAR2-GFP *Sz.pombe* can be segmented using the existing seeded growth method (Figure 6.2a) by selecting an intensity threshold which allows separation of the nuclei from the cell bodies. There is a sufficiently large intensity difference between nuclei and the rest of the image (Figure 6.3) which enables such a threshold to be readily obtained, either interactively (using the 'Preview' function of the software, see Appendix A.5), from visual inspection of the plot profile (Figure 6.3b), or more rigorously using a method such as 'K-Means Clustering' [73, 120].

The GAR2-GFP *Sz.pombe* images can readily be separated into three distinct regions as described above. Applying the K-Means clustering on the pixel intensities using k=3 places each pixel into one of three 'bins' which is effectively a multi-value global threshold. The mean intensity values of each cluster can subsequently be used to calculate the segmentation thresholds:

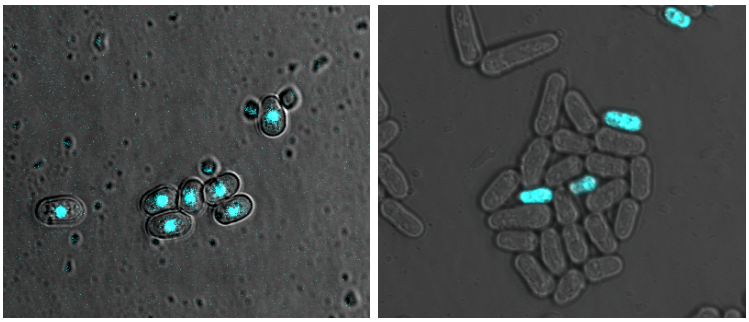
$$\text{Mean Intensity for cluster } i = \mu_i$$

Where $i = 0$ for background, $i = 1$ for cell body, $i = 2$ for nuclei.

$$\text{Threshold for nuclei} = 1 - \frac{\mu_2 - \mu_0}{\mu_2}$$

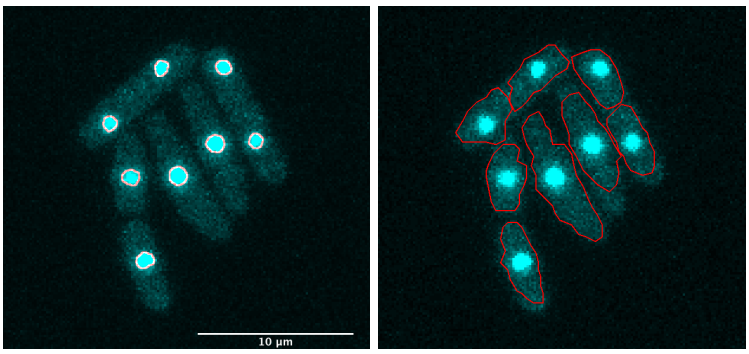
$$\text{Threshold for cell bodies} = \frac{\mu_1 - \mu_0}{\mu_2}$$

In Figure 6.3c, the yellow cell bodies are not completely separate and a seeded growth may incorrectly segment the areas where cells touch. This becomes more likely when part of the cell is closer to a neighbouring nucleus



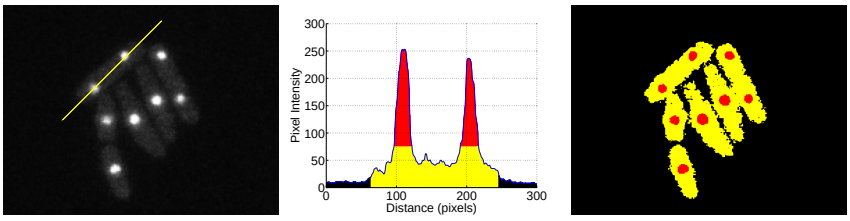
a) Hoechst-stained *Sz.pombe* b) After 24 hours of growth

Figure 6.1: Investigating the Hoechst staining of *Sz.pombe*. a) Stained for 4 hours at 1:2000 dilution in minimal media. b) After growing for 24 hours, cells had either died or ejected the Hoechst.



a) Nucleus segmentation b) Full cell segmentation

Figure 6.2: Segmentation of yeast cells expressing GAR2-GFP. a) Limiting segmentation to the bright nucleus b) Lowering the brightness threshold to allow segmentation to select the entire cell.



a) Position of intensity profile b) Intensity profile across cell c) K-means clustering of pixel intensities

Figure 6.3: Intensity profile across a *gar2*-GFP expressing *Sz.pombe* cell which is undergoing mitosis. There is sufficient difference in intensity between image background, cell body and nucleus to allow them to be detected individually. a) Location of the intensity profile measurement. b) Intensity profile, with different regions coloured; black: image background, yellow: cell body, red: cell nucleus. c) K-means clustering of pixel intensities, colour coded the same as (b)

than the correct nucleus, so when the growth step occurs the pixel is visited and claimed by the neighbour nucleus. Additional checks are required during growth to prevent this from taking place.

The intensity gradient of the image may be used as a guide to which nucleus is associated with a particular point. The Radial Growth algorithm was extended to take advantage of this to investigate whether it would be useable for segmentation. The magnitude and direction of the gradient is calculated as given in Equations 6.1–6.2.

$$\text{Magnitude } = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \tag{6.1}$$

$$\text{Direction } = \tan^{-1}\left(\frac{\partial I}{\partial x}\right) \Big/ \left(\frac{\partial I}{\partial y}\right) \tag{6.2}$$

The gradient is calculated on a smoothed copy of the image (Gaussian kernel, $\sigma = 3$). At each point p during the growth stage, a vector $\overrightarrow{p}$ is obtained pointing back along the 'spoke' towards the nucleus. A vector $\overrightarrow{g}$ is obtained from the gradient. The angle between the vectors is calculated using the dot product (Equation 6.3).

$$\theta = \cos^{-1}(\overrightarrow{p} \cdot \overrightarrow{g}) \tag{6.3}$$

This angle is used to adjust the rate of growth of the cell in that direction, where the spoke increases by

$$1 - 0.5\theta^2 \tag{6.4}$$

which favours growth in the direction of the image gradient and thus allows cells to grow more rapidly along the major axis and decreases growth in areas where cells are in contact, which reduces the 'hijacking' of pixels from neighbouring cells.

6.3 Adapting the existing Lineage Construction methods

The cells of *Sz.pombe* are non-motile so any movement observed will be caused by either growth or proliferation of cells which causes a cluster of cells to expand outward, or movement of the microscope stage. The latter are often due to vibrations or temperature fluctuations but these are minimised by using a thermostatically controlled environment and a damped optical table.

The positions and intensities of the nuclei, rather than the whole cells, are employed for tracking since they are smaller and brighter and therefore the positions can be measured with greater accuracy. The tracking method described in Section 3.4.1 is used to follow the nuclei. No changes are necessary apart from a smaller distance threshold, due to the decreased motion.

A different division detection method is required since there is no observed drop in intensity during mitosis. A suspected division event occurs when a cell appears which does not have a precursor in the previous frame. If the cell is within the distance-threshold of the edge of the frame, it is disregarded since at such close proximity it will be difficult to unambiguously determine whether the cell is the result of division of a cell just outside the frame and therefore not part of an existing lineage.

The lineage construction method can be described as 'Orphan-Adoption' since it first identifies a cell which has no 'parent', then calculates which parent is the best match. For the calculation described below, the first daughter cell is the one assigned during the tracking step, since the tracking automatically assigns the best-match cell. The second daughter will only be assigned if the matching criteria are reached.

Additional position features are available for consideration in identifying parent and daughter cells, in part due to the elongated cells providing a major axis. During division the nucleus divides to produce the two daughter nuclei which then move to opposite ends of the cell before the formation of a septum which leads to the cell dividing in two [121].

There is low correlation for the distance between the nuclei before and after division, when comparing the positions of the two daughter nuclei (Figure 6.5a, R^2=0.06). The cell axis orientation is better conserved during division, where

the daughter nuclei lie along the axis of the mother cell, as measured in the previous frame (Figure 6.5b).

The previous frame is examined for potential parents and each parent cell is given a score which is calculated as follows:

- If potential daughter cells are on opposite sides of parent, increase the score by 1 for both x & y axes.
- Measure the angle difference between the major axis of the parent cell and parent-daughter positions. Increase the score using Equation 3.4, using 15 degrees as threshold.

The parent with the highest score is assigned to the orphan daughter and the algorithm moves on to the next un-assigned cell.

6.3.1 Obtaining *Sz.pombe* Lineages

The lineages are stored as lists where each entry represents a cell at a particular frame. Each entry in the lineage keeps a record of the cells in the next and previous frames and also any daughter cells where divisions occur (see Appendix A.2.1 for a visual representation). This allows traversal of the list or tree to rapidly count the number of divisions or intervals between them. Data gathered in this manner may be used to construct a lineage tree for a cell (see Figure 6.6 – the tree graphic was created using Graphvis*, see Appendix B.8 for further details).

The sample lineage presented here was selected from a larger population which was allowed to grow and proliferate over a 12 hour period. The cell count and cell cycle lengths are presented in Figure 6.7. The mean generation time measured from the lineage was 4.6 hours with the mode = 3.8 hours. For wild-type *Sz.pombe*, the typical generation time is 2–4 hours, with an observed division every 3 hours [122] at 25 °C

$$\text{no. of cells} = 2^{(t-t_0)/c} \tag{6.5}$$

The doubling time measured by fitting an exponential growth curve (Equation 6.5, where t is the time in minutes, c is the doubling time, and t_0 is the

*`http://www.graphviz.org/`

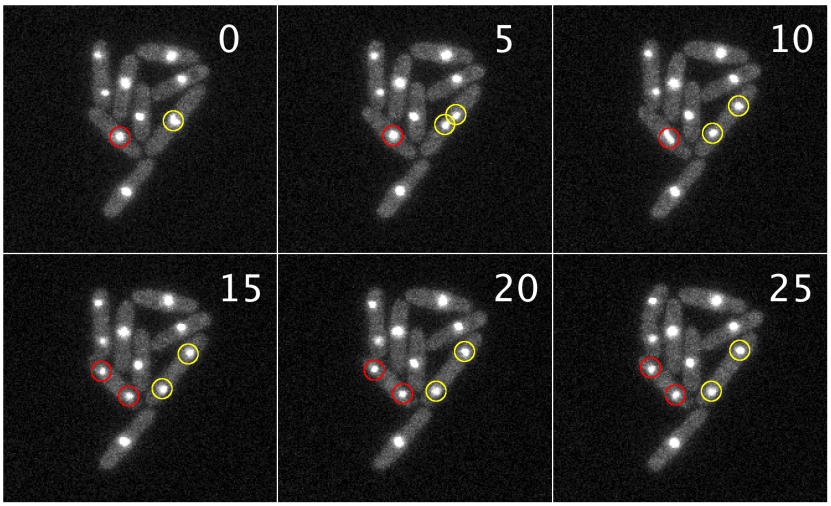
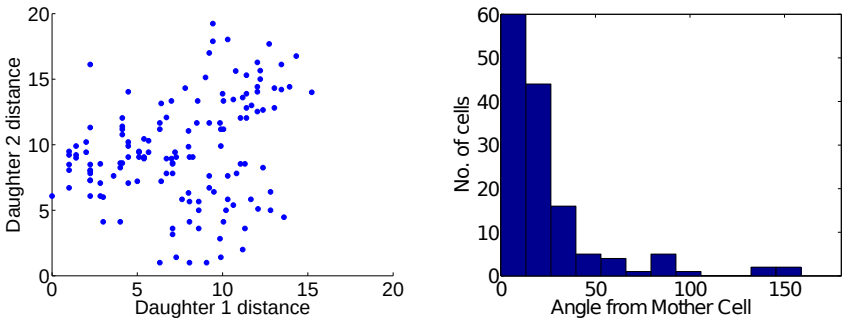


Figure 6.4: Dividing yeast cells, time displayed in minutes. Circles highlight the dividing nuclei.



a) Distance of daughter nuclei following division

b) Angle of daughter nuclei.

Figure 6.5: Daughter cell position relative to mother cell for *Sz.pombe*. a) Distance of the two daughter nuclei from the last measured position of the mother cell nucleus. b) Angle of daughter cell positions, in relation to the parent cell nucleus in the previous frame.

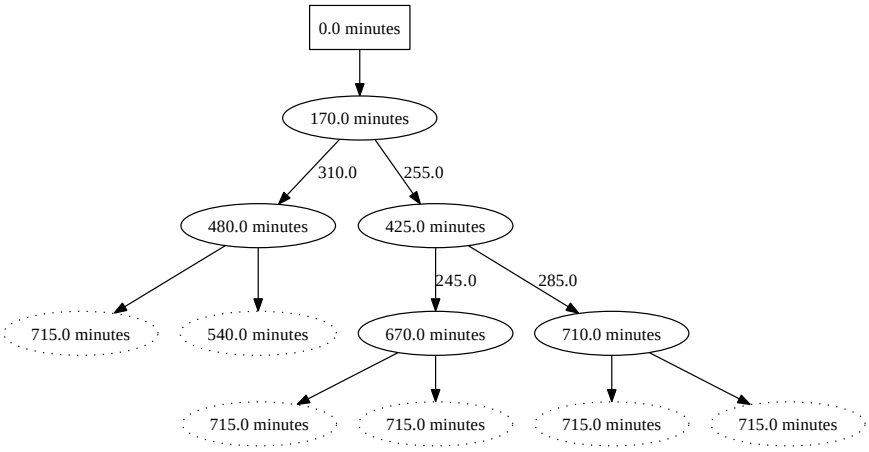
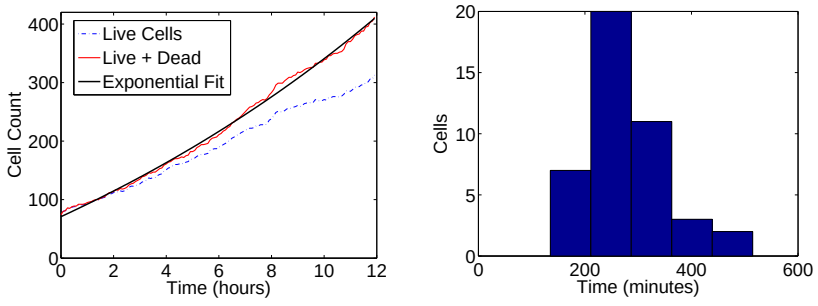


Figure 6.6: Sample *Sz.pombe* lineage tree. Rectangles indicates start of tracking, ovals represent divisions, dotted ovals are when tracking ended. Times are given in minutes from the start of the experiment. Times written alongside arrows display intervals between divisions.



a) Cell Proliferation b) Cell Cycle Lengths

Figure 6.7: *Sz.pombe* proliferation and cell cycle lengths. a) Proliferation of cells during the time-series. The blue dotted line is the number of live cells at any one time, with the red line indicating live cells plus all cells which have died or left the field of view. An exponential growth-curve is superimposed. b) Cell cycle lengths obtained from the constructed lineage.

extrapolated time when the population would have begun from a single cell) was calculated as 5.5 hours, which is significantly longer than the observed doubling time of 4.6 hours. One possible reason could be that the cell count is affected by cells which were not observed to divide during the timecourse. Of the 74 cells present at the start, 32 did not divide. There were 50 cells in total which were lost when they left the field of view but these were included in the count of 'dead' cells.

6.4 The LineageTracker software is suitable for *Sz.pombe* cells

The methods developed for C2C12 have been demonstrated to also work for *Sz.pombe*. The lineage construction methods were adapted to accommodate the presence of the major axis which aided detection of daughter cells. This allowed accurate lineages to be constructed and individual generation times to be obtained.

Chapter 7

Cell Cycle Analysis

There have been many published studies involving the cell cycle and oscillating proteins where cells are required to be observed over lengthy time periods or over multiple generations [26, 85, 123]. In the absence of single-cell imaging, many studies synchronise cells by, for example changing media [24], to facilitate fluorescence measurement by measuring bulk intensity of a cell population. This approach can be problematic when attempting to measure over multiple generations since correlation of cycling proteins between daughter cells has been observed to decrease rapidly [124].

7.1 The FUCCI Markers indicate phase of the Cell Cycle

The tracking and segmentation methods were originally developed for cells which exhibited a permanent nuclear marker (Chapters 2 and 3). There will be cases where it is not possible to add such a marker or analysis is required on images which have already been obtained. To demonstrate the feasibility of using the software without a continuously visible fluorescent marker, it is used here to obtain intensity profiles of zebrafish embryonic PAC2 cells expressing FUCCI cell cycle markers (**F**luorescent **U**biquitination-based **C**ell **C**ycle **I**ndicator) which are visible for the most of the duration of the cell cycle. Later, this is applied to study the relationship between cell cycle and circadian clock in NIH3T3 cells (Section 7.3)

The FUCCI cell cycle markers were developed at the RIKEN Institute in Japan [66, 95] and uses two fluorescent dyes to visualise cells in different stages

of the cell cycle. The markers consist of two ubiquitin ligase substrates, Cdt1 and geminin, which are expressed during different phases of the cell cycle and have been fused with red- and green-emitting fluorescent proteins. Cdt1 is a regulating factor in DNA replication [125] and is involved in the formation of the pre-replication complex which is assembled during the early stages of the G1 phase. Geminin is a negative regulator of Cdt1 so its appearance during the S phase causes a reduction in Cdt1. Truncated versions of the proteins were selected since they need to become targets for the ubiquitin mediated proteolysis but should not be active as regulators themselves. The truncated Cdt1 and geminin were fused to red-emitting and green-emitting fluorescent proteins respectively.

Newly divided cells start in the G1 phase which appear red and change to green during the S, G2 and M phases of the cell cycle. There is an overlap during the G1 to S transition where both markers are visible, giving the nuclei a yellow colour (Figures 7.1 and 7.2).

At mitosis, there is a rapid decrease in intensity in the green channel, but there is a short delay before the cell becomes visible in the red channel. Because of that delay, there is insufficient difference between daughter cells and background for accurate automatic detection, so it is often necessary to manually track a short section of each lineage (Figure 7.3 and Table 7.1).

7.2 Obtaining Cell Cycle in Zebrafish embryo cells

Fluorescent timecourse images of Zebrafish PAC2 embryonic fibroblast cells, which had been transfected with FUCCI markers as described in Section 2.3.1, were obtained from Kathy Tamai. These images were used to demonstrate the effectiveness of the LineageTracker software in following cells without a permanent nuclear marker.

7.2.1 Cyclic markers pose challenges to segmentation

The PAC2 cells were segmented using a modified version of the Seeded Growth algorithm (see Section 3.1.4) which features improved separation between cells fluorescing in different channels.

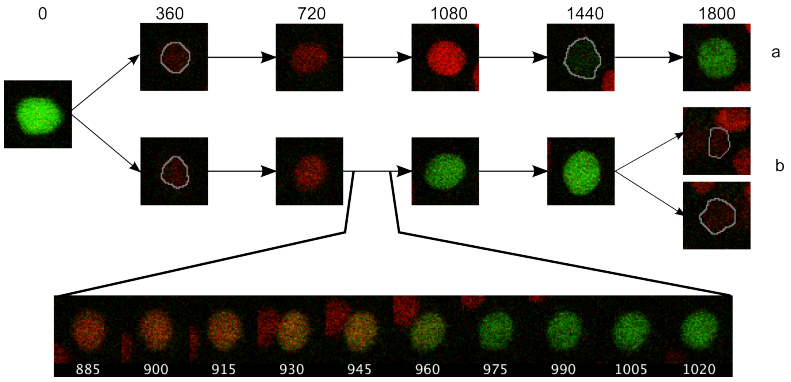


Figure 7.1: Colour changes during the cell cycle indicated by FUCCI markers in two daughter cells labelled 'a' and 'b' (see also Figure 7.2). Time is in minutes following division. The overlap in the red and green fluorescence (transition between G1 and S phase) is shown for cell 'b' (bottom panel). White outlines are given for nuclei showing weak fluorescence. *Adapted from Downey et al. PlosOne (2011)*

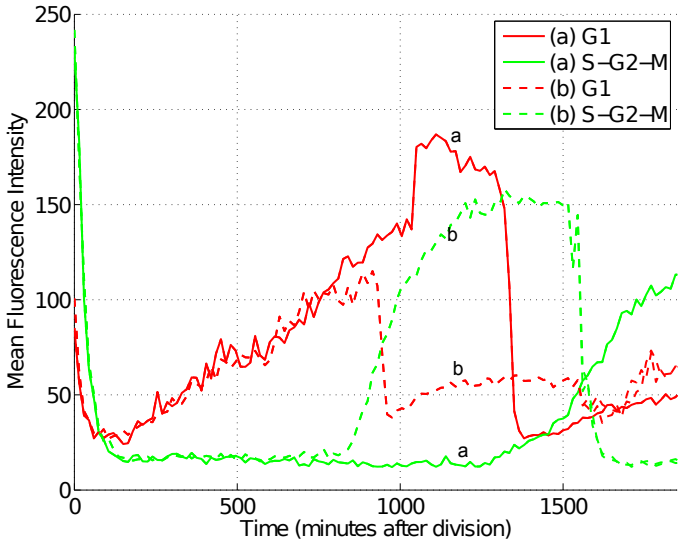
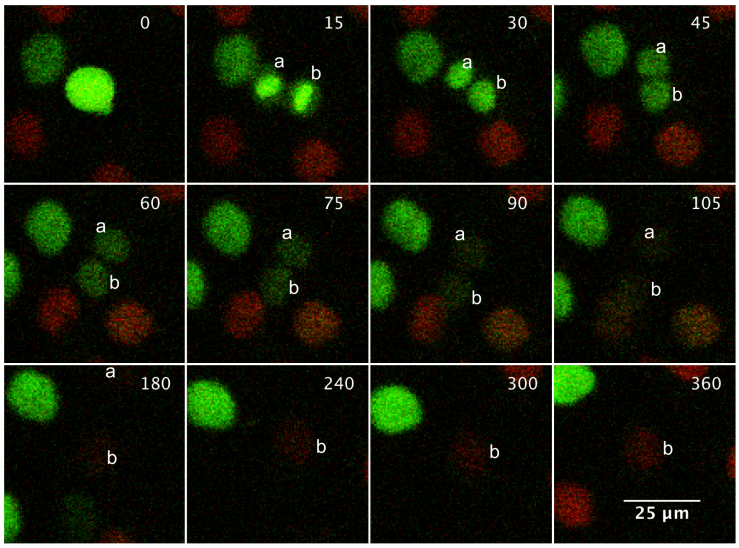
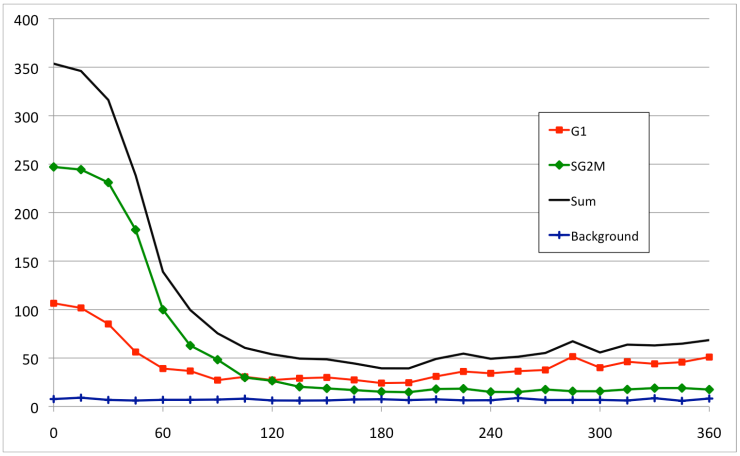


Figure 7.2: Intensities of the FUCCI markers following cell division. Fluorescence intensity following cell division for the two daughter cells in Figure 7.1. The two FUCCI channels have been shown for an entire cell cycle. The G1 signal (red) increases gradually following mitosis, then decreases following a rise in S-G2-M signal (green). A magnified view of the first 3 hours is shown in Figures S11 and S12. *Adapted from Downey et al. PlosOne (2011)*



a) Dividing cell visualised using FUCCI markers. The green FUCCI S-G2-M marker fades after mitosis followed by a slow increase in red G1 marker. Time displayed in minutes same as 7.3b. Two daughter cells, labelled 'a' and 'b' have been tracked.



b) Intensity drop following division for zebrafish PAC2 cells, following daughter cell 'b'. The image background intensity and sum of image channels for the measured cell are also plotted.

Figure 7.3: Dividing zebrafish PAC2 cells visualised using FUCCI cell cycle markers. *Adapted from Downey et al. PlosOne (2011)*

As mentioned in Section 7.1, there is a delay before the increase in fluorescence, as illustrated in Figure 7.3. Although the cell fluorescence intensity is above the background, it is insufficient for the 'maxima detection' algorithm to locate cells without there being excessive numbers of false positives detected. Such cells may be segmented using the existing Seeded Growth method but the seed position must be provided manually. This has been implemented as an interactive plugin for the LineageTracker software (see Figure A.5) where the cell outline is grown according to the procedure described in Section 3.1.3. The parameters relating to the cell outline (such as intensity threshold and 'colour difference') can be adjusted and a preview of the outline is displayed so that the best fit can be selected.

7.2.2 Tracking Invisible Cells based on position

The cell tracking algorithm was originally developed to track fluorescent cells where there will always be a measurable intensity to aid tracking. Where cells consistently have low intensities it will be necessary to use lower weights for the intensity features during tracking so that cell position and area become the major features considered in the calculation.

Cell ID	Lineage Length (Frames)	Segmentation Adjustments	Tracking Adjustments	Longest Cont. Sequence
1	203	18.2%	11.8%	15.8%
2	68	4.4%	5.9%	48.5%
3	408	6.6%	4.4%	20.8%
4	485	8.5%	7.2%	13.8%
5	586	9.4%	6.0%	18.6%
6	425	3.8%	2.8%	22.8%
7	758	19.4%	6.2%	11.9%
8	477	5.7%	3.4%	42.3%
9	91	0.0%	2.2%	69.2%

Table 7.1: Tracking precision for zebrafish PAC2 cells visualised using FUCCI markers. The segmentation and tracking adjustments represent the percentage of frames which required manual intervention to preserve accurate tracking. The longest continuous sequence was observed with cell 8 at over 50 hours without corrections. Following division, daughter cells fade to close to background intensity requiring cells to be manually segmented. *Adapted from Downey et al. PlosOne (2011)*

In the absence of a permanent nuclear stain, the initial automatic segmentation and tracking will not be able to detect all cells (see Table 7.1 for examples). There are two possible routes to obtaining a corrected lineage:

manually correcting segmentation then re-running automatic tracking, and manually correcting both segmentation and tracking. Re-running tracking on the re-segmented Zebrafish PAC2 images gave a cell-linkage accuracy of 98.5% with a mean length for longest trajectory of 75.1%, which is below the values typically obtained for Hoechst-stained nuclei. This is likely to be due to additional ambiguity caused by attempting to identify very faint cells.

7.3 The Cell Cycle and the Clock

In addition to the cell cycle, there is an additional periodic oscillator in cells: the circadian clock [126–128]. This is familiar to most of us through the sleep cycle and the body's response to changing daylight patterns, especially when caused by jet-lag due to travelling to a different time-zone. There is also considerable evidence that the body exhibits different response to drugs at different times of the day [129].

Most of the circadian control in mammals is regulated by the suprachiasmatic nucleus part of the hypothalamus in the brain, which synchronises oscillators elsewhere in the body [128]. In some unicellular organisms, such as algae and cyanobacteria, cell divisions occur in synchronisation with the circadian cycle [130, 131]. Similar linkages have been found in mammalian cells [69] where the cell division timing appears to be gated by the circadian clock [132]. The time of division occurs at regular intervals following peaks in expression of Rev-erbα, a transcription factor controlling rhythmic expression of downstream targets in mammals.

A subsequent study [24] has claimed that mitosis can be independent of circadian clock in rat fibroblasts, which is counter to previous results which show circadian gating of division [133–135].

The work in the following section attempts to confirm whether a correlation is observed between the clock and circadian cycle in mouse fibroblasts [136].

7.3.1 The C5Sys Project

The C5Cys project* (Circadian and Cell Cycle Clock Systems in Cancer) is a project within the *ERASysBio+* initiative and involves researchers in the UK,

*`http://www.erasysbio.net/index.php?index=272`

France and The Netherlands. The aim of the project is to increase understanding of the interactions between the cell cycle and circadian oscillators, especially how disruptions in either can affect cell survival and proliferation within cancer cell populations.

The work described here is a collaboration with David Rand and Peter Krusche, of the Systems Biology Department at the University of Warwick, and Filippo Tamanini of the Erasmus University Medical Centre, Rotterdam.

Images were provided by Filippo Tamanini and Shoko Saito of Erasmus University Medical Center, Rotterdam, using the procedure described in Section 2.3.2.

Dividing cells were tracked based on the technique described in Sections 7.2.1–7.2.2. Lineages were collected for a total of 26 cells, which contained 39 cell divisions.

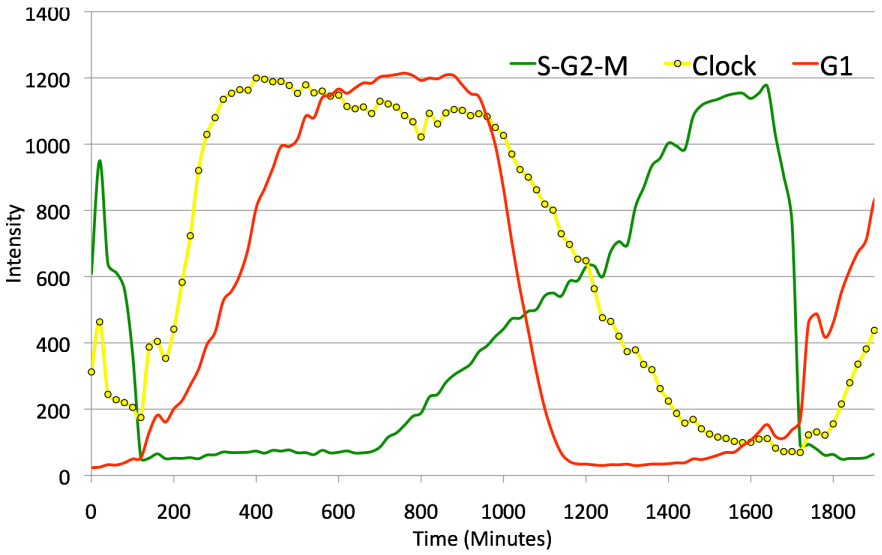


Figure 7.4: Fluorescence intensity for FUCCI and clock markers for a single cell.

The analysis steps required are:

- Fitting curves to obtain the frequency or period of oscillations.
- Obtaining times of division.
- Obtaining the individual phases of the clock or circadian oscillators at the moment of division.

7.3.2 Fitting Periodic Data to oscillating measurements

The three fluorescent channels corresponded to the clock signal and the G1 & S-G2-M FUCCI signals. The clock signal is a degraded nuclear localised Venus Fluorescent Protein similar to the one described in Section 2.1.2. The Nuclear Localisation Sequence (NLS) causes the Venus to appear colocalised with the FUCCI markers which also contain a similar NLS. The PEST sequence increases degradation of Venus, shortening the half-life and therefore improving measurement of a rapidly changing signal, allowing direct measurement of the changing signal without requiring temporal deconvolution.

The discrete sampling and limited duration of measurements place accuracy constraints on the calculated period [137]. The upper limit is based on the Nyquist frequency which states that the sampling frequency must be at least twice that of the highest frequency to be captured, to prevent *aliasing*, where higher frequencies will appear as a much lower frequency due to phase changes greater than π between samples.

Ideally the oscillations should be observed for multiple cycles to obtain an accurate measurement of period. The Fast Fourier Transform (FFT) is unsuitable for measuring such low cycle numbers. For example, consider a dateset containing 288 discrete measurements at 5 minute intervals. The FFT requires input lengths as powers of two so only $2^8 = 256$ consecutive values can be used from the data. The output frequencies will be placed in $2^7 = 128$ bins with a maximum frequency of one oscillation every 10 minutes (given by the Nyquist frequency). The bin width (resolution) is:

$$\frac{60/10}{128} = 0.047 \text{ hour}^{-1}$$

which is therefore lowest measurable frequency, and is equivalent to

$$24 \times 0.047 = 1.125 \text{ cycles per day}$$

The second lowest frequency is $2 \times 1.125 = 2.25$ cycles per day which is insufficient resolution since it only allows periods of 21.3 and 10.7 hours to be measured with no values between those ranges.

A similar constraint is present using the classic Fourier Transform, where there is an 'uncertainty principle' where accuracy of frequency is inversely

proportional to the window width. The measured frequency will therefore have an associated width which will be related to this uncertainty. Since only a single frequency was required from the oscillating signals, a sinusoidal curve fitting method was used.

The fluorescent intensities of the tracked cells were obtained from the LineageTracker software as described in Section 4.3.2. The raw fluorescence data is quite noisy (see Figure 7.4), partly due to the stochastic nature of biological processes in single cells [25] and partly due to the noise inherent in the imaging and measurement processes.

The signal is first smoothed using LOESS fitting with a 2^{nd} degree polynomial. Each of the periodic signals can be approximated as a sine curve so the smoothed curve is then fitted to an equation of the form:

$$\text{Fitted Curve}, c = \text{offset} + \text{scale} \times \sin(t \times \omega + \phi) \tag{7.1}$$

using the Matlab 'ezfit' toolbox which is based on the Nelder-Mead method to minimise sum of squared residuals. The smoothing and fitting process is illustrated in Figure 7.5. In addition to the frequency, ω and phase shift, ϕ, the fitting also reports the R^2 parameter as the quality of fit.

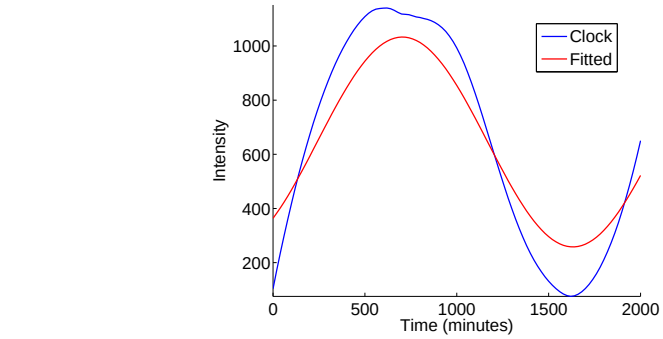
The frequency obtained from Equation 7.1, ω, is the angular frequency, which is $2\pi f$ where f is the frequency in minutes^{-1}. The period is then obtained using Equation 7.2.

$$\text{period} = \frac{2\pi}{\omega} \text{ minutes, or } \frac{2\pi}{60\omega} \text{ hours} \tag{7.2}$$

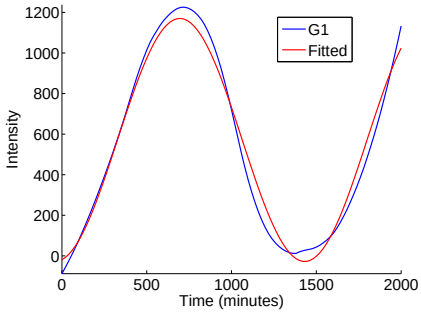
7.3.3 Comparing Cell Cycle and Circadian Oscillator

The periods obtained from the clock fitting are plotted in Figure 7.6. Each data point is coloured according to the quality of the curve fitting, using the R^2 values for each of the fitted curves. Red points indicate a poor fit, graduating through to blue for closer fits. Since each point is the result of two data fittings, the colour is based on the lowest fit value of the two.

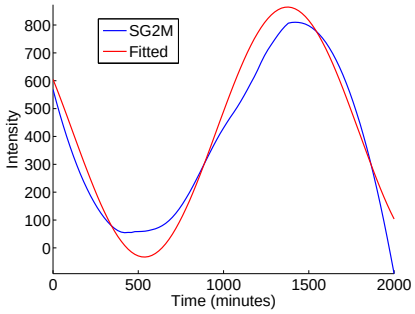
The size of each point is based on the length of data used to obtain the period, with larger points indicating longer sets of measurements.



a) Fitting the Clock curve



b) G1 cell cycle

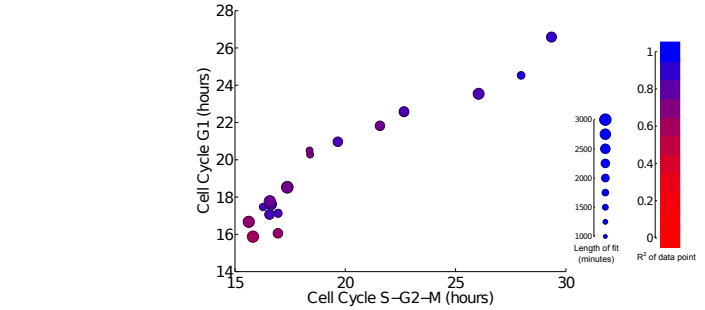


c) S-G2-M cell cycle

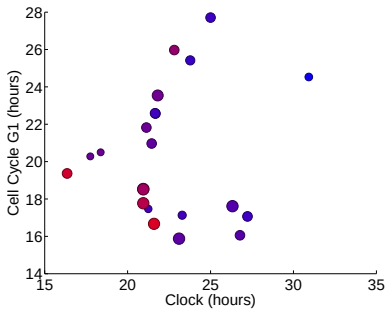
Figure 7.5: A smoothing filter is applied to the intensity data from Figure 7.4 before fitting a sinusoid. Blue: LOWESS smoothed signal, Red: fitted periodic curve.

The oscillations obtained from the G1 and S-G2-M signals should be the same duration but shifted in phase, since they both measure components of the same cell cycle mechanism. The cycle lengths obtained from this method are plotted in Figure 7.6a as a 'sanity check' of the data fitting (the 'reddest' point is from a fit with an R^2 value of 0.52, whereas the 'bluest' point came from a curve with a fit of 0.99). Any deviations from linear would indicate errors in the data fitting. The resulting correlation (R^2=0.91) suggests the data fitting is reasonable.

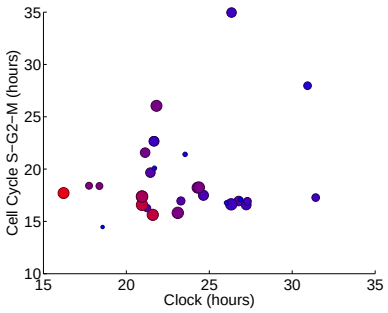
There is no apparent correlation between either Clock/G1 (R^2=0.04) and Clock/S-G2-M (R^2=0.01) periods which suggests that the clock and cell cycles are independent of each other, and while both cycles oscillate with similar periods (in the range 15–30 hours), the two are not synchronised in the cells.



a) G1 vs S2GM



b) Clock vs G1



c) Clock vs S2GM

Figure 7.6: Fitted cycle lengths from measured intensities. Each point is measured from the data-fitting on the Clock and FUCCI cycles. Size of each point is related to the length of the signal used to calculate the cycle length. Colour of each point is based on the minimum R^2 values for the data fitting. a) R^2=0.91 b) R^2=0.04 c) R^2=0.01

7.3.4 Does Cell Division occur at particular phases of the Clock?

The phases of the clock and cell cycle signals were calculated using Equation 7.3.

$$\theta = \tan^{-1}\left(\frac{c}{\mathrm{d}c/\mathrm{d}t}\right) \tag{7.3}$$

where c is the fitted clock or FUCCI signal. Figure 7.7a illustrates the relationship between the oscillating signal and the phase.

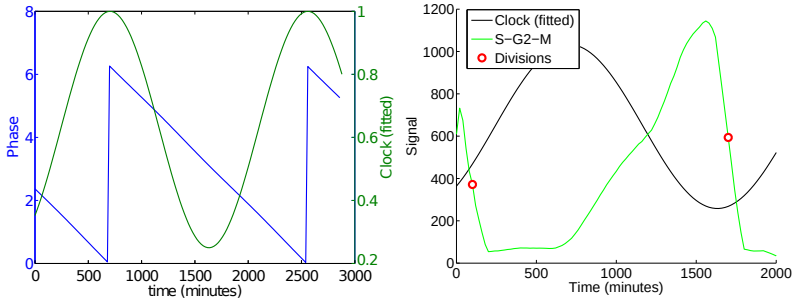
The division times were obtained from the LineageTracker software, taken from the positions of branches in the lineage data structure (Appendix A.2.1). The phases of each oscillator were obtained at these timepoints.

Given that the FUCCI markers indicate the phase of the cell cycle, it is expected that the cell divisions occur at defined phases of the FUCCI signals (at the end of the M-phase). This is illustrated in Figure 7.7b, where the division time, as obtained from the lineage data, occurred during the steep drop in the S-G2-M marker.

The phases were measured for all 39 divisions identified earlier. Figure 7.8a displays the phases of the FUCCI cycle for all divisions. Since the cell divisions occur during the drop in S-G2-M, it would be expected that the measured phases would be between 4 and 2π. All of the divisions did occur within, or close to, this range: below 0.5 radians (equivalent to 27°) or above 4.2 (245°). The divisions occurred while the G1 signal was in the phase range 1.84–3.33 (equivalent to 105.5°–190.6°).

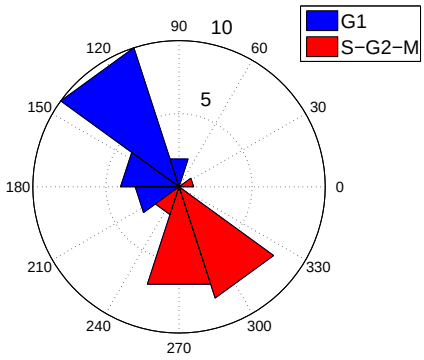
The phases of the clock at point of division are shown in Figure 7.8b. Although divisions occur at all phases of the clock, most are concentrated in a peak between 2–4 radians.

A Rayleigh Test was performed to test whether the phase distribution of the divisions is randomly distributed. Lower p-values from this test indicate that the data exhibits a unimodal deviation from uniformity [138]. The results from the FUCCI measurements (Figure 7.8a) are 5.7×10^{-9} for the G1 signal and 2.0×10^{-12} for the S-G2-M signal. Low values such as these verify that the measurements are reporting realistic or consistent phase angles since divisions will only occur at the termination of the M-phase.

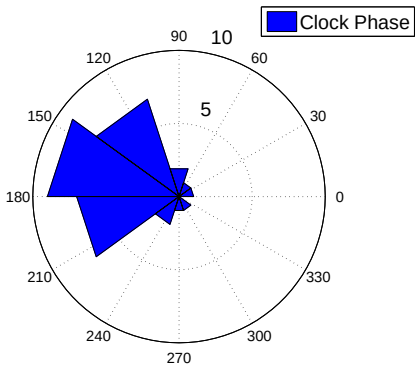


a) Relationship between signal and phase b) Cell division takes place during drop in S-G2-M

Figure 7.7: a) Phase (in radians, from $0 \rightarrow 2\pi$) takes the form of a sawtooth wave. b) Sudden drop in the S-G2-M FUCCI marker from a single cell during division. Times of division are indicated by red circles over the FUCCI signal.



a) Phase of FUCCI marker at division



b) Clock phase at cell division

Figure 7.8: Phases of the FUCCI marker and cell cycle at division. a) The phases of the G1 and S-G2-M FUCCI markers at the point of division. b) The phase of the clock marker at division.

The Rayleigh p-value obtained from the Circadian signal was 1.2×10^{-6}. This indicates that divisions do not occur uniformly throughout the circadian cycle but since this is higher than the p-values for the FUCCI signals, the distribution is not as narrow.

7.4 Circadian and Cell Cycles are not independent of each other

Although the durations of the circadian and FUCCI oscillations were found to be unrelated, the cell divisions predominantly occurred during a restricted range of phases of the clock. This is in agreement to previous studies [139] which have observed that divisions occur preferentially at particular times of day, and is contrary to the observations of Yeom et.al [24].

Chapter 8

Discussion

The cell segmentation and tracking methods described here are similar to methods which have been published elsewhere [50, 81, 82, 84, 91, 92, 140] but the implementation is unique in that it provides a flexibility to choose from a range of methods (similar to CellProfiler [59]) but with the novel feature of interactive modifications to the segmentation, tracking and lineages.

Since a automated system rarely reaches 100% accuracy compared to ground-truth data, the ability to correct the automatic analysis will increase the numbers of cells available for analysis and allow complete timecourses and lineages to be obtained for cells which would be impossible to analyse otherwise, such as the Zebrafish FUCCI cells described in Chapter 7 where the low contrast between cells and background pose too great a challenge for automatic segmentation.

The original analysis methods were developed for Hoechst-stained C2C12 cells but a change in circumstances during the PhD resulted in the necessity of applying the software to other cell types. Few changes were required to allow the segmentation and tracking to perform successfully on *Sz.pombe* cells, the major modifications being to the lineage construction element of the tracking, since the movement of the nuclei during mitosis were largely constrained along the cell axis, different daughter cell identification methods were required. Ultimately the software demonstrated its utility and flexibility in the analysis of 4 different cell types, extracting fluorescence timecourse, lineage and oscillatory information from the cells.

The segmentation and tracking systems are both built using a modular

approach, where different parts of the task are handled as separate steps. This allows different methods to be used as more sophisticated techniques become available. The first step in segmentation is cell detection which uses two very simple operations, a Gaussian convolution followed by maxima detection. This provides a very rapid cell detection which, as shown in Chapter 5, also leads to good accuracy. The high cell densities and rapid motion rule out methods which rely on cell positions or shape within a previous frame, as often utilised in active-contour segmentation and tracking systems. The cell detection can perform poorly if a bright cell overlaps a faint cell, where the Gaussian convolution can mask the maxima which would be present from the fainter companion. Situations such as these remain challenging to any segmentation method.

The tracking system is also implemented as a two stage modular system where the first stage builds the transition matrix while the second module assigns the trajectories based on the matrix. A similar approach is used by a number of other systems [46, 50, 57], where short tracks are often created then linked together to create full lineages. The current system solves the optimisation problem for each frame by creating single links connecting adjacent frames. It may be possible to solve the global tracking for multiple frames but this may be impractical for large populations where the computational cost will rapidly rise to unmanageable levels.

8.1 LineageTracker is a unique solution to HTS

The LineageTracker software described here attempts to solve a problem which is addressed by other software, such as CellProfiler. One disadvantage with existing systems is the lack of ability to correct mistakes made by the computer. There have been attempts at solving this problem by rejecting unreliable tracks [108] but this increases overall accuracy at the expense of volume of data. The interactive data viewer in LineageTracker allows any inevitable segmentation or tracking errors to be corrected, increasing both quantity and quality of data, the tradeoff being time required for analysis.

The analysis of circadian systems in single cells is reliant on the accurate measurement of fluorescent oscillators in those cells. While LineageTracker was in development, a dedicated circadian analysis package was released [85] which

was also based around ImageJ. While the latter software would seem to offer sufficient tools to enable it to be used in the C5Sys analysis, the handling of cell division was incomplete, with only a single daughter being followed, which would restrict its use where lineage construction is required.

New software has recently (September 2011) been made available which provides a similar solution. Cell Evaluator [141] utilises seed based or threshold based segmentation and tracking is provided by minimising a cost matrix in a similar manner as described in Section 3.4.8. This software works in a similar manner to LineageTracker: it is installed as an ImageJ plugin but largely operates as a self-contained application. Additionally it provides rudimentary cell editing to correct for errors made by the segmentation algorithms.

8.2 Performance and Accuracy

Chapter 5 describes the performance and accuracy of the methods developed and compares them to existing implementations. The tracking accuracy was found to be equivalent or better than the available alternatives, and when the option of manual intervention is considered, this increases the total possible accuracy of any results in favour of LineageTracker.

The LineageTracker software features the ability to use segmentation and tracking information from a range of sources, including CellProfiler and several ImageJ plugins, which will increase the functionality of both LineageTracker and the third party applications. Currently the design of the software requires that the tracking and division detection are part of the same module. Future versions of the software may allow different tracking and division modules to be selected independently. The matrix or graph holding the assigned trajectories will be made available to the division module which may then assign lineages a-posteriori.

Cell Motion

The cell motion analysis revealed two distinct motion types within one of the experiments, where half of the measured cells were moving in a persistent migratory style with smaller direction changes with the remaining cells revealing an untargeted random-walk motion.

There are some possible modifications which may be made to the tracking which could prove useful in certain situations. Currently when searching for a matching cell, no attempt is made to account specifically for cells dying, entering or leaving the frame, or re-connecting trajectories where gaps appear. The trajectory construction simply ceases if there is no suitable matching cell available. The reason for this was due to the high cell densities which led to multiple cell overlaps. It became difficult to determine whether a cell had disappeared or simply became temporarily obscured, and the large random component of the motion made re-linking trajectories problematic.

Different subpopulations were identified in Section 5.1 where cell motion was either migratory or seemingly random or untargeted. Additional experiments were analysed to determine whether this behaviour was widespread or particular to the original experiments. These results are presented in Appendix B.7, where 3 of the 4 extra experiments show a similar mixture of migratory and random motion.

This effect could be investigated further by preparing a line of knockout cells where aspects of the chemotaxis signalling pathway have been disabled and analyzing the motion of these cells using the same methods.

MSX1-GFP partitioning during division

The analysis of the *Msx1* expression in C2C12 cells revealed that asymmetric divisions were uncommon in vitro [142], with only a single such division observed. Fluorescent protein was very evenly distributed between daughter cells and the fluorescence recovery (measured as sum of daughter cell intensity compared to intensity of mother cell) was high, indicating that the fluorescence measurements were not affected by experimental or instrumental factors such as detector saturation during imaging.

Schizosaccharomyces pombe Lineage Construction

The LineageTracker software was used to track yeast cells expressing GAR2-GFP, a fluorescently tagged nuclear protein. The generation time was extracted from the constructed lineage and found to be 4.6 hours. The doubling time, as measured from the cell proliferation data, was slightly longer at 5.5 hours. A

significant fraction of the cells (32 from the initial population of 74) did not divide during the timecourse, which may account for the discrepancy.

Cell Cycle and the Clock

The cell cycle analysis in Chapter 7 investigated the connection between the cell cycle and circadian oscillators in mouse fibroblasts in vitro. The two oscillators were found to have different, and uncorrelated, periods, but there was evidence of divisions being gated since the majority of divisions occurred at a specific phase of the circadian system. If there was no connection the divisions would be more evenly distributed. This is in agreement with previous observations [136,139] and does not support a contradictory study [24] which concluded that mitosis occurs completely independently of the circadian clock.

Currently the experimental data used to draw these conclusions is limited to only a small number of time-series, where only a subset of cells divided during the course of the recording and also gave a full measurable cycle in all three oscillating components. The images were provided to determine whether the LineageTracker software would be suitable for extracting and analyzing the data. Future data may expand the study beyond the current pilot-scale.

8.3 Future Applications

Additional applications may be identified outside of the original goal of cell tracking. Tracking systems are often used in the study of microtubules in the cell cytoskeleton. Fluorescently modified tip-tracking proteins (such as 'End Binding Proteins' which bind to growing mictotubules [143]) allow growth to be monitored. Software such as PlusTipTracker [144] provides detection and tracking which has been tailored to the particular requirements of microtubule analysis. This software provides automatic identification and tracking but lacks an easy to use way of interacting with the results. The user is also unable to correct mistakes apart from rerunning analysis under different settings or excluding particular cells or regions of cells. It would be possible to provide support for microtubule analysis in Lineage tracker either by providing segmentation and tracking methods optimised for the task, or adding the ability to import tracking results from PlusTipTracker to allow better interactivity with

the results.

The software which developed during the project has been released as Open Source and can be downloaded from the University of Warwick website*, and is compatible ImageJ version 1.44 or newer.

The modular nature of the design and the plugin architecture allow other data analysis, segmentation and tracking methods to be added by following the specifications given in Appendix A.

*http://go.warwick.ac.uk/lineagetracker

Appendix A

LineageTracker Software

A.1 Overview of the Software

The LineageTracker software is written in Java and performs segmentation and tracking of cells, constructing lineages and extracting fluorescent timecourse data. It is run from within ImageJ and uses the functionality of the parent program to provide image loading, saving, a basic image processing library and visual display.

A.1.1 LineageTracker installs three ImageJ Plugins

The software is run by selecting one of three options in the ImageJ Plugins menu:

Segmentation: This allows creation of new experiments or editing of existing experiments by adding or removing image channels. Segmentation methods and parameters can be selected.

Create from Open: If the images have already been loaded into ImageJ (for example using the BioFormats importer), this plugin creates a new experiment then passes control to the 'Segmentation' plugin.

Experiment Viewer: This is the main viewing and editing window where lineages can be traced and fluorescence data can be exported. Segmentation and lineage editing takes place here.

A.2 Software Components

All cell information is stored in two data types: The *Cell* object, described below, stores measured features as recorded during the segmentation step. The cell/nuclear outlines are stored using the ImageJ *Roi* (Region of interest) object.

There are three public Interfaces which the software uses to perform task (Segmentation, Tracking and Analysis/Editing). Java classes which follow this specification will be loaded by LineageTracker and can be used to extend the functionality of the software by adding additional features. These interfaces (which are all implemented as abstract classes) describe the methods which must be present within the class – these methods will be called by the software as required to perform the functions.

A.2.1 Data is stored in a *Cell* object

These hold the position and size of cell (frame, x,y, width, height, area, major and minor axis and angle), the intensity of each channel as well as texture features on first channel.

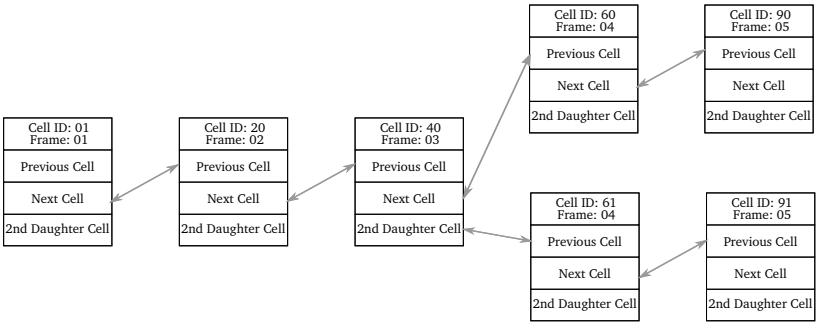


Figure A.1: Storing tracking and lineage information in the Cell objects

The Cells store lineage information as pointers to other cells: cell in previous frame, next cell in tracking, 2nd daughter cell (if any). The first daughter is stored as the continuation of tracking (see Figure A.1). During tracking the cell trajectory is stored as a Doubly-Linked List [145] where each node holds pointers to both the next and previous nodes. Branches are populated when a cell division is stored, giving the form of a 'Sparse Tree', unlike a traditional binary tree where every node is connected to two branches.

A.2.2 Writing Segmentation Plugins

The `ProcessingPlugin` abstract class provides the following public methods and fields:

`protected transient ImagePlus segmentFrame;`

Holds the sum of all image channels for the particular frame.

`protected List<ImageProcessor> allChannels;`

Implemented as an ArrayList, which holds each image channel (for methods which handle channels separately)

`protected int channels;`

Counter holding how many channels were loaded.

`public float[] getPixelVector(int x, int y, int radius)`

Returns an array holding the average pixel intensities in a (2r+1) sized box for each image channel.

`public float[] getPixelVector(int x, int y)`

Returns the intensities at a single pixel location.

`public static[] float normVect(float[] v)`

Calculates a unit vector in the same direction as v[]

`public static double vectorDistance(float[] p1, float[] p2)`

Calculates the distance between two points

`public static double colourDistance(float[] fg, float[] fg2)`

Calculates the colour difference between two pixels, whose channel intensities are stored in the two arrays.

`public static double sumColours(float[] v)`

Returns the sum of the intensity components in v[]

The following methods must be written by the user, to implement the desired functionality of the plugin:

`public abstract String getName();`

Returns the name of the plugin, as appears in the list of available segmentation methods, or `null` if the method is unavailable.

`public abstract void runSegmentation(int frameNo);`

Perform the actual segmentation and create a cell mask of cell or nucleus shapes.

`public abstract ImagePlus cellMask();`

Return the segmentation mask prepared during runSegmentation()

`public abstract boolean colourCoded();`

Returns true if the segmentation mask uses pixel values to identify different cells. If false, individual cells cannot be touching. If true, pixel number is used to separate touching masks.

`public abstract boolean editSettings(Frame parentWindow);`

Display a dialog box to edit the segmentation parameters. Return true if 'OK' was selected.

A.2.3 Writing Tracking Plugins

All tracking methods must extend the AbstractTracker class. This provides the following fields and methods:

`protected static String dir;`

The root directory where the experiment data is stored.

`protected static String exptName;`

The name of the experiment.

`protected static String trackingFile;`

The filename to save the tracking data (or `null` or empty string if not saved).

`protected List<Cell> cellsToTrack;`

The full list of cells within the experiment.

`protected ExptData expLoader;`

All of the experiment details, including image scale and number of frames.

`protected TrackingParameters tp;`

Tracking parameters, such as thresholds and weights, if required.

The user must implement the following methods:

`public TrackingParameters getTrackingParameters()`

Optional: returns the tracking parameters used. Override this (to return null) in methods which do not use parameters.

`public boolean trackable(File[] filesinDir)`

Optional: For trackers which load tracking data instead of implementing a tracker themselves, this method returns `true` if there is tracked data in the correct format, otherwise return `false`.

`public abstract String getName();`

As above, return name or `null`.

`public abstract List<Cell> run() throws IOException;`

Run the tracking and return the new list of cells, where the cell linkages have been connected to provide the tracking and lineages.

A.2.4 Writing Data Analysis & Manipulation Plugins

Plugins which extend the `AnalysisPlugin` class are called from the experiment viewer. They have full access to the data and display and can perform any data analysis or manipulation. The following fields and methods are provided:

`protected ExptData exp;`

All of the experiment details, including image scale and number of frames.

`protected List<Cell> cellsToTrack;`

The full list of cells within the experiment.

`protected Image5DWithOverlay screen;`

Holds the Image5D representation of the cell images and provides methods for plotting and annotating the display.

`public Point lastClickedPosition;`

The last-clicked position within the image window.

`public boolean acceptsDummyCells = false;`

The plugin needs to set this flag to `true` if the analysis methods can be run using arbitrary positions in the window. Otherwise the methods will only be run if the user clicks within a cell.

`protected Map<Integer, List<Roi>> roiLists;`

Holds the ROIs for all cells in the experiment.

`public abstract String getName();`

As above, return name or `null`.

`public abstract void setup();`

Called when the 'Setup Plugin' button is clicked, allowing any plugin parameters to be adjusted. Often used to display a help screen for plugins which do not have editable parameters.

`public abstract void analyze(Cell currentCell);`

Called when the 'Analyse' button is clicked in the control panel. Holds the last clicked cell, or `null` if no cells have been clicked on yet.

`public abstract void cellClicked(Cell currentCell);`

Called every time a cell is clicked on in the window.

A.2.5 Global methods available to all Plugins

The methods described here are a selection of method calls which are available to all plugins.

`static boolean LineageTracker.newerThan(java.lang.String version)`

Checks whether the current version is newer than the given one.

`static Roi RoiTools.findRoi(int x, int y, java.util.List<Roi> roiList)`

Returns the Roi from the list which contains the point (x,y)

`static double TrackedCell.findDistance(Cell c1, Cell c2)`

Calculates the distance between two cells.

`static Cell TrackedCell.findNearestCell(frame, x, y, List cellList)`

Find the cell nearest to the given co-ordinates.

A.2.6 Example of a Threshold based Segmentation Method

The threshold segmentation method below demonstrates the implementation of segmentation plugins.

```
package segmentation;

import ij.*;
import ij.process.ImageProcessor;
import java.awt.Frame;
/**
```

```
 * Simple example of writing a Processing Plugin.
 * @version 14-Jan-2011
 * @author Mike Downey
 */
public class ThresholdSegmentation extends ProcessingPlugin{

    ImageProcessor toSegment;
    ImagePlus mask;

    @Override
    public void imageToSegment(ImageProcessor img) {
        toSegment = img;
    }

    @Override
    public ImagePlus cellMask() {
        return mask;
    }

    @Override
    public boolean colourCoded() {
        return false;
    }

    @Override
    public void runSegmentation(int frameNo) {
        // Take the ImageProcessor and convert into a binary image.
        ImagePlus imp = new ImagePlus("temp",toSegment);
        IJ.setAutoThreshold(imp, "Li dark");
        IJ.run(imp, "Convert to Mask", "");
        mask = imp;
    }

    @Override
    public String getName() {
        return "Simple Auto Threshold";
    }

    @Override
    public boolean editSettings(Frame parentWindow) {
        IJ.showMessage("No settings to edit.");
        return false;
    }
}
```

A.2.7 Example of a Simple Tracking Method

The Simple nearest tracking method is a very crude implementation designed to demonstrate writing a tracking plugin.

```
package tracking;

import util.Cell;
```

```
import util.GenericDataLoader;
import java.io.IOException;
import java.util.List;

/**
 * Locates the nearest cell in the next frame and if it hasn't already
 * been assigned into a trajectory, add it to the current cell.
 * Simply presented as a demonstration and not intended for actual use.
 * This plugin is very slow when used with large cell populations.
 * @author Mike Downey
 * @version 16 Feb 2011
 */
public class SimpleNearest extends AbstractTracker{

    @Override
    public List<Cell> run() throws IOException {
        // Load in the segmented cell data
        cellsToTrack = GenericDataLoader.loadData(expLoader);

        int storeOld = TrackedCell.maximumCellSeparation;
        TrackedCell.maximumCellSeparation=50;

        for(Cell c : cellsToTrack){
            int f = c.getFrame();
            if(f<expLoader.getFrames()){
                Cell nearest = TrackedCell.findNearestCell(
                            f+1, c.getX(), c.getY(), cellsToTrack);
                if(nearest!=null && nearest.getPreviousCell()==null)
                    c.setNextCell(nearest);
            }
        }
        TrackedCell.maximumCellSeparation = storeOld;
        return cellsToTrack;
    }

    @Override
    // Method does not use tracking parameters so return null.
    public TrackingParameters getTrackingParameters() {
        return null;
    }

    @Override
    public String getName() {
        return "Simple Nearest Cell";
    }
}
```

A.2.8 Examples of a Data Analysis Plugin

This plugin demonstrates the AnalysisPlugin interface and interaction with the image window.

```
package analysis;
```

```
import analysers.AnalysisPlugin;
import ij.IJ;
import ij.gui.GenericDialog;
import java.awt.Color;
import util.Cell;

/**
 * Plot the cell number alongside each cell
 * @version 04-May-2011
 * @author Mike Downey
 */
public class NumberCells extends AnalysisPlugin {

    double fontSize=12;
    int[] cellsInFrame;

    @Override
    public void setup() {
        GenericDialog gd = new GenericDialog("Select Text Size");
        gd.addNumericField("Size in Pixels", fontSize, 0);
        gd.showDialog();
        if(gd.wasCanceled())
            return;
        if(gd.wasOKed()){
            fontSize = gd.getNextNumber();
        }
    }

    @Override
    public void analyze(Cell currentCell) {
        countCells();
        for(Cell c: cells)
        cellClicked(c);
        for (int i = 0; i < cellsInFrame.length; i++) {
            IJ.log(Integer.toString(i + 1) + " , " + cellsInFrame[i]);
        }
    }

    /**
     * Plots the cell number.
     * @param currentCell
     */
    @Override
    public void cellClicked(Cell currentCell) {
        if(screen!=null){
        screen.plotNumber(currentCell.getFrame(), currentCell.getX(),
            currentCell.getY(), currentCell.getCellID(),
            fontSize/10, Color.white, false);
        }
    }

    @Override
    public String getName() {
```

```
        return("Draw Cell Numbers");
    }

    private void countCells() {
        cellsInFrame = new int[exp.getFrames()];
        for (Cell c : cells) {
            cellsInFrame[c.getFrame() - 1]++;
        }
    }
}
```

A.3 Data Structure and Data Storage

Cellomics datasets are based around 96-well plates with wells labelled from A1–H12 and a variable number of fields per well. All wells have a unique WellID which identify them within the experiment. Each image is assigned a FieldID which increases for each time-point in the experiment so, for example, Field 01 in Well C04 at timepoint 1 will have a different FieldID to the same well and field at a subsequent time-point.

A.3.1 The Cellomics Database

The Cellomics database is stored as a *Microsoft Access* file. There are two versions, depending on whether the level of processing which has been performed on the experimental data. The first version of the database contains the experiment conditions and references to all of the image data. The second version is an extension of the first with additional cell segmentation and intensity information. A list of main database tables is given in Table A.1.

A.3.2 The Cellomics Image Format

Cellomics 'DIB' files were stored as uncompressed 16-bit data with a 52 byte header which holds the image details as 2 or 4 byte words (see Table A.2). Individual pixels were stored as two bytes. All numbers are stored as 'little-endian', i.e. least significant byte followed by most significant byte.

A.3.3 Accessing LineageTracker Experiment Data

The experiment data (as stored in the ExptData object) is saved as an `xml` file. This data is available to plugins using the methods described in Table A.3. Any

Table	Contents
asnProtocol	Number of image channels and objective lens used.
asnPlate	Experiment start time, number of wells in experiment and number of fields per well.
FImage	Filenames, Well ID and Field number for all images.
wField	WellIDs for each well used in the experiment.
Cell	For every segmented cell, holds the FieldID, CellID, cell position and size within the frame.
CellFeature	Holds multiple rows for each cell holding the cell intensity with one row per cell per image channel.
AsnFeatureType	Holds the 'key' values which identify which rows of the CellFeature and asnWellFeature tables holds the cell number and WellID number.
asnWellFeature	Holds the WellID numbers which for each image channel.

Table A.1: Major tables within the Cellomics database

Offset	Size (bytes)	Data
4	4	Image Width (pixels)
8	4	Image Height
14	2	Bits per pixel
24	4	Scale (pixels per meter)

Table A.2: Header structure for the Cellomics DIB format. Missing values are either unpopulated or ignored by the software.

methods in *italics* are used internally during loading and saving data and should not be called from plugins.

A.3.4 Positions and intensity values are stored in text files

The cell and tracking information is stored in a series files as described in Tables A.4 and A.5 where the names begin with the experiment name followed by an underscore then one of: `celldata.txt`, `intens.txt`, `tracked.txt`.

There are two versions of the tracking data file: the original version which identifies tracked cells using Cell ID, the second version uses cell positions and is used when segmentation has changed.

Parameter	Getter	Setter
Storage for any additional data generated by plugins. The must be Serializable	retrieve(String)	store(String,Object)
Height of pixel Width of pixel Measurement units for height and width	getPixelWidth getPixelHeight getScaleUnits()	setPixelHeight(double) setPixelWidth(double) setScaleUnits(String)
Width of frame (pixels) Height of frame (pixels)	getWidth() getHeight()	*setWidth(int)* *setHidth(int)*
Number frames in experiment Frame interval Interval Description	getFrames() getFrameInterval() getIntervalUnits()	*setFrames(int)* setFrameInterval(double) setIntervalUnits(String)
Channel names (start at 0) Number of Channels	getChannel(int) getNChannels()	 *setNChannels(int)*

Table A.3: Accessing the experiment data

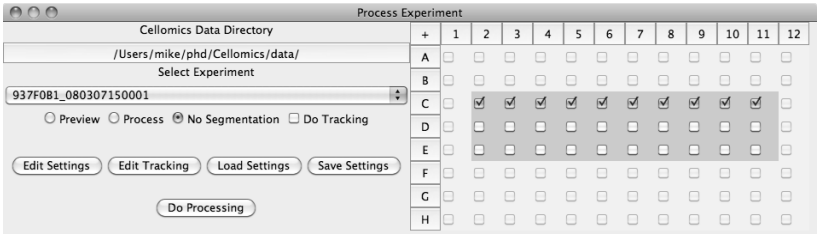
`_cellintens.txt`		`_celldata.txt`	
Column	Data	Column	Data
1	Cell ID	1	Cell Area
2	Frame number	2	'Validated' flag
3–9	First 8 intensity channels	3	Major axis length
10	Y	4	Minor axis length
11	X	5	Major axis angle
12	Height	6	Circularity
13	Width	7	2^{nd} Intensity moment
14	unused (holds 0)	8	Median Intensity
15	unused (Cell ID)	9	Standard Deviation
		10	Kurtosis of Intensity

Table A.4: Cell feature data file formats

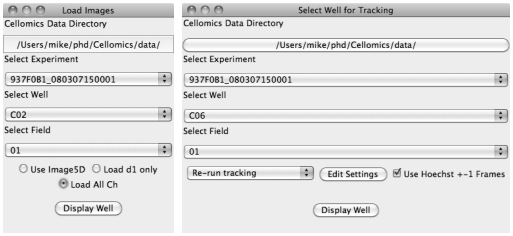
`_tracked.txt`		`_modified.txt`	
Column	Data	Column	Data
1	Frame number	1	Frame number
2	Cell ID	2–3	X,Y Position
3-4	X,Y position	4	Frame no. of 'Next'
5	'Next' Cell ID	5–6	Daughter X,Y
6	Daughter Cell ID	7	Frame no. of Daughter
7	Movement Score	8	Movement Score

Table A.5: Tracking data file formats. 'NA' is used in any columns where there is no tracking information.

A.4 The Cellomics Experiment Viewer Software



a) Select Wells for Segmentation



b) Image Loader c) Tracking Viewer

Figure A.2: The user-interface for Segmentation, Tracking and Viewing the Cellomics experiment data.

A.5 An Example of using LineageTracker

After obtaining the images, there are several steps required before any fluorescent timecourse data can be extracted.

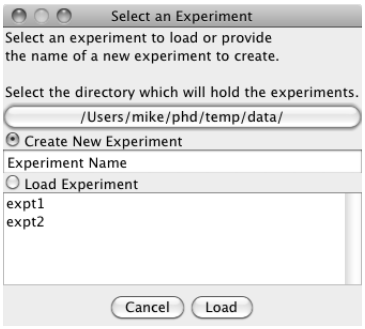
This section describes a worked example from image data through to obtaining a spreadsheet holding timecourse data, including several steps describing manual corrections which may be required if under challenging circumstances.

The first step is to create the 'LineageTracker' project on disk.

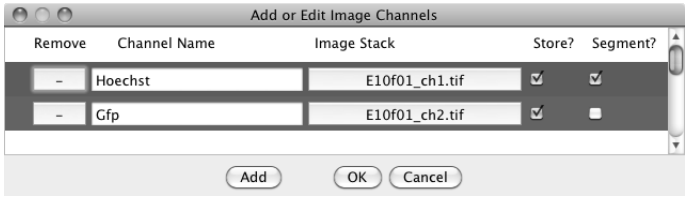
There are two ways of doing this, depending on whether the images are already loaded into ImageJ:

1. If the images are not loaded, select 'Segmentation'. When the 'Select Experiment' dialog box appears (Figure A.3a), choose the directory where you wish to store the data, along with a name for the experiment. After clicking on OK, the 'Edit Channels' dialog box will appear where you can add the image channels to the experiment (Figure A.3b).

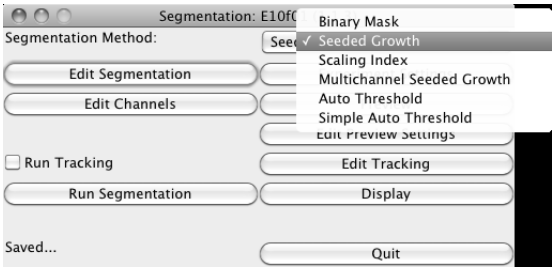
2. If the images are already in ImageJ, select 'Create From Open' from the menu, in ImageJ→Plugins→LineageTracker. If any of the images contain multiple image channels or z-stacks, you will be prompted to split the channels or run a maximum projection.



a) Select an Experiment



b) Edit Channel dialog box



c) Choose segmentation method

Figure A.3: The LineageTracker user-interface for segmentation and editing image channels

The 'Edit Channels' dialog box is used to add, remove or rename image channels. To add a channel. click on the 'Add' button, which will add a blank line to the window. Then click on the button in the 'Image Stack' column and select the images. If the 'Store' tick-box is selected, the intensities of this channel will be recorded during the 'Segmentation' step. If the 'Segment' box is ticked, this image channel will be passed to the segmentation module: segmentation

based on a single channel will sum together all ticked channels, methods which are aware of multi-channel data will receive all image channels.

When the 'Segmentation' window appears (Figure A.3c), the first thing to do is select the segmentation method.

Clicking on 'Edit Segmentation' opens up a dialog box to change any parameters used by the chosen segmentation method, for example the 'Seeded Growth' method has the following settings:

- Segmentation Settings Noise Tolerance: Intensity of a peak before it is detected by the ImageJ 'Find Maxima' method.
- Growth Threshold: Smaller values allow lower intensity pixels during the 'growth' of the cell. Values are between 0–1.0
- Growth Iterations: The number of passes of the 'grow' method. Sets the maximum size which will be created by the method.
- Blur Radius: The size of the Gaussian kernel applied to the image before the 'Find Maxima' is called. Larger values will suppress noise but may prevent small cells from being detected.
- Sharpening Factor: An Unsharp Mask may be applied to separate touching cells. This value is the 'weight' of the mask. Recommended values are between 0-0.7
- Smooth before growth: If selected the seeded growth is performed on the Gaussian-smoothed image, otherwise the growth is performed on the original image.

Before clicking on 'Preview', click on 'Edit Preview Settings' and choose the speed (or number of frames to skip), otherwise the preview will take the same amount of time as a full segmentation. Use the 'Preview' and 'Edit Segmentation' buttons to obtain acceptable cell outlines.

To run the tracking, make sure the 'Run Tracking' tick-box is selected, then click on 'Edit Tracking' to choose the tracking method:

Run Tracking is the method optimised for the C2C12 Hoechst-stained cells.

Tracking (No Divisions) is a general-purpose tracking method based on 'Run Tracking' but with cell divisions removed.

Minimum Cost Tracking is a general-purpose tracking without automatic cell division detection, based on the Hungarian Algorithm.

The 'Edit Settings' dialog box will then appear to allow the tracking parameters to be modified. These include the threshold and weight values which control how cells are identified during tracking. The most important parameters are:

Distance Threshold – controls how far away the tracking will look to locate cells. Use higher values if cells are moving rapidly.

Intensity Weights – If one image channel contains a permanent stain (such as Hoechst), give this channel a higher weight.

Clicking on 'Run Segmentation' will perform the cell detection and tracking. When that has finished, click on 'Display' to open the experiment viewer (Figure A.4).

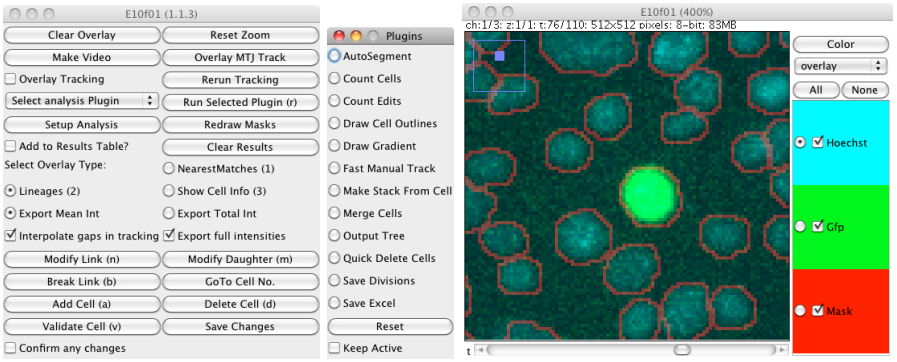
The initial segmentation and tracking may have missed some cells so before exporting any fluorescence timecourse data, it is recommended to check any cells of interest first. Click on a cell to select it, then move through the time-series to check that the tracking is correct and that the cell outline is accurate. If the tracking needs fixing, click on the cells which need linking together and then click on 'Modify Link' or 'Modify Daughter'.

If the cell outlines has been either missed or drawn incorrectly, this can be corrected by deleting the old outline (by clicking on the cell or drawing an ROI outline around one or more cells, and pressing 'Delete'). The new outline can be added in one of two ways:

- Drawing an outline ROI such as oval or freehand, then clicking on 'Add Cell'
- Selecting the 'AutoSegment' plugin then clicking on the cell (see Figure A.5). Adjust the sliders to get an accurate outline.

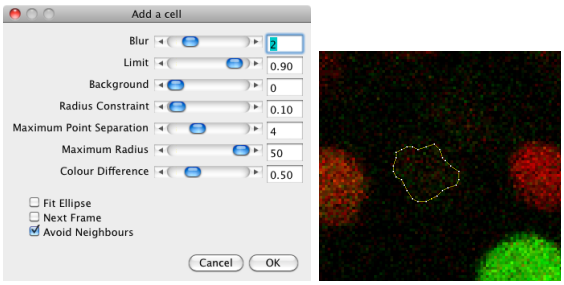
Fluorescence timecourse information for individual cells is exported by selecting 'Add to Results Table', then clicking on the cell of interest. The results table can then be saved to disk. Remember to close or clear the results table before clicking on the next cell.

If any changes have been made to the cells or tracking, don't forget to click on 'Save Changes' before closing.



a) Main control panel b) Plugins c) Experiment Viewer Window

Figure A.4: The user-interface for the LineageTracker Interactive Experiment Viewer, including the data analysis and editing plugins.



a) Add a cell b) Cell Outline

Figure A.5: Adding individual cells using the 'Auto Segment' plugin to determine the cell outline. This method can outline cells which are too faint to be detected using the fully automatic Seeded Growth method. Selecting 'Next Frame' will move to the next frame and automatically add a tracking link between cells if 'OK' is selected.

Control	Function
Control Panel Window:	
Clear Overlay	Remove any overlaid highlighting from the window.
Reset Zoom	Reset to 100% magnification.
Make Video	Create a time-series image complete with highlighted cells and tracks.
Load MTJ Track	Load and overlay an MTrackJ trajectory.
Overlay Tracking	When a tracked cell is selected, draw the full trajectory from current timepoint onwards.
Rerun Tracking	Allows tracking to be run with a different method or parameters.
Select Analysis Plugin	If any additional analysis plugins have been installed, they can be selected from here. When a cell is clicked in the viewer window, the plugin will be called for that cell.
Do Analysis	Call the selected analysis plugin for all cells in the experiment.
Setup Analysis	Edit any settings (if any) for the analysis plugin. Also used to display a brief description of the plugin.
Add to Results Table	When a cell is selected, add fluorescence intensity data to an ImageJ Results Table
Clear Results	Clear any currently open Results Tables.

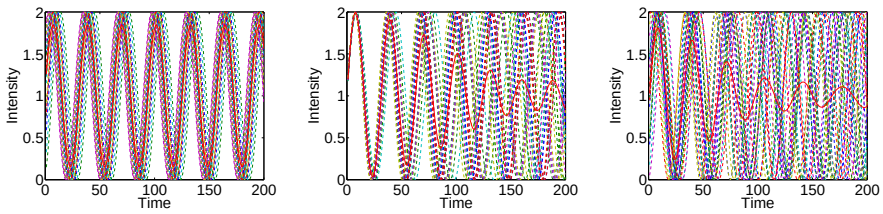
Control	Function
Overlay Types: the next 3 buttons control the information displayed or added to a results table when a cell is highlighted in the window.	
1) Nearest Matches	Overlay the Movement Scores for trajectories leaving the highlighted cell.
2) Lineages	Display tracking, including daughter cells.
3) Cell Info	Display fluorescence information for current timepoint.
Export Total/Mean Int	Selects between exporting the integrated or mean cell intensity to the results table.
Interpolate Gaps	Interpolates the intensity if tracking skips a frame.
Export Full Intensities	ON: Exports the full lineage intensity for daughter cells
	OFF: Only export intensities from division onwards for daughter cells.
The next three buttons modify tracking based on the previous two highlighted cells. To alter the link between two cells (in adjacent frames), click on one cell then move to the next frame and click on the second cell. Finally select one of the following buttons.	
Modify Link	Add the two selected cells as a trajectory
Modify Daughter	Add a trajectory, marking it as a cell division. If the parent cell already has a tracked next cell, that is marked as the other daughter cell.
Break Link	Remove the tracking between the two selected cells.
Go To CellID	Select a cell based on its ID number
Save Changes	Save any segmentation or tracking changes.
Add Cell	Expects a closed ROI (such as oval or polygon). Calculates the cell features and adds a new cell.
Delete Cell	Deletes the currently selected cell.
Validate Cell	Marks the currently selected trajectory as validated. The validated cells are drawn with filled-in masks for easy identification.
Confirm Changes	Prompts the user before any actions which change the cell data.

Control	**Function**
Cell Image Window	
The tick-boxes show or hide the following image channels.	
Time t	The primary fluorescence channel
Mask	Segmentation outlines
GFP	Any additional fluorescence channels.
Plugins Window	
This window lists all available Analysis Plugins and allows easy selection between them. Two plugins come built-in to the software.	
AutoSegment	After selecting this and clicking on a cell in the image window, the plugin will attempt to calculate the outline of the cell and add it to the experiment data. A dialog box appears allowing the segmentation parameters to be adjusted.
Merge Cells	Requires an ROI to be drawn in the image window. When Run Selected Plugin is clicked, any cells within the ROI will be merged together.
Reset	Un-selects the current selected plugin.
Keep Active	If un-selected, the current plugin will be used for the next click in the cell image window. Selecting this will allow the plugin to be run several times on different cells.

Appendix B

Additional Material

B.1 Whole Frame Intensities require good cell synchronization



a) Constant λ b) Varying λ c) Phase and λ varying

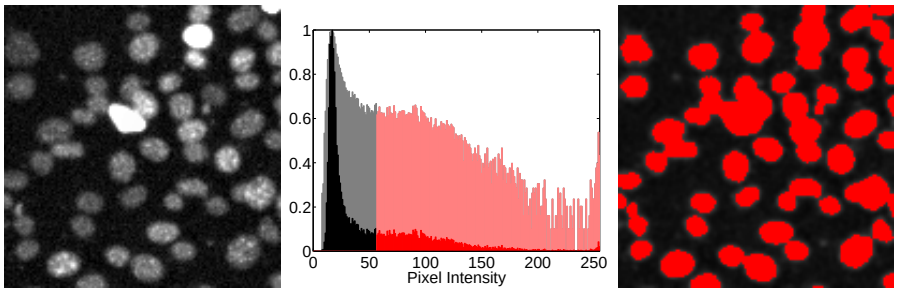
Figure B.1: Comparison of single cell fluorescence measurements with whole frame intensities. 20 individual cell intensities are plotted using dashed lines with the mean intensity plotted in a solid red line. a) All cells oscillate with the same frequency but different phase ($\pm\frac{\pi}{4}$). b) Different frequencies (varying by $\pm 7.5\%$ of base frequency, normally distributed). Cells start off in phase. Amplitude of oscillations appears to decay. c) Phase and frequency both changing.

B.2 Description of Segmentation Methods used in the Performance Testing

The Seeded Growth and Scaling Index methods were developed as described in Materials & Methods chapter. The threshold methods were part of the standard distribution of ImageJ or Fiji.

Threshold Segmentation

The image was de-noised by convolving with a Gaussian kernel ($\sigma = 0.9$) followed by background subtraction (rolling ball background from ImageJ,



a) b) c)

Figure B.2: Using thresholding to separate cells from the background. a) The original black & white cell images. a) The histogram of this image (standard histogram in black, log histogram in grey), segmented image. Pixels above the threshold are shown in blue on both the histogram and the segmented image. The histogram shows a dip between the background pixels (peak on left) and the cell pixels. This intensity value is used as the threshold. c) Thresholded image.

radius = 50 pixels). For the global threshold, a different threshold value was used for each frame. A binary image was produced where pixel intensities lower than the threshold were set to zero. Finally the ImageJ watershed transform was applied to separate touching or clumped cells. The binary images were then size-filtered to remove particles or debris smaller than 10 pixels in area.

Local threshold segmentation followed the same procedure with Niblack adaptive thresholding (available with the Fiji distribution of ImageJ) applied to each frame.

Seeded Growth

A Gaussian blur ($\sigma = 2$) was applied to the image then centres of the nuclei were detected by locating local maxima using the ImageJ maxima finder, which takes a 'tolerance value' N, and locates individual peaks or plateaux with intensity at least N greater than neighbouring minima.

Scaling Index

After background subtraction the scaling index method was run with radii $r_1 = 3$ & $r_2 = 9$. The nuclei appeared as low dimension objects. Each frame was thresholded, followed by the Watershed transform to separate touching cells. The resulting segmentation mask was filtered according to size using the same method employed in the threshold segmentation.

CellProfiler

The images were convolved with a Gaussian kernel ($\sigma = 1$ pixel) followed by background subtraction (block size=50 pixels, polynomial smoothing). Cells were identified using the Background Adaptive threshold method. Clustered cells were split apart using the 'Shape' method which uses a distance-transform of the binary image to detect touching objects.

B.3 Tracking Parameters used in performance tests

Tracking Parameter	*Threshold*	*Weight*
Distance	9.94 pixels	0.70
Area	45%	0.19
Intensity	51%	0.45
Intensity Moment	46%	0.50
Standard Deviation	48.5%	0.30

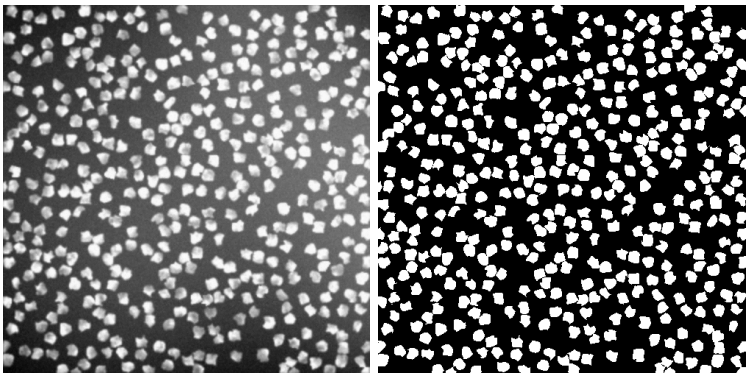
Table B.1: Tracking Parameters (Thresholds and Weights)

B.4 Generating Artificial Cell Images using SIMCEP

The SIMCEP* software is controlled by a script file containing the parameters which specify the number, size and shape of the cells or nuclei. The `imgLevel` parameter in the following script is the frame number of the 5 image Gold Standard set. The population size was chosen to closely match the manually annotated Gold Standard.

```
population.template = ones(500);
% Set window size to 500, then increase to 512 in ImageJ - to avoid
% having cells at the edge of the screen which will be handled
% differently in different segmentation methods.

% Number of cells simulated in the image
population.N = 200+225*sqrt(imgLevel);

% Amount of clusters
population.clust = 0;

% Probability for assigning simulated cell into a cluster. Otherwise
```

*http://www.cs.tut.fi/sgn/csb/simcep/tool.html



a) Artificial Cell Image b) Ground Truth Image

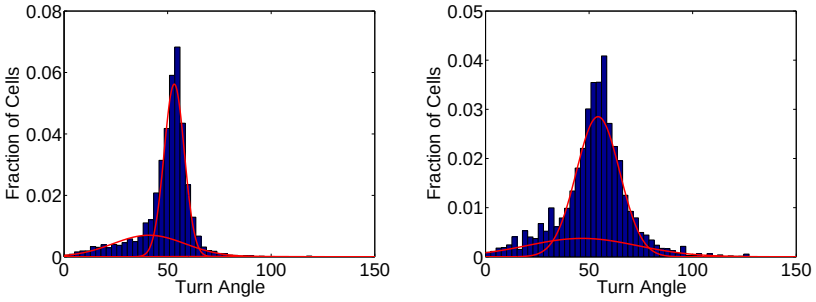
Figure B.3: Images generated using SIMCEP to test segmentation methods. The Ground Truth is simply a binary image with no delineation between cells so there is no way of accurately assigning boundary pixels at the overlap of touching cells. *Adapted from Downey et al. PlosOne (2011)*

```
% cells are uniformly distributed on the image.
population.clustprob = 0;

% Variance for clustered cells
population.spatvar = 0;

% Amount of allowed overlap for cells [0,1]. For example,
% 0 = no overlap allowed and 1 = overlap allowed.
population.overlap = 0;

% Is the overlap measured on nuclei (=1), or cytoplasm (=2)
population.overlap_obj = 1;

% Parameters for the measurement system

% Energy of illumination compared to the energy of cells
measurement.illumscale = 0.25;

% Misalignment of illumination source in x and y direction
measurement.misalign_x = 0;
measurement.misalign_y = 0;

% Energy of autofluorescence compared to the energy of cells
measurement.autofluorscale = 0.25;

% Variance of noise for ccd detector
measurement.ccd = 0.001;

% Amount of compression artefacts
measurement.comp = 0.0;

% Change the first lines in generate_measurement.m to read:
```

```
%   O_S = measurement.kernel;
%   O_V = measurement.variance;
% These parameters control the position in the focal plane.
measurement.kernel = 5;
measurement.variance = 0.1;

% Is cytoplasm included in the simulation ( 0 = no, 1 = yes)
cell_obj.cytoplasm.include = 0;

% Is nucleus included in the simulation ( 0 = no, 1 = yes)
cell_obj.nucleus.include = 1;

% Nucleus radius
cell_obj.nucleus.radius = 8;

% Parameters for random shape
cell_obj.nucleus.shape = [0.1 0.5];

% Parameters for texture: persistence, 1st octave, last octave,
% and intensity bias
cell_obj.nucleus.texture = [0.5 2 5 0.2];
```

B.5 Angle distributions change during the experiment

The turn angles of the cells were measured during the experiment used to create the Gold Standard 2 test set. The distribution of the angles changes during the experiment is shown in Figure B.4. The cell density increases (Figure B.5) which will restrict the available space for the cells to move into.



a) First Quartile b) Last Quartile

Figure B.4: The distribution of turn angles changes during the Gold Standard 2 experiment.

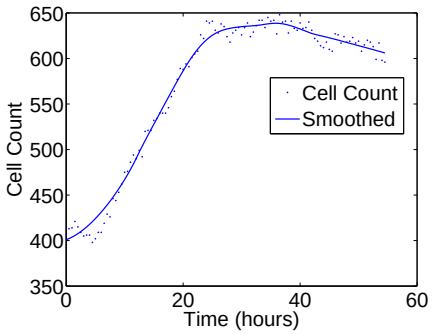


Figure B.5: Cell counts for the experiment used to construct the Gold Standard 2 test set. The population increases rapidly until confluence is reached and cells eventually start to die off.

B.6 Distance-Pathlength Heatmaps as an alternative view of cell motility

An alternative method of visualising the cell motion was developed, using a scatter plot where total distance travelled was plotted against the straight-line

distance moved since the start of measurement. A cell which moved in a straight line would lie on the diagonal of such a plot, where walk length = distance from origin.

A random walk with where both step size and direction are chosen from a uniform random distribution is shown in Figure B.6a, with a heatmap representation in Figure B.6b to show the plot density more easily. In this plot, the cells 'diffuse' out from the origin where the mean distance from the origin is proportional to the square root of the time. In the simulation presented here, the cell step size was randomly chosen in the range 0–10 and measured over 50 frames. The angle change was randomly selected from the full range 0–360°. The mean total distance covered by the cells will therefore be

$$50 \times \frac{(10-0)}{2} = 250$$

while the distance moved away from the origin will be given by

$$\sqrt{50 \times \left(\frac{10-0}{2}\right)^2} = 35.4$$

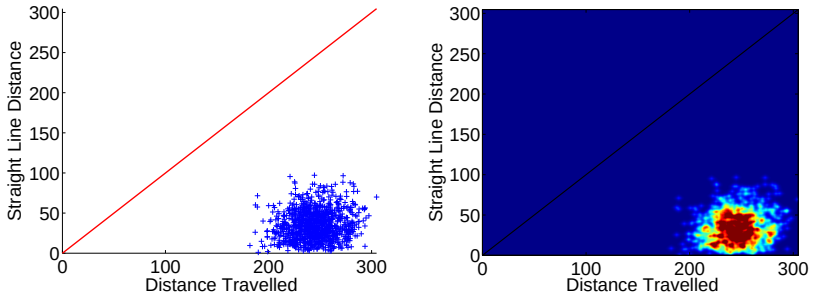
The mean distance travelled for the simulated cells in Figure B.6a–b is 244.9 while the mean straight line distance is 35.2, which agree closely with the values calculated above.

The Distance-Pathlength plots for the two Gold Standard data-sets are presented in Figures B.6c–d. These show noticeably different behaviour, where Set 1 lies closer to the diagonal, indicating that the cells moved in straighter paths than Set 2, which had more of a 'horizontal' component, suggesting more direction changes.

An attempt was made to simulate this behaviour by modelling different types of random walk where the step and direction changes were limited. In Figure B.6e, the cells were initialised with a random direction and with a speed randomly selected from a Gaussian distribution ($\mu = 4, \sigma = 4$). For each step, the angle was varied by adding a random amount of ± 0.5 radians. The speed was chosen from a Gaussian distribution with μ =original speed and $\sigma = 0.5\times$ original speed. The Distance-Pathlength plot followed a similar pattern to the Gold Standard 1, with most cells closely following the diagonal.

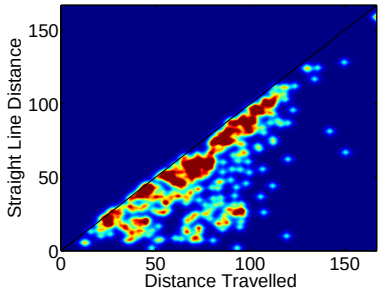
A second motion-constrained random walk is shown in Figure B.6f. The step size was kept constant, with the initial values selected from a uniform

distribution. The angles were changed by ±0.5 radian (± 29°), again from a uniform distribution. These later simulated walks show similar properties to the measured cell motion.

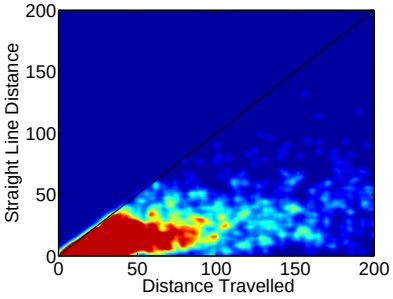


a) Random Walk scatterplot

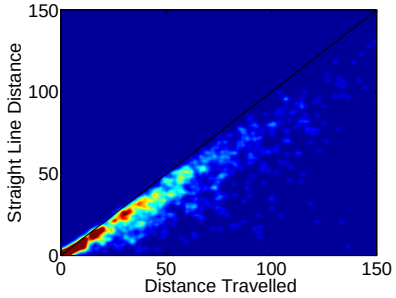
b) Random Walk heatmap



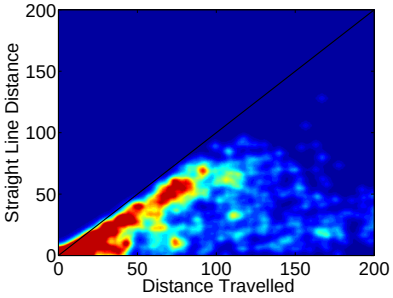
c) Gold standard 1 (10 minute intervals)



d) Gold standard 2 (30 minute intervals)



e) Gaussian random walk



f) Constrained random walk

Figure B.6: Distance-Pathlength plots for different classes of random walks and the two Gold Standard tracked sets. Points on the diagonal line indicate particles which have moved in a straight line throughout the simulation. a) & b: Classic random walk with unit length displacement in a random direction. c) Results from Gold Standard 1. d) Results from Gold Standard 2. e) Random walk where the change in angle and speed are randomly selected from a Gaussian distribution. f) Random walk where the step size was kept constant but angle changes by a random amount up to ± 0.5 radians.

B.7 Angle Distributions from additional experiments

Section 5.1 presented two experiments and measured different distributions of turn angle, where one set of cells had predominantly large turn angles whereas the second set had a subpopulation with smaller turns, suggesting that some cells were moving in straighter lines.

Four additional experiments were measured and the angle distributions are presented below in Figure B.7. Three of the experiments displayed similar 'straight line' subpopulations while the experiment shown in Figure B.7b was predominantly composed of turning cells.

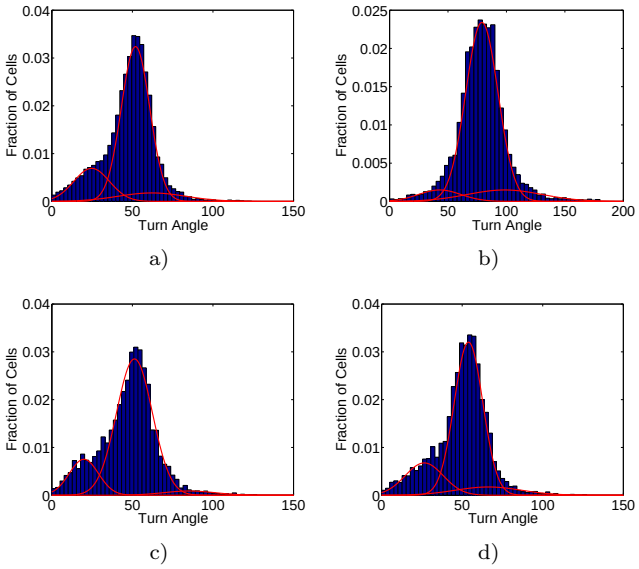


Figure B.7: Four additional cell experiments were tracked and analysed. A 3-component Gaussian fit is superimposed on each. Frame interval was 10 minutes in all cases.

B.8 Graphical representation of *Sz.pombe* Lineage Trees

The lineage trees are drawn using the open source Graphvis software, which creates graphs and trees based on a script which describes the nodes and connections.

The lineage tree in Section 6.3.1 was created from the following script:

```
digraph lineage {
    cell2 [label="0.0 minutes" ,shape=box]
    cell3418 [label="170.0 minutes"]
    cell2 -> cell3418
    cell14527 [label="480.0 minutes"]
    cell3418 -> cell14527 [label="310.0"]
    cell27306 [label="715.0 minutes" ,style=dotted]
    cell14527 -> cell27306
    cell17545 [label="540.0 minutes" ,style=dotted]
    cell14527 -> cell17545
    cell12037 [label="425.0 minutes"]
    cell3418 -> cell12037 [label="255.0"]
    cell24626 [label="670.0 minutes"]
    cell12037 -> cell24626 [label="245.0"]
    cell27311 [label="715.0 minutes" ,style=dotted]
    cell24626 -> cell27311
    cell27310 [label="715.0 minutes" ,style=dotted]
    cell24626 -> cell27310
    cell27002 [label="710.0 minutes"]
    cell12037 -> cell27002 [label="285.0"]
    cell27309 [label="715.0 minutes" ,style=dotted]
    cell27002 -> cell27309
    cell27307 [label="715.0 minutes" ,style=dotted]
    cell27002 -> cell27307
}
```

This script is build up recursively by following the links in the Cell objects until a division is reached. The time since the last division is calculated. The two daughter cells are then followed until the tracking ends (the 'leaves' of the tree are reached, where the cells do not point to any 'next' cells).

Appendix C

Publications

- Mike Downey, Keith W. Vance, and Till Bretschneider. Lineagetracker: A statistical scoring method for tracking cell lineages in large cell populations with low temporal resolution. *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 1913 –1916 (2011). doi:10.1109/ISBI.2011.5872783. [146]

- Mike J. Downey, Danuta M. Jeziorska, et al. Extracting fluorescent reporter time courses of cell lineages from high-throughput microscopy at low temporal resolution. *PLoS ONE* 6(12): e27886. doi:10.1371/journal.pone.0027886. [147]

LINEAGETRACKER: A STATISTICAL SCORING METHOD FOR TRACKING CELL LINEAGES IN LARGE CELL POPULATIONS WITH LOW TEMPORAL RESOLUTION

Mike Downey, Keith W Vance, Till Bretschneider

Warwick Systems Biology Centre, Coventry House
University of Warwick, Coventry CV4 7AL, United Kingdom

ABSTRACT

Automated high-throughput analysis of single-cell timecourse data presents a major bottleneck in live cell imaging. We present LineageTracker, an ImageJ framework to track expression of fluorescent gene reporters over multiple cell divisions. It is able to perform automatic segmentation and tracking, and allows viewing and editing of tracks. The main feature of the tracking algorithm is a statistical scoring method which takes into account characteristic intensity and size changes to classify dividing and non-dividing cells. By including such dynamic features, the method can identify dividing cells in time series with 30 min frame intervals, and handle large cell displacements between frames. We created a manually validated data set of mouse C2C12 cells expressing a fluorescent protein targeted to the cell nucleus which we will make available for benchmarking different segmentation and tracking methods.

Index Terms— *High-throughput live cell imaging, cell tracking, cell lineage profiling*

1. INTRODUCTION

Computerized microscopy in combination with live cell fluorescence imaging has become a valuable technique for measuring the regulation of gene expression in single cells. Single cell studies allow measuring the characteristics and effects of noise in transcriptional control. For further computational analyses temporal profiles of different cells can be synchronized in silico using curve-fitting techniques [1]. One example we present here is Msx1 expression in C2C12 mouse cells (Figure 1). Msx1 is a transcription factor that plays a role controlling pluripotency in mesenchymal stem cells [2]. Here we are specifically interested in how Msx1 expression is regulated in daughter cells.

Segmentation of nuclei to determine intensities of nuclear localized fluorescent proteins is a critical step, but since nuclei have a consistent shape (convex and roughly circular) standard methods such as seeded growth work reasonably well. Touching objects can be separated using conventional watershed methods. The main difficulty is to track cells over time, which has been addressed by some authors [3, 4], and construct mother-daughter cell lineages, a much less well studied problem [5, 6].

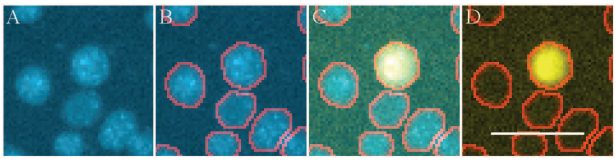


Figure 1: C2C12 cells labelled with Hoechst and expressing vGFP under the control of a Msx1 regulatory promoter sequence. A: Hoechst stained nuclei. B) Overlaid with segmentation mask. C) vGFP fluorescence channel overlaid. D) Segmentation mask showing area of measurement for fluorescence intensity. Scale bar: 50 micron.

Although some algorithms do handle divisions, they are usually optimized for short frame intervals making use of nearest distance information. The nuclear stain (Hoechst) is used here to track cells which do not express the fluorescent protein marker, and to identify cell divisions. Hoechst is excited by UV light, however, which is toxic to cells. Increased time intervals (10-30 min) to reduce long-term exposure (over 2.5 days) impose severe constraints on tracking and lineage construction as cells exhibit considerable motion between frames.

2. IMAGE ACQUISITION AND SEGMENTATION

Dual color image time series (currently Hoechst and GFP) were acquired using a Cellomics KineticScan KSR machine with a 10x 0.4NA objective at 512x512 resolution every 30 minutes. A custom import module was written to import Cellomics data into ImageJ.

A seeded growth algorithm [7] is used after correcting for non-uniform illumination using ImageJ's rolling ball background subtraction (radius = 25 pixels). Following Gaussian convolution of the Hoechst channel (sigma = 2.5 pixels) the centers of nuclei are located as local maxima in the image [8]. Outlines of nuclei are grown radially outwards from the seeds identified above, until either a dynamic threshold is reached, which is ½ of the peak intensity, or they touch a neighbor cell.

3. TRACKING

After segmentation we compute for each cell a number of features which include position, area, multi-channel

intensities, standard deviation, intensity moments, shape parameters (circularity, minor & major axes).

Our scoring based tracking algorithm follows mainly ideas on multi-feature based cell-type classification as used by Murphy et al. and Loo et al. for static images [9, 10]. Here we extend this approach to include dynamic features in order to identify dividing cells in time series. Tracking is based on combining the most informative features to compute a cell transition matrix which holds probabilities for cell-cell linkages between frames (**Figure 2**) [5].

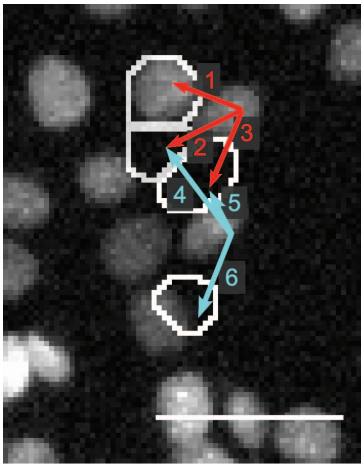


Figure 2: Computing statistical transition scores for linking cells between frames. Shown in white are outlines of possible target cells in the subsequent frame. Numbered red and blue arrows indicate the three most likely moves for each of two cells. See Table 1 below for associated movement scores. In accordance with the manually validated sequence, arrows 3 & 6 are the correct linkages. Scale bar: 50 micron.

Transition	1	2	3	4	5	6
Score	0.579	0.484	0.742	0.619	0.549	0.712

Table 1: Examples of transition scores. Higher numbers indicate a better match.

In brief, feature selection is based on principal component analysis. In the next step weighting of parameters is based on a brute-force optimization algorithm. In our example the first three components account for 76% of the data variance. The most significant features are Hoechst mean intensity, standard deviation of Hoechst intensity, 2nd order intensity moment and nucleus area.
The steps involved in tracking are illustrated in **Figure 3**A.

3.1 Construction of a Transition Matrix

We compute a movement score M for each cell in a given frame, and its potential target in the subsequent frame. M is based on differences in the measured features between frames. A threshold and weight are associated with each feature to determine the contribution to the overall score. For each individual feature, *f*, the movement score is:

$$M(f) = 1 - \left(1 + e^{\alpha\left(\frac{T(f)-D(f)}{D(f)}\right)}\right)^{-1}$$

where *T(f)* is a statistically determined threshold level for the feature in question, *D(f)* is the difference in feature values for the two cells, and α controls the rate of decrease in M. The sigmoid shape of the function penalizes only large enough changes in feature values.
The overall movement score for a cell is then:

$$M = \prod_f \left(1 - W(f) + W(f) \times M(f)\right)$$

where *W(f)* is the weight, $0 \le W \le 1$, associated with each feature.
All movement scores are assembled in form of a matrix consisting of transition scores for each cell and its potential target.

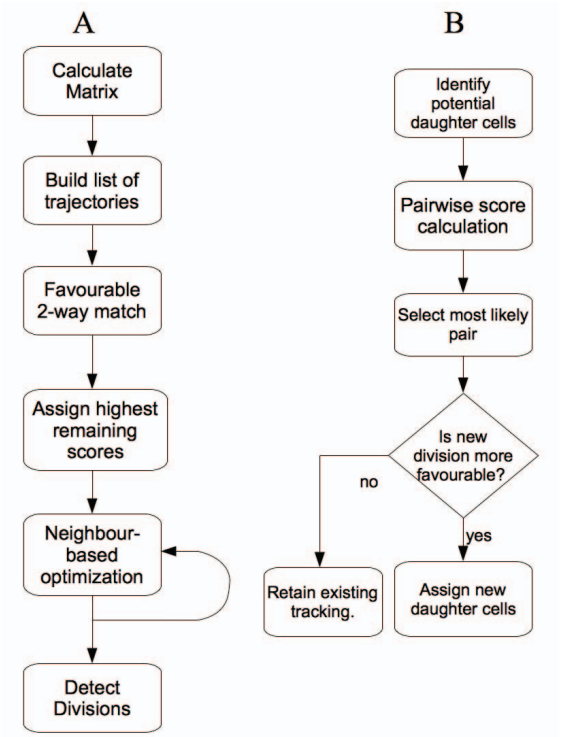


Figure 3: A) Flow diagram for tracking algorithm (details see text). **B)** Expanded view of the "Detect Divisions" module.

3.2 Construction of Trajectories

Trajectories are constructed as follows and stored as a doubly-linked tree to enable traversal forwards and backwards through cell lineages.

The first step builds up a list of most likely target cells for each cell according to the transition matrix together with the movement scores, in both the forward (t→t+1) and backward (t+1→t) directions.

Initial trajectories are assigned using a 'co-operative best match', which selects the transition where the highest scoring transition in the forward direction is also the highest scoring backward pointing transition.

The third step completes any remaining unassigned trajectories by assigning the highest scoring transitions. The final step performs an optimization by maximizing the summed movement scores of all linked cells. A new total score is calculated based on a pair-wise exchange of trajectories. The new trajectories are retained if the new score is greater than the existing score.

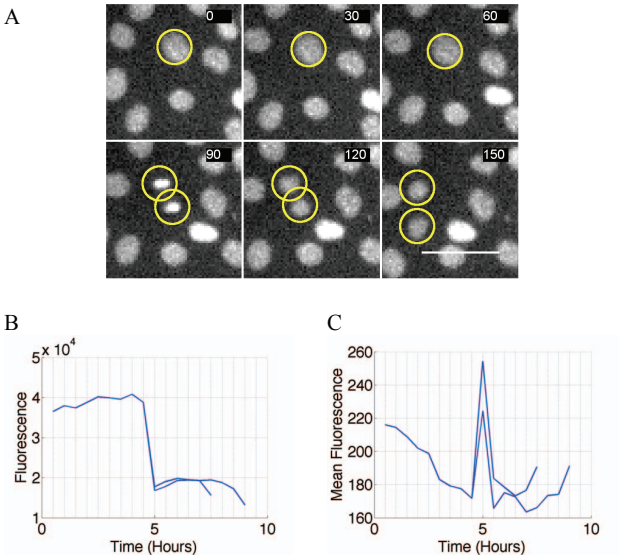


Figure 4: A) sequence of a dividing cell showing Hoechst enrichment prior to division. Time in minutes, scale bar: 50 micron. **B)** Total Hoechst intensity showing equal distribution over the two daughter cells. **C)** Mean Hoechst intensity over time for the cell given in (A). Chromatin condensation causes the mean intensity to peak during cell division.

3.3 Detection of Divisions

Potential cell divisions are detected by identifying cells newly appearing during tracking. During cell division the Hoechst is partitioned between the two daughter nuclei. This leads to a reduction in the integrated Hoechst intensity (**Figure 4**B).

Statistical testing shows a close relationship between integrated intensity of the mother cell and the sum of intensities of the two daughter cells (R^2=0.92), and between integrated intensities of the two daughter cells (R^2=0.95).

The Detect Divisions module is summarized in **Figure 3**B. Potential daughter cells are identified by a characteristic change in Hoechst intensity or area (**Figure 4**C), or when tracking is lost due to size or intensity of the daughter cells being too dissimilar to be recognized for continued tracking.

All cells within a specified radius of the last known position are examined and similarity scores are calculated between each pair of potential daughter cells. The combined intensities are compared to the potential mother cells and a new movement score is calculated. If the tracking has been lost or the new score is more favorable than the existing tracking score, the two daughter cells are linked to the mother cell.

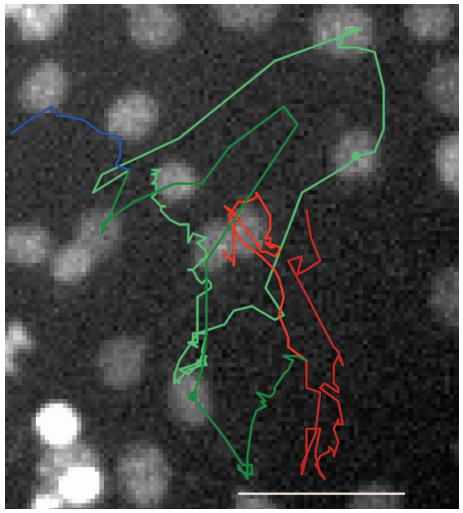


Figure 5: Result of tracking and lineage construction. Progenitor cell trajectory is displayed in blue. Following division, daughter cells are displayed in green. Filled circles on green lines indicate the actual cell positions of the displayed frame. One daughter (dark green) further divides and daughter cell trajectories are shown in red. The displayed trajectory is a manually corrected lineage tree. The longest automatically tracked section was 82 frames, including one division. Total duration of tracking was 110 frames. Scale bar: 50 micron.

4. RESULTS

The segmentation accuracy has been measured using a manually annotated gold standard where centers of nuclei were recorded and a synthetic data gold standard [11] for which nuclei positions and ground-truth pixel data are available. The manual gold standard consists of five frames with a total of 2932 cells at increasing densities. A cell is considered to be detected correctly if it is found within less than 1 cell radius of the manually marked position. Any cells detected at a greater distance from a known cell are recorded as false positives.

Our segmentation method yields a 95% detection rate with a TP:FP (true positive : false positive) ratio of 18.3:1 . This compares to a 93% detection rate and a ratio of 9:1 found if the same set of images are analyzed using CellProfiler, an

established open standard for high-throughput cell profiling [3]. CellProfiler's lower ratio is due to a higher number of false positive detections with the low contrast/high cell density images.

The synthetic images contain a total of 1837 non-overlapping nuclei. These were used to measure the pixel-accuracy of the segmentation to determine whether the method successfully reproduced the outlines of detected nuclei. The F-score using a weighting of $\beta = 1$ was calculated as follows (FN=false negative):

$$\text{Precision (P)} = \frac{TP}{TP + FP} = 0.86$$

$$\text{Recall (R)} = \frac{TP}{TP + FN} = 0.76$$

$$\text{F - score} = \frac{(1+\beta^2)PR}{\beta^2(P+R)} = 0.82$$

CellProfiler's F-score was computed as 0.88 for comparison.

The tracking accuracy was measured by manually constructing cell lineages for a time-series consisting of 110 frames taken at 30-minute intervals. The standard contained 7317 cell-cell transitions and 100 divisions (**Figure 5**). The individual transitions were detected with 97% accuracy. The longest successfully tracked section of each lineage was measured. In the manually tracked data set trajectories contain between 5 and 111 transitions. On average each lineage was tracked along 88% of its total length before tracking was lost or interrupted. By comparison CellProfiler detected 96% of all transitions, tracking an average of 85% of each trajectory. The large frame intervals made detecting divisions challenging. 65 of the 100 marked divisions were correctly identified.

5. CONCLUSIONS

Lineage tracking in an emerging field and there is no commercial software available. Cell detection rates and segmentation accuracies of our software are comparable to CellProfiler software. CellProfiler's F-score (accuracy of segmentation) was slightly higher for the artificial data whereas our seeded growth performed better on actual cell images for which the detection rate has been optimized. LineageTracker offers an ImageJ based framework that is easily extendible and has the capability to track cell lineages while being specifically designed to handle large cell displacements between frames.

Because of cell trajectories crossing over, an average tracked length of 88% can be considered as acceptable given that we aim at tracking hundreds or thousands of cells. The same applies to the seemingly low success rate of 65% for detecting divisions.

Although fully automated reconstructions of cell lineages in experiments with low temporal resolution are currently not within reach, our framework enables experimentalists to track lineages much more easily and efficiently compared to manual tracking.

The software is still under development and will be released from http://go.warwick.ac.uk/lineagetracker as open source.

Its implementation in ImageJ will allow different segmentation and lineage construction algorithms to be substituted and evaluated. The viewer/editor allows the segmentation or tracking to be manually edited or corrected where required. All data are stored in text files which enables further analysis by software such as R or Matlab™.

6. ACKNOWLEDGEMENTS

We would like to thank S. Ott (University of Warwick) for critical comments and discussions in the early stages of this project, and Cheng-jin Du (University of Warwick) for critical comments on the manuscript.

REFERENCES

1. Sigal, A., et al., *Dynamic proteomics in individual human cells uncovers widespread cell-cycle dependence of nuclear proteins.* Nature Methods, 2006. **3**(7): p. 525-531.
2. Bendall, A.J. and C. Abate-Shen, *Roles for Msx and Dlx homeoproteins in vertebrate development.* Gene, 2000. **247**(1-2): p. 17 - 31.
3. Carpenter, A.E., et al., *CellProfiler: image analysis software for identifying and quantifying cell phenotypes.* Genome Biology, 2006. **7**(10): p. R100.
4. Du, C.-J., et al., *Interactive Segmentation of Clustered Cells via Geodesic Commute Distance and Constrained Density Weighted Nyström Method.* Cytometry Part A, 2010(In Press).
5. Al-Kofahi, O., et al., *Automated Cell Lineage Construction: A Rapid Method to Analyze Clonal Development Established with Murine Neural Progenitor Cells.* Cell Cycle, 2006. **5**(3): p. 327-335.
6. Li, K., et al., *Cell population tracking and lineage construction with spatiotemporal context.* Medical Image Analysis, 2008. **12**(5): p. 546-566.
7. Mehnert, A. and P. Jackway, *An improved seeded region growing algorithm.* Pattern Recognition Letters, 1997. **18**(10): p. 1065-1071.
8. Fenistein, D., et al., *A fast, fully automated cell segmentation algorithm for high-throughput and high-content screening.* Cytometry Part A, 2008. **73A**(10): p. 958-964.
9. Loo, L.-H., et al., *An approach for extensibly profiling the molecular states of cellular subpopulations.* Nat Meth, 2009. **6**(10): p. 759-765.
10. Murphy, R.F., M. Velliste, and G. Porreca, *Robust Numerical Features for Description and Classification of Subcellular Location Patterns in Fluorescence Microscope Images.* The Journal of VLSI Signal Processing, 2003. **35**(3): p. 311-321.
11. Lehmussola, A., et al., *Synthetic Images of High-Throughput Microscopy for Validation of Image Analysis Methods.* Proceedings of the IEEE, 2008. **96**(8): p. 1348-1360.

OPEN ACCESS Freely available online

PLoS one

Extracting Fluorescent Reporter Time Courses of Cell Lineages from High-Throughput Microscopy at Low Temporal Resolution

Mike J. Downey[1], Danuta M. Jeziorska[2], Sascha Ott[3], T. Katherine Tamai[4], Georgy Koentges[2], Keith W. Vance[3¤], Till Bretschneider[3*]

1 Molecular Organisation and Assembly in Cells, University of Warwick, Coventry, United Kingdom, **2** School of Life Sciences, University of Warwick, Coventry, United Kingdom, **3** Warwick Systems Biology Centre, University of Warwick, Coventry, United Kingdom, **4** Department of Cell and Developmental Biology, University College London, London, United Kingdom

Abstract

The extraction of fluorescence time course data is a major bottleneck in high-throughput live-cell microscopy. Here we present an extendible framework based on the open-source image analysis software ImageJ, which aims in particular at analyzing the expression of fluorescent reporters through cell divisions. The ability to track individual cell lineages is essential for the analysis of gene regulatory factors involved in the control of cell fate and identity decisions. In our approach, cell nuclei are identified using Hoechst, and a characteristic drop in Hoechst fluorescence helps to detect dividing cells. We first compare the efficiency and accuracy of different segmentation methods and then present a statistical scoring algorithm for cell tracking, which draws on the combination of various features, such as nuclear intensity, area or shape, and importantly, dynamic changes thereof. Principal component analysis is used to determine the most significant features, and a global parameter search is performed to determine the weighting of individual features. Our algorithm has been optimized to cope with large cell movements, and we were able to semi-automatically extract cell trajectories across three cell generations. Based on the MTrackJ plugin for ImageJ, we have developed tools to efficiently validate tracks and manually correct them by connecting broken trajectories and reassigning falsely connected cell positions. A gold standard consisting of two time-series with 15,000 validated positions will be released as a valuable resource for benchmarking. We demonstrate how our method can be applied to analyze fluorescence distributions generated from mouse stem cells transfected with reporter constructs containing transcriptional control elements of the Msx1 gene, a regulator of pluripotency, in mother and daughter cells. Furthermore, we show by tracking zebrafish PAC2 cells expressing FUCCI cell cycle markers, our framework can be easily adapted to different cell types and fluorescent markers.

Citation: Downey MJ, Jeziorska DM, Ott S, Tamai TK, Koentges G, et al. (2011) Extracting Fluorescent Reporter Time Courses of Cell Lineages from High-Throughput Microscopy at Low Temporal Resolution. PLoS ONE 6(12): e27886. doi:10.1371/journal.pone.0027886

Editor: Daniela Cimini, Virginia Tech, United States of America

Received June 21, 2011; **Accepted** October 27, 2011; **Published** December 15, 2011

Funding: MJD was funded by the EPSRC (www.epsrc.ac.uk) through the MOAC Doctoral Training Centre, University of Warwick. KWV was funded by a Senior Research Fellowship from the University of Warwick. The experimental work was funded by WellcomeTrust (WT 066790/E/02/Z and 066745/Z/01/Z, www.wellcome.ac.uk) and HFSP (RGP0029/2007 C, www.hfsp.org) programme grants awarded to GK. Fucci data provided by TKT was obtained through funding by BBSRC/ERASysBio+ (BB/I004823/1). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: T.Bretschneider@warwick.ac.uk

¤ Current address: Department of Physiology, Anatomy and Genetics, University of Oxford, Oxford, United Kingdom

Introduction

Live cell fluorescent reporter-based techniques reveal the dynamics of gene expression under the control of different regulatory promoters, in individual cells and over periods of several days. Destabilized reporters with short half-lives of ~30 minutes not only show when genes are turned on, but also how long expression lasts and possible periodic or random repetitions, either self-stimulated or induced. Single cell studies uncover the characteristics and effects of noise in transcriptional control by making it possible to synchronize temporal expression profiles *in silico* [1–3], contrary to population assays where individual responses are averaged out [4,5]. Much progress has been made in high-throughput microscopy of tissue culture systems to study cells through several rounds of division [6,7], with great potential to investigate differential gene expression in self-renewing and differentiating stem cells.

Commercial platforms are available that offer integrated setups containing a fluorescence microscope connected to a high resolution CCD camera with autofocus, a humidified incubator, liquid handling robots and computer systems allowing the automated imaging of thousands of cells [8–11]. A major limitation of current single cell approaches is, however, the identification and tracking of cells in time-series, both through cell divisions and in confluent cultures.

Identifying cells using nuclear markers

The requirement to generate multiple clonal cell lines containing targeted insertion of reporter plasmids limits the use of stable transfections in large scale synthetic biology promoter studies. Transient transfection of fluorescent reporters represents a rapid alternative and is therefore the method of choice for analysing multiple promoters and regulatory elements. Transient

transfections are also advantageous as onset rates of transcription can be measured by introducing a naked DNA template into live cells on which transcriptional complexes can assemble [12]. The latter is particularly important in cells that continuously express genes under the control of endogenous promoters. To capture the onset of expression, we must ensure all cells are labelled using an independent marker, so that cells can be tracked before expression of any fluorescent marker sets in. Identifying cells with nuclear markers, such as Hoechst, abolishes the need for co-transfection (of a second constitutively active fluorescent colour for tracking purposes), thus facilitating experiments with primary cells and comparative expression analyses of different promoter constructs. Another important aspect for our analyses is that during cell divisions the chromatin marker segregates into the two daughter cells, which aids in identifying cell divisions and assigning mother and daughter cells. Since Hoechst is excited with UV light, photodamage has to be kept to a minimum. To image over long periods of time (days) with minimal cell death, we tested UV exposure times empirically and determined 30 minute intervals to be optimal for transfected C2C12 mouse mesenchymal stem cells. During that time interval, cells exhibit significant motion, thereby greatly challenging the reliability of any tracking method.

Segmentation of nuclei is discussed in Text S1 (see also Figure S1).

Cell tracking

Recently, software has become available for high resolution cell tracking and spatiotemporal analysis of protein dynamics in subcellular compartments (QuimP [13], CellTracker [14]). However, as these methods are designed to track cell boundaries in great detail, they require cells to only move by small amounts. Conventional tracking methods still require at least a minimum overlap to link cell positions between consecutive frames, measured either in absolute pixel counts, or relative to object size. This is the approach used by CellID [15], CellTracer [16], and Overlap-Based Cell Tracker [17]. If cells exhibit persistent motion and cell collisions are infrequent, 'keyhole' tracking algorithms can be applied, which calculate the probability of finding matching cells in a particular direction [18].

A number of single particle tracking methods have also been developed recently, which are able to track multiple non-overlapping objects and can, in principle, be applied to tracking cells [19]. Altinok et al. [20] have used spatiotemporal graph matching for tracking microtubule tips. Similarly, particle filter methods have been developed for tracking objects [6,21,22]. Future positions of objects are predicted using a motion model, and then matched with objects at the real positions. This usually involves solving a global linear assignment problem [23]. Both graph-based and hidden Markov model approaches can easily be extended to include additional object features, such as shape, size, colour, or texture. However, for large-scale problems, including time-series with thousands of cell positions, global optimization approaches are computationally very costly. Furthermore, particle filters only work for small displacements where motion between frames is highly correlated. In time-series with low temporal resolution and considerable cell motion, these approaches generally perform poorly.

Instead of solving a global optimization problem, we formulate here a statistical scoring approach in a less rigorous and formal way, which was briefly introduced in [24]. It is based on a similarity matrix, where scores are calculated for possible target cells within a maximum distance that can be covered by a cell in a given time interval. Relevant similarity features are selected from a larger list of possible features based on principal component analysis (PCA), similar to methods used in multi-feature cell-profiling [25,26]. Computational demand for this local optimization problem simply scales linearly with the number of cells to be tracked.

Constructing cell lineages

There have been some approaches to lineage construction based on the appearance or behaviour of cells during mitosis [7]. Debeir [27] computes tracking in reverse from the final frame. Divisions are detected by the merging of two daughter cells. As the cells approach mitosis, their size decreases and the two daughter cells come closer. When size and distance are below a threshold, the 'reverse mitosis' event has completed. Wang [28] calculates texture based features and uses feature reduction methods, including PCA to reduce 145 features to 15–20. Divisions are detected by treating each stage of the mitosis event as a hidden state in a Markov chain. A training set was used to calculate the probabilities for the chains. Similarly, Markov trees were used in [29] to map cell states to lineages.

Al-Kofahi et al. [30] construct lineages by calculating a significance score based on the observation that daughter cells have a similar size. The Ellenberg group has developed a powerful framework for automatic detection of cell divisions and chromosome phenotypes [31,32]. Their approach, which is based on 3D time-series with stacks captured at 5–7 minute intervals, makes use of region adaptive thresholding and a feature point tracking method. Probabilities for detecting mitosis events are based on size and distance of chromosome sets for which weights are determined empirically. Li et al. [6] and a more advanced version by Bise et al. [33] use phase contrast images for cell segmentation and detection of mitosis events, which appear brighter in phase contrast. Cell trajectories are assembled into shorter fragments first, so called tracklets, which are stitched together by using a global optimisation problem a posteriori. Accuracies achieved are 87% for tracking (correctly identified cell-cell linkages between frames) and 68% for detecting divisions correctly.

Padfield et al. [34] also make use of a Hoechst label to segment nuclei, although imaging at a higher frame-rates of 6 or 15 minutes. They use a wavelet based method for cell segmentation. Subsequently, a graph flow method is used for tracking cells, and they report 99.2% of cells tracked with complete accuracy (with an average track length of 13 frames) and 97.8% correctly identified divisions, validated using 104,000 cell positions. Although the methods by Bise and Padfield are both considered state of the art, they result in markedly different detection rates and accuracies. It is difficult to pinpoint a single cause for this, but most likely it is due to experimental differences in cell density, movement and clustering. For example, the net translocation of cells observed by Padfield is small (after correction for stage drift) and thus, makes validation of large numbers of cells comparatively easy.

Comparison of different methods is almost impossible, since many of them are only available as part of an integrated commercial platform or publicly not available. Often, precision of different segmentation routines is not validated based on objective ground-truth using synthetic data, but by human observers [34], and it is difficult to obtain a comprehensive list of all parameters being used. Since there is currently no standard for exchanging track-data for evaluating different methods, we set out here to develop a new software framework using ImageJ which allows comparisons of different segmentation and tracking routines. Furthermore, we will make available validated tracked data sets at different temporal resolutions (10 and 30 min), which can be used as a benchmark test for others. The method we present here incorporates the tracking of cell lineages in our statistical scoring framework for cell tracking. It makes use of dynamic feature changes, such as characteristic

changes in Hoechst distribution and nuclear size. The experimental data we make available are challenging as they are subject to considerable noise, and there is a huge variation in nuclear size and shape when compared to the examples given in Padfield [34]. Also, large cell displacements between frames make tracking by eye and validation of large numbers of cells more difficult. The clustering of cell nuclei found in our experiments poses a particular challenge when reconstructing cell lineages, as it obscures mother-daughter cell relationships.

Current software toolkits

A software framework specifically tailored for high-throughput single cell studies is the open source image analysis platform CellProfiler [35]. CellProfiler is highly flexible and supplies all of the above mentioned segmentation methods, as well as several tracking methods including a multi-object tracker based on the method by Jaqaman [21], which accounts for splitting and merging of objects. Other tracking methods within CellProfiler utilize features such as object overlap, distance or any other measurements (intensity, morphology). A version of CellProfiler has been used for single-cell tracking by Alon et al. [3].

Here we use an alternative platform, ImageJ, which is widely used and easily extendible by Java plugins. Existing cell tracking methods for ImageJ are currently very limited, however. The Particle Tracker plugin is an implementation of Feature Point Tracking [36] and provides both segmentation and tracking based on the intensity moment of the particle images. Mtrack2 performs tracking and requires the segmentation to be performed beforehand. Trajectories are assigned by selecting the nearest particle in the following frame.

Msx1 expression profiling

The software we developed was initially designed to measure the activity of fluorescent reporters driven by transcriptional control elements from the Msx1 gene in C2C12 mouse mesenchymal stem cells. The Msx1 protein is involved in regulating pluripotency of mesenchymal stem cells [37]. It is a member of the homeobox family of transcription factors involved in vertebrate craniofacial and muscle development. Expression of Msx1 during embryogenesis maintains progenitor cells in their undifferentiated state and mutations in the Msx1 gene lead to cranial and dental defects [38], including cleft palate. Several control elements of Msx1 have been identified by others and ourselves (Vance et al., submitted), and a key objective for the development of our analysis method was to quantify the role these elements play upon transcription rates by using fluorescent reporters. Expression levels are proportional to the amount of reporter protein provided the measured intensity is within the linear range of the imaging system. Fluorescent reporters were modified by the addition of a nuclear localization sequence (nls), which led to post-translational targeting to the nucleus. Segmentation based on Hoechst can therefore be used to measure reporter intensities in the nucleus. Ideally, we want to determine reporter levels during the lifetime of individual cells in order to avoid transgenerational inaccuracies or differences in reporter activity due to asymmetric fate choices. For this reason, methods are needed to determine reporter fluorescence between two automatically recognized cell division events in entire clonal populations.

Materials and Methods

Imaging of mouse C2C12 cells

C2C12 mouse myoblast cells (ECACC, Catalogue No. 91031101) were grown in DMEM supplemented with 10% foetal bovine serum at 37°C in an atmosphere of 5% CO_2. For transient transfections, the cells were transferred to a 96-well plate at a density of 1.25×10^4 cells per well. Hoechst 33342 (Invitrogen) 400 ng/ml in DMEM was added and incubated at 37°C for 30 minutes. Cells were then washed twice with PBS, and DMEM (without phenol red) was added. Cells in each well were subsequently transiently transfected with 200 ng of reporter plasmid using Lipofectamine 2000 (Invitrogen) according to the manufacturer's instructions.

Images were obtained using a Cellomics KineticScan KSR machine with a 10× NA 0.4 objective at a resolution of 512×512 pixels. Two colour channels (Hoechst and vGFP) were obtained every 30 minutes using the XF100 filter set. A custom import module was written to import Cellomics data (version 1.35) into ImageJ using Jackcess (version 1.1.21, http://jackcess.sourceforge.net), a library for reading and writing Microsoft Access databases.

Imaging of zebrafish PAC2 cells

Zebrafish PAC2 cells derived from 24-hour embryos were transfected with FUCCI constructs mKO2-zCdt1(1/190)/pT2KXIGΔin and mAG-zGeminin(1/100)/pT2KXIGΔin [39,40] and plasmid pcDNA3.1/myc-His A (Invitrogen), as previously described [41]. After neomycin selection, single cells were sorted sequentially for orange fluorescence (mKO2) and then green fluorescence (mAG) by fluorescence-activated cell sorting. A clonal FUCCI cell line was established and cultured as previously described [41]. For time-lapse analysis, FUCCI cells were plated at a density of 100,000–150,000 cells/ml onto a 35 mm glass-bottomed dish (Wilco), maintained at 28°C and imaged with a 10× NA 0.3 objective lens on an inverted Leica SPE confocal microscope. Images were captured every 15 minutes for a total of 65 hours using sequential fast scanning.

Software design and implementation

The software was written in Java as a set of ImageJ plugins and uses the image manipulation routines available within ImageJ. The Image Viewer requires the Image5D plugin to be installed, which is available separately or bundled with the 'Fiji' version of ImageJ (available from http://rsb.info.nih.gov/ij/and http://pacific.mpi-cbg.de/wiki/index.php/Fiji). There are separate plugins for segmentation/tracking and viewing/editing the data.

The segmentation software can handle any image format which can be imported into ImageJ. The user selects the location to store the data and loads the image sequence into ImageJ. The segmentation parameters can be adjusted with a preview available.

The viewer allows the user to visually interact with the segmentation and tracking, and perform minor edits to the data. The application is compatible with tracking information from CellProfiler and the ImageJ plugins MTrackJ and ParticleTracker. Fluorescence time course data and cell division data can be exported as spreadsheet files. Tracking videos can be exported with highlighted cells overlaid.

Results

Figure 1 and Figure S2 summarise the problem of tracking individual cells moving in crowded environments, and show segregation of the nuclear marker during cell divisions. Figure 1A,B show the Hoechst and GFP channels for an image with a cell density of 1300 cells/mm^2 typically reached at t = 40 hours after transfection. The close up in Figure 1C illustrates the basic idea behind statistical scoring mechanisms for identifying matching cells in subsequent frames. For each of two example cells, three arrows point to possible target cells (white outlines) in the subsequent frame. Differently coloured arrows (e.g. red 3 and

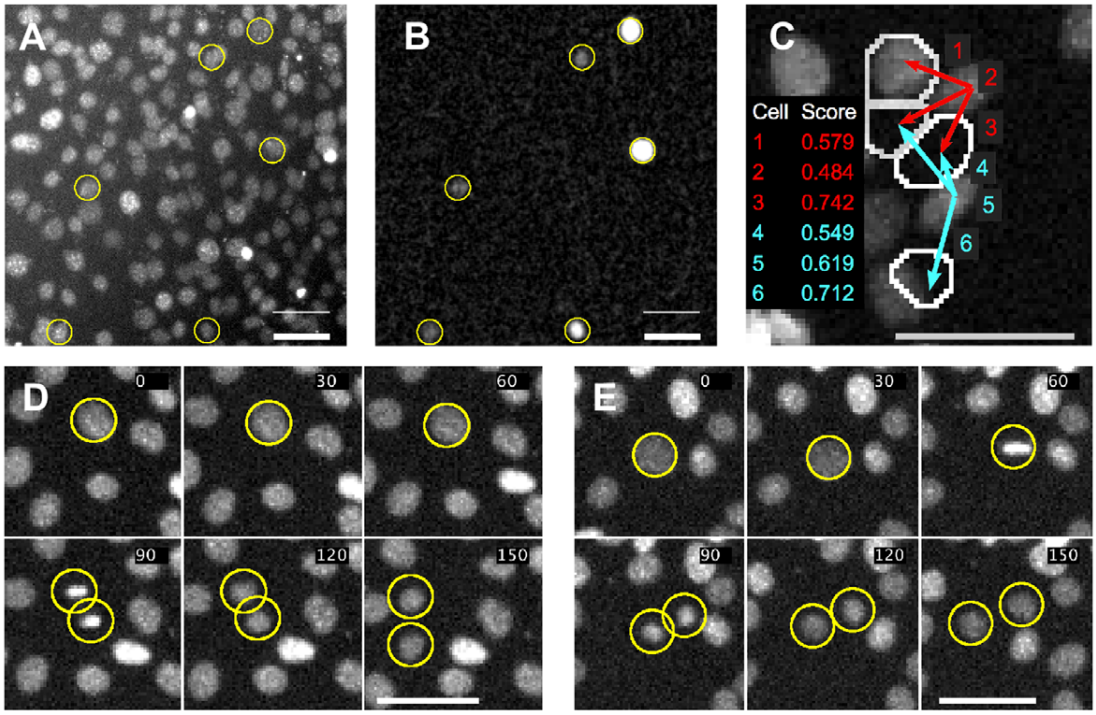


Figure 1. Magnified section of an image obtained from the Cellomics automated microscope. A) C2C12 cells labelled with Hoechst stain. B) Same view showing expression of GFP driven by a Msx1 promoter. GFP expressing cells have been highlighted in yellow in A and B. C) Potential ambiguity in linking cells in subsequent frames (white outlines). Arrows represent potential trajectory assignments with numbers representing the calculated score for each potential assignment. D and E) Cell divisions exhibiting chromatin condensation close to the point of division. Time is displayed in minutes. Scale bar in all images is 50 microns. (C and D have been adapted from [24], © 2011 IEEE).
doi:10.1371/journal.pone.0027886.g001

blue 4) pointing to the same target cell in the centre of the image make it obvious that positional information alone is not sufficient to discriminate which of the possible target cells is the correct one. Although connection 4 is the shortest, it turns out that connection 3 achieves the highest red score and is preferred over 4, while the highest blue score is 6. Figure 1D,E show characteristic condensation of the Hoechst marker during cell division (90 and 60 min frames), followed by segregation into daughter cells. This is an essential feature, which is used to identify cell divisions, as will be shown later on.

In the following section, we compare the efficiency and accuracy of a commercial solution, Cellomics, with different segmentation methods (for details of segmentation see Text S2). We then describe the development of the statistical scoring method for cell lineage tracking, which will be validated using a manually tracked gold standard.

Segmentation accuracy

Two different methods were used to evaluate segmentation results, each using a different gold standard set of artificial and real cells.

Firstly, we measured the pixel-accuracy of segmentation using artificial ground truth images created by Simcep software [42]. Five frames with 2885 cell nuclei in total (at densities between 425 and 703 cells per frame to match experimentally observed cell densities) were created along with binary images, which partition the image into foreground or background. There is no additional information regarding which cell a pixel belongs to (Figure S3A, B). The F-score indicates the overall accuracy of the segmentation according to this foreground/background partitioning, but does not penalize methods which fail to separate clustered or touching cells. The precision and recall values indicate whether a segmentation method consistently over- or under-estimates the size of the detected objects. The method counts the True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) pixels.

$$\text{Precision (P)} = \frac{TP}{TP+FP}$$

$$\text{Recall (R)} = \frac{TP}{TP+FN}$$

$$\text{F-score} = \frac{(1+\beta^2)PR}{\beta^2(P+R)}$$

A weighting factor of $\beta = 1$ was chosen to give an equal weight to precision and recall, as a combined F-score usually was found to be a good indicator of overall segmentation accuracy. The F-score performance of the different segmentation methods that have been tested is illustrated in Figure S3C. Surprisingly, the Global Threshold (Li automatic threshold from ImageJ) resulted in the highest F-values ($\sim$0.95) for all cell densities, while the more sophisticated regional adaptive Seeded Growth and Scaling Index methods performed poorly on the artificial data ($0.85 < \text{F-score} < 0.91$).

Using the kappa index to evaluate segmentation accuracy for the Simcep data, we obtain values of KI = 0.90 (for the Seeded Growth algorithm) compared to values between 0.81 and 0.96 reported in [34]. The kappa index measures the degree of overlap

between two sets:

$$\mathrm{KI} = 2\left(\frac{|A \cap B|}{|A| + |B|}\right)$$

A and B are ground truth and segmented pixel data, respectively.

To demonstrate that segmentation results at higher spatial resolution are comparable to the 10× NA 0.4 images used in the rest of the paper, Figure S4 shows an image of segmented cells using a 20× NA 0.75 objective.

The second method measured positional accuracy and used images of Hoechst stained nuclei. A set of 4 frames was selected from a 48-hour period of a single experiment (frame interval 30 minutes, 110 frames in total). The images exhibited a range of cell densities from 437–730 cells per image (902–1507 cells/mm^2); 1500 cells/mm^2 yield 25–30% total area covered by nuclei measured using the Hoechst channel, which approximately corresponds to 90–100% cell confluency.

The nuclei were manually located using the CellCounter plugin in ImageJ. The locations as determined by regional adaptive and non-adaptive segmentation methods were then compared with these ground-truth locations. For the Seeded Growth and Scaling Index segmentation methods, we developed custom-written ImageJ plug-ins. Threshold segmentation used existing methods available in ImageJ or Fiji.

To determine positional accuracy, we define a cell as true positive when being within 1 radius of a ground-truth cell. Cells which cannot be matched are classified as false positive. Cells in the ground truth data set which remain unassigned are classified as false negative. Figures 2A–H show common problems with over- and undersegmentation encountered with different methods. Generally, it turns out that there is not a single method which outperforms all others for all cell densities (Figure 2I, and

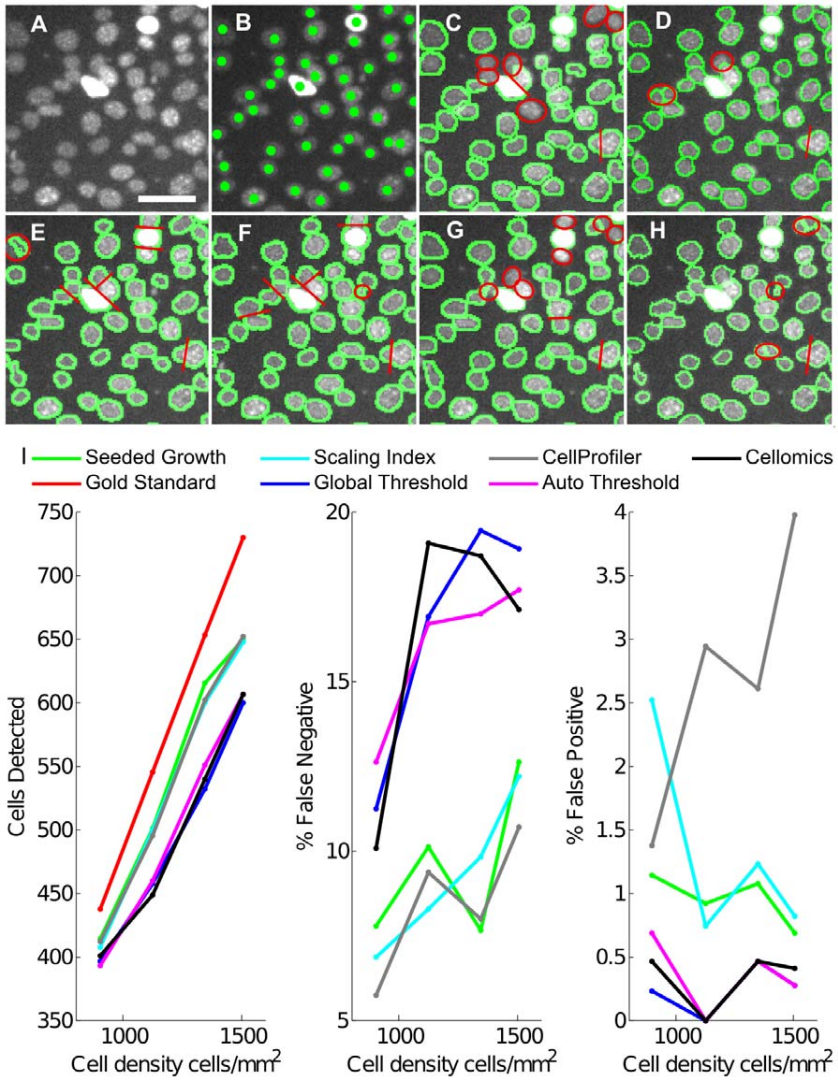


Figure 2. Segmentation of cell nuclei. A) Original nuclei (scale bar 50 microns) taken from the gold standard data set, cell density 1150 cells/mm^2. B–H) Nuclei with segmentation examples overlaid. Ellipses indicate segmentation errors. Lines indicate unresolved clusters of cells. B) Manually marked cell position. C) Cellomics segmentation. D) Seeded Growth. E) Global Threshold. F) Local Threshold. G) Scaling Index. H) CellProfiler. I) Cell detection accuracy measurements: Total cell count, false negatives and false positives comparing different segmentation methods to the gold standard.

doi:10.1371/journal.pone.0027886.g002

additional methods in Figure S3D), and above 1400 cells/mm^2, detection rates decline. The Seeded Growth and Scaling Index algorithms and CellProfiler perform slightly better regarding false negatives, which are consistently below 13%. However, the simpler threshold based methods (Cellomics, Global and Auto Threshold) yield numbers of false positives (below 1%), which are well below the Scaling Index and the CellProfiler Background Adaptive method.

The large number of missed cells at high cell densities means there is currently no reliable method that can work in an unsupervised manner when cultures become confluent (in Text S3 we describe a graphical user interface for validating cell positions and eliminating falsely classified cells). We here decided to use the Seeded Growth method as it provides a good balance between false positives and negatives for different cell densities.

Identifying features for cell tracking

During segmentation, several numerical features of nuclei are measured, similar to feature-based cell-type classification methods developed by Murphy et al. and Loo et al. [25,26,43], or recent methods for predicting cell fates of retinal progenitor cells using measurements of cell motion and phenotype [44].

All of the features are measured on the Hoechst nuclear channel. Additionally, the integrated intensity values are measured on the GFP channel. Our tracking algorithm combines the most informative features to compute probabilities for cell-cell transitions, which are stored in a matrix.

For the 7221 tracked positions, the measured features from Table 1 were examined using Principal Component Analysis. The first 5 principal components accounted for 74% of the variance in the Hoechst channel with the major contributions coming from mean intensity, 2nd intensity moment (divided by area), nuclear area and standard deviation.

The tracking algorithm relies on features remaining similar from frame to frame. Therefore, correlation scatter plots were produced, which compared the values of the features across successive frames (see Figure 3 and Figure S5). Daughter cells following division are plotted in red. For calculating correlation scores, dividing and non-dividing cells were treated separately. Dynamic features were plotted where the difference in feature value was calculated. Good features to use in tracking are ones where the values cover a wide range, while the correlation between cells in adjacent frames is good (see Table 1 for R^2 values). According to the outcomes of principal component and correlation analysis, the following 5 features were selected for tracking: distance moved, nuclear area, mean intensity, standard deviation of intensity, 2nd intensity moment (normalized to area). The feature selection was confirmed by comparing tracking accuracies for different sets of features.

Constructing the transition matrix

Tracking is calculated on a per-frame basis with individual trajectories linking a cell in one frame with a matching cell in the next frame. For each frame, a matrix is created where the rows represent cells in the current frame and columns represent cells in

Table 1. Measured and derived features used in tracking.

Feature	*Cumulative components*			*Correlation (R^2)*
	1	*2*	*3*	
Mean Hoechst intensity‡	**46.95**	**97.97**	**97.97**	**0.94**
Integrated Hoechst Intensity	84.29	97.25	97.26	0.97
Median Hoechst Intensity	45.63	78.05	78.05	0.86
Standard Deviation Hoechst intensity	**40.64**	**91.41**	**91.42**	**0.92**
Relative standard deviation‡	5.70	58.04	58.08	0.50
2nd Intensity Moment	94.76	95.06	95.09	0.85
2nd Moment (Intensity Normalized)‡	40.86	95.82	95.91	0.78
2nd Moment (Area*Intensity Normalized)‡	47.55	91.95	92.05	0.80
2nd Moment (Area Normalized)‡	**95.33**	**97.44**	**97.46**	**0.90**
Nucleus Area	**57.43**	**92.49**	**92.60**	**0.84**
Integrated GFP Intensity	**16.89**	**30.11**	**30.26**	**0.91**
Major Axis Angle	0.04	0.09	0.09	0.20
Axis Ratio	0.24	1.04	1.08	0.37
Circularity	46.95	97.97	97.97	0.16
Centre co-ordinates of nucleus	**N/A**	**N/A**	**N/A**	**1.00**
Δ Hoechst	6.54	9.56	9.57	0.00
Δ Area	0.22	0.23	68.22	0.01
Δ 2nd Intensity Moment	0.11	0.17	80.40	0.04
Δ Hoechst Standard Deviation	0.16	0.30	83.68	0.00
Δ Integrated GFP Intensity	0.00	0.23	0.43	0.07
Δ Circularity	0.05	0.15	43.87	0.18

Principal Component Analysis was used to determine which features contributed most to the tracking accuracy. The cumulative components columns specify how much variance of each feature is described by the first 3 principal components. Features in bold are used in the tracking system.
‡Derived from other features. R^2 values are given for non-dividing cells only.
doi:10.1371/journal.pone.0027886.t001

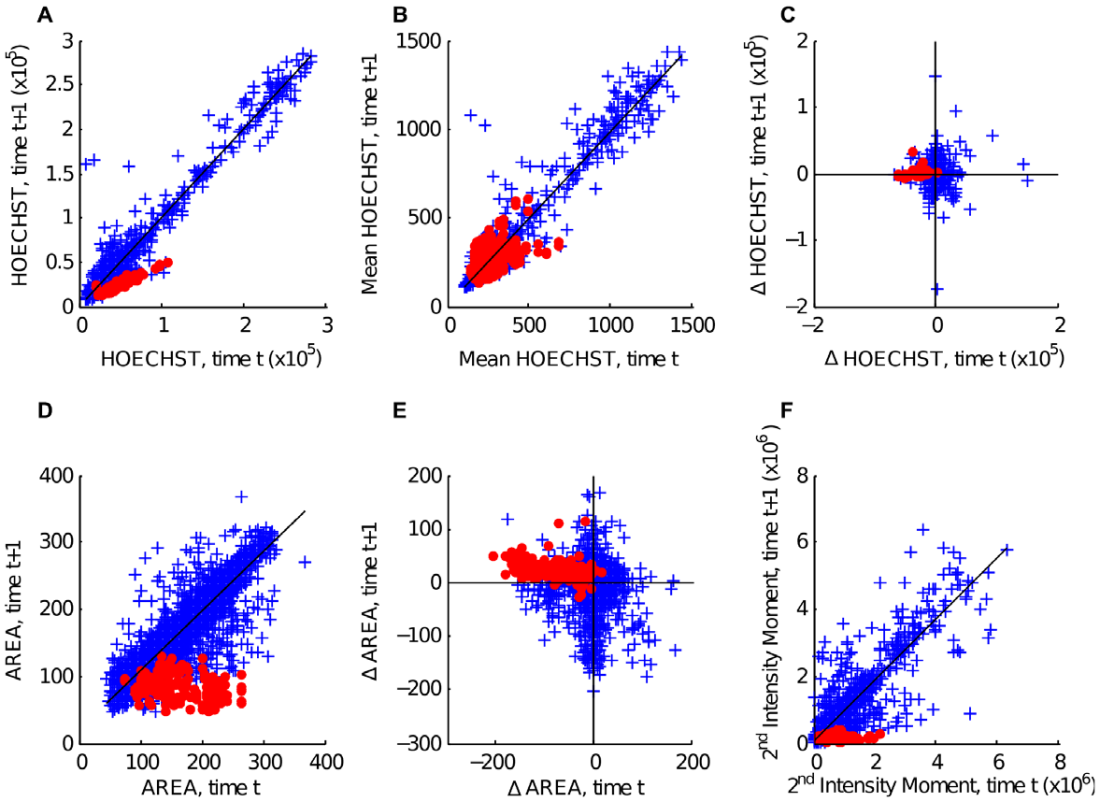


Figure 3. Correlations of different features between consecutive frames. Tracked cells are plotted in blue. Cells that divided between consecutive frames are plotted as red circles. R^2 values are given only for very highly correlated values. A) Integrated Hoechst intensity. Non-dividing cells show a very high correlation in Hoechst between frames (blue $R^2 = 0.97$). Red cells show that Hoechst levels are halved during division (red $R^2 = 0.90$). B) Mean Hoechst intensity (blue $R^2 = 0.94$). C) Change in Integrated Hoechst. D) Nucleus area. (blue $R^2 = 0.84$). E) Change in nucleus area. F) 2nd Intensity moment (measured on Hoechst channel, blue $R^2 = 0.85$).
doi:10.1371/journal.pone.0027886.g003

the subsequent frame. Each element in the matrix holds a *movement score* representing the similarity in position and measured feature values between the cells. A value of 1 indicates that the position and feature values are unchanged between frames.

Each cell in the current frame 't' is compared to the cells in the following frame 't+1' and a potential trajectory is computed for each pair. Individual movement score contributions are calculated for each feature by computing the differences between the features. A threshold value determines the range over which the feature is active.

$$M(f) = 1 - \left(1 + e^{s(f)}\right)^{-1} \quad (1)$$

where

$$s(f) = \alpha\left(\frac{T(f) - D(f)}{D(f)}\right)$$

The movement score for an individual feature is given in equation (1), where $T(f)$ is the threshold, $D(f)$ is the difference between the values of a particular feature f as found in Table 1, and α determines the steepness of the curve (value to be obtained through optimization). The sigmoidal shape penalizes large changes in feature value, greater than the threshold T.

Threshold values are obtained by performing an initial tracking followed by analysis of the change in features (see Figure S6 and Table S1). A threshold can be selected by choosing a high percentile (95^{th}–99^{th}) as a cut-off, which will give a value suitable for the majority of cells in the experiment.

Each of the features has a weight which is proportional to the contribution towards the total movement score for the trajectory. Initial estimates of the weight values are obtained by determining the relative importance of each feature according to the strength of the correlation (see Figure 3, and R^2 values in Table 1). The features with the highest correlation values (coordinates and intensity) were assigned an initial weight of 0.9 with the other features assigned weights of 0.5.

Weights and thresholds are subsequently optimized by locally varying them in an iterative manner, while maximizing the tracking performance. Each parameter is perturbed in turn by a small amount ($\pm 1\%$ of the parameter range) with the new values retained if the tracking score is improved. The optimizer attempts to avoid local minima by gradually increasing the scale of the perturbations if repeated iterations fail to improve the score.

The individual scores are combined using equation (2) as the product of all feature weights and movement scores.

$$M = \Pi_f(1 - W(f)(1 - M(f))) \quad (2)$$

Assigning trajectories

Assigning movements is a four-stage process (see Figure S7). The first step builds a list of potential target cells in the adjacent frames according to the movement scores in the transition matrix. Each cell holds a list of highest scoring cells in both the forward ($t \rightarrow t+1$) and backward ($t \rightarrow t-1$) directions.

The second stage assigns a trajectory if the highest scoring forward transition agrees with the highest scoring inbound transition of the target cell at t+1 (see Figure S8). Step 2 is performed repeatedly until all such transitions have been assigned. The third step completes any remaining links by assigning the highest forward pointing transition.

The final step optimizes the tracking by calculating the sum of transition scores for each frame. If two cells share potential targets, a new transition score is calculated based on exchanging the trajectories. The new trajectories are retained if the exchange improves the total score.

The method of assigning trajectories may be replaced with the Hungarian Algorithm [45,46], while retaining the initial matrix calculation. The Hungarian method requires a square matrix; therefore an additional step is required to pad the matrix where there are different numbers of cells in adjacent frames. Although the tracking accuracies with the Hungarian method are very similar, the main advantage of our custom assignment is that it is capable to account for the detection of cell divisions.

Detection of divisions

The large frame intervals used in the C2C12 experiments lead to difficulties in identifying cell divisions. The M-phase of the cell cycle is relatively brief and can occur between frames; therefore, the change in appearance of the nucleus during M-phase cannot be relied upon to detect divisions. Also, directional information about daughter cells moving in opposite directions during division could not be used, as there was no significant correlation observed between frames.

The first step in locating divisions is to identify cells which may have divided by making use of dynamic features obtained during tracking, in particular, characteristic changes in intensity and nuclear area (Figure 3 C,E), which both decrease by at least 25% during cell division (Figures S9 and S10).

The integrated intensity of the parent cell is very closely retained in the daughter cells ($R^2 = 0.95$, sum of daughter intensities is $100 \pm 1.5\%$ of parent cells, errors indicate standard error of the mean, $n = 100$ cell divisions), and there is a close correlation between the two daughter cells ($R^2 = 0.92$, mean difference between daughter cells $6.0 \pm 0.5\%$). The daughter cells in the frame immediately following a division were of a similar size to each other (average difference $12.6 \pm 1.0\%$), and for the sum of daughter cell areas we obtain an average total $110 \pm 4.3\%$ of parent cell area. There were some cases where a daughter cell was larger than the final measured area of the parent cell due to the long frame interval and chromatin condensation occurring during the previous frame. Because of this and the larger variation obtained for the area, cell size (weight 0.25) is weighted lower than intensity (weight 1).

Potential daughter cells are selected by examining cells within a certain distance of the parent cell. These cells are examined one pair at a time, and a similarity score is calculated using equation (2) based on intensity and size only. The most favourable daughter pairs are compared to the parent cell by re-evaluating equation (2) using a 'composite cell' where the area and intensities are the sums of the daughter values, again using weights of 1 and 0.25 for intensity and area, respectively. Finally, daughter cells with the highest score are selected.

Tracking accuracy

To compare tracking accuracies of our method with CellProfiler and ImageJ's Particle Tracker (https://weeman.inf.ethz.ch/ParticleTracker), we used an experiment with 24 frames in total (frame intervals of 10 minutes). The average cell movement between frames was 3.9 pixels, with a maximum of 28 pixels (average nucleus diameter was 11 pixels). The cell density (1300 cells/mm^2) was in the middle of the range of our 30 minute experiment described earlier. We created a gold standard, whereby the segmentation and tracking were manually adjusted until at least 50% of the visible cell nuclei had been tracked. The gold standard contains 7017 individual cell to cell linkages between frames, with 359 tracks ranging from 5 to 23 frames (average 19). The tracking accuracy was measured by counting the number of individual links that were correctly identified using the automated methods and the longest continuously tracked section (Table 2, Figure 4).

Table 2. Results of gold standard tracked sets.

Experiment:	*24 frames (10 minute interval), gold standard.*	*110 frames (30 minute interval), gold standard.*
Validated Positions	7321	7417
Validated Trajectories	359	157
Frame to Frame links	6886	7221
Average track length	19	46
Tracking Scores:		
LineageTracker (Custom assignment)	97.7/91.8	97.2/85.3
LineageTracker, (Hungarian Assignment)	98.1/94.2	96.9/89.1
CellProfiler*	95.9/88.3	96.1/85.4
Particle Tracker (ImageJ)	92.3/82.9	86.4/64.1
Cellomics	n/a	85.9/55.9

Two numbers are given for each measurement: total number of correctly tracked steps and longest continuously tracked section (as percentage of total steps). For the 10 minute interval experiment, the seeded growth algorithm was used, and segmentations were manually edited, so that 50% of cells with positively validated segmentations were included in the tracking gold standard. The 30 minute interval experiment is based on the Cellomics segmentation, as to allow comparison of the Cellomics tracking routines with other ones.

*CellProfiler tracking using LAP (Linear Assignment Problem) tracking.

doi:10.1371/journal.pone.0027886.t002

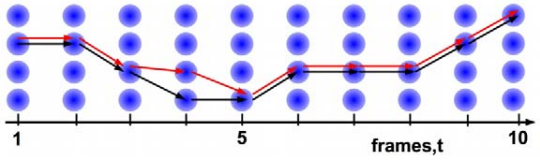


Figure 4. Measuring tracking accuracy. Horizontal axis shows time with the vertical axis representing cells in the frames. The red line is the manually tracked 'gold standard' route marked through the cells, and the black line is the calculated tracking. Tracking accuracy is measured by counting the total number of steps which match the gold standard and the longest continuous chain of correct steps.
doi:10.1371/journal.pone.0027886.g004

While our custom method with 97.7% correctly identified linkages compares similarly to CellProfiler (95.9%), ImageJ's Particle Tracker more generic feature point tracking, which like our method also includes intensity and higher order intensity moments as features, has a slightly lower detection rate of 92.3%. Next, a tracking 'gold standard' was created using the longer 48 hour time-series data with 30 minute frame intervals from the same experiment used for the segmentation standard. 157 cell trajectories were created in our tracking viewer/editor containing a total of 7221 individual steps. Track lengths range from 5 to 110 frames (average 46). Average cell movement was 3.8 pixels per frame (maximum 29 pixels per frame, average cell diameter of 14 pixels). Additionally, this experiment includes 100 cell divisions. Results for our method and CellProfiler are very similar to the previous experiment, whereas the Particle Tracker plugin shows a markedly decreased rate of accuracy for the longest continuously tracked section (Table 2), possibly because of higher cell densities encountered in the 30 min interval experiment.

Execution times are comparable for all methods, taking approximately 1–2½ minutes on a 2.4GHz Intel Core i5 running OSX 10.6.7. These times decrease for the custom tracking when an optimized value for the Distance Threshold is used, to below 10 seconds for the custom assignment and approximately 1 minute for the Hungarian assignment.

Division accuracy and daughter cell fluorescence

The main purpose of our software development was to create a framework that allowed tracking of cells through cell divisions. To determine the accuracy of detecting cell divisions, we considered the 110-frame experiment. Out of the 100 manually annotated cell divisions, 80 were correctly identified by the software. There were 16 false positive divisions detected: two where a division was correctly identified, but the daughter cells were assigned incorrectly, and the remaining 14 where a division was detected and none occurred. In a series of additional experiments, our software was used to study the partitioning of a cis-regulatory module promoter driven GFP between daughter cells for dividing C2C12 cells. Transient transfections were performed with reporters containing four different Msx1 transcriptional regulatory regions (A–D) upstream of the Msx1 promoter and the promoter alone (Vance et al., submitted). The fluorescence activity of mother and daughter cells was measured for a total 96 divisions. These cells were manually validated. The partitioning between daughters is summarized in figure 5A ($R^2 = 0.92$). The high correlation in the partitioning means that for all the different Msx1 promoter constructs driving GFP expression, we find that fluorescence is symmetrically distributed in the two daughter cells with a high degree of accuracy, ensuring that in most cases Msx1 levels are maintained during cell divisions to prevent differentiation. The total fluorescence recovery (measured as the percentage of fluorescence in the daughter cells compared to the mother cell) is summarized in figure 5B, C. A correlation between mother fluorescence and total daughter fluorescence yields an R^2 value of 0.86. This lower value most likely reflects degradation of GFP during cell division, when transcription of GFP under the control of the Msx1 promoter ceases.

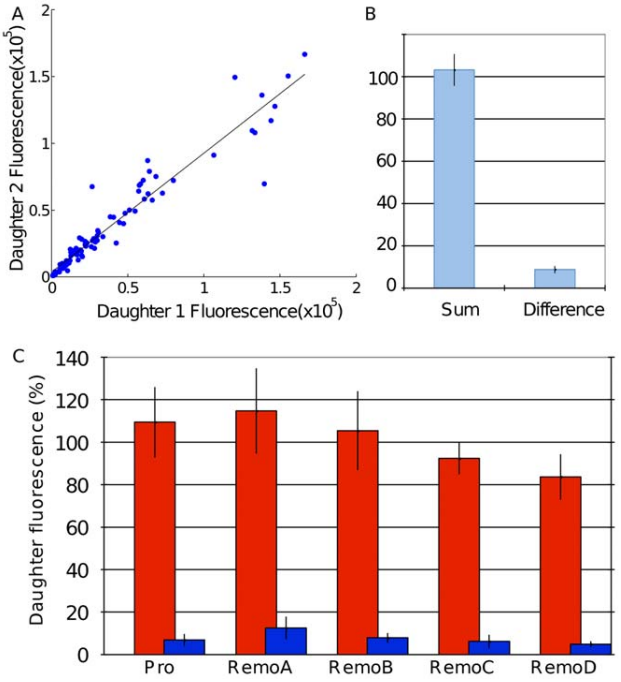


Figure 5. GFP Fluorescence measurements across cell divisions. A) Correlation plots of daughter fluorescence ($R^2 = 0.92$) taken from the 5 Msx1 ReMo constructs. B) Sum of daughter fluorescence and difference between daughter fluorescence, as a percentage of parent fluorescence. C) Breakdown of sum and difference of intensities for the 5 different Msx1 ReMo constructs.
doi:10.1371/journal.pone.0027886.g005

Tracking cells without a permanent nuclear marker

The software was originally designed to track cells which contained a continuously visible fluorescent marker. To show that this is not an absolute requirement, we use it here to obtain intensity profiles of zebrafish embryonic PAC2 cells, expressing FUCCI cell cycle markers visible for the most of the duration of the cell cycle. The markers consist of two ubiquitin ligase substrates, which are expressed during different phases of the cell cycle [39] and have been fused with red- and green-emitting fluorescent proteins [40]. The nuclei of cells in the G1 phase appear red and change to green during the S, G2 and M phases of the cell cycle (Figures 6 and 7). There is an overlap during the G1 to S transition where both markers are visible, giving the nuclei a yellow colour (Figure 6, bottom panel). At mitosis, there is a rapid decrease in intensity in the green channel, but there is a short delay before the cell becomes visible in the red channel. Because of that delay, there is insufficient difference between daughter cells and background for accurate automatic detection, so manual intervention is required for a short section of each lineage (Figures S11 and S12 and Table S2). As described in Text S2, differences in the colour channels inform the seeded growth algorithm, as well as the tracking module in order to facilitate discrimination

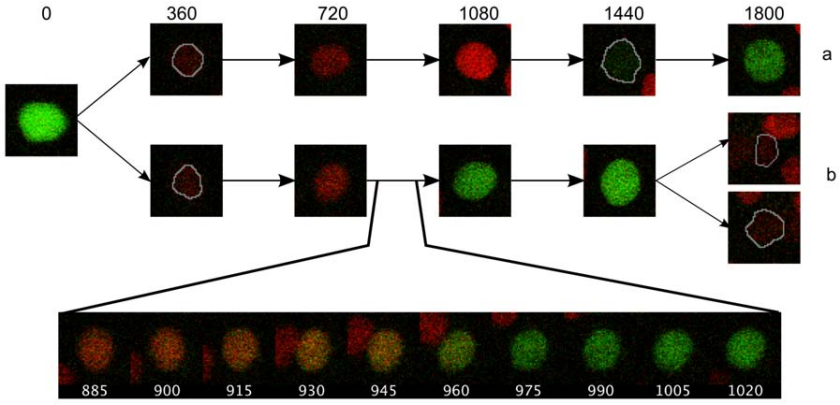


Figure 6. Colour changes during the cell cycle indicated by FUCCI markers in two daughter cells labelled a and b (see also Figure 7). Time is in minutes following division. The overlap in the red and green fluorescence (transition between G1 and S phase) is shown for cell b (bottom panel). White outlines are given for nuclei showing weak fluorescence.
doi:10.1371/journal.pone.0027886.g006

between nearby cells at different phases of cell cycle (see also Figure S13).

Conclusions

Currently, there are few alternatives for automated cell tracking that are freely available, such as CellTracker, CellID, CellProfiler, CellTracer, and Overlap-Based Cell Tracker. All of them have shortcomings with large cell displacements between frames, and do not allow for automated cell lineage construction. Our method, which is based on the ImageJ plugin architecture, has demonstrated a similar performance to CellProfiler when it comes to cell segmentation, but has the added feature of cell lineage construction capabilities, and the advantage to interactively correct segmentation or tracking mistakes.

It can read data files produced from CellProfiler to allow visualization and editing of segmentation and tracking output, in order to compare between different tracking solutions implemented in CellProfiler and ImageJ. The Seeded Growth segmentation we used detected cells with 92% accuracy with <1% false positives. Cell tracking followed entire trajectories (of mean length 45 cell-cell transitions) with 85% accuracy. This is similar to results in [33], but does not reach the higher accuracies reported in [34], in which cells exhibit less motion between frames and are less clustered. The gold standard we release (15,000 validated cell positions) has a longer average of 19 and 46 tracked frames for the 10 min and 30 min interval experiments with 359 and 157 tracks for each of the experiments when compared to an average track length of 13 frames in [34]. We found for different Msx1 promoter constructs that there is a high level of accuracy when distributing GFP fluorescence to daughter cells during cell divisions. Additionally, as shown in the example of FUCCI cell cycle markers, our software can be easily adapted to different cell types and fluorescent markers.

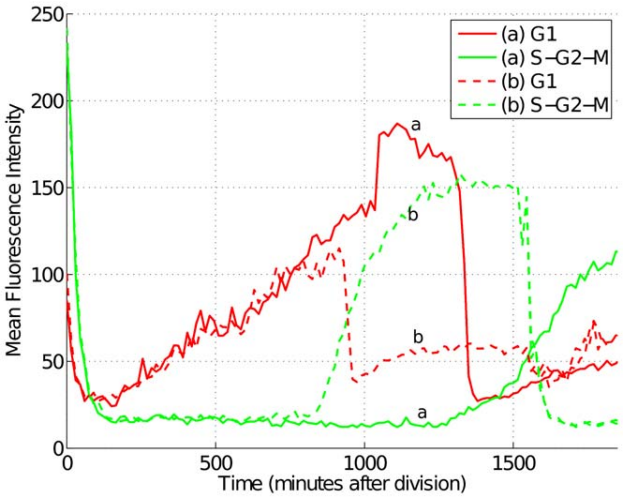


Figure 7. Intensities of the FUCCI markers following cell division. Fluorescence intensity following cell division for the two daughter cells in figure 6. The two FUCCI channels have been shown for an entire cell cycle. The G1 signal (red) increases gradually following mitosis, then decreases following a rise in S-G2-M signal (green). A magnified view of the first 3 hours is shown in Figures S11 and S12.
doi:10.1371/journal.pone.0027886.g007

Availability and future directions

The software and source code can be downloaded from http://go.warwick.ac.uk/lineagetracker. Additional segmentation or tracking methods are possible by adding modules for tracking or lineage construction within the software. Current segmentation methods have been optimized for circular nuclei. Different methods could be substituted for segmenting different shapes, such as rod-shaped yeast or bacterial cells, or when using different fluorescent stains, such as GFP-histone for labelling cell nuclei [47].

The tracking comparison and benchmarking software will be made available from the lineagetracker website.

Our statistical scoring framework can, in principle, be translated into a more formal framework of a graph based problem, as used by Padfield [34] or others. Here we have chosen it for the simplicity with which it can be implemented and the ease in which dynamic features can be incorporated.

Supporting Information

Figure S1 Distribution of nuclei sizes follows a gamma distribution. A) 110 frames (30 min intervals) experiment of C2C12 cells (n = 62586 , $\gamma = 7.4$, $\beta = 20.2$). B) Analysis of the first three frames of the sequence showing the distribution of all nuclei that have been automatically identified using the built-in Cellomics segmentation (1235 cells, blue and red), Blue is a subset of nuclei that have been manually validated to be non-overlapping

(n = 1198). The corresponding gamma curve has parameters $\gamma = 11.1$ and $\beta = 12.0$. Red contains nuclei that have been confirmed to be overlapping by visual inspection (35 nuclei, 2.8% of total), i.e. where two nuclei were reported as one. 1 nucleus was oversegmented, i.e. falsely reported as two.
(TIF)

Figure S2 Example of C2C12 cell motion. The highlighted cell has been tracked through multiple frames. Scale bar is 50 microns. Time is displayed in minutes. A) Hoechst channel B) GFP Channel.
(TIF)

Figure S3 Segmentation score plots. A) Artificial cell images from Simcep [42]. B) Ground Truth image. C) Precision, Recall & F-Score for the SimCep images. D) Comparison of cell detection accuracies for various segmentation methods.
(TIF)

Figure S4 Segmentation of C2C12 cells at a higher resolution, obtained using a 20× NA 0.75 objective.
(TIFF)

Figure S5 Correlation plots with dividing cells coloured in red. Top: Change in Hoechst intensity, Change in 2nd order intensity moment, Correlation in standard deviation. Bottom: intensity correlations for daughter cells, parent fluorescence against sum of daughter fluorescence, parent cell area against sum of daughter areas.
(TIF)

Figure S6 Measuring changes in features for cell-cell transitions during tracking. A) Change in cell areas (pixels) in adjacent frames. B) Distance moved by non-dividing cells in one frame. C) Percent change in Hoechst fluorescence for non-dividing cells. D) Distribution of daughter cell distances (in pixels) from parent cell in the frame immediately following a division.
(TIF)

Figure S7 A) Tracking flow chart. B) Expanded flow chart for the Detect Divisions module. (Adapted from [24] © 2011 IEEE).
(TIF)

Figure S8 Demonstration of three iterations of the assignment step. 1, 2 & 3 represent three cells in time t, a, b & c are three cells at time t+1. Numbers on arrows indicate movement scores. A) The highest scoring link between 2-c is selected. B) Links to and from cells 2 & c are removed. The highest scoring link 3-b is selected. C) Links involving cells 3 & b are removed, leaving 1-a.
(TIF)

Figure S9 The cell divisions from figure 1B, showing changes in Hoechst intensity. For each row, the left plot displays the integrated Hoechst intensity; the right plot displays mean Hoechst intensity. (S9A adapted from [24] © 2011 IEEE).
(TIF)

Figure S10 Cell tracked across 3 generations. A) Intensity profile of the lineage showing GFP fluorescence. B&C) Highlighted sections of the cell trajectory. Tracks are colour coded to match the intensity plot. Inset shows the cell highlighted.
(TIF)

Figure S11 Intensity drop following division for zebrafish PAC2 cells. The image background intensity and sum of image channels for the measured cell are also plotted.
(TIF)

Figure S12 Dividing cell visualised using FUCCI markers. The green FUCCI S-G2-M marker fades after mitosis followed by a slow increase in red G1 marker. Time displayed in minutes same as Figure S11 above.
(TIF)

Figure S13 Segmentation of zebrafish PAC2 cells using the 'Multi-Channel Segmentation' method.
(TIF)

Table S1 90–99th percentile values for change in area, frame to frame displacement during tracking, and parent-daughter distance following cell division. These values (measured in pixels) are used to select the initial threshold parameters used for tracking.
(PDF)

Table S2 Tracking precision for zebrafish PAC2 cells visualised using FUCCI markers [39–41]. The segmentation and tracking adjustments represent the percentage of frames which required manual intervention to preserve accurate tracking. The longest continuous sequence was observed with cell 8 at over 50 hours without corrections. Following division, daughter cells fade to close to background intensity requiring cells to be manually segmented.
(PDF)

Text S1 Segmentation of cell nuclei.
(PDF)

Text S2 Description of algorithms and parameters used for segmentation.
(PDF)

Text S3 Description of LineageTracker software user interface.
(PDF)

Acknowledgments

We would like to acknowledge Professor Atsushi Miyawaki and colleagues at the Riken Brain Science Institute for the zebrafish FUCCI plasmids, and the FACS facility at Cancer Research UK for their cell sorting expertise. We would also like to give special thanks to Kara L. Cerveny for helping to generate the zebrafish FUCCI cell line and making the time-lapse movie.

Author Contributions

Conceived and designed the experiments: MJD DMJ SO GK KWV TB. Performed the experiments: MJD DMJ KWV. Analyzed the data: MJD TB. Contributed reagents/materials/analysis tools: DMJ SO GK KWV TKT. Wrote the paper: MJD TB.

References

1. Cohen AA, Geva-Zatorsky N, Eden E, Frenkel-Morgenstern M, Issaeva I, et al. (2008) Dynamic Proteomics of Individual Cancer Cells in Response to a Drug. Science 322: 1511–1516.
2. Longo D, Hasty J (2006) Dynamics of single-cell gene expression. Molecular Systems Biology 2.
3. Sigal A, Milo R, Cohen A, Geva-Zatorsky N, Klein Y, et al. (2006) Dynamic proteomics in individual human cells uncovers widespread cell-cycle dependence of nuclear proteins. Nature Methods 3: 525–531.
4. Spiller DG, Wood CD, Rand DA, White MRH (2010) Measurement of single-cell dynamics. Nature 465: 736–745.
5. Rosenfeld N (2005) Gene Regulation at the Single-Cell Level. Science 307: 1962–1965.
6. Li K, Miller ED, Chen M, Kanade T, Weiss LE, et al. (2008) Cell population tracking and lineage construction with spatiotemporal context. Medical Image Analysis 12: 546–566.
7. Meijering E, Dzyubachyk O, Smal I, van Cappellen WA (2009) Tracking in cell and developmental biology. Seminars in Cell & Developmental Biology 20: 894–902.
8. Giuliano KA, Cheung WS, Curran DP, Day BW, Kassick AJ, et al. (2005) Systems Cell Biology Knowledge Created from High Content Screening. ASSAY and Drug Development Technologies 3: 501–514.

9. Haney SA, LaPan P, Pan J, Zhang J (2006) High-content screening moves to the front of the line. Drug Discovery Today 11: 889–894.
10. Thomas N (2010) High-Content Screening. Journal of Biomolecular Screening 15: 1–9.
11. Pepperkok R, Ellenberg J (2006) High-throughput fluorescence microscopy for systems biology. Nat Rev Mol Cell Biol 7: 690–696.
12. Finkenstadt B, Heron EA, Komorowski M, Edwards K, Tang S, et al. (2008) Reconstruction of transcriptional dynamics from gene reporter data using differential equations. Bioinformatics 24: 2901–2907.
13. Tyson RA, Epstein DBA, Anderson KI, Bretschneider T (2010) High Resolution Tracking of Cell Membrane Dynamics in Moving Cells: an Electrifying Approach. Math Model Nat Phenom 5: 34–55.
14. Shen H, Nelson G, Nelson DE, Kennedy S, Spiller DG, et al. (2006) Automated tracking of gene expression in individual cells and cell compartments. Journal of The Royal Society Interface 3: 787–794.
15. Gordon A, Colman-Lerner A, Chin TE, Benjamin KR, Yu RC, et al. (2007) Single-cell quantification of molecules and rates using open-source microscope-based cytometry. Nature Methods 4: 175–181.
16. Wang Q, Niemi J, Tan CM, You L, West M (2010) Image segmentation and dynamic lineage analysis in single-cell fluorescence microscopy. Cytometry Part A 77A: 101–110.
17. Chalfoun J, Cardone A, Dima AA, Halter M, Allen DP (2010) Overlap-Based Cell Tracker, NIST Interagency/Internal Report (NISTIR) - 7663.
18. Reyes-Aldasoro CC, Akerman S, Tozer GM (2008) Measuring the velocity of fluorescently labelled red blood cells with a keyhole tracking algorithm. Journal of Microscopy 229: 162–173.
19. Erik Meijering E, Smal I, Dzyubachyk O, Olivo-Marin J-C (2008) Time-Lapse Microscopy Imaging. In: Qiang W, Merchant F, Castleman K, eds. Microscope Image Processing. Waltham, MA: Elsevier Academic Press. 576 p.
20. Altinok A, El-saban M, Peck AJ, Wilson L, Feinstein SC, et al. (2006) Activity analysis in microtubule videos by mixture of hidden markov models. IEEE CVPR 2: 1662–1669.
21. Jaqaman K, Loerke D, Mettlen M, Kuwata H, Grinstein S, et al. (2008) Robust single-particle tracking in live-cell time-lapse sequences. Nat Meth 5: 695–702.
22. Smal I, Draegestein K, Galjart N, Niessen W, Meijering E (2008) Particle Filtering for Multiple Object Tracking in Dynamic Fluorescence Microscopy Images: Application to Microtubule Growth Analysis. Medical Imaging, IEEE Transactions on 27: 789–804.
23. Schrijver A (2003) Combinatorial Optimization. Berlin, Germany: Springer-Verlag. 1800 p.
24. Downey M, Vance KW, Bretschneider T (2011) LineageTracker: A Statistical Scoring Method For Tracking Cell Lineages In Large Cell Populations With Low Temporal Resolution. 2011 8TH IEEE INTERNATIONAL SYMPOSIUM ON BIOMEDICAL IMAGING: FROM NANO TO MACRO, IEEE International Symposium on Biomedical Imaging. pp 1913–1916.
25. Loo L-H, Lin H-J, Steininger RJ, Wang Y, Wu LF, et al. (2009) An approach for extensibly profiling the molecular states of cellular subpopulations. Nat Meth 6: 759–765.
26. Murphy RF, Velliste M, Porreca G (2003) Robust Numerical Features for Description and Classification of Subcellular Location Patterns in Fluorescence Microscope Images. The Journal of VLSI Signal Processing 35: 311–321.
27. Debeir O, Milojevic D, Leloup T, Van Ham P, Kiss R, et al. (2005) Mitotic Tree Construction by Computer In Vitro Cell Tracking: a Tool for Proliferation and Motility Features Extraction. EUORCON 2005: The International Conference on the Computer as a Tool. 21–24 Nov. 2005, Belgrade, Yugoslavia. pp 951–954.
28. Wang M, Zhou X, King R, Wong S (2007) Context based mixture model for cell phase identification in automated fluorescence microscopy. BMC Bioinformatics 8: 32.
29. Olariu V, Coca D, Billings SA, Tonge P, Gokhale P, et al. (2009) Modified variational Bayes EM estimation of hidden Markov tree model of cell lineages. Bioinformatics 25: 2824–2830.
30. Al-Kofahi O, Radke RJ, Goderie SK, Shen Q, Temple S, et al. (2006) Automated Cell Lineage Construction: A Rapid Method to Analyze Clonal Development Established with Murine Neural Progenitor Cells. Cell Cycle 5: 327–335.
31. Harder N, Mora-Bermudez F, Godinez WJ, Wunsche A, Eils R, et al. (2009) Automatic analysis of dividing cells in live cell movies to detect mitotic delays and correlate phenotypes in time. Genome Research 19: 2113–2124.
32. Walter T, Held M, Neumann B, Hériché J-K, Conrad C, et al. (2010) Automatic identification and clustering of chromosome phenotypes in a genome wide RNAi screen by time-lapse imaging. Journal of Structural Biology 170: 1–9.
33. Bise R, Yin Z, Kanade T (2011) Reliable Cell Tracking By Global Data Association. In Proc.ISBI 2011. pp 1004–1010.
34. Padfield D, Rittscher J, Roysam B (2010) Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. Medical Image Analysis 21: 374–385.
35. Carpenter AE, Jones TR, Lamprecht MR, Clarke C, Kang I, et al. (2006) CellProfiler: image analysis software for identifying and quantifying cell phenotypes. Genome Biology 7: R100.
36. Sbalzarini IF, Koumoutsakos P (2005) Feature point tracking and trajectory analysis for video imaging in cell biology. Journal of Structural Biology 151: 182–195.
37. Bendall AJ, Abate-Shen C (2000) Roles for Msx and Dlx homeoproteins in vertebrate development. Gene 247: 17–31.
38. Lidral AC, Reising BC (2002) The Role of MSX1 in Human Tooth Agenesis. Journal of Dental Research 81: 274–278.
39. Sakaue-Sawano A, Kurokawa H, Morimura T, Hanyu A, Hama H, et al. (2008) Visualizing Spatiotemporal Dynamics of Multicellular Cell-Cycle Progression. Cell 132: 487–498.
40. Sugiyama M, Sakaue-Sawano A, Iimura T, Fukami K, Kitaguchi T, et al. (2009) Illuminating cell-cycle progression in the developing zebrafish embryo. Proceedings of the National Academy of Sciences 106: 20812–20817.
41. Tamai TK, Young LC, Whitmore D (2007) Light signaling to the zebrafish circadian clock by Cryptochrome 1a. Proceedings of the National Academy of Sciences 104: 14712–14717.
42. Lehmussola A, Ruusuvuori P, Selinummi J, Rajala T, Yli-Harja O (2008) Synthetic Images of High-Throughput Microscopy for Validation of Image Analysis Methods. Proceedings of the IEEE 96: 1348–1360.
43. Hu Y, Osuna-Highley E, Hua J, Nowicki TS, Stolz R, et al. (2010) Automated analysis of protein subcellular location in time series images. Bioinformatics 26: 1630–1636.
44. Cohen AR, Gomes FL, Roysam B, Cayouette M (2010) Computational prediction of neural progenitor cell fates. Nat Methods 7: 213–218.
45. Jonker R, Volgenant T (1986) Improving the Hungarian assignment algorithm. Operations Research Letters 5: 171–175.
46. Wright MB (1990) Speeding up the hungarian algorithm. Computers & Operations Research 17: 95–96.
47. Kanda T, Sullivan KF, Wahl GM (1998) Histone-GFP fusion protein enables sensitive analysis of chromosome dynamics in living mammalian cells. Current Biology 8: 377–385.

C.1 Supporting tables and figures for Downey *et al.* PLoS ONE (2011)

Figures not reproduced as part of the thesis are included below:

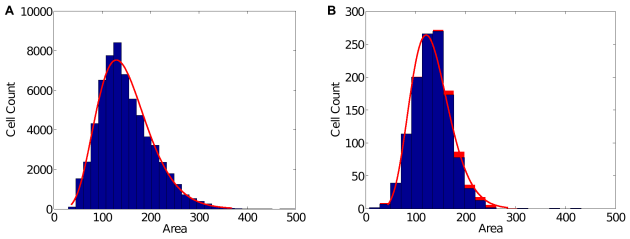


Figure S1: Distribution of nuclei sizes follows a gamma distribution.(Figure S1A reproduced as Figure 3.4)

Figure S2: Example of C2C12 cell motion (reproduced as Figure 3.13).

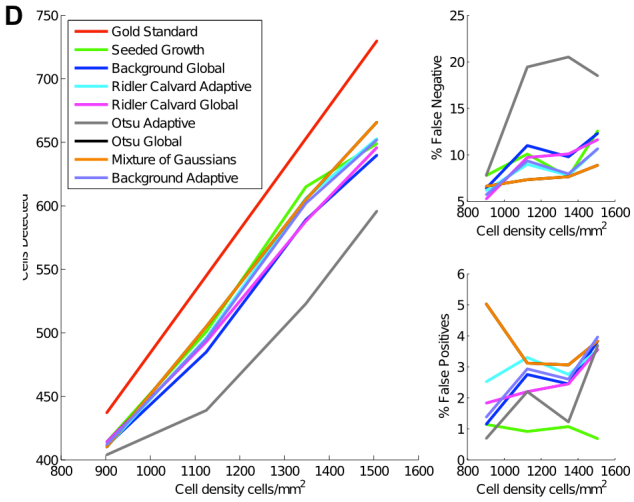


Figure S3: Segmentation score plots. (A+B reproduced as Figure B.3, C as Figure 5.7, D reproduced above.)

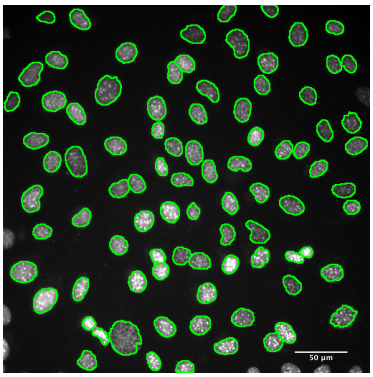


Figure S4: Segmentation of C2C12 cells at a higher resolution, obtained using a 20x NA 0.75 objective.

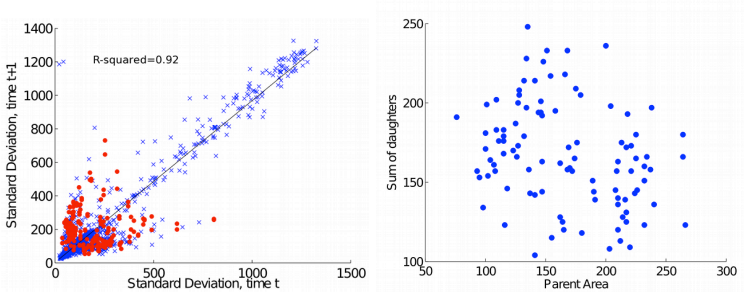


Figure S5: Correlation plots with dividing cells coloured in red. (A-B reproduced as part of Figure 3.9, D-E as Figure 4.3, C & F reproduced above.)

Figure S6: Measuring changes in features for cell-cell transitions during tracking. (Reproduced as Figure 3.12)

Figure S7: A) Tracking flow chart. B) Expanded flow chart for the Detect Divisions module. (Reproduced as Figure 3.16b)

Figure S8: Demonstration of three iterations of the assignment step. (Reproduced as Figure 3.14)

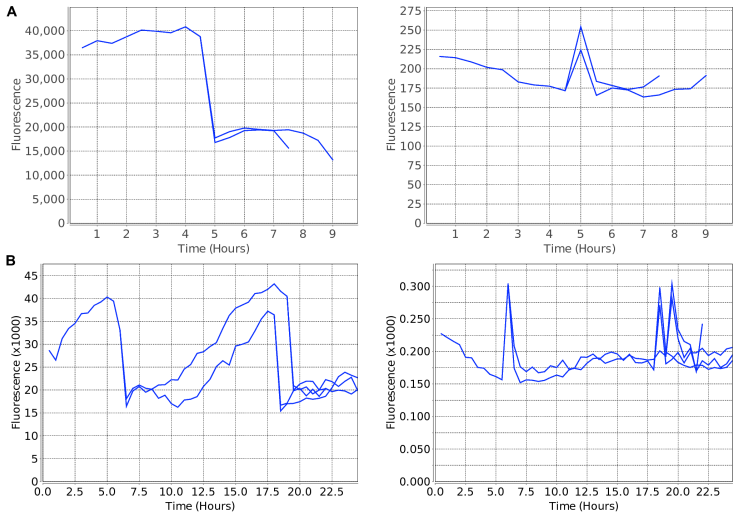


Figure S9: The cell divisions from figure 1B, showing changes in Hoechst intensity.

Figure S10: Cell tracked across 3 generations.(Reproduced as Figure 3.8)

Figure S11: Intensity drop following division for zebrafish PAC2 cells. (Reproduced as Figure 7.3b)

Figure S12: Dividing cell visualised using FUCCI markers. (Reproduced as Figure 7.3a)

Figure S13: Segmentation of zebrafish PAC2 cells using the Multi-Channel Segmentation method. (Reproduced as Figure 3.5)

Table S1: 90-99th percentile values for change in area. (Reproduced as Table 3.2)

Table S2: Tracking precision for zebrafish PAC2 cells visualised using FUCCI markers.(Reproduced as Table 7.1)

Text S1: Segmentation of cell nuclei (See Section 3.1)

Text S2: Description of algorithms and parameters used for segmentation (See Appendix B.2)

Text S3: Description of LineageTracker software user interface (See Appendix A.5)

Bibliography

[1] Xavier Darzacq, Jie Yao, et al. Imaging Transcription in Living Cells. *Annual Review of Biophysics*, 38(1):173–196 (2009).

[2] David G. Spiller, Christopher D. Wood, David A. Rand, and Michael R. H. White. Measurement of single-cell dynamics. *Nature*, 465(7299):736–745 (2010).

[3] Diane Longo and Jeff Hasty. Dynamics of single-cell gene expression. *Molecular Systems Biology*, 28:1–10 (2006). doi:doi:10.1038/msb4100110.

[4] James C. W. Locke and Michael B. Elowitz. Using movies to analyse gene circuit dynamics in single cells. *Nat Rev Micro*, 7(5):383–392 (2009).

[5] D F Pisani, C Cabane, B Derijard, and C A Dechesne. The topoisomerase 1-interacting protein btbd1 is essential for muscle cell differentiation. *Cell Death and Differentiation*, 11(11):1157–1165 (2004).

[6] Alasdair MacKenzie, Lorna Purdie, et al. Two enhancer domains control early aspects of the complex expression pattern of *Msx1*. *Mechanisms of Development*, 62(1):29 – 40 (1997). doi:DOI:10.1016/S0925-4773(96)00646-6.

[7] Martin Chalfie, Yuan Tu, et al. Green Fluorescent Protein as a Marker for Gene Expression. *Science*, 263(5148):802–805 (1994).

[8] Måns Ehrenberg. The Nobel Prize in Chemistry. Technical report, The Royal Swedish Academy of Sciences (2008).

[9] Jennifer Lippincott-Schwartz and George H. Patterson. Development and Use of Fluorescent Protein Markers in Living Cells. *Science*, 300:87–91 (2003).

[10] Takeharu Nagai, Keiji Ibata, et al. A variant of yellow fluorescent protein with fast and efficient maturation for cell-biological applications. *Nat Biotech*, 20(1):87–90 (2002).

[11] Xianqiang Li, Xiaoning Zhao, et al. Generation of destabilized green fluorescent protein as a transcription reporter. *J. Biol. Chem.*, 273(52):34970–34975 (1998).

[12] Scott Rogers, Rodney Wells, and Martin Rechsteiner. Amino acid sequences common to rapidly degraded proteins: The pest hypothesis. *Science*, 234(4774):364–368 (1986).

[13] Jyrki Selinummi, Pekka Ruusuvuori, et al. Bright field microscopy as an alternative to whole cell fluorescence in automated analysis of macrophage images. *PLoS ONE*, 4(10):e7497 (2009). doi:10.1371/journal.pone.0007497.

[14] Zhaozheng Yin and T. Kanade. Restoring artifact-free microscopy image sequences. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 909 –913 (2011). doi:10.1109/ISBI.2011.5872551.

[15] Smadar Ben-Tabou de Leon and Eric H. Davidson. Gene Regulation: Gene Control Network in Development. *Annual Review of Biophysics and Biomolecular Structure*, 36(1):191–212 (2007). doi:10.1146/annurev.biophys.35.040405.102002.

[16] J.L. Rubio-Guivernau, M.A. Luengo-Oroz, et al. Combining sea urchin embryo cell lineages by error-tolerant graph matching. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 5918 –5921 (2009). doi:10.1109/IEMBS.2009.5334851.

[17] Philipp J. Keller, Annette D. Schmidt, Joachim Wittbrodt, and Ernst H.K. Stelzer. Reconstruction of zebrafish early embryonic development by scanned light sheet microscopy. *Science*, 322(5904):1065–1069 (2008). doi:10.1126/science.1162493.

[18] Nicolas Olivier, Miguel A. Luengo-Oroz, et al. Cell lineage reconstruction of early zebrafish embryos using label-free nonlinear microscopy. *Science*, 329(5994):967–971 (2010). doi:10.1126/science.1189428.

[19] M.A. Luengo-Oroz, T. Savy, et al. Processing pipeline for digitalizing the lineage tree of early zebrafish embryogenesis from multiharmonic imaging. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 1561 –1564 (2011). doi:10.1109/ISBI.2011.5872699.

[20] J.E. Sulston, E. Schierenberg, J.G. White, and J.N. Thomson. The embryonic cell lineage of the nematode Caenorhabditis elegans. *Developmental Biology*, 100(1):64–119 (1983).

[21] C. Thomas, P. DeVries, J. Hardin, and J. White. Four-dimensional imaging: Computer visualization of 3d movements in living specimens. *Science*, 273(5275):603–607 (1996). doi:10.1126/science.273.5275.603.

[22] Nick Thomas. High-Content Screening. *Journal of Biomolecular Screening*, 15(1):1–9 (2010).

[23] Nitzan Rosenfeld, Theodore J. Perkins, et al. A Fluctuation Method to Quantify In Vivo Fluorescence Data. *Biophysical Journal*, 91(2):759–766 (2006).

[24] Mijung Yeom, Julie S. Pendergast, Yoshihiro Ohmiya, and Shin Yamazaki. Circadian-independent cell mitosis in immortalized fibroblasts. *Proceedings of the National Academy of Sciences*, 107(21):9665–9670 (2010). doi:10.1073/pnas.0914078107.

[25] Nitzan Rosenfeld, Jonathan W. Young, et al. Gene Regulation at the Single-Cell Level. *Science*, 307(5717):1962–1965 (2005).

[26] Alex Sigal, Ron Milo, et al. Dynamic proteomics in individual human cells uncovers widespread cell-cycle dependence of nuclear proteins. *Nat Meth*, 3(7):525–531 (2006).

[27] A. A. Cohen, N. Geva-Zatorsky, et al. Dynamic Proteomics of Individual Cancer Cells in Response to a Drug. *Science*, 322(5907):1511–1516 (2008).

[28] Robert F. Murphy, Meel Velliste, and Gregory Porreca. Robust numerical features for description and classification of subcellular location patterns in fluorescence microscope images. *The Journal of VLSI Signal Processing*, 35(3):311–321 (2003).

[29] Estelle Glory and Robert F. Murphy. Automated subcellular location determination and high-throughput microscopy. *Developmental Cell Review*, 12 (2007).

[30] Shann-Ching Chen, Ting Zhao, Geoffrey J. Gordon, and Robert F. Murphy. Automated image analysis of protein localization in budding yeast. *Bioinformatics*, 23(13):i66–71 (2007). doi:10.1093/bioinformatics/btm206.

[31] Eberhard Krausz. High-content siRNA screening. *Mol. BioSyst.*, 3(4):232–240 (2007).

[32] Urban Liebel, Vytaute Starkuviene, et al. A microscope-based screening platform for large-scale functional protein analysis in intact cells. *FEBS Letters*, 554(3):394–398 (2003).

[33] Z. E. Perlman, T. J. Mitchison, and T. U. Mayer. High-content screening and profiling of drug activity in an automated centrosome-duplication assay. *Chem. Eur. J. of Chem. Bio.*, 6(1):145–151 (2005).

[34] Beate Neumann, Michael Held, et al. High-throughput RNAi screening by time-lapse imaging of live human cells. *Nat Meth*, 3(5):385–390 (2006).

[35] Holger Erfle, Beate Neumann, et al. Reverse transfection on cell arrays for high content screening microscopy. *Nat. Protocols*, 2(2):392–399 (2007).

[36] R.A. Tyson, D.B.A. Epstein, K.I. Anderson, and T. Bretschneider. High resolution tracking of cell membrane dynamics in moving cells: an electrifying approach. *Math. Model. Nat. Phenom.*, 5(1):34–55 (2010).

[37] H. Shen, G. Nelson, et al. Automatic tracking of biological cells and compartments using particle filters and active contours. *Chemometrics and Intelligent Laboratory Systems*, 82(1-2):276–282 (2006).

[38] Andrew Gordon, Alejandro Colman-Lerner, et al. Single-cell quantification of molecules and rates using open-source microscope-based cytometry. *Nat Meth*, 4(2):175–181 (2007).

[39] Q. Wang, J. Niemi, et al. Image segmentation and dynamic lineage analysis in single-cell fluorescence microscopy. *Cytometry*, 77A(1):101–110 (2010).

[40] J. Chalfoun, A. Cardone, et al. Overlap-based cell tracker, nist interagency/internal report (nistir) - 7663. Technical report, NIST (2010).

[41] C. C. Reyes-Aldasoro, S. Akerman, and G. M. Tozer. Measuring the velocity of fluorescently labelled red blood cells with a keyhole tracking algorithm. *Journal of Microscopy*, 229(1):162–173 (2008).

[42] E Meijering, I Smal, O Dzyubachyk, and J.C Olivo-Marin. Time-Lapse Microscopy Imaging. *Microscope Image Processing*, pages 401–440 (2008).

[43] I.F. Sbalzarini and P. Koumoutsakos. Feature point tracking and trajectory analysis for video imaging in cell biology. *Journal of Structural Biology*, 151(2):182–195 (2005).

[44] John C. Crocker and David G. Grier. Methods of Digital Video Microscopy for Colloidal Studies. *Journal of Colloid and Interface Science*, 179(1):298–310 (1996).

[45] Y. Hu, E. Osuna-Highley, et al. Automated analysis of protein subcellular location in time series images. *Bioinformatics*, 26(13):1630–1636 (2010).

[46] Kang Li, Eric D. Miller, et al. Cell population tracking and lineage construction with spatiotemporal context. *Medical Image Analysis*, 12(5):546–566 (2008).

[47] I. Smal, K. Draegestein, et al. Particle filtering for multiple object tracking in dynamic fluorescence microscopy images: Application to microtubule growth analysis. *Medical Imaging, IEEE Transactions on*, 27(6):789–804 (2008).

[48] Alexander Schrijver. *Combinatorial Optimization*, volume 24 of *Algorithms and Combinatorics*. Springer (2003).

[49] Alphan Altinok, Motaz El-Saban, et al. Activity analysis in microtubule videos by mixture of hidden markov models. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:1662–1669 (2006). doi: http://doi.ieeecomputersociety.org/10.1109/CVPR.2006.48.

[50] Omar Al-Kofahi, Richard J. Radke, et al. Automated cell lineage construction: A rapid method to analyze clonal development established with murine neural progenitor cells. *Cell Cycle*, 5(3):327–335 (2006).

[51] Randy W King Meng Wang, Xiaobo Zhou and Stephen TC Wong. Context based mixture model for cell phase identification in automated fluorescence microscopy. *BMC Bioinformatics*, 8 (2007).

[52] Erik Meijering, Oleh Dzyubachyk, Ihor Smal, and Wiggert A. van Cappellen. Tracking in cell and developmental biology. *Seminars in Cell & Developmental Biology*, 20(8):894–902 (2009).

[53] O. Debeir, D. Milojevic, et al. Mitotic tree construction by computer in vitro cell tracking: a tool for proliferation and motility features extraction. In *The International Conference on Computer as a Tool*, volume 2, pages 951–954. IEEE (2005). doi:10.1109/EURCON.2005.1630104.

[54] V. Olariu, D. Coca, et al. Modified variational Bayes EM estimation of hidden Markov tree model of cell lineages. *Bioinformatics*, 25(21):2824–2830 (2009).

[55] N. Harder, F. Mora-Bermudez, et al. Automatic analysis of dividing cells in live cell movies to detect mitotic delays and correlate phenotypes in time. *Genome Research*, 19(11):2113–2124 (2009).

[56] Thomas Walter, Michael Held, et al. Automatic identification and clustering of chromosome phenotypes in a genome wide rnai screen by time-lapse imaging. *Journal of Structural Biology*, 170(1):1–9 (2010).

[57] R. Bise, Zhaozheng Yin, and T. Kanade. Reliable cell tracking by global data association. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 1004 –1010 (2011). doi:10.1109/ISBI.2011.5872571.

[58] Dirk Padfield, Jens Rittscher, and Badrinath Roysam. Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. *Medical Image Analysis*, In Press, Corrected Proof:– (2010).

[59] Anne Carpenter, Thouis Jones, et al. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, 7(10):R100 (2006). doi:10.1186/gb-2006-7-10-r100.

[60] W.S Rasband. ImageJ. `http://rsb.info.nih.gov/ij/` (1997–2011).

[61] Erik Meijering, Oleh Dzyubachyk, and Ihor Smal. Methods for cell and particle tracking. In P. Michael conn (editor), *Imaging and Spectroscopic Analysis of Living Cells Optical and Spectroscopic Techniques*, volume 504 of *Methods in Enzymology*, chapter 9, pages 183 – 200. Academic Press (2012). doi:10.1016/B978-0-12-391857-4.00009-4.

[62] D. Sage, F.R. Neumann, et al. Automatic Tracking of Individual Fluorescence Particles: Application to the study of chromosome dynamics. *IEEE Transactions on Image Processing*, 14(0):1372–1383 (2005).

[63] Keith W. Vance, Dan J. Woodcock, et al. Cis-regulatory function, evolutionary robustness and divergence inside single vertebrate cells. Submitted to PlosBiol.

[64] Mark A. Gregory and Stephen R. Hann. c-Myc Proteolysis by the Ubiquitin-Proteasome Pathway: Stabilization of c-Myc in Burkitt's Lymphoma Cells. *Mol. Cell. Biol.*, 20(7):2423–2435 (2000).

[65] Robert H. Newman and Jin Zhang. Fucci: Street Lights on the Road to Mitosis. *Chem Biol*, 15(2):97–98 (2008).

[66] Marcel Mchali and Malik Lutzmann. The Cell Cycle: Now Live and in Color. *Cell*, 132(3):341–343 (2008).

[67] T. Katherine Tamai, Lucy C. Young, and David Whitmore. Light signaling to the zebrafish circadian clock by cryptochrome 1a. *Proceedings of the National Academy of Sciences*, 104(37):14712–14717 (2007).

[68] Mayu Sugiyama, Asako Sakaue-Sawano, et al. Illuminating cell-cycle progression in the developing zebrafish embryo. *Proceedings of the National Academy of Sciences*, 106(49):20812–20817 (2009). doi:10.1073/pnas.0906464106.

[69] Emi Nagoshi, Camille Saini, et al. Circadian gene expression in individual fibroblasts: Cell-autonomous and self-sustained oscillators pass time to daughter cells. *Cell*, 119(5):693–705 (2004).

[70] Shailja Singh, B.S. Dwarakanath, and T. Lazar Mathew. DNA ligand Hoechst-33342 enhances UV induced cytotoxicity in human glioma cell lines. *Journal of Photochemistry and Photobiology B: Biology*, 77(1-3):45 – 54 (2004). doi: 10.1016/j.jphotobiol.2004.08.009.

[71] C.H. Li and C.K. Lee. Minimum cross entropy thresholding. *Pattern Recognition*, 26(4):617 – 625 (1993). doi:10.1016/0031-3203(93)90115-D.

[72] Nobuyuki Otsu. A threshold selection method from gray-level histogram. *IEEE Trans. on Systems, Man, and Cybernetics*, 9:62–66 (1979). doi:10.1109/TSMC.1979.4310076.

[73] Mehmet Sezgin and Bulent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–168 (2004). doi:10.1117/1.1631315.

[74] Ping sung Liao, Tse sheng Chen, and Pau choo Chung. A fast algorithm for multilevel thresholding. *Journal Of Information Science And Engineering*, 17:713–727 (2001).

[75] Stuart P Lloyd. Least squares quantization in pcm. *IEEE Transactions On Information Theory*, IT-28(2):129–137 (1982).

[76] Wayne Niblack. *An introduction to digital image processing.* Strandberg Publishing Company, Birkeroed, Denmark, Denmark (1985). ISBN 87-872-0055-4.

[77] Jos B. T. M. Roerdink and Arnold Meijster. The watershed transform: definitions, algorithms and parallelization strategies. *Fundam. Inf.*, 41(1-2):187–228 (2000).

[78] F. Jamitzky, R. W. Stark, et al. Scaling-index method as an image processing tool in scanning-probe microscopy. *Ultramicroscopy*, 86(1-2):241–246 (2001).

[79] Christoph Räth, Wolfram Bunk, et al. Analysing large-scale structure – i. weighted scaling indices and constrained randomization. *Monthly Notice of the Royal Astronomical Society*, 337(2):413–421 (2002).

[80] Benoit B. Mandelbrot. *Fractals: Form, Chance and Dimension.* W.H.Freeman & Co Ltd. (1977).

[81] R. Adams and L. Bischof. Seeded region growing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(6):641–647 (1994). doi:10.1109/34.295913.

[82] Andrew Mehnert and Paul Jackway. An improved seeded region growing algorithm. *Pattern Recognition Letters*, 18(10):1065–1071 (1997).

[83] C. Wählby, I-M. Sintorn, et al. Combining intensity, edge and shape information for 2d and 3d segmentation of cell nuclei in tissue sections. *Journal of Microscopy*, 215(1):67–76 (2004).

[84] D. Fenistein, B. Lenseigne, et al. A fast, fully automated cell segmentation algorithm for high-throughput and high-content screening. *Cytometry Part A*, 73A(10):958–964 (2008).

[85] D. Sage, M. Unser, P. Salmon, and C. Dibner. A software solution for recording circadian oscillator features in time-lapse live cell microscopy. *Cell Division*, 5(17) (2010).

[86] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(6):583–598 (1991). doi:10.1109/34.87344.

[87] M. Tsukahara, S. Mitrović, et al. Coupled tomography and distinct-element-method approach to exploring the granular media microstructure in a jamming hourglass. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 77(6) (2008). Implementation available from `http://bigwww.epfl.ch/sage/soft/watershed/`.

[88] Khuloud Jaqaman, Dinah Loerke, et al. Robust single-particle tracking in live-cell time-lapse sequences. *Nat Meth*, 5(8):695–702 (2008).

[89] András Frank. On Kuhn's Hungarian Method – A tribute from Hungary. *Naval Research Logistics (NRL)*, 52(1):2–5 (2005). doi:10.1002/nav.20056.

[90] Hang T. Lau (Ed: Kenneth H. Rosen). *A Java library of Graph Algorithms and Optimization*. Taylor & Francis Group (2007).

[91] Roy Jonker and Ton Volgenant. Improving the Hungarian assignment algorithm. *Operations Research Letters*, 5(4):171–175 (1986).

[92] M. B. Wright. Speeding up the Hungarian algorithm. *Computers & Operations Research*, 17(1):95–96 (1990).

[93] Lit-Hsin Loo, Hai-Jui Lin, et al. An approach for extensibly profiling the molecular states of cellular subpopulations. *Nat Meth*, 6(10):759–765 (2009).

[94] 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).

[95] Asako Sakaue-Sawano, Hiroshi Kurokawa, et al. Visualizing Spatiotemporal Dynamics of Multicellular Cell-Cycle Progression. *Cell*, 132(3):487–498 (2008).

[96] Takayuki Takahashi, Charanjeet Guron, et al. A Minimal Murine Msx-1 Gene Promoter. *Journal of Biological Chemistry*, 272(36):22667–22678 (1997). doi:10.1074/jbc.272.36.22667.

[97] Vincent Castronovo, Masami Kusaka, et al. Homeobox genes: Potential candidates for the transcriptional control of the transformed and invasive phenotype. *Biochemical Pharmacology*, 47(1):137–143 (1994).

[98] M T Murtha, J F Leckman, and F H Ruddle. Detection of homeobox genes in development and evolution. *Proc Natl Acad Sci U S A.*, 88(23):10711–10715 (1991).

[99] A. J. Bendall and C. Abate-Shen. Roles for *Msx* and *Dlx* homeoproteins in vertebrate development. *Gene*, 247(1-2):17 – 31 (2000). doi:DOI:10.1016/S0378-1119(00)00081-0.

[100] Gezhi Hu, Hansol Lee, et al. Msx homeobox genes inhibit differentiation through upregulation of cyclin d1. *Development*, 128(12):2373–2384 (2001).

[101] Jinhee Park, Kyoungsook Park, Sunghoon Kim, and Je-Ho Lee. Msx1 gene overexpression induces g1 phase cell arrest in human ovarian cancer cell line ovcar3. *Biochemical and Biophysical Research Communications*, 281(5):1234 – 1240 (2001). doi:10.1006/bbrc.2001.4474.

[102] A.C. Lidral and B.C. Reising. The Role of MSX1 in Human Tooth Agenesis. *Journal of Dental Research*, 81(4):274–278 (2002).

[103] Ichiro Satokata and Richard Maas. Msx1 deficient mice exhibit cleft palate and abnormalities of craniofacial and tooth development. *Nat Genet*, 6(4):348–356 (1994).

[104] Heleni Vastardis, Nadeem Karimbux, et al. A human msx1 homeodomain missense mutation causes selective tooth agenesis. *Nat Genet*, 13(4):417–421 (1996).

[105] Anoop Kumar, Cristiana P. Velloso, Yutaka Imokawa, and Jeremy P. Brockes. The Regenerative Plasticity of Isolated Urodele Myofibers and Its Dependence on *Msx1*. *PLoS Biology*, 2(8):e218 (2004).

[106] Shannon J. Odelberg, Angela Kollhoff, and Mark T. Keating. Dedifferentiation of Mammalian Myotubes Induced by msx1. *Cell*, 103(7):1099–1109 (2000).

[107] Colin Dingwall, Stephen V. Sharnick, and Ronald A. Laskey. A polypeptide domain that specifies migration of nucleoplasmin into the nucleus. *Cell*, 30(2):449 – 458 (1982). doi:10.1016/0092-8674(82)90242-2.

[108] Daniel H. Rapoport, Tim Becker, et al. A novel validation algorithm allows for automated cell tracking and the extraction of biologically meaningful parameters. *PLoS ONE*, 6(11):e27315 (2011). doi:10.1371/journal.pone.0027315.

[109] Gyu-Un Bae, Ursula Gaio, et al. Regulation of myoblast motility and fusion by the cxcr4-associated sialomucin, cd164. *Journal of Biological Chemistry*, 283(13):8301–8309 (2008). doi:10.1074/jbc.M706730200.

[110] Daniel Campos, Vicenç Méndez, and Isaac Llopis. Persistent random motion: Uncovering cell migration dynamics. *Journal of Theoretical Biology*, 267(4):526–534 (2010). doi:10.1016/j.jtbi.2010.09.022.

[111] A. Lehmussola, P. Ruusuvuori, et al. Synthetic images of high-throughput microscopy for validation of image analysis methods. *Proceedings of the IEEE DOI - 10.1109/JPROC.2008.925490*, 96(8):1348–1360 (2008).

[112] Jesse Davis and Mark Goadrich. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 233–240. ACM, New York, NY, USA (2006). ISBN 1-59593-383-2. doi:http://doi.acm.org/10.1145/1143844.1143874.

[113] A.J. Hand, T. Sun, et al. Automated tracking of migrating cells in phase-contrast video microscopy sequences using image registration. *Journal of Microscopy*, 234(1):62–79 (2009).

[114] Jo Wixon. Featured Organism: *Schizosaccharomyces pombe*, the fission yeast. *Comparative and Functional Genomics*, 3(2):194–204 (2002). doi:10.1002/cfg.92.

[115] Valerie Wood, Midori A. Harris, et al. PomBase: a comprehensive online resource for fission yeast. *Nucleic Acids Research* (2011). doi:10.1093/nar/gkr853. `http://www.pombase.org`.

[116] Adam B. Shapiro, Adam B. Corder, and Victor Ling. P-glycoprotein-mediated hoechst 33342 transport out of the lipid bilayer. *European Journal of Biochemistry*, 250(1):115–121 (1997). doi:10.1111/j.1432-1033.1997.00115.x.

[117] I. Tatischeff, M. Bomsel, et al. Dictyostelium discoideum cells shed vesicles with associated dna and vital stain hoechst 33342. *Cellular and Molecular Life Sciences*, 54:476–487 (1998). 10.1007/s000180050176.

[118] Marie-Pierre Gulli, Jean-Philippe Girard, et al. gar2 is a nucleolar protein from *Schizosaccharomyces pombe* required for 18Sr RNA and 40S ribosomal subunit accumulation. *Nucleic Acids Research*, 23:1912–1918 (1995).

[119] Hélène Sicard, Marlène Faubladier, et al. The Role of the *Schizosaccharomyces pombe* gar2 Protein in Nucleolar Structure and Function Depends on the Concerted Action of its Highly Charged N Terminus and its RNA-binding Domains. *Molecular Biology of the Cell*, 9(8):2011–2023 (1998).

[120] Anil K. and Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651 – 666 (2010). doi:10.1016/j.patrec.2009.09.011.

[121] Jacqueline Hayles and Paul Nurse. A review of mitosis in the fission yeast *Schizosacchromyces pombe*. *Experimental Cell Research*, 184(2):273 – 286 (1989). doi:10.1016/0014-4827(89)90327-3.

[122] P.A. Fantes. Control of cell size and cycle time in *Schizosaccharomyces pombe*. *Journal of Cell Science*, 24(1):51–67 (1977).

[123] Ariel Aharon Cohen, Tomer Kalisky, et al. Protein Dynamics in Individual Human Cells: Experiment and Theory. *PLoS ONE*, 4(4):e4901– (2009).

[124] Naama Geva-Zatorsky, Nitzan Rosenfeld, et al. Oscillations and variability in the p53 system. *Molecular Systems Biology*, 2:– (2006).

[125] H. Nishitani, Z. Lygerou, T. Nishimoto, and P. Nurse. The Cdt1 protein is required to license DNA for replication in fission yeast. *Nature*, 404(6778):625–628 (2000). doi:10.1038/35007110.

[126] David Zwicker, David K. Lubensky, and Pieter Rein ten Wolde. Robust circadian clocks from coupled protein-modification and transcription-translation cycles. *Proceedings of the National Academy of Sciences*, 107(52):22540–22545 (2010). doi:10.1073/pnas.1007613107.

[127] Catherine Oikonomou and Frederick R Cross. Frequency control of cell cycle oscillators. *Current Opinion in Genetics & Development*, 20(6):605 – 612 (2010). doi:10.1016/j.gde.2010.08.006.

[128] SM Reppert and DR Weaver. Coordination of circadian timing in mammals. *Nature*, 418(6901):935–41– (2002).

[129] Lynne Lamberg. Time of Day Medicine Dose Is Taken May Boost Its Efficacy, Cut Toxicity. *JAMA: The Journal of the American Medical Association*, 275(15):1143–1144 (1996). doi:10.1001/jama.1996.03530390007002.

[130] Tim Hunt and Paolo Sassone-Corsi. Riding Tandem: Circadian Clocks and the Cell Cycle. *Cell*, 129(3):461 – 464 (2007). doi:10.1016/j.cell.2007.04.015.

[131] Peter Lenz and Lotte Søgaard-Andersen. Temporal and spatial oscillations in bacteria. *Nat Rev Micro*, 9(8):565–577 (2011).

[132] Riaan Conradie, Frank J. Bruggeman, et al. Restriction point control of the mammalian cell cycle via the cyclin e/cdk2:p27 complex. *FEBS Journal*, 277(2):357–367 (2010). doi:10.1111/j.1742-4658.2009.07473.x.

[133] Judit Zámborszky, Christian I. Hong, and Attila Csikász Nagy. Computational analysis of mammalian cell division gated by a circadian clock: Quantized cell cycles and cell size control. *Journal of Biological Rhythms*, 22(6):542–553 (2007). doi:10.1177/0748730407307225.

[134] Bin Kang, Yuan-Yuan Li, et al. Modeling the effects of cell cycle m-phase transcriptional inhibition on circadian oscillation. *PLoS Comput Biol*, 4(3):e1000019 (2008). doi:10.1371/journal.pcbi.1000019.

[135] Michele A. Gauger and Aziz Sancar. Cryptochrome, Circadian Cycle, Cell Cycle Checkpoints, and Cancer. *Cancer Research*, 65(15):6828–6834 (2005). doi:10.1158/0008-5472.CAN-05-1119.

[136] Carl Hirschie Johnson. Circadian clocks and cell division: What's the pacemaker? *Cell Cycle*, 9(1538-4101):3864–3873 (2010).

[137] M. Unser. Sampling—50 Years After Shannon. *Proceedings of the IEEE*, 88(4):569–587 (2000).

[138] Philipp Berens. CircStat: A MATLAB Toolbox for Circular Statistics. *Journal of Statistical Software*, 31(10):1–21 (2009).

[139] Takuya Matsuo, Shun Yamaguchi, et al. Control mechanism of the circadian clock for timing of cell division in vivo. *Science*, 302(5643):255–259 (2003). doi:10.1126/science.1086271.

[140] O. Konevsky and Yu. Stepanets. An algorithm for segmenting cytological images. *Pattern Recognition and Image Analysis*, 18(1):93–100 (2008).

[141] Simon Youssef, Sebastian Gude, and Joachim O. Radler. Automated tracking in live-cell time-lapse movies. *Integr. Biol.*, 3:1095–1101 (2011). doi:10.1039/C1IB00035G.

[142] Shahragim Tajbakhsh, Pierre Rocheteau, and Isabelle Le Roux. Asymmetric Cell Divisions and Asymmetric Cell Fates. *Annual Review of Cell and Developmental Biology*, 25(1):671–699 (2009). doi:10.1146/annurev.cellbio.24.110707.175415.

[143] Anna Akhmanova and Michel O. Steinmetz. Microtubule +TIPs at a glance. *J Cell Sci*, 123(Pt 20):3415–3419 (2010). doi:10.1242/jcs.062414.

[144] Kathryn T. Applegate, Sebastien Besson, et al. plusTipTracker: Quantitative image analysis software for the measurement of microtubule dynamics. *J Struct Biol*, 176(2):168–184 (2011). doi:10.1016/j.jsb.2011.07.009.

[145] Timothy Budd. *Classic Data Structures in Java*. Addison Wesley (2000). ISBN: 978-0201700022.

[146] M. Downey, K.W. Vance, and T. Bretschneider. Lineagetracker: A statistical scoring method for tracking cell lineages in large cell populations with low temporal resolution. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 1913 –1916 (2011). doi:10.1109/ISBI.2011.5872783.

[147] Mike J. Downey, Danuta M. Jeziorska, et al. Extracting fluorescent reporter time courses of cell lineages from high-throughput microscopy at low temporal resolution. *PLoS ONE*, 6(12):e27886 (2011). doi:10.1371/journal.pone.0027886.