Quantitative analysis of cell function and death in label-free high content screening



Thomas Akwesi Nketia

University of Oxford
Kellogg College

A thesis submitted for the degree of

Doctor of Philosophy

Trinity, 2016.

Acknowledgements

I owe my mum, Sabina a debt of gratitude for her dedication and personal sacrifice to support my education and personal development. I thank my wife, Gifty for all her companionship and our extended families for their continual support.

I am very grateful to my supervisors, Prof. Jens Ritscher, Prof. Alison Noble and Prof Andrew Zisserman for the guidance, support, and patience that enabled the completion of this thesis. I am also grateful to all members and staff of the Biomedical Image Analysis (BioMedIA) Lab for the discussions, suggestions, and maintenance of a conducive work setting over the years.

Dedication

This thesis is dedicated to my son, *Samuel*.

Abstract

Time-lapse data is increasingly being used to conduct more detailed high throughput studies. Quantitative information that is being derived from vast and complex image data sets is essential in our understanding of basic cellular processes. Advances in cell culturing methods combined with sophisticated molecular probes allow the monitoring of a broad array of cellular functions *in vivo*. These experiments provide a wealth of multi-channel time-lapse data that renders traditional manual interpretation infeasible. The work outlined in the thesis is aimed at quantitative analysis of such time-lapse data resulting in statistical analysis methods and software tools that offers an improvement in the efficiency and impact of phenotypic screening experiments. The research contributions towards the stated aim is outlined in three (3) main areas: cell morphology and lineage, population-based summaries for cell function, and cell state labelling.

Firstly, cell morphology and lineage tasks address the two fundamental tasks in time-lapse quantitative biological image analysis; segmentation and tracking. Cell segmentation presents a key challenge in any image-based single cell analysis as most further analysis is heavily dependent on this step. Details of a segmentation approach based on an existing light phase retardation model and results on sample data of phase contrast HeLa cervical cells is presented. Using phase retardation feature extraction to precondition images for deep learning is also explored. Cell shape and texture features from segmentation are then used in the tracking to obtain a lineage metric to assess the progression of cell morphology changes and motion over time. Details of a tracking scheme based on coupled minimum cost flow network is presented.

Secondly, metrics that quantify cell function of a population such as proliferation, viability and migration based on a summary of single cell measurements are common.

Such metrics however generally do not account for segmentation errors resulting from cell crowding and overlapping cell boundaries. Details of an approach that allows incorporating the confidence in segmentation accuracy for each single cell in the population metric is presented. Analysis on simulated data shows that the proposed method provides better summary that is representative of the cell population and hence could improve conclusions made from quantitative analysis of cell populations.

Determining the variability in the mode of death of cells is important in multiple live cell phenotypic toxicity screens. The mechanism associated with the mode a cell progresses towards death can be observed as a sequence of morphological events associated with cell death. As these events are observed morphologically, changes in shape and texture features can be used to model the time series process. Here, such temporal evolution is modelled as hierarchical dirichlet process of a hidden markov model (HDP-HMM). The model eliminates the limit on the number of states representing morphological events and hence allows new states to be discovered as more cell data from new screens are added. Details also include sequence analysis methods used to group similar cells based on the temporal changes of their morphological features.

Overall, the proposed work provides methods and software tools that enable the efficient interpretation of patterns and groups in large amounts of data obtained from high throughput phenotypic screens that otherwise would be infeasible to obtain manually. Hence, the work improves efficiency of analysis as well as help obtain objective and repeatable conclusions from live cell experiments.

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Thomas Akwesi Nketia, Kellogg College

Table of Contents

Acknowledgements i

Dedication ii

Abstract iii

Table of Contents vi

List of Figures x

1 Introduction 1

1.1 Motivation 1

1.2 Live cell imaging spectrum 3

1.3 Research objectives 4

1.4 Thesis structure and contributions 6

1.4.1 Cell detection, segmentation and boundary estimation 7

1.4.2 Cell tracking 8

1.4.3 Temporal modelling of cell sequences 8

1.4.4 Related publications/presentations 8

1.5 Overview of developed software 9

1.6 Impact 9

2 Background 11

2.1 Live cell imaging 11

2.2 Computational methods 13

2.2.1 Cell Detection and Segmentation 15

2.2.2 Single Cell Tracking 20
2.2.2.1 Learning-based cell tracking 22
2.2.2.2 Global object association tracking 24
2.2.3 Cell state labelling 25
2.2.3.1 Hidden Markov Models 26
2.2.3.2 Hierarchical Dirichlet process HMM (HDP-HMM) 28
2.3 Software tools 33
2.3.1 General Tools 33
2.3.2 Toolkits & Libraries 34
2.4 Conclusion 36

3 Cell Segmentation in Phase contrast 39
3.1 Introduction 39
3.2 Method 42
3.2.1 Phase retardation feature extraction 42
3.2.2 Generating superpixels over phase retardation features 46
3.2.3 Multiclass clustering and region aggregation 47
3.2.3.1 Neighbourhood feature augmentation 49
3.2.3.2 GMM clustering and boundary estimation 50
3.3 Experiments and Results 51
3.3.1 Parameter Selection 51
3.3.2 Evaluation 53
3.4 Preconditioning for deep learning 53
3.4.1 Feature map activations 55
3.5 Conclusion 55

4 Boundary Estimation in Overlapping Cells 59
4.1 Introduction 59
4.2 Method 62
4.2.1 Datasets 63

4.2.1.1 Experimental datasets 64
4.2.1.2 Reference datasets 64
4.2.2 Segmentation and boundary estimation 65
4.2.3 Feature density estimation and weighting 66
4.2.4 Segmentation confidence and weight assignment 68
4.2.5 Assessing the impact of weighting 68
4.3 Results 69
4.4 Conclusion 71

5 CMCFtrack: Single cell tracking 73
5.1 Introduction 73
5.2 Coupled minimum-cost flow tracking 74
5.2.1 The flow network 75
5.2.2 Extending the graph for modelling cell behaviours 78
5.2.2.1 Appearing and disappearing cells 78
5.2.2.2 Splitting and merging cells 79
5.3 Software implementation 81
5.3.1 Data structures 82
5.3.1.1 Vertices 82
5.3.1.2 Edges and costs 82
5.3.2 Solving the coupled minimum-cost flow problem 84
5.3.2.1 Pruning the network 84
5.3.2.2 Edge coupling 86
5.3.2.3 Optimisation 86
5.3.3 Output and visualisation 88
5.3.4 Installation and usage 90
5.4 Conclusion 90

6 Cell State Labelling 93
6.1 Introduction 93

6.2 Transforming time lapse to state sequences . . . 95
6.2.1 Segmentation and tracking . . . 95
6.2.2 Feature extraction and formulation . . . 96
6.2.3 State transition modelling . . . 97
6.3 Analysing state sequences . . . 98
6.3.1 State sequence characteristics . . . 99
6.3.1.1 Distribution of states . . . 99
6.3.1.2 Mean duration of states . . . 102
6.3.1.3 Entropy of states . . . 103
6.3.2 State transitions . . . 104
6.4 Clustering sequences . . . 105
6.4.1 Clustering state sequences . . . 106
6.5 Extension to cell population dynamics . . . 106
6.5.1 Population state labelling . . . 107
6.5.2 Population state sequence analysis . . . 109
6.5.3 Clustering population state sequences . . . 109
6.6 Conclusion . . . 109

7 Summary and future work . . . 113
7.1 Cell segmentation . . . 113
7.2 Cell boundary estimation . . . 114
7.3 Cell tracking . . . 115
7.4 Cell sequence labelling . . . 115
7.5 Future work . . . 116

Bibliography . . . 117

List of Figures

1.1 Live cell imaging analysis spectrum 5

2.1 Schematic of a controlled environment microscope incubator. *Image source: Nikon MicroscopyU* 12

2.2 Sample images of cells in phase contrast and fluorescence. (a) Phase contrast image of HeLa cells showing no labelling and lower contrast between cell regions and background. Image also shows some imaging artefacts such halos (b) Fluorescence image of HeLa cells showing multiple staining and a higher contrast between cell regions and background . . . 14

2.3 (a) Tree structure of candidate nested MSERs. (b) Segmentation result showing delineated cell boundaries and training dot annotations 17

2.4 Simplified setup of bright field and phase contrast microscopes 19

2.5 (a) Sample phase contrast image. (b) Three-component reconstructed image using phase retardation model 21

2.6 Schematic showing tracking framework used by Lou and Hamprecht [41] showing the assignment representation in a pair of adjacent frames 24

2.7 Overview of tracklet association method used by Bise *et al* [48] 25

2.8 Graphical model representation of (a) DP mixture model and (b) hierarchical DP mixture model [69]. 30

2.9 Graphical model representation of the HDP-HMM [69]. 31

2.10 (a) Multinomial observation sequence; (b) true state sequence; (c)-(d) estimated state sequence after 30,000 Gibbs iterations for the original and sticky HDP-HMM, respectively, with errors indicated in red. Without an extra self-transition bias, the HDP-HMM rapidly transitions among redundant states 32

3.1 The general overview of the proposed method for phase contrast cell segmentation. For every image, phase retardation features are computed for every pixel by image reconstruction based on a dictionary of diffraction patterns. The pixel-wise features are ranked according to the most discriminative by a graph cut score. Pixels are then aggregated into SLIC superpixels based on their features to obtain an oversegmentation. The superpixels are then grouped into regions and the regions that correspond to cells are extracted and merged boundaries are split by a watershed algorithm 42

3.2 Diffraction pattern kernel of phase contrast imaging, airy(r), obscured Airy pattern with (a) 3D surface view and (b) 2D top-view. Image from [15] . 44

3.3 $DPK(m)$, the diffraction patterns with different phase retardations, θ_m. Image from [15] 45

3.4 Initial processes resulting in cluster regions: (a) Original phase contrast image; (b) The top three(3) principal features for each pixel mapped onto RGB of a colour image 46

3.5 Superpixel oversegmentation of phase retardation features showing effect of compactness parameter, c: (a) number of segments, $U = 2500$, compactness, $c = 10$ (b) zoomed section showing closer view of boundaries when $c = 10$; (c) number of segments, $U = 2500$, compactness, $c = 100$; (d) zoomed section showing closer view of boundaries when $c = 10$ 48

3.6 Clustering of superfeatures 50

3.7 Final processing leading to final segmentation output: (a) *a priori* knowledge of cell location as dots (b) Cell region mask; (c) Euclidean distance transform map; (d) Segmentation contours overlaid on image . . 52

3.8 VGGNet16 network architecture. Image by Zhicheng Yan et al.[107] . . . 54

4.1 Sample binary image with two cells touching and occluding sections of their boundaries in (a) alongside transform operations A-D leading to the estimated boundary in (b). *Image courtesy: MathWorks* 61

4.2 Sample simulated cell images showing (a) clustered and overlapping cells in a conventional mono-culture and (b) a co-culture of two cell types where the larger cell type shows clustering and overlapping whereas smaller cell type are evenly dispersed. 62

4.3 Separating overlapped cells in (a) using Voronoi diagram over seed nuclei detections in (b) to obtain the estimated boundaries shown in (c). 65

4.4 Kernel density estimates for area shown in (b) and (d) of the reference datasets for mono-culture cells in (a) and co-culture cells in (c) respectively. 67

4.5 Confidence weights 69

4.6 Sample distributions of area metric showing components obtained from the least confident and most confident segmentations as well as the mixture distributions 70

4.7 Results showing the KL divergence from the true distribution of the equally weighted and confidence weighted for different fraction of overlapping cells. (a),(c), (e) and (b),(d),(f) shows the divergence values for mono-cultured and co-culture scenarios respectively. It is observed that in both cases the distance measured by KL of the unweighted from the true distribution is higher compared to the weighted distribution 72

5.1 Simple bipartite graph. The black vertices on the left represent the cells on the previous frame, and the black vertices on the right represent the cells on the current frame. The left vertices are fully connected to the right vertices. Image courtesy: [16] 76

5.2 The flow network showing the the source ($T+$) and target ($T-$) vertices added to convert the bipartite graph to a minimum-cost flow problem. Image courtesy: [16] 77

5.3 Flow network with additional vertices A and D for appearing and disappearing cells. Image courtesy: [16] 79

5.4 Flow network with additional vertices M and S split/merge cells. 80

5.5 Structure of vertex `dictionary` showing `key` and `value` pairs with associated data types. 83

5.6 Flow network and corresponding incidence matrix. Image courtesy: [16] . 85

5.7 The coupling process and the corresponding coupled matrix 87

5.8 Data structure of cell tracks output. 89

5.9 The web-based visualisation tool is shown running from an Amazon S3 server in browser. The layout allows different aspects of the user interface to be separated into panels to improve usability. **(a)** shows a fully functional initial implementation. The tool can be accessed at `https://s3.amazonaws.com/cmcf-tracker/visualiser/index.html`. **(b)** shows the current new interface. 91

6.1 Frames from a single nucleus sequence showing the changes in morphology and texture over time towards death in 375 frames. From left to right, Frame 9, 18, 72, 216 95

6.2 Frames from a single nucleus sequence showing the changes in intensity of applied labels. From left to right, Frame 1, 3, 10, 18 96

6.3 Sample segmentation using simple adaptive thresholding of nuclei in selected frame 97

6.4 State sequences obtained after fitting sticky HDP-HMM. Each colour block represents a state. 98
6.5 Manual multi-parameter analysis of the forms of cell death in pancreatic β-cells. As adopted from [18]. 100
6.6 State sequences of 100 cell sequences. Each number/color block represents a state 101
6.7 Distribution of states at each frame in the sequence. 102
6.8 Relative duration of states. 103
6.9 Entropy of states at each frame. 104
6.10 Top 5 most frequent event transitions with transition state pairs are denoted as $s_i > s_{i+1}$ 105
6.11 State distribution within each cluster 106
6.12 Representative sequence(s) of each of 6 clusters 107
6.13 Sample frames from a 193-frame video sequence 108
6.14 State sequence of 30 videos 108
6.15 State sequence characteristics 110
6.16 Clusters in population state sequences 111

Chapter 1

Introduction

1.1 Motivation

High throughput screening is a drug-discovery process widely used in the pharmaceutical industry to quickly assay the biological or biochemical activity of a large number of drug-like compounds through automation of the assay process [1, 2, 3]. High throughput screening of cells has become important in pharmaceuticals due to the increasing number of candidate active compounds and the corresponding wide variety of cell lines and parameters that need to be investigated [3, 4]. Phenotypic screening and target-based screening constitute the two main approaches to drug discovery.

Target-based screening involves modification of drug targets, typically proteins that appear to have a role in the pathogenesis of a particular disease and therefore requires knowledge of the molecular mechanisms of the disease [5][6]. Techniques for studying such protein analysis by mass spectrometry, or expression and transcription profiles do not provide adequate temporal or spatial resolution nor directly show whether the identified molecules have a functional role in the cellular process (i.e. disease pathogenesis) under investigation [7][5].

Phenotypic screening which involves assessing the observable characteristics such as morphology however could be performed with limited knowledge of the molecular mechanism of the cellular process and hence a specific molecular mechanism of action

hypothesis is not required [5]. Also, high-throughput screens that use fixed-cell or endpoint assays do not provide any temporal information and results risk being misinterpreted if the final state of cells is affected by an event upstream in the sample preparation process or the cell development cycle [8]. Live cell imaging involves capturing images of live cells in a culture medium at given time points to generate a time-lapse image sequence or video. Such imaging and the corresponding quantitative analysis overcomes the challenges of the lack of spatial and temporal resolution associated with endpoint assays and provides more detailed phenotypic information over time.

A common and key goal in drug discovery is the identification of new compounds that can be engineered into safe and effective drugs. Large-scale *in vitro* experiments play a critical role in translating new biological discoveries into tangible patient benefits. Experimental studies in industrial and academic research labs make increasing use of robotic high-content imaging systems. Here, thousands of experiments can be conducted in a massively parallel fashion to explore the various interactions between molecular targets (i.e. proteins) under various different environmental conditions.

The relevance of these *in vitro* studies hinges on the predictive value of cell based model systems. Mechanism-based testing methods are necessary to characterize the cellular response to novel chemical tool compounds and contribute to our fundamental understanding of drug induced toxicity. Advancing technologies that provide better predictive data at an earlier stage of the drug development process are needed to avoid costly failures of drug development programmes.

Quantitative information that is being derived from vast and complex image data sets is essential in this process. Time-lapse data is increasingly being used to conduct more detailed mechanistic studies. Advances in cell culturing methods combined with sophisticated molecular probes allow the monitoring of a broad array of cellular functions in models that replicate the biochemical characteristics of their *in vivo* counterparts. These experiments provide a wealth of multi-channel time-lapse data that prevents any traditional manual interpretation. Hence, there is a growing need to develop quantitative analysis methods and software tools that enable efficient exploration and inference from

the large volumes of data generated.

1.2 Live cell imaging spectrum

The nature of image data obtained from live cell experiments varies widely depending on the cell line, fluorescent markers used, and the specific goal of the experiment. Some experiments seek insights into cell state and function at the cellular scale while others measure interactions at the sub-cellular level. Single cell experiments tend to be phenotypic, focusing on changes in cell morphology and structure as well as overall population dynamics. This thesis focuses on computational analysis of image data from phenotypic screening at the single cell level.

From a computational image analysis perspective, cell level phenotypic data present in three main forms. For purposes of discussions here, these are labelled **Type A**, **Type B** and **Type C** as shown in Figure 1.1. Each dataset requires a different set of computational tools to extract the right biological insight though the general toolkit is mostly shared across the types. To analyse dynamics of individual cells from timelapse image sequences or videos requires three main complementary elements in the computational toolkit: segmentation, tracking and temporal modelling. Segmentation involves detecting and delineating the boundaries each cell from the image background. Tracking involves building a time trail of each segmentated cell through all frames in a timelapse sequence. Temporal modelling can take a variety of forms, but the key here is to obtain a computational model that describes the temporal dynamics of the morphological phases exhibited by cells in a given study.

In Type A datasets, there usually exist good contrast between cells and background. Cells tend to be sparsely distributed in the image with no overlaps or occlusion of neighbourhood cells as well as good temporal resolution. Such datasets are usually obtained from fluorescent-labelled experiments where uptake of the stain/label is very good within the entire cell body. Segmentation and tracking in such datasets can be achieved with good performance with current conventional methods. As single cell

trajectories can be extracted efficiently, fitting temporal models to single cells is also feasible. Cell cycle phase labelling usually falls within this category.

In other dataset forms as in Type B, there is also a good cell-background contrast barrier. However cell density in frames tend to be higher with significant overlapping and occlusion of individual cells. Accurate single cell segmentation in such images is usually more involved requiring more sophisticated methods to separate overlapping cells. Tracking in the such datasets also more challenging. In some cases, the goal is to understand overall expression of a given marker of the population of cells rather than single cell behaviour. Population level analysis tends to provide sufficient insight.

Other timelapse live cell datasets do not possess similar properties as in A and B as evidenced in Type C. Segmenting and tracking single cell morphology is infeasible considering individual single cell boundaries can not be clearly identified and biological insights at single cell level is not clear or even required. However, computational analysis of whole video frames to measure cell mass dynamics can be performed. Such analysis is however beyond the scope of this thesis. The focus here is on temporal cell morphology analysis at the single cell level or population level in datasets similar to those shown in Type A and B datasets.

1.3 Research objectives

The overall aim of the work outlined in this thesis is to build a toolkit of methods and pipelines that enable the efficient computational analysis of timelapse image data obtained from high throughput phenotypic screening for drug discovery.

This involves developing algorithms and corresponding software that extract information relating to cellular function and mechanisms of death from time lapse images of the different screening assays that will is conducted by biologists. The overall aim is categorised into specific research objectives (RO) and tasks as outlined below:

RO-1: Cell morphology and lineage - This involves implementing segmentation and tracking algorithms to obtain morphology descriptors for each cell and their temporal

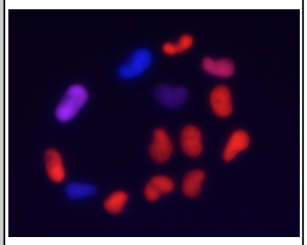
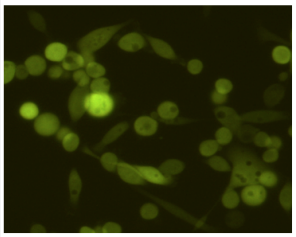
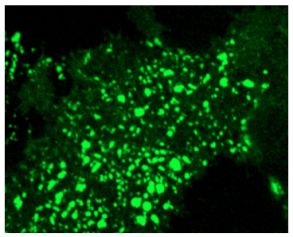
	Type A	Type B	Type C
Data			
Preprocessing Methods	Segmentation, Tracking	Segmentation	-
Analytics input	Single cell lineages	Time sequence of foreground regions	Whole frames of videos
Insight/Results	Single cell dynamics	Population dynamics	Cell mass dynamics

Figure 1.1: Live cell imaging analysis spectrum

trajectories for the entire image sequence of any given dataset. To achieve these objectives, a number of core tasks need to be completed:

Task-1: Detection and segmentation - The task involves implementing algorithms for delineating cell contours as foreground structures out of the background. The outcome of this task will be a separation of cell regions (cytoplasm and nucleus) from background and image artefacts.

Task-2: Single cell morphology - The task is towards investigating various metrics to quantitatively characterize the shape of the segmented regions of each cell and nucleus. This provides an objective numerical descriptor of the various shape transformations of a cell and its nucleus over the course of a live imaging experiment. Choi *et al* [9] has shown that shape metrics such as solidity, circularity, eccentricity, normalized perimeter and length of principal axes of nuclei could be used to distinguish diseased cells from normal cells.

Task-3: Cell tracking - This involves implementing appropriate object tracking algorithms to generate a lineage of the various features of the segmented cell regions and associated morphology descriptors through time as captured by the the frames of a live imaging time lapse. By this, a dynamic summary of the changes in individual cell and nuclear morphology could be obtained for further population-based analysis.

RO-2: Quantifying cell population characteristics - With segmentation, shape metrics and individual cell lineage obtained, individual cell features need to be aggregated for specific cell populations to determine a population metric that describes the group of cells in any given dataset. This involves developing a framework to aggregate features from different groups of cells into one clear distribution that accurately describes the population of cells.

RO-3: Cell state labelling - Building temporal models that describe cell behaviour pattern through computed morphological and structural features over time is key. For instance, initiation and progression of cell death - (apoptosis) could be observed through various appearance and morphological events including membrane blebbing, binucleation, nuclear fragmentation and cell membrane rupture [10] [11]. By analysing patterns based on the computed shape and appearance metrics along the lineage of cells, the particular nature of cell death induced by a candidate compound can be inferred from new sequences, given ground truth sequences. Having a quantitative model of various events that characterise cell death, the specific or dominant modes induced by a candidate compound in a toxicity screen could be obtained and quantified in a cell population. This provides a phenotypic and quantitative insight into how such compounds interfere in the normal life cycle of the treated cells.

1.4 Thesis structure and contributions

Time lapse image data obtained from live cell experiments vary widely depending on the cell line, microscope used, and the specific needs and research questions the experiment

seeks to address. Results from experiments are usually analysed at either the single cell level or cell population level. In this respect, quantitative analysis of static and dynamic cell state and function in this thesis follows two main themes; single cell progression and population dynamics. Chapter 2 of this thesis provides the background and review of the literature of related work in areas of phenotypic live cell screening and the key computational tools required for analysis of the image and video data generated.

Most live cell screening experiments seeking insights at the single cell level usually involve culturing and imaging multiple cells. Such experiments include cell cycle and cell death assays. Hence, the resulting time lapse or video shows multiple objects corresponding to cells and some artefacts. Quantitative assessment of single cells over time requires cell regions to be accurately segmented from the background, and artefacts excluded. Segmented regions corresponding to cells then need to be tracked over time to obtain a lineage of the cell trajectory. Each cell lineage is made up of the computed feature of the cell at all given time points. Computational analysis that shows the cell state and progression in time can then performed. Such computation also allows cell trajectories to be compared in an efficient way. The contributions of this thesis aims at addressing the computational needs described in three main themes: **(i)** cell detection, segmentation and boundary estimation; **(ii)** single cell tracking; and **(iii)** temporal modelling of cell sequences.

1.4.1 Cell detection, segmentation and boundary estimation

Chapter 3 details cell segmentation approach in phase contrast microscopy using hand-crafted features inspired by modelling the light diffraction patterns in the optical image formation process in phase contrast microscopy. The work shows how physically-inspired featured alongside partial weak annotation of cells achieves good performance in segmenting cells in phase contrast microscopy.

In many segmentation tasks, cells in an image tend to form clusters or overlap and occlude sections of individual boundaries. Methods to separate overlapping boundaries but quantification of this process in downstream analysis is rare. Chapter 4 provides a

framework for incorporating the different levels of confidence expressed in segmentation accuracy affected by approximating overlapping and occluded boundaries.

1.4.2 Cell tracking

Dynamic temporal modelling of individual cells requires individual cell trajectories across all frames of the sequence. Chapter 5 details the Python implementation of a network cost flow tracking algorithm with a web-based visualisation interface that delivers efficient cell tracking to enable temporal sequence modelling. The implementation was completed in a collaborative work with Alan Aberdeen, a fourth year engineering undergraduate who contributed to this work as part of his final undergraduate project.

1.4.3 Temporal modelling of cell sequences

Live cells show different morphological states for a variety of life processes. In cell cycle, this is demonstrated in mitotic phases characterised by changes in morphology of the overall cell structure or sub-cellular structures. Chapter 6 explores frameworks for constructing models that computationally describe the states and the order of states associated with dynamic cell behaviour using non-parametric Bayesian clustering techniques. The chapter explores approaches to cluster state sequences to cluster dynamic patterns in individual cells. Beyond individual cell sequence analysis and clustering, the work presents approaches to group cell populations aimed at comparing the effects of different drug conditions on dynamic cell population behaviour.

1.4.4 Related publications/presentations

This section outlines the publications that have featured work that is presented in this thesis. Initial work using the implementation of the phase retardations for cell segmentation extended in Chapter 3 is published in the Proceedings of the Medical Image Understanding and Analysis (MIUA) 2014 conference following a poster presentation[12].

An earlier version of the boundary estimation work in Chapter 4 is published in IEEE Xplore under proceedings of the IEEE International Symposium for Biological Imaging

(ISBI) 2015 [13] following an oral presentation. Preliminary results of this work also featured as an abstract publication and oral presentation at the Quantitative Bio Imaging (QBI) 2015 conference. The work in cell state labelling work presented in Chapter 6 is currently in preparation for publication in a journal.

1.5 Overview of developed software

To obtain the results presented in this thesis, a variety of open software tools were used out-of-the-box or as an enabling tool to develop other software. The opensource Python programming language and its associated core packages including `Numpy`, `Scipy`, `Matplotlib`, `Pandas`, `Scikit-learn` and `Scikit-image` provided the base platform on which other bespoke modules for preprocessing and visualising image data was developed.

Aside snippets and modules developed, two main software implementations of published models and algorithms were completed: **(**i) the Python open source implementation of the phase retardation feature extraction method proposed by Yin and colleagues [14, 15] and featured in Chapter 3; and **(ii)** the implementation of the coupled minimum cost flow network tracking algorithm proposed by Padfied *et al.* [16]. Details of the implementation and usage is provided in Chapter 5

1.6 Impact

The work outlined in this thesis provides a basis for extensive application of quantitative methods to the analysis of biological image data obtained from large scale phenotypic screens. The work shows that such computational methods can discover known and new patterns in cell dynamic behaviour.

Chapter 2

Background

2.1 Live cell imaging

Live cell imaging involves capturing a sequence of images in time of a cell as it undergoes biochemical life processes in a live state. As opposed to fixed end point imaging where cells are fixed on slides and imaged, live cell imaging requires the cell to be kept alive and nourished while being imaged. In many modern setups, microscopes are enclosed in cell incubators to keep cells alive during observation to ensure a favourable controlled environment. Figure 2.1 shows a schematic of a microscope enclosed in an incubation chamber.

The protocol and equipment required for efficient and accurate imaging of live cells is generally more complex than fixed cells and endpoint assays. This complexity is inherent in the number and variety of condition parameters that will have to be considered and controlled effectively to obtain the desired images while keeping cells in a healthy state and functioning normally while imaging. This involves maintaining a delicate balance of optimum ranges for temperature, pH, oxygenation, humidity and osmolarity within the controlled imaging environment. The media buffer in which the cells grow also needs to chosen carefully. Also determining the optimum sensitive detector (i.e camera), how often to image the specimen and limiting photodamage by shutting off illumination when not in active imaging are very challenging procedures to conduct effectively [8].

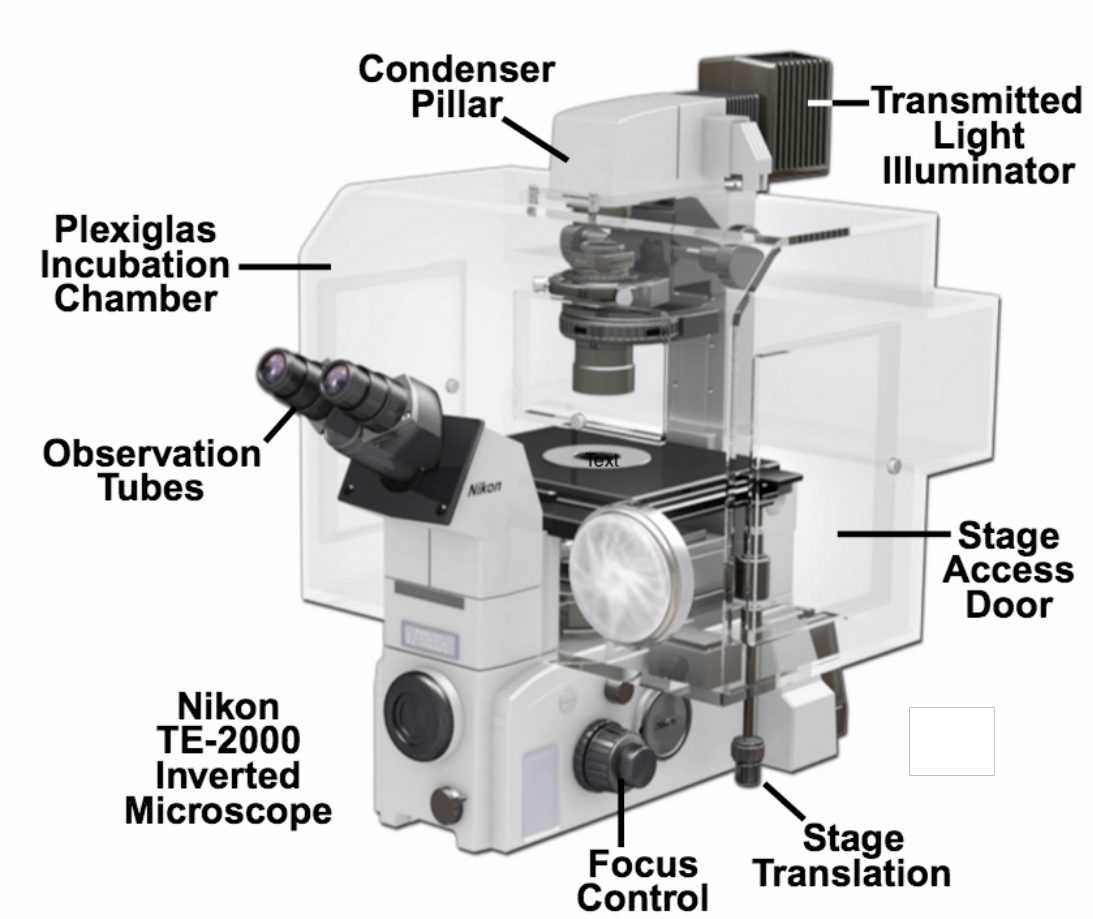


Figure 2.1: Schematic of a controlled environment microscope incubator. *Image source: Nikon MicroscopyU*

However, microscopes and biological sample preparation methods are improving with new specialised microscope systems being developed such as the Incucyte® Live Cell Analysis System by Essen Bioscience. Such systems and methods are fuelling a rising interest in the live cell imaging for pharmaceutical high-content screens.

The actual imaging of live cells is a complementary combination of fluorescence, which requires labelling with fluorescent-labelled proteins (such as GFP), and transmitted-light (brightfield) techniques which are label-free. Fluorescence methods provide information on specific biochemical processes whereas brightfield approaches give insight into morphological properties such as cell shape, position, and motility that are vital in the study of the mechanisms of cell division (*mitosis*) and cell death (*apoptosis*) [8].

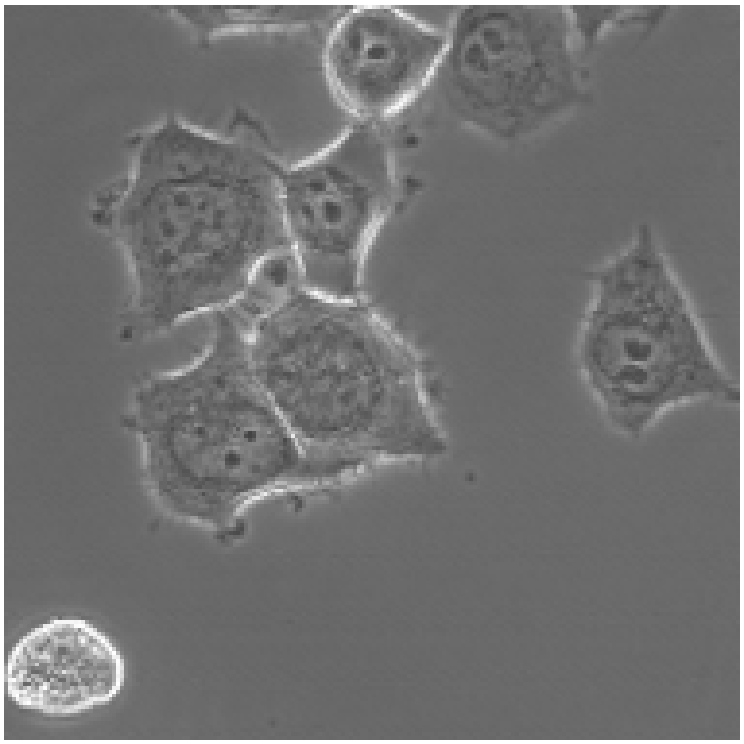
Live cell imaging is instrumental in revealing temporal dynamics in the study of cells

that enables insights that otherwise could not be obtained from fixed point experiments. Using live cell imaging, Chan *et al.*[17] were able to identify distinct colony types among human fibroblast cells undergoing reprogramming. Such cells resemble each other morphologically but differ in molecular phenotype and differentiation ability over time. From the study, Chan *et al.* could distinguish true induced-pluripotent stem (iPS) cells from partially reprogrammed cells via live cell imaging. Similarly, Yang *et al.* have used live cell studies to show the different modes of cell death in pancreatic β-cell death[18] as well as reveal the role of Na^+ channels in cytokine-induced β-cell death[19].

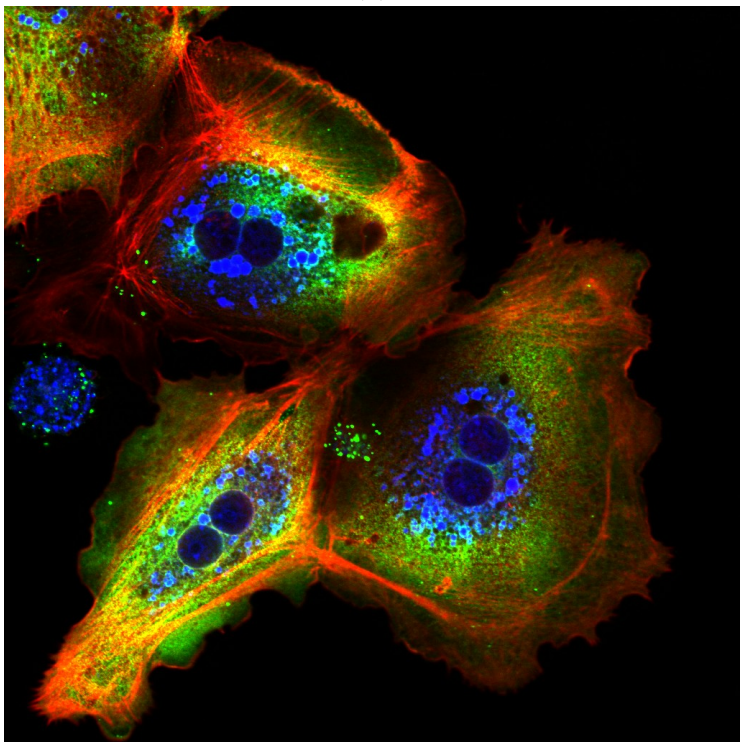
To capture such information in fluorescence usually requires multiple labels or stains for the cytoplasm, nucleus and sometimes specific organelles of the cell. A variety of fluorophores that bind to proteins in cells has been used in labelling cells for a spectrum of applications and cell components. Other agents, such as quantum dots have also been shown to be effective [20]. Label-free methods also allow cells to be studied in their natural state without the need for additional chemical or biological components which are often expensive and could also be cytotoxic [21]. The most widely used brightfield modalities are phase contrast and differential interference contrast (DIC), with the former being more common. Figure 2.2 shows an example of phase contrast and fluorescence images.

2.2 Computational methods

Live cell screening generates large volumes of digital image data at a rate that far exceeds conventional fixed point alternatives. A typical high resolution live cell screen could involve imaging cells in about 40-50 different field views every 10 minutes for 2-3 days, resulting in at least approximately 11,520 images to be analysed [11]. Each image will usually have multiple cells that either progress similarly or differently in time. To understand the dynamic behaviour of single cells and determine group patterns and outliers, an expert would have to observe and document this large number of frames carefully.



(a)



(b)

Figure 2.2: Sample images of cells in phase contrast and fluorescence. (a) Phase contrast image of HeLa cells showing no labelling and lower contrast between cell regions and background. Image also shows some imaging artefacts such halos (b) Fluorescence image of HeLa cells showing multiple staining and a higher contrast between cell regions and background

The volumes of data that need to be analysed in such screens that are often run in parallel, renders efficient manual analysis infeasible. There is therefore the need for automatic, computerized image processing methods to assist biologists to assign an unbiased quantitative measure to cell state at a given point in time or behaviour over a period of time. Image processing methods, including machine learning and computer vision routines provide standardized parameters that could be subjected to rigorous statistical evaluation and improve further modelling of the morphological features obtained [7]. Such image processing methods have been applied in a variety of single images [22, 23, 24, 15] yet application in high-throughput screens to extract patterns is rare. Such computational methods could help in preliminary assessments and summary of data that could shorten the timespan of experiments and help minimise early failure rates in drug discovery.

Computational methods and associated software tools required to analyse the progression of single cells as observed in frames of videos obtained from live cell imaging experiments can be categorised into three main task groups: segmentation, tracking and state labelling. Cell segmentation methods provide approaches to detect and delineate the boundaries of individual cells from the image background in each frame. With cell boundaries determined across frames, tracking methods provide a trajectory of each cell across all frames. Having obtained the trajectory of each cell, state labelling models partition trajectories into temporal states corresponding to homogenous morphological regions in time. The order of such morphological states is analysed to discover patterns in cell behaviour and function.

2.2.1 Cell Detection and Segmentation

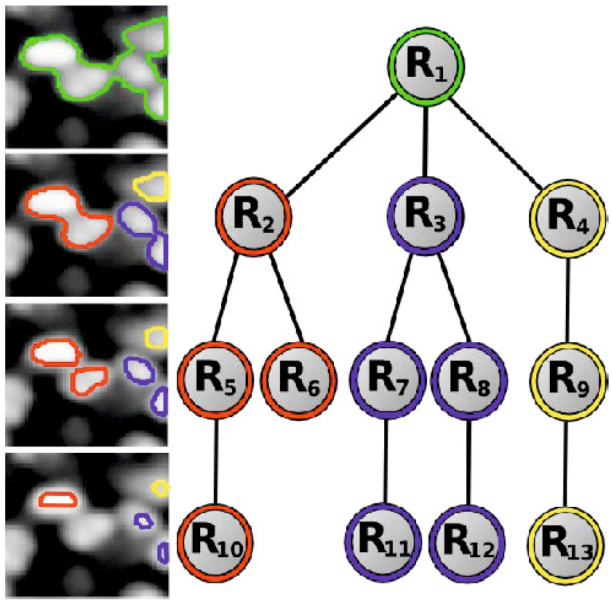
One fundamental task in quantitative biological imaging is cell segmentation. Segmentation delineates the boundaries of each cell region from the background. Hence, to perform any analysis on the location, shape or texture of individual cells, segmentation must be obtained. Consequently, accuracy of further analysis largely depends on the quality of segmentation obtained [25]. This makes segmentation a very important basic

step in any accurate quantitative analysis of cell images. While a number of sophisticated and robust segmentation techniques have been developed for fluorescent microscopy [26][27], cell segmentation in label-free modalities such as phase contrast and DIC remains challenging.

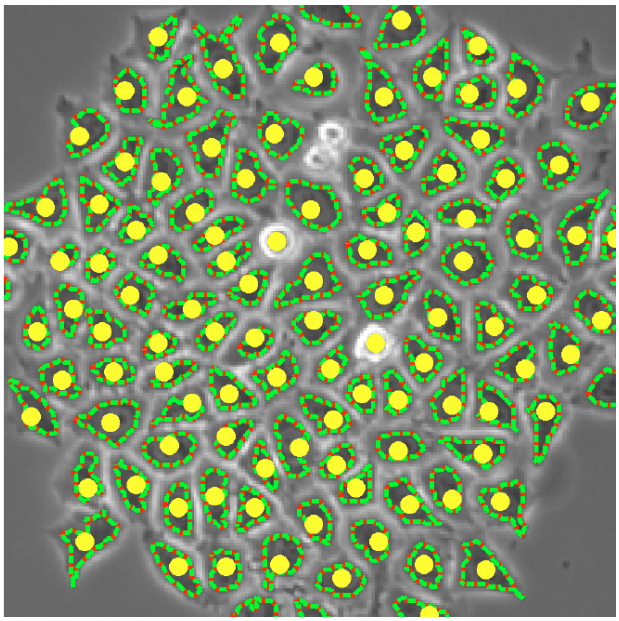
The challenge in obtaining accurate and efficient cell segmentation in label-free images can be attributed to a variety of factors. Key among these include the low contrast between cells and background, presence of image artefacts, dissimilarity among sub-cellular structures within and across cells as well as occlusion. With poor contrast variability of sub-cellular structures, the intensity profile of cell regions compared to background does not show a clear separation, resulting in poor performance of intensity-based thresholding methods. Image artefacts are either introduced by the specific optics of the imaging hardware setup or non cell particles in the culture media . Artefacts are usually observed as halo around cell boundaries and shade off effects in the image [15]. The lack of clear boundaries among densely cluttered cells also poses additional challenges in separating individual cells.

Various methods have been proposed to achieve good segmentation in phase contrast microscopy [24][15][28][23]. In most cases, the distinguishing factor across the various proposed methods is the feature extraction method applied. Ali *et al* [24] combines local phase analysis and level set techniques to automatically extract cells from background. In this approach, a cell is first detected by Otsu thresholding [29] a strongly defocused version of the original image. The monogenic signal framework [30] is then used to generate local phase and local orientation images that are used as initialisation to guide the evolution of a level set-based active contour to trace the cell boundary. Such level set-based methods, including that in [16] and [31], usually tend to be sensitive to initialisations and hence heavily dependent on the accuracy of the initialisation method used.

Methods using machine learning approaches in addition to image analysis routines have also shown success in some applications. Wang and colleagues [28] extract texture and edge features that represent specific appearance, structural and edge profiles observed in phase contrast data. They then apply AdaBoost and support vector machines



(a)



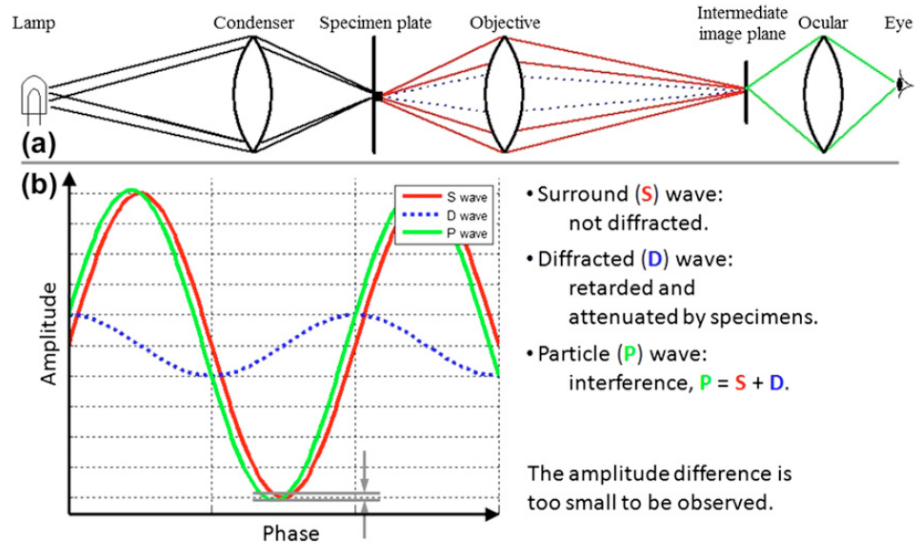
(b)

Figure 2.3: (a) Tree structure of candidate nested MSERs. (b) Segmentation result showing delineated cell boundaries and training dot annotations

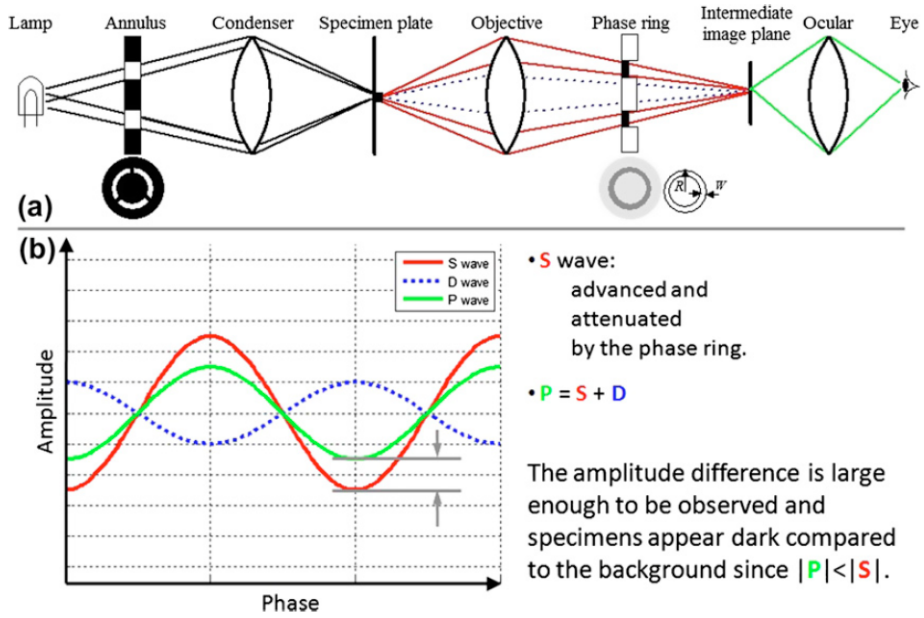
(SVM) to *learn* these profiles from annotated samples. Cell regions are then predicted in test images using the learned model. Arteta *et al* [23] follows a similar machine learning approach with a three-stage framework. First, a set of nested overlapping candidate cell regions are detected using maximally stable extremal regions (MSER) [32] feature extraction as shown in Figure 2.3a. This is followed by scoring the candidate regions using a structured SVM that is trained using simple dot annotations on cell positions. Finally, dynamic programming is used to select the optimal set of non-overlapping extremal regions from the tree of nested regions.More recently, convolutional neural networks(CNN) have been applied to cell detection/counting [33] as well as boundary segmentation[34] with state-of-the-art performance. Though such learning-based approaches report good performance, they require annotated training data that is often expensive to obtain.

A recently developed image restoration method [15][14] seeks to extract features which are intrinsic to phase contrast microscopy. The method exploits the optics of the image formation process to build a model that delivers features that can separate cell regions and boundaries from the background in an image. Phase contrast microscopy is essentially a contrast-enhancing technique that employs an optical mechanism to translate small variations in phase (diffraction pattern) of the transmitted light into corresponding changes in amplitude. The difference in amplitude can then be visualized as differences in image intensity.

In a conventional brightfield microscope as shown in the setup in Figure 2.4a, light from an illuminated source is focused and transmitted through the specimen plate. The primary source, P when transmitted through the specimen plate is divided into two components; S, the light that goes through areas surrounding the cell and D, the light that goes through the denser cell region is attenuated in amplitude and its phase retarded. The difference in amplitude between S and P translates to the observed difference in intensity between cell regions and background. This amplitude difference is generally small hence lower contrast. Phase contrast imaging introduces additional hardware, the condenser annulus and phase ring as shown in Figure 2.4b. The new setup results



(a) Bright field microscope optics setup



(b) Phase contrast microscope optics setup

Figure 2.4: Simplified setup of bright field and phase contrast microscopes

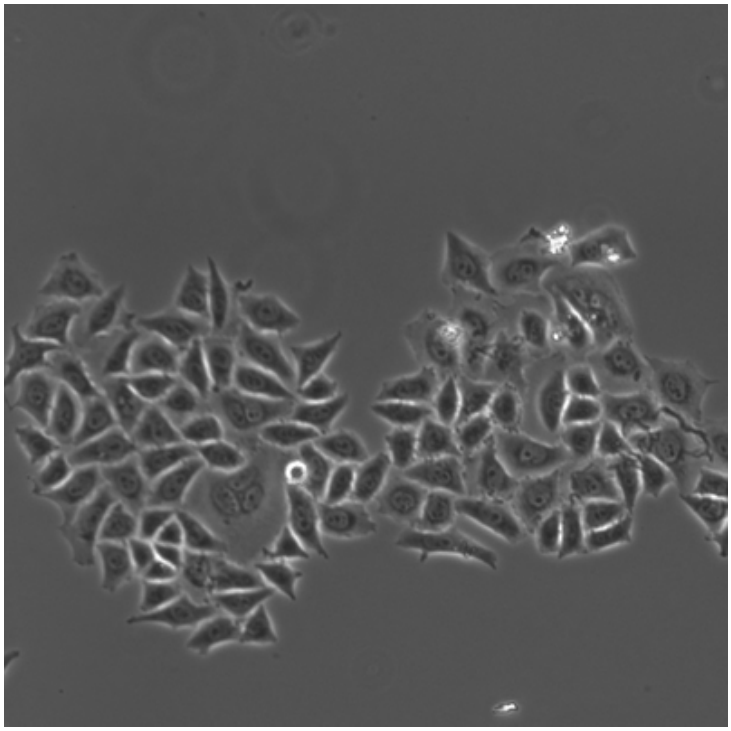
in advancing the phase and attenuation of the S wave hence increasing the amplitude difference and consequently the image contrast.

Su *et al.*[15, 14] proposed an image reconstruction model, similar to wavelet decomposition of images [35], that exploits the optics of the image formation process in phase contrast microscopy. The reconstruction model is used to extract dictionary-based features based on the diffraction pattern and phase retardation at each pixel in a grayscale phase contrast image. Figure 2.5 shows a three-component feature decomposition of a sample phase contrast image. The model produces vectorized features that distinguish the cell regions from the background. Details of the application of this feature extraction method in cell segmentation is provided in Chapter 3.

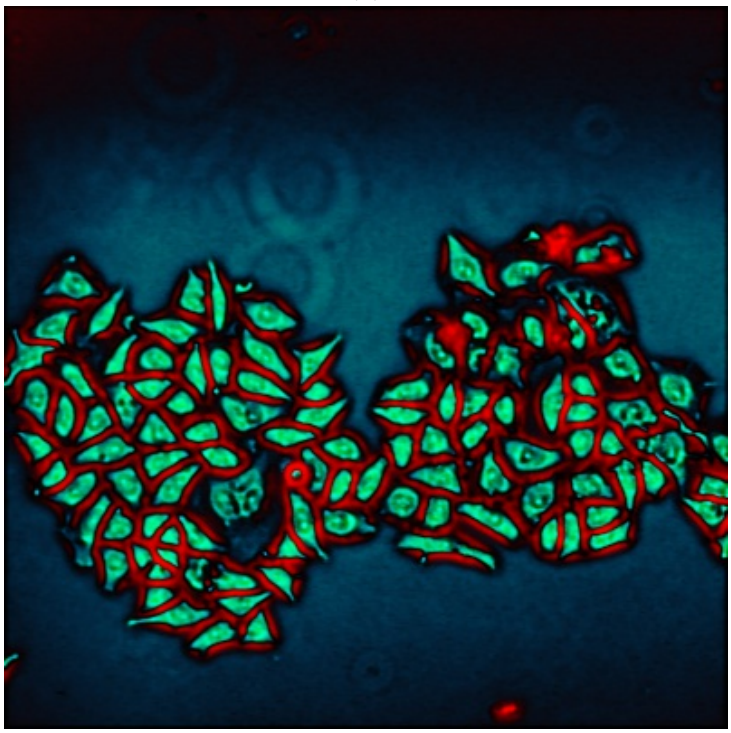
2.2.2 Single Cell Tracking

Cell tracking is essential in understanding the temporal dynamics of cell behaviour in time-lapse sequences. Cell density in time-lapse sequences obtained from typical biological experiments often tend to be high resulting in very closely-packed cells in most frames. In addition, equipment and biological limitations including risk of photodamage reduce the frequency at which the images are captured over time. This results in time-lapse data with significantly low temporal resolution compared to that of conventional video. As a result, motile cells in the time series data seem to *jump* in random fashion between consecutive frames showing minimal spatial overlap. These make cell tracking more complex and challenging than conventional object tracking in video [36]. This is further compounded by sudden changes in cell morphology over time [37]. Current methods for cell tracking could be broadly categorized into deformable models, state space models, and segmentation-based object association.

Deformable models involve a contour evolution approach, usually level set and active contour methods to obtain the contour of an object in the current frame by evolving the contour in the previous frame [38, 39, 40]. In practice, it involves initializing the segmentation of objects in the first frame and updating in subsequent frames. Such contour evolution models fail in applications where objects undergo sufficiently large



(a)



(b)

Figure 2.5: (a) Sample phase contrast image. (b) Three-component reconstructed image using phase retardation model

displacements or show little overlap between consecutive frames [41]. Such deformable models could also struggle to accurately track dividing cells and cells that leave or enter the field of view. Li *et al.* in [39] attempt to address this by associating new cells with a parent cell as daughter using a local association scheme.

State space models for tracking often employ stochastic filtering techniques that rely on the motion or appearance model of the tracked object. In this regard, the model expects the object to follow some assumed motion pattern but do not require an accurate segmentation of the tracked objects. They tend to be computationally demanding due to the large number of motion hypothesis involved yet allow for complex observation models [42]. They are however better suited to handle larger displacements [43][44][45].

Segmentation-based object association tracking models involve two major processes: object detection or segmentation in all frames and the association of objects in different frames to obtain a lineage. This allows cell tracking to be handled as two separate tasks; detection and association. Such association methods have been shown to be effective for cell tracking, scale well and achieve high accuracy in overall cell tracking [41][46][47][16][48][49]. The segmentation step vary widely depending on application or imaging modality. The methods used for such segmentation may include any of the methods discussed in Section 2.2.1.

Another method that could be loosely associated with the object association model is template matching with image registration between time points. The registration could be applied at the pixel level [50], at feature points [51], or for entire cell objects [52]. The remainder of this section will focus on various approaches used in object association models as these have been most effective for cell tracking in time lapse phase contrast data [49][16][48].

2.2.2.1 Learning-based cell tracking

Tracking by assignment usually involves optimization of terms that contain parameters that need to be adjusted to obtain good assignments. Such parameters could be learned from training data by casting tracking as a learning problem [41]. Such methods have

been applied to tracking pedestrians in conventional video with Conditional Random Field (CRF) [53] and HybridBoost [54] techniques. The main challenge with learning based methods is obtaining accurate training data which usually involves costly annotations. In this regard, a cell labelling and active learning framework [55] has been applied to minimize the cost of annotation that is required in [41].

Lou and Hamprecht [41] present a more comprehensive learning-based formulation of the minimum cost flow theory method proposed by Padfield *et al.* [16] which is purpose-built for cell tracking. In this formulation, given sets of detected objects $\{C, C'\}$ from two consecutive frames, the goal is to find the optimum assignment of all the possible event hypotheses between $c \in \mathcal{P}(\mathcal{C})$ and $c' \in \mathcal{P}(\mathcal{C}')$. The compatibility of an assignment with an event $e \in E$ is expressed as an inner product, $\langle f^e_{c,c'}, w^e \rangle$, of a feature vector $f^e_{c,c'}$ that characterizes the discrepancy between object candidates c,c' and a parameter vector w^e that encodes what is learned from the training data for every event. Hence, considering all objects in both frames and all types of events, results in the compatibility function;

$$\mathcal{L}(x, z; w) := \sum_{e \in E} \sum_{c \in \mathcal{P}(\mathcal{C})} \sum_{c' \in \mathcal{P}(\mathcal{C}')} \langle f^e_{c,c'}, w^e \rangle z^e_{c,c'} \tag{2.1}$$

where $\mathbf{z} = \{z^e_{c,c'}\}$ is the set of binary variables whose element indicates whether a hypothesis is accepted. Figure 2.6 shows the schematic of the tracking framework developed by Lou *et al.*[41].

The approach *learns* the parameters associated with tracking events from training data to improve robustness and eliminate parameter tweaking. The learning of parameters hence provides a more robust approach to estimating model parameters. Such a learning-based approach however requires ground truth training data that contains frame-to-frame association pairs of all transition events for all cells. In the standard frame transition events, a cell may *move* or *divide*, and can *appear* or *disappear* from a frame. Accounting for appearing or disappearing cells is important as cells near the boundary of the field of view of the camera could move out and in depending on the direction of motion of the cells. Also, by introducing additional events *merge* and *split*, Lou *et al* [41]

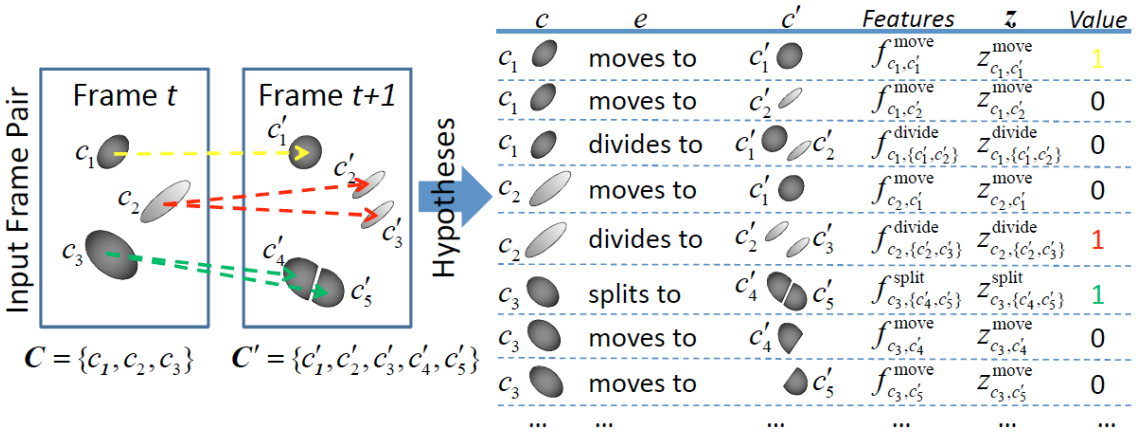


Figure 2.6: Schematic showing tracking framework used by Lou and Hamprecht [41] showing the assignment representation in a pair of adjacent frames

account for the under-segmentation and over-segmentation errors typically observed in the detection step of closely-packed objects.

2.2.2.2 Global object association tracking

As indicated earlier, the object association methods have shown to be effective in time-lapse. However, the challenge of accurately accounting for cells that leave or appear in a frame as well as merges and splits over many consecutive frames still remains. To resolve this, Bise *et al.*[48] proposes a global object association model rather than the local association methods involving association between consecutive frames only as in [16] and [41].

Global association methods associate objects over multiple frames by joining multi-frame trajectories. This could be achieved via *tracklet stitching* [56][57]. This involves first generating reliable fragments of object tracks (tracklets) and then stitching these tracklets together. Connecting tracklets is achieved in [56] via the Hungarian algorithm [58] and by dynamic programming in [59].

The global association methods mentioned above are built for general object tracking and do not address specific cell behaviour such as division. The formulation proposed by Bise *et al* [48] however accounts for division and possible cell segmentation errors. The tracklet association problem is formulated as a maximum-a-posteriori (MAP) problem

and solved by linear programming to obtain the cell trajectories and lineage trees. Figure 2.7 shows the overview of the tracklet association approach.

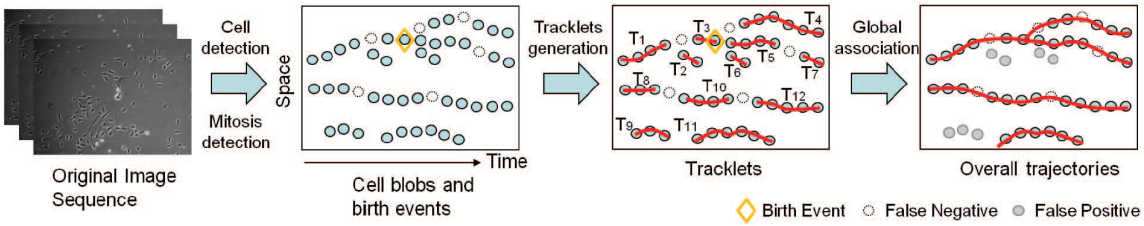


Figure 2.7: Overview of tracklet association method used by Bise *et al* [48]

2.2.3 Cell state labelling

The morphology of a cells in phenotypic experiments changes over time. Such changes are characterised by events or states that are known to be biologically relevant in cell division cycle, cell death or other cell behaviourial patterns. By segmenting, tracking and computing relevant features of individual cells from videos as described above, a temporal sequence of the features that encode these states at the single cell level is obtained. A transition model is then required to partition the sequence of features from cell trajectories into states that could be analysed to understand the behaviour patterns of the cells under study. Each state then represents a morphology profile where similar morphological features belong to the same state. Hence frames where cells show similar appearance and encoded in their computed features will be more likely to be assigned to the same state.

Categorising time points in the cell trajectories into states provides an effective way of grouping similar cell features in time. Depending on the experiment or application, the features of interest that are partitioned in time could be cell morphology, image intensity of a marker or both. In label-free phenotypic analysis, the key features are morphology-based; including cell structure and texture. In multi-parameter experiments, expression of various fluorescent indicators in addition to morphology are monitored.

Mitotic phase labelling is the most widely performed cell state labelling in dynamic cell image analysis. Mitosis, the process by which individual cells divide into daughter cells is

characterised by a number of observable phases; interphase, prophase, prometaphase, metaphase, anaphase, and telophase. Given the video or track of a dividing cell, with associated features for each time point, the task is to accurately label the state corresponding to each phase. To achieve this quantitatively requires temporal models. The most common model used is the hidden markov model (HMM) [60]. The HMM, as well as its more flexible extension, hierarchical dirichlet process (HDP) HMM, are reviewed below.

2.2.3.1 Hidden Markov Models

Cell processes can can be categorised into temporal states or phases that generally follow some underlying order based on the observed features at each time point. The next state in a pattern is usually dependent on the previous observed state or set of states. Markov and semi-Markov models provide the framework for modelling such phenomena.

The Hidden Markov Model(HMM) is the most widely used transition model and has previously been used for modelling cell cycle [61]. Although initially introduced in the 1960s [62], HMMs remain a popular modelling approach in the quantitative study of cell dynamics [61, 63, 64, 65].

The state model provides a framework to infer a set of *hidden* state variables from a sequence of *observed* variables. The hidden state variables are linked by a transition matrix that defines the probability of reaching any state given the present state. Hence, the HMM is a Markov chain where the probability of the next state, given the discrete observations y, is dependent only on the current state and not on any past states:

$$p(x_{t+1}|x_1, x_2, ..., x_t) = p(x_{t+1}|x_t) \tag{2.2}$$

where x_t is the hidden state variable at a discrete time, t. The hidden state variable here, x_t is also discrete and can take on K values denoted by integers denoted by integers $\{1, ..., K\}$. In the classic HMM, the value K, representing the number of hidden states involved, is provided as a parameter input to the model.

To formally define an HMM for any particular application requires the following set

of elements to be determined:

1. N - the number of states in the model. This is usually the first design parameter to be selected depending on the needs of application. In the case of the mitotic phase modelling, this will be 6; one for each of the states listed in Section 2.2.3 above. The set of states are denoted $S = \{S_1, S_2, ..., S_N\}$

2. M - the number of distinct observation symbols per state. This relates to the number of features that are being observed to infer the state and hence determined from the length of the feature vector for a cell.

3. $A = \{a_{ij}\}$ - the state transition probability distribution defines the probability of reaching a given state, S_j given the current state, S_i. A is a matrix with entries, a_{ij} specifying the probability value for pairs of states. For example, in mitotic phase labelling, if S_i is state for interphase and S_j is the state for prophase, a_{ij} is the probability of transitioning from interphase to prophase.

4. $B = \{b_j(k)\}$ - the observation symbol probability distribution in state j. This represents the distribution of features variables within a given state, S_j. For example, the distribution of cell area within interphase.

5. $\pi = \{\pi_i\}$ - the initial state distribution. This represents the likelihood of each state before features are observed. This usually comes from knowledge of the biology of the states involved.

Given the appropriate model specification values of N, M, and $\lambda = (A, B, \pi)$, the following problem of finding the states from a sequence of features can be posed as:

given observation sequence of features, $O = (O_1, O_2, ...O_T)$ *and the model* λ*, how do we choose a corresponding optimal state sequence* $Q = (q_1, q_2, ..., q_t)$. This is solved by applying the forward-backward algorithm [60]

In the analysis of cell state from live cell imaging, HMMs have been applied mostly in mitotic phase labelling [66, 61]. Such studies usually require annotation of the known cell states (e.g mitotic phases) in order to model meaningful states and transitions. In

the wider context of live cell studies, the dynamics of the cell phenotype is not strictly known, and hence the task is usually exploration aimed at pattern discovery. To model the changes in cell morphology dynamics, Zhong *et al.*[65] proposed an unsupervised modelling of the dynamics of cell morphology using a combination of their temporarily constrained combinatorial clustering (TC3), Gaussian mixture modelling (GMM), and HMM. In this approach, TC3 is to initialise GMM to cluster frames into groups based on cell morphology with transition modelling achieved via the HMM. Though their approach partitioned cell morphology trajectory into uniform states over time, a user-defined number of states is still required for TC3. Similar to other HMM-based approaches, the user-defined number of states is usually unknown or poorly understood biologically.

In studies such as cell cycle phase labelling where the number of hidden states, K is known or well-understood, El-Labban *et al.* [67] has shown that the classic, as well as the adapted HMM applies well. The work by Harder *et al.* [63, 64] further shows that a generalised finite state modelling achieves a transition model for cell phenotypes associated with mitosis. While the study of cell cycle in [67, 64] provide a strong set of priors for the state distribution, other studies where the phases are not as well defined or known are more challenging to model with classic HMMs. This is due to the difficulty in selecting an appropriate input value for K. In the exploratory analysis of the morphological states in a variety of timelapse cell experiments, the number of states is usually unknown or poorly understood across cell lines and varying treatments. The states hence need to be *discovered* from the morphological features over the frame sequences. The classical HMM is therefore insufficient for such analyses.

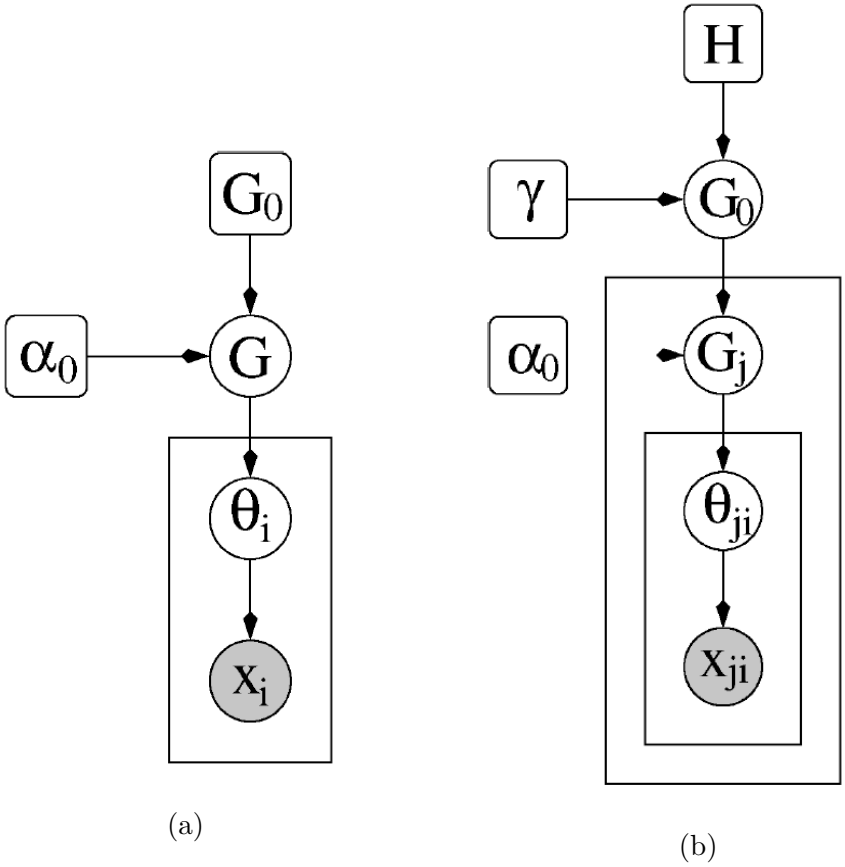
2.2.3.2 Hierarchical Dirichlet process HMM (HDP-HMM)

HMMs provide a clear intuitive approach for modelling dynamic cell behaviour as a probabilistic transition of states. A significant drawback of the model arises in exploratory analysis when the number of cell states is unknown. Choosing the appropriate number of states for any given dataset is challenging. However, model selection and cross validation techniques can be used to choose the *best* number of states from a range of values based

on some evaluation metric. A more flexible alternative is the class of models considered *infinite* HMMs, in which the number of states of the HMM is allowed to be countably infinite [68, 69]. In such modelling, the number of states is inferred from the clusters that form within the data. This proves more useful in understanding single cell dynamics and and patterns present in a given dataset in explorative analysis.

The Dirichlet process (DP) is a stochastic process used in nonparametric Bayesian modelling of data. It is a distribution over distributions, such that each draw from a Dirichlet process is itself a distribution [70]. The DP is defined as $G \sim DP(\alpha_0, G_0)$, where G is the distribution of a random probability measure over a measurable space, (Θ) such that for any finite partition $(A_1, A_2, ..., A_r)$ of Θ, the random vector $(G(A_1), ..., G(A_r))$ is distributed as a finite-dimensional Dirichlet distribution with parameters $(\alpha_0 G_0(A_1), ..., \alpha_0 G_0(A_r))$ [70, 71]. In this formulation, G_0 is the base measure representing the mean of the DP whereas α_0 is the concentration parameter that describes the concentration of mass around the mean of the DP. The DP is used as a nonparametric prior on parameters of mixture models, particularly DP mixture models. A DP mixture model can be derived as the limit of a sequence of finite mixture models, where the number of mixture components is taken to infinity. A hierarchical DP is defined as a measure where the base measure is also distributed as a DP [69]. Figure 2.8 shows the graphical model representation of the DP mixture model and hierarchical DP mixture model.

As described in Section 2.2.3.1, the classic HMM is a doubly stochastic Markov chain in which the sequence of multinomial state variables, $Q = (q_1, q_2, ..., q_t)$ are linked through a transition matrix such that each observation, O_t is drawn independently from other observations, $O = (O_1, O_2, ..., O_T)$ conditional on y_t. This casts the HMM as a dynamic set of finite mixture models where each state corresponds to a mixture model. Teh *et al.* [69] define a nonparametric model with countably infinite states by replacing the set of conditional finite mixture models in the classical HMM with a hierarchical Dirichlet process mixture model. The resulting model is referred to as a hierarchical Dirichlet process hidden Markov model (HDP-HMM). Figure 2.9 shows the graphical



(a) (b)

Figure 2.8: Graphical model representation of (a) DP mixture model and (b) hierarchical DP mixture model [69].

model representation of the HDP-HMM. The HDP-HMM eliminates the need to select a prior number of states for the model.

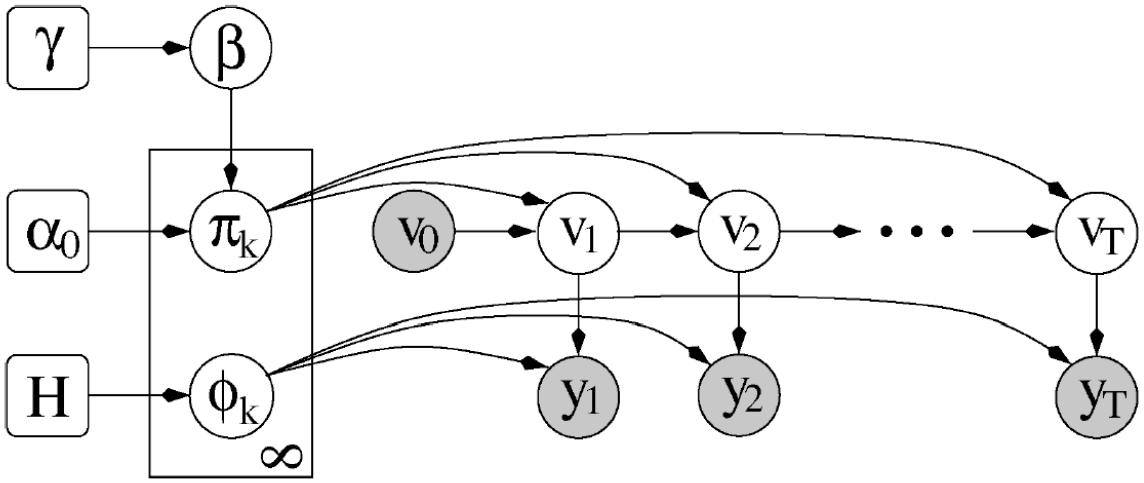


Figure 2.9: Graphical model representation of the HDP-HMM [69].

Fox *et al.* [72] however further shown that the original formulation of the HDP-HMM where the number of states, K grows infinitely could be prone to the proliferation of a large number of redundant states with rapid switching among states. In proposing the *sticky* HDP-HMM, which was originally used to solve the speaker diarization problem, Fox *et al.* show that by introducing a bias parameter, rapid transitions among redundant states can be minimised, leading to better performance. Figure 2.10 adopted from [72] shows the effect of the bias parameter in minimising redundant states and improving performance in the speaker diarization problem.

The sticky HDP-HMM generative process $(\gamma, \alpha, \kappa, G_o)$ can be summarised as

$$
\begin{aligned}
&\beta \sim GEM(\gamma), \\
&\pi_i \overset{iid}{\sim} DP(\alpha + \kappa, \beta + \kappa\delta_i) \qquad \theta_i \overset{iid}{\sim} G_o \qquad i = 1, 2, ..., \\
&z_t \sim \pi_{z_{t-1}}, \\
&y_t \sim f(\theta_{z_t}) \qquad t = 1, 2, ..., T,
\end{aligned}
$$

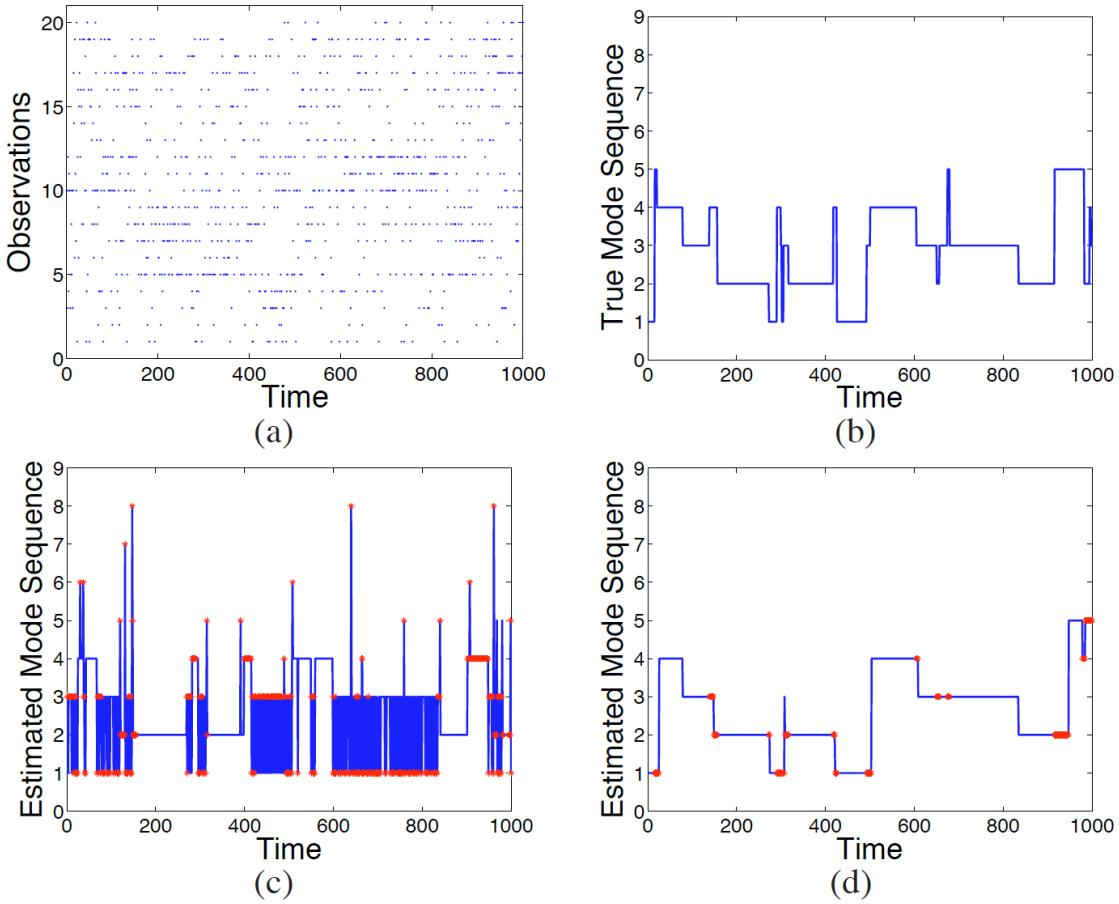


Figure 2.10: (a) Multinomial observation sequence; (b) true state sequence; (c)-(d) estimated state sequence after 30,000 Gibbs iterations for the original and sticky HDP-HMM, respectively, with errors indicated in red. Without an extra self-transition bias, the HDP-HMM rapidly transitions among redundant states

where GEM (Griffiths, Engen, and McCloskey) is a *stick breaking* process [73], DP is a Dirichlet process and f is an observation distribution parameterised by draws from G_o, the base distribution. Each transition distribution from state i, π_i is a DP draw and the HDP plays the role of a prior over infinite transition distributions. The π_i are linked by being DP draws parameterised by and hence dependent on the same discrete measure β, thus $\mathbb{E}[\pi_i] = \beta$. Values of $\kappa > 0$ imply increasing the prior probability $\mathbb{E}[\pi_{ii}|\beta]$ of self-transitions.

2.3 Software tools

Eliceiri *et al.* [74] provide a comprehensive overview of all the different software tools that are necessary for the implementation of an image informatics workflow. Here image acquisition, storage, analysis and image and data visualisation need to be taken into account. This section focuses on image analysis tools with an emphasis on available open source packages. It should be mentioned that environments such as OMERO [75] and Bisque [76], which were originally developed for storing and retrieving microscopy image data sets have now been extended to provide client-server software visualisation, analysis and management of microscope images. While industrial labs and larger centres have the capacity to customise these solutions, a more modular approach might suit other users. In this presentation we differentiate between software that is designed for biologists and toolkits for developers of algorithms or workflows.

2.3.1 General Tools

Undoubtedly ImageJ and the recently enhanced distribution of the same software, Fiji [77], are the oldest and most widely used tools for scientific image analysis. The software can be extended with help of a large number of plug-ins ranging from cell counting to co-localisation analysis. From the quick assessment of image quality to the scripted analysis of smaller image data sets it can be used for a wide range of tasks. As it is written in Java it can be easily deployed on all popular operating systems. While very

versatile, some of the more advanced algorithms for segmentation and tracking are not available and it is probably less suited for the routine analysis of large data sets.

Cellprofiler [78] has been developed to facilitate the interactive data exploration, analysis, and classification of large biological image sets. Originally written in MATLAB, the software has been re-written in Python and includes a number of powerful features such as a range of machine learning tools. Cellprofiler provides an open source alternative to commercially available high-content image analysis software packages such as Cellomics (ThermoFisher Scientific), Harmony (Perkin Elmer) or IN Cell Investigator (GE Healthcare Life Sciences). As an open source tools it provides the advantage that it can be extended very easily but on the other hand it is not very well integrated with high-throughput microscopy hardware.

While Cellprofiler [79] is more geared towards 2D high-throughput screening data, Icy [79] provides a very comprehensive set of tools for the analysis of multi-channel 3D images. Developed and maintained by the biomedical image analysis group at the Institut Pasteur (Paris, France) it builds on a number of existing open source libraries and can also be extended through a broad range of plugins. The ilastic [80] platform is a simple, user-friendly tool for interactive image classification, segmentation and analysis. It provides access to some very advanced machine learning based algorithms for segmentation and tracking.

2.3.2 Toolkits & Libraries

The National Library of Medicine Insight Segmentation and Registration Toolkit (ITK) [81, 82] is the most widely used open source medical image analysis toolkit. It is designed to support N-dimensional images. Working and extending the C++ based libraries does require training. However, as the algorithms that have been included in the toolkit have been carefully validated, ITK made an invaluable contribution to the scientific community. SimpleITK [83] now provides a scripting interface to the underlying libraries. The newest release now includes access to the registration algorithms which will be of interest to a number of groups. Farsight [84, 85], which is a more specialised set of tools

for the analysis of multi-channel fluorescent microscopy images is build directly on ITK.

Python is increasingly viewed as the lingua franca of data analytics and provides a great environment for developing analysis scripts that can be deployed on end user computers as well as specialised compute clusters. Rather than entering into a debate whether or not is should be used instead of statistics package R we argue that these two languages, which are both distributed under open source licences, complement each other. With the Python interface of OpenCV [86] and the scikit-image libraries [87] Python provides a very rich set of image analysis tools. To date there are not many libraries that address microscopy specific analysis solutions. The Python Microscopy Environment (PyME) [88] is being developed to address this gap. It is targeted to high-resolution microscopy. In addition, all the underlying software modules provided by Cellprofiler can be accessed as Python libraries. SimpleITK can now also be installed directly as part of leading Python distributions. Scripting languages not only provide the advantage of fast prototyping, they also allow the systematic integration of well established libraries for numerical computing, linear optimisation and now machine learning. With the increase in data volume it will be necessary to process data sets on dedicated compute clusters. The Python environment does allow to make this transition. Possible performance bottlenecks can be eliminated through targeted optimisation.

Commercial software are often pre-installed, marketed and sold alongside hardware microscope systems. Most commercial software hence combine the image acquisition software that drives the microscope hardware in acquiring the images with analysis and storage following acquisition. These commercial software are developed by microscope manufacturing companies and hence supplied with microscopes. include NIS-Elements from Nikon, cellSens from Olympus, Volocity, Acapella and Columbus from Perkin-Elmer, InCell Investigator from GE, , and AxioVision from Zeiss. Upgrades and functional add-ons to such commercial software is controlled by internal teams at the manufacturing companies and offered to existing users for free or for additional fees.

Alternatively, open source software are not tied to a given microscope company but rather developed by collaborating research teams. The software is usually downloadable

free online and the source code is available and modifiable to fit specific needs that may not be included. Added functionality is developed by research interest groups and released free to the user community. Open source software for biological imaging is dominated by analysis software. Popular open source projects include CellProfiler[89], CellProfiler Analyst[90], Icy[91], CellCognition[61], ImageJ[92], and Fiji[93]. Table 2.1 summarizes the key uses and origin of some of the popular open source cell image analysis software.

2.4 Conclusion

Live cell imaging experiments are generally more complex compared to their static endpoint alternatives. Conventionally, cell biology researchers interested in understanding phenotypic transformations observe cell morphology changes under the microscope. With the advent of high throughput phenotypic screening enabled by new live cell culture and imaging systems, the effect of a number of candidate compounds are being studied using millions of cells. The data output in such high throughput live cell imaging experiments a vast set of time lapse sequences.

Analysing dynamic behaviour of cells based on time-lapse images at the high throughput scale manually by a human observer is infeasible. Quantitative approaches are therefore required to automatically extract insights from such dynamic data. In this regard, three main tasks are required; cell segmentation, single cell tracking and temporal modelling. A number of computational methods and software tools have been developed to enable efficient and accurate analysis of live cell imaging. Open source software tools for general cell image analysis include CellProfiler, ImageJ, CellCognition, and Icy. Commercial software often marketed and supplied with microscopes also include Acapella, Columbus, cellSens, InCell investigator, and NIS-Elements.

Most of the existing tools cater for segmentation and tracking tasks, whereas temporal modelling tools and methods are rare. A number of research teams, including the team responsible for CellCognition[61], have been working on addressing this gap[65][94][95][96]. Considering most of the approaches develop temporal models for

Table 2.1: Summary of open source software

Software	Key capabilities	Origin	Website
CellProfiler	Object counting, segmentation and tracking	Carpenter Lab at Broad Institute of Harvard and MIT	www.cellprofiler.org
CellProfiler Analyst	Interactive data exploration, Machine Learning tools for classification and analysis of large image datasets	Carpenter Lab at Broad Institute of Harvard and MIT	www.cellprofiler.org/cp-analyst
CellCognition	Object detection, feature extraction, tracking, cell transition analysis	IMBA Vienna, Institut Curie Paris, EMBL Heidelberg, and ETH Zurich	www.cellcognition.org
Icy	3D Image analysis, object tracking, Sequence analysis, Matlab and ImageJ integration	Quantitative Image Analysis Unit at Institut Pasteur	www.icy.bioimageanalysis.org
CellOrganiser	Cell shape modelling, cell time series analysis	Center for Bioimage Informatics at Carnegie Mellon University	www.cellorganizer.org
ImageJ	Multidimensional image analysis, segmentation, tracking, shape analysis	National Institutes of Health	www.imagej.net
Fiji	Variant of ImageJ, additional analysis plugins pre-packaged	National Institutes of Health	www.fiji.sc

specific applications such as cell cycle, the work presented here seeks to generalised temporal state space modelling that allows fitting a unified model to explore a wide variety of cell dynamic behaviour.

Chapter 3

Cell Segmentation in Phase contrast

3.1 Introduction

High-content microscopy plays an increasingly important role in drug development and basic life science research. There has been growing interest in over the last decade in enabling live cell imaging [8] experiments that reveal more detailed information about cellular function than standard end point assays. Label free imaging, in particular phase contrast microscopy, plays an important role in studying live cells since it minimizes the need for potentially cytotoxic fluorescent stains and labels. Considering the volume and complexity of the image data sets obtained from such studies, analysis tools that measure parameters associated with cell division, motility, mechanisms of death and population statistics are required.

One fundamental task in quantitative biological imaging is cell segmentation. While a number of sophisticated and robust segmentation techniques have been developed for fluorescent microscopy, cell segmentation in phase contrast image sequences is still challenging. Artefacts that are introduced by the specific optics, observed as halo and shade off effects, as well as the lack of clear boundaries among cluttered cells, poor contrast between cells boundaries need to be addressed to obtain an accurate separation of cells from background. A variety of methods have been developed to segment cells from background under these conditions. Wang *et al.* [97] apply machine learning to learn the

specific edge profiles observed in phase contrast data whereas Arteta *et al.*[23] employs a structured learning scheme with dot annotations over non-overlapping extremal region features.

A key component of machine learning approaches applied to object detection and segmentation is extracting the most relevant and information-rich features from which a model can be obtained that distinguish cell regions from background and other artefacts or objects within any given image. Conventional feature extraction methods for object detection and segmentation in images generally use the pixel values and representation in an image to extract colour, edges, corners and textures that are relevant to differentiating objects and background. Such methods do not specifically account for the optics of the image formation process and therefore tend to perform well when the images have little or no artefacts resulting from imaging system. In recent years, model-based approaches to exploit image formation has been proposed. Ali *et al.* [24] use local phase analysis to extract features in DIC cell images and then applies level set techniques to automatically extract the cells from background in low contrast images.

This chapter focuses on a model-based feature extraction method proposed and developed by Yin *et al.* and Su *et al.* for phase contrast images [15, 14]. The approach extracts rich structural features from raw greyscale images using a physical model of the image formation process in phase contrast microscopy. At the core of this model is optical phase retardation, which is intrinsic to the optical image formation process in phase contrast microscopy. Details of the model and implementation is reviewed in later sections.

Recent advances in deep learning architectures have delivered state-of-the-art performance across benchmark object detection datasets. Deep learning architectures are composed of a multi-layer set of convolutional neurons and non-linear filters that transform an input image into a series of volumes resulting in object or pixel labels at the output layer. Through the series of transformations, deep learning methods *learn* relevant features corresponding to colour, edges, corners, and textures at each layer. This eliminates the need to designed *hand-crafted* features that are relevant in detecting and

segmenting objects from background in an image.

A significant drawback of such deep learning architectures is the large number of labelled samples and computational resources required to trained the resulting model. Various deep learning methods have been applied to cell detection [33, 34]. In practice, it is generally rare to have such large labelled datasets to train a deep convolutional neural network (CNN) from scratch for every given application. An alternative approach involve adopting pre-trained models to initialise new models for specific applications or using the pre-trained as a feature extractor for new datasets that can then be classified with conventional machine learning methods such as support vector machines. These approaches fall under the broad area of *transfer learning*. The feature representation activated at given layers of a deep pre-trained can be obtained from known architectures given the learned model weights. Most deep architecture models for image classification are trained on the large labelled dataset provided in the large public image repository, ImageNet[98]. To motivate the extraction of phase retardation features as a preconditioning step in phase contrast, it is shown here that preconditioned image obtained provides richer activations in a pre-trained model of the popular deep learning architecture, VGGNet16 [99] compared to the raw pixel image. Details of this is presented in Section 3.4.

Using the phase retardation features, a weakly supervised machine learning approach is proposed that achieves good segmentation in phase contrast images. In this approach, multi-dimensional phase retardation features are extracted for each pixel in the given image. Pixel features are aggregated into SLIC superpixels [100] to obtain homogenous local clustering of neighbourhood pixel features. This contributes to a significant reduction of the number of samples to classify, reduces model complexity and hence improves computational speed. A neighbourhood graph of superpixels is constructed and used in clustering and watershed segmentation to find an accurate delineation of the cell boundaries. Weak *a priori* knowledge about a single point location of the individual cells is also used to achieve robustness obtaining cell regions in the presence of artefact objects in the image. Details of the approach are presented in Section 3.2. Figure 3.1 shows the

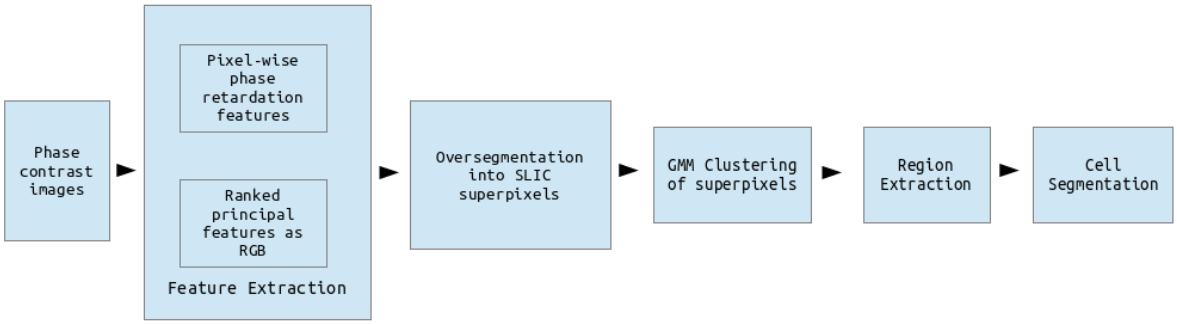


Figure 3.1: The general overview of the proposed method for phase contrast cell segmentation. For every image, phase retardation features are computed for every pixel by image reconstruction based on a dictionary of diffraction patterns. The pixel-wise features are ranked according to the most discriminative by a graph cut score. Pixels are then aggregated into SLIC superpixels based on their features to obtain an oversegmentation. The superpixels are then grouped into regions and the regions that correspond to cells are extracted and merged boundaries are split by a watershed algorithm

segmentation pipeline.

3.2 Method

Extracting accurate morphometric information for each individual cell requires reliable detection of object boundaries. Given the nature of phase contrast microscopy, the appearance of such object boundaries depends on the specimen itself as well as the local imaging conditions. This motivates why we utilize the phase retardation features, which are then aggregated into superpixels. Clustering is used to label the superpixels, which results in an over-segmentation of the image. The identification of the individual cell bodies is explained in Section 3.2.3.

3.2.1 Phase retardation feature extraction

Phase contrast microscopy is essentially a contrast-enhancing technique that employs an optical mechanism to translate small variations in phase (diffraction pattern) into corresponding changes in amplitude, which can then be visualized as differences in image

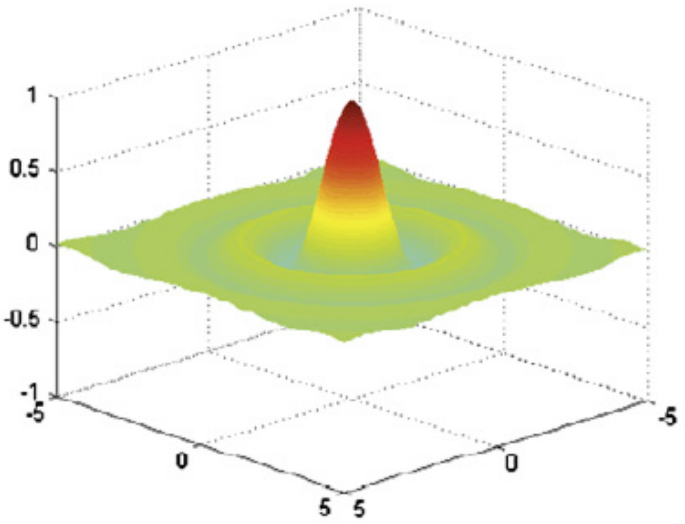
intensity. The key difference in setup of a phase contrast microscope and a conventional brightfield microscope is the introduction of the a phase ring and annulus that enables the contrast obtained in the phase of light to be effected into light intensity to be observed. Details of this setup is reviewed in Section 2.2.1. This image formation process is exploited by Yin, Su and colleagues [15, 14] in an image reconstruction method to obtain features for each pixel. The multidimensional features at each pixel is dictionary of diffraction patterns quantifying the phase retardation experienced at the pixel. The features for an N-pixel grayscale image expressed as:

$$g = \sum_{m=0}^{M-1} H_m \psi_m, \quad s.t.\ \psi_m \geq 0, \tag{3.1}$$

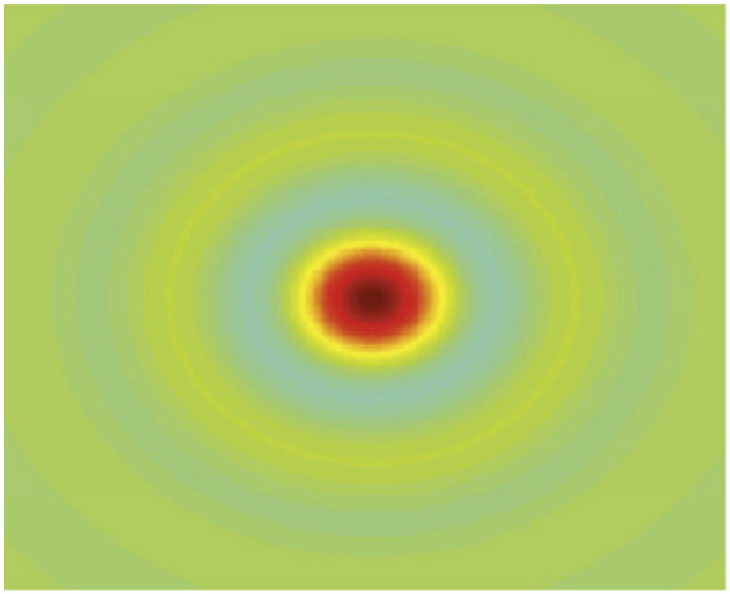
where g is the vectorized image intensities $(N \times 1)$ expressed as a linear combination of a dictionary of M bases $\{H_m\}_{m=0}^{M-1}$. The vectorized coefficients, Ψ_m for each pixel corresponds to a diffraction pattern with phase retardation angle $\theta_m = \frac{2\pi m}{M}$. $m = \{1, 2, ..., M\}$, providing a set of M diffraction angles. Figure 3.2 shows the diffraction pattern kernel in 3D and 2D. The diffraction pattern, H_m is function of the phase retardation, θ_m computed from a diffraction pattern kernel, $DPK(\theta_m)$. Figure 3.3 shows the $DPK(\theta_m)$ at different retardations, θ_m. Hence, the n^{th} row of an H_m $(N \times N)$ matrix corresponds to the vectorized kernel function obtained from a diffraction pattern kernal around pixel location x_n. The vectorized coefficients for each grayscale pixel are then utilized as the M-dimensional feature vector for that pixel.

As outlined above, the grayscale pixel values of a phase contrast image can be expressed a linear combination of coefficients, psi_m and the M bases set. The feature vector of each pixel is composed of the M coefficients. Hence, given the grayscale value for each pixel, x_n and the bases set, H the coefficients psi can be computed. This is posed as an optimization problem aimed at obtaining the coefficient values that minimize the reconstruction residual error between the linear combination values and the actual pixel values.

For a given set of N pixel values from a phase contrast image, there are K subset representative diffraction patterns from the $\{H_m\}_{m=0}^{M-1}$ bases that contribute principally



(a)



(b)

Figure 3.2: Diffraction pattern kernel of phase contrast imaging, airy(r), obscured Airy pattern with (a) 3D surface view and (b) 2D top-view. Image from [15]

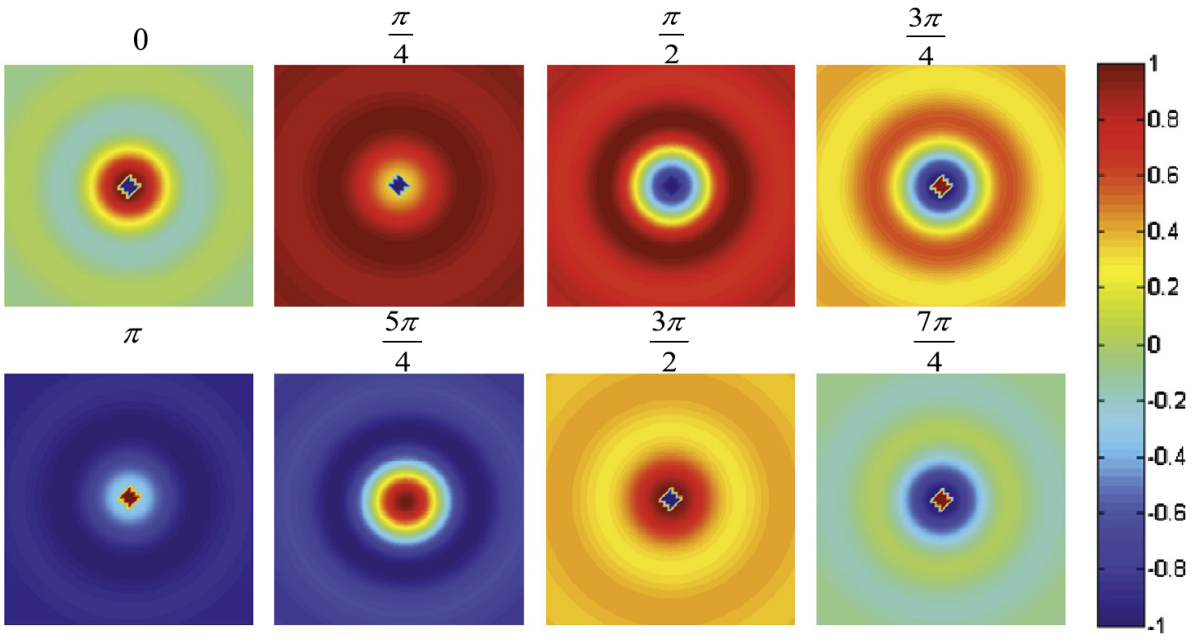


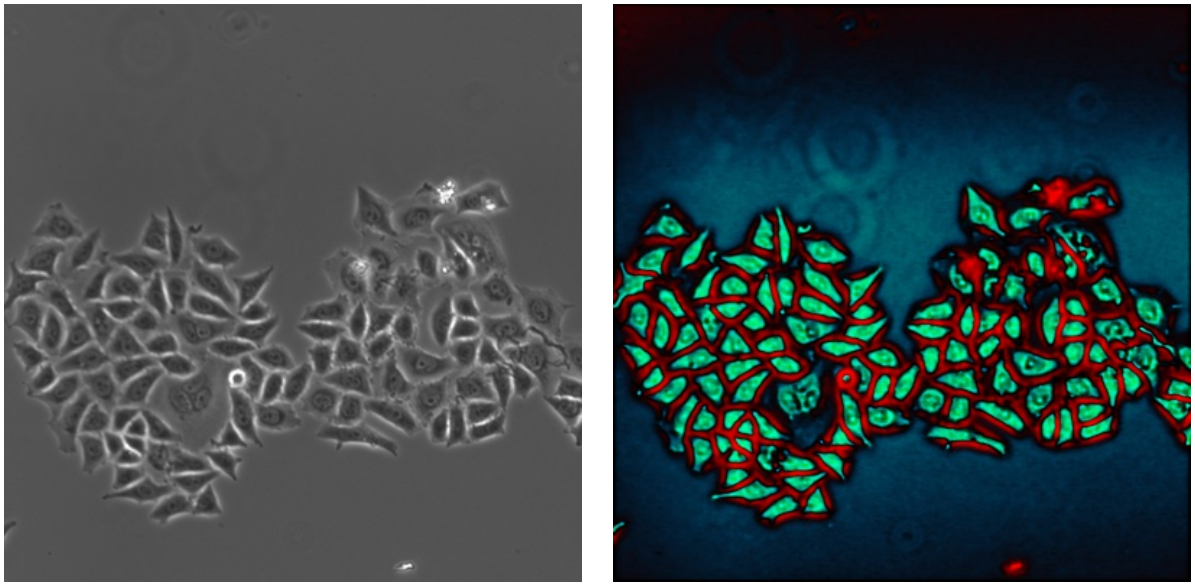
Figure 3.3: $DPK(m)$, the diffraction patterns with different phase retardations, θ_m. Image from [15]

to the visual representation of the image. The problem of finding the K representative bases with the associated coefficients for a given set of pixels by minimizing the image reconstruction residual error is formalised as:

$$\begin{aligned} \min_{\psi_m, K} \quad & \|g - \sum_{k=0}^{K-1} H_m^{(k)} \psi_k\|_2^2 + \omega_r \sum_{k=0}^{K-1} (\psi_k^T \Lambda \psi_k + \|\psi_k\|_1) \\ \text{subject to} \quad & \psi_k \geq 0 \end{aligned} \tag{3.2}$$

where ω_r is the regularization weight parameter, Λ is a Laplacian matrix that encodes the similarity between spatial neighbourhood pixels, and $\|\psi_k\|_1$ is a spatial smoothness regularization parameter. $\|\psi_k\|_1$ is a sparseness regularization term that penalizes the small values of ψ_k, enforcing values of ψ_k corresponding to background pixels to zero.

The extracted features for each pixel, g can be visualised by plotting the top three ψ_k that produce the least residual errors when used in the reconstruction of the original image as outlined above are extracted. The three principal features are observed to preserve the structure of the original image and therefore can be mapped onto the RGB. The coefficients of each basis of the three bases maps directly to a specific structural form or intensity profile of sections of the image of the original image. Figure 3.4 shows the



(a) (b)

Figure 3.4: Initial processes resulting in cluster regions: (a) Original phase contrast image; (b) The top three(3) principal features for each pixel mapped onto RGB of a colour image

RGB colour image image representation of a sample phase contrast image of HeLa cells.

The principal features shown are rich and clearly discriminates cell structures from image background. This provides and good preconditioning of the raw grayscale image for building a robust model for detecting and segmenting cells from background.

3.2.2 Generating superpixels over phase retardation features

The goal of segmentation following extraction of dicriminative features from the raw images is to construct a model that partitions the pixel features into cell regions or background. The visualisation of the features expose the neighbourhood redundancy at the pixel level as neighbouring pixels show similar feature profile. Aggregating the pixel features into local homogenous regions eliminates the local neighbourhood redundancy in the feature space and reduces the computational cost of partitioning the image into the required cell and background regions. Oversegmentation of the phase contrast image using the phase retardation features by simple linear iterative clustering (SLIC) [100] achieves the aggregation of features into local homogenous regions called *superpixels*.

For an image with N pixels, U superpixels which consist of small regions with an

approximate size N/K that do not cross cell object boundaries in the original image as shown in Figure 3.5. Hence, for roughly equally-sized superpixels there would be a superpixel center at every grid interval $Q = \sqrt{N/U}$. Pixel features are aggregated into superpixel regions based on a distance measure encoding similarity between neighbourhood pairs of pixel features. A lower distance between pixels corresponds to high similarity and vice versa. For an image with phase retardation features $\{\Psi_x, \Psi_y\}$ for arbitrary pixel locations $\{x, y\}$, the similarity distance measure for aggregating the pixels, d_s is defined as:

$$d_s(x, y) := (\Psi_x - \Psi_y) + \frac{c}{Q}||x - y||_2 \tag{3.3}$$

where d_s accounts for both feature variation and distance in pixel location. The compactness, c determines the extent of emphasis on spatial proximity. The higher its value, the more spatially compact the superpixels. Figure 3.5 shows the effect of the compactness, c on the superpixel boundaries when increased from $c = 10$ to $c = 100$ with the same number of segments, $U = 2500$.

3.2.3 Multiclass clustering and region aggregation

Phase retardation neighbourhood superpixels need to be further aggregated into cell regions and background. This is achieved through clustering on the features associated with each superpixel to group the segments into regions that correspond to the main region partitions expected in a typical phase contrast image. A typical phase contrast image has three main observable regions. These include dark cells, bright cells, and halos. Cells that appear dark are often the predominant forms in a phase contrast image. Halos are bright regions observed at the boundaries of cells and are considered an imaging artefact relating to diffraction at the cell-background border. Bright cells are observed in a variety of instances, most commonly when cells are at the onset of mitotic division or apoptotic death. Hence in static phase contrast image, any of these profiles could be observed. The task of segmentation, however, is to determine all regions associated with cells while rejecting background and imaging artefacts. The clustering of superpixels is

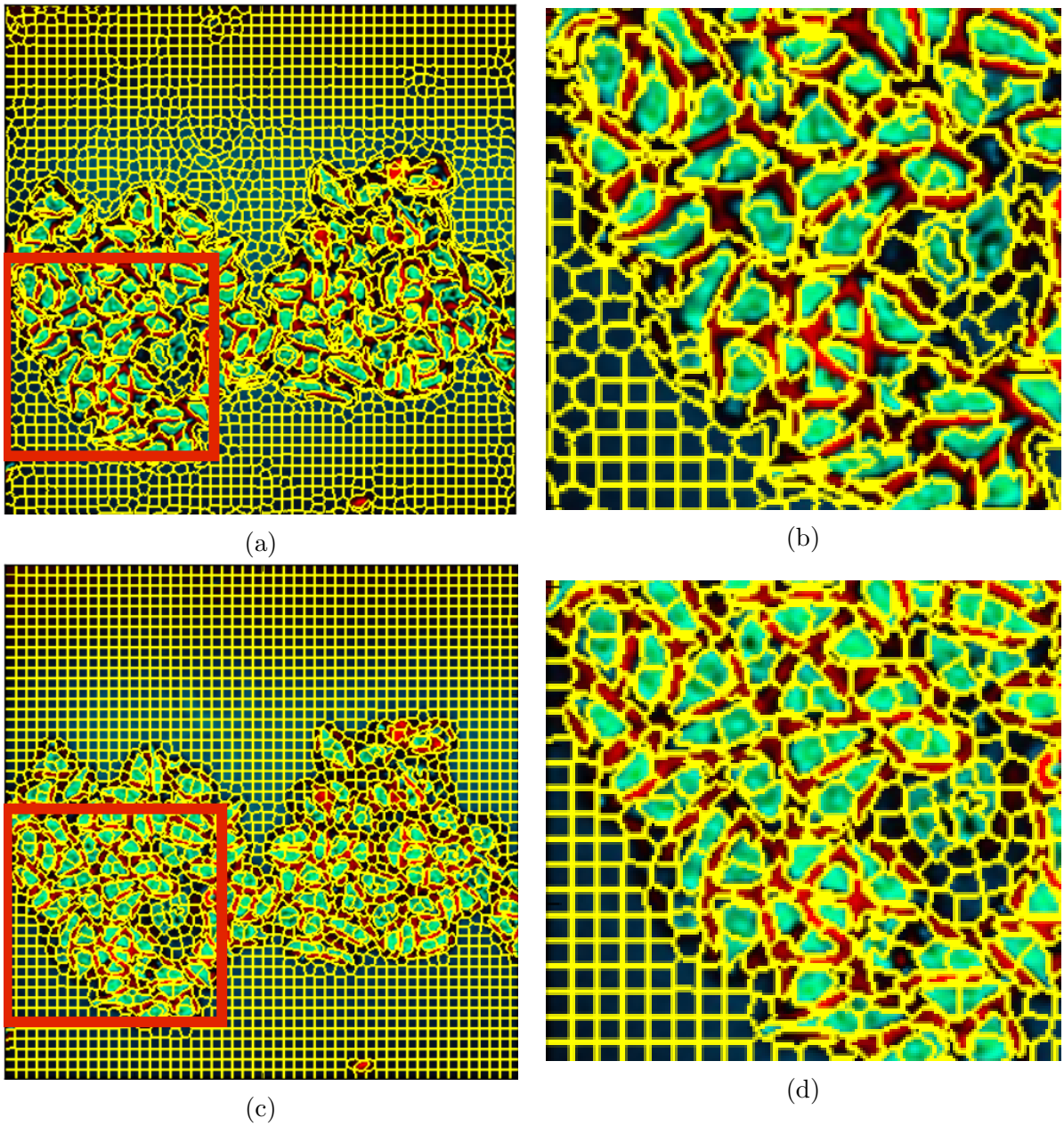


Figure 3.5: Superpixel oversegmentation of phase retardation features showing effect of compactness parameter, c: (a) number of segments, $U = 2500$, compactness, $c = 10$ (b) zoomed section showing closer view of boundaries when $c = 10$; (c) number of segments, $U = 2500$, compactness, $c = 100$; (d) zoomed section showing closer view of boundaries when $c = 10$

hence a multi-class clustering problem to obtain all the partitions expected in the image.

3.2.3.1 Neighbourhood feature augmentation

To compute features for superpixels, each superpixel is initially characterized by the mean and covariance of the multi dimensional pixel phase feature vector within it. It is observed that bright cells and halos are both represented by similar phase retardation features (shown in red in Figure 3.4). Hence bright cells and halos cannot be distinguished by using only the features within each superpixel. As these are usually surrounded by different structures, the feature vector of each superpixel is augmented by incorporating neighbourhood information. To include neighbourhood information, the adjacency graph, $G(V, E)$, of the superpixels $s_i \in S$ is constructed, where E is the set of edges formed between pairs of adjacent superpixels (s_i, s_j) in the image.

By estimating the dissimilarity between the feature distribution, P_s, in a given superpixel, s, and the feature distribution, $P_{N(s)}$, in its neighbourhood, $N(s)$, some basic context information is captured. Dissimilarity is measured by the Jensen-Shannon divergence (JSD) [101], which is a symmetrized and smoothed version of the Kullback-Leibler divergence, KL, is defined as:

$$JSD(P_s||P_{N(s)}) = \frac{1}{2}KL(P_s||M) + \frac{1}{2}KL(M||P_{N(s)}) \quad \text{where} \quad M = \frac{1}{2}(P_s + P_{N(s)}) \ . \quad (3.4)$$

where $KL(P_s||P_{N(s)}) = \sum_i P_s(i) \log \frac{P_s(i)}{P_{N(s)}(i)}$. The divergence between P_s and $P_{N(s)}$ is used to select the most distinctive neighbour feature, which is used as an additional feature for each superpixel s. As a result, an augmented feature vector for each superpixel is constructed that consists of its mean and covariance, and that of its most distinctive neighbour. This ensures that neighbourhood information that is included is the one that shows the most deviation from the distribution of a given superpixel. In essence, each superpixel is now modelled as a multivariate Gaussian with a mean vector and covariance. The newly constructed superpixel features is named *superfeatures*.

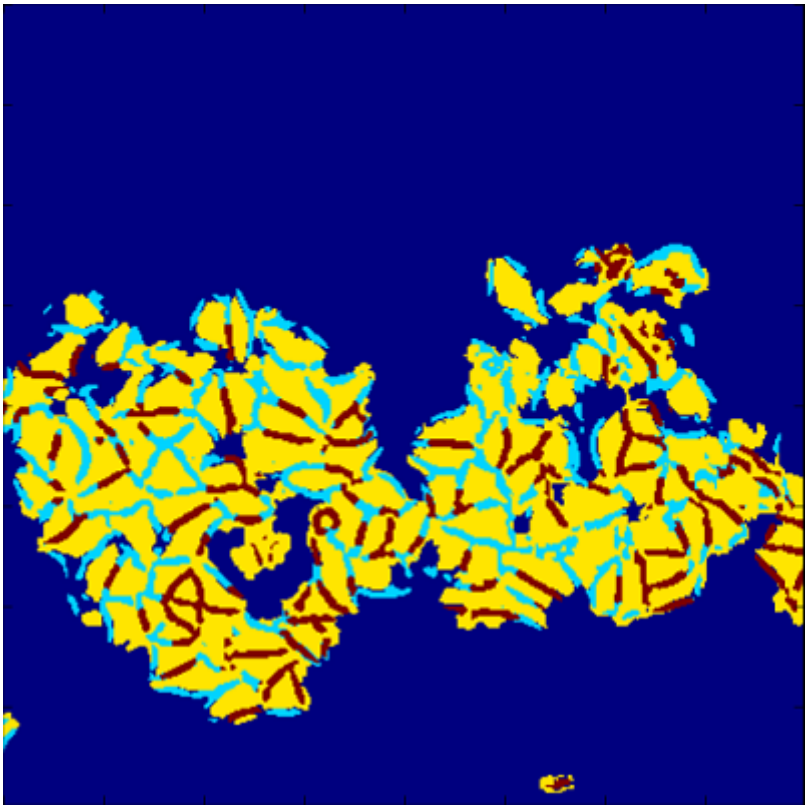


Figure 3.6: Clustering of superfeatures

3.2.3.2 GMM clustering and boundary estimation

Considering every superpixel is modelled by the mean, μ and covariance, σ of a multivariate Gaussian, we proceed to make the assumption that groups of superpixel features that share similar mean and covariance matrix could be modelled as the components of a Gaussian Mixture Model (GMM). The superfeatures are fit to q components of a GMM via expectation maximization (EM). This amounts to clustering the superpixels in a given image into q clusters. Figure 3.6 shows the output of the clustering.

It is observed that the clusters partition the image into various regions. Considering this partitioning is completely unsupervised, the label associated with cell regions is obtained by incorporating prior weak knowledge of some pixel coordinates indicating cell location. A single user *mouse* click in any part of a cell region corresponding to a coordinate per cell is required for only a few cells in the image as shown in Figure 3.7(a)) to train the association of cells with the appropriate label. A binary mask is then extracted,

as shown in Figure 3.7, of cell regions and background for each image by labelling all regions that have the same label as the known cell pixel points to $l = 1$ (foreground) and all other regions to $l = 0$ (background).

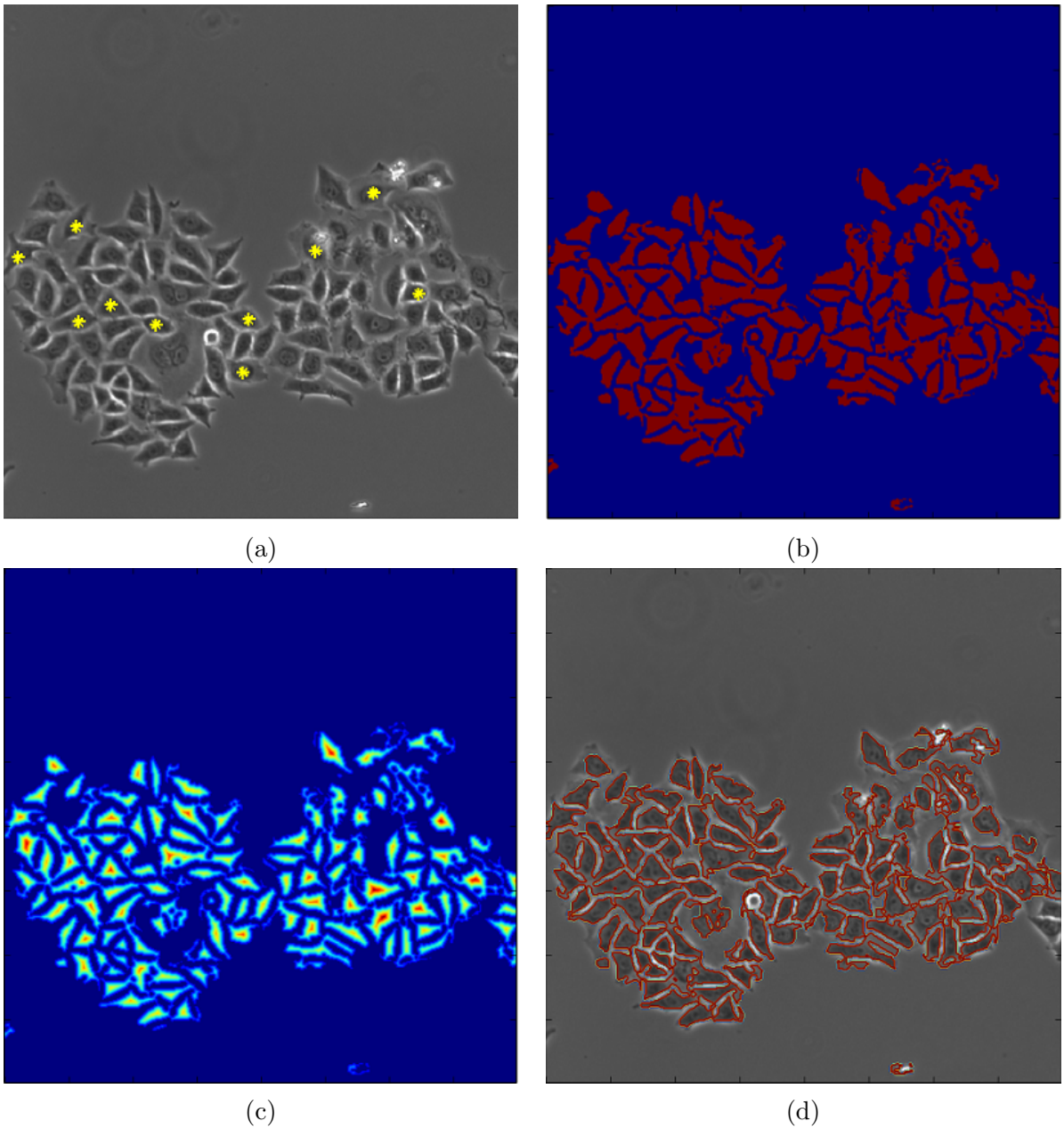
It is further observed that some cell regions (especially for cluttered cells) in the mask can merge along some length of their shared boundary. To split such regions, the Euclidean distance transform on the mask is computed such that the value of each pixel is replaced by its distance to the nearest background pixel. The distance transform on the mask is shown in Figure 3.7). Local peaks in the distance transform is determined and labelled as markers. A marker-controlled watershed routine is then applied to find the *watershed line* that separates adjoining regions.

3.3 Experiments and Results

Phase contrast images of cervical cancer cell colonies of the HeLa cells were used to evaluate the proposed algorithm. The cells contained in the data set vary in size and shape. Each image is 400×400 pixels in size and contains approximately a 100 cells each imaged at a magnification of 10X.

3.3.1 Parameter Selection

The various parameter settings adopted for the experiments run on the data are detailed here. In computing the phase retardation features for each pixel, the length of the dictionary of diffraction patterns is set as, $M = 20$ resulting in a 20-dimensional feature vector for each pixel. Also, set the number of superpixels, $K = 2500$ for each image with a corresponding high compactness, $c = 100$ to emphasize spatial compactness that ensures that the superpixels do not spread over boundaries in the image. For a 400×400 image with $N = 160000$ pixels, we obtain an average superpixel size, $N/K = 64$ pixels and grid size $Q = 8$. For multiclass clustering using GMM, $q = 4$ components representing one each for dark cell regions, bright cell regions, halos and background.



(a) (b)

(c) (d)

Figure 3.7: Final processing leading to final segmentation output: (a) *a priori* knowledge of cell location as dots (b) Cell region mask; (c) Euclidean distance transform map; (d) Segmentation contours overlaid on image

3.3.2 Evaluation

A quantitative assessment of the segmentation approach on data is obtained by computing the pixel-wise Jaccard Index (JI) introduced in [102] as:

$$JI = \frac{L \cap L_o}{L \cup L_o} \tag{3.5}$$

where L is the predicted label obtained from the final segmentation mask and L_o is the ground truth label obtained by manually segmenting all cells in all images used.

We achieve an average JI of 0.94 ± 0.03. Table 3.1 compares the result to that of similar work conducted by Arteta *et al.* [23] on the same dataset. Good performance is achieved despite the weak cell location annotation provided, with dots for only about 5% of cells in an image compared to 100% in [23].

Method	**Tanimoto Coefficient**
Our proposal	0.94 ± 0.03
Arteta *et al.* [23]	0.86 ± 0.02

Table 3.1: Quantitative comparison with similar work

The approach however is dependent on the majority of the labelled positions being correctly grouped into the cluster that corresponds to cells. In the event of poor clustering, there is a risk of poor delineation of boundaries when the majority of extracted labels from *a priori* locations in the multiclass clustering do not correspond to that cells regions. This is however unlikely to occur in our case unless the features are poorly extracted and the clustering fails completely.

3.4 Preconditioning for deep learning

CNN models have achieved state-of-the-art performance on a variety of object detection and segmentation applications in natural image datasets as well as biological sample images[34, 33, 99, 103]. Though computationally expensive in general, CNNs eliminate the need for hand-crafted features since the models learn relevant features from raw pixel

values along the layers of the deep multi-layer network. In their current configuration, CNNs for object detection and image classification take raw pixel values as input output appropriate labels based on the training set. The layers of CNNs can be viewed as sequential filters that transform the input image pixel values in a series of feedforward operations into a label vector. It has been shown that the filters at the layers are essentially edge, corner, colour and texture features [104].

This section explores the effect of using the RGB phase retardation feature representation as the input image to a CNN rather the raw grayscale pixel values. Transforming the grayscale phase contrast image into an RGB channels while preserving the structural components of the original image presents a richer representation of the original grayscale image. This amounts to preconditioning the raw pixel values as a preprocessing and data augmentation step before automatic features are learnt from a deep CNN. The aim is to obtain much richer activations at different layers of the deep network.

There exist a wide variety of deep CNN architectures. The most commonly known include AlexNet, GoogLeNet, VGGNet, and ResNet [104, 105, 99, 106]. Each architecture proposes a different configuration of layers to improve computational expense and improve accuracy. Here, the VGGNet16 architecture adopted because it is widely used and implemented on multiple platforms and programming languages. Figure 3.8 shows the layers of of VGGNet16 architecture.

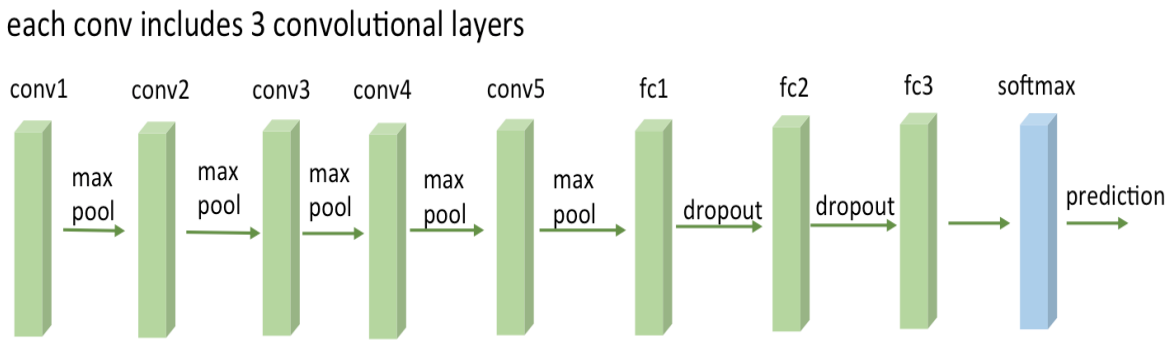


Figure 3.8: VGGNet16 network architecture. Image by Zhicheng Yan et al.[107]

3.4.1 Feature map activations

A pre-trained VGGNet model trained on the large repository of natural images on ImageNet is adopted here. In a pre-trained network, the parameter weights for all filters at each layer has been computed via stochastic gradient descent optimization. Training such deep networks requires a large set of labelled images as well as expensive CPU and GPU hardware. It is known that lower layers of networks trained on natural images tend to extract more lower level image features such as corners, edges and textures with later layers activating more abstract semantic features such as object parts. The pre-trained VGGNet16 used here is originally trained on natural images from ImageNet.

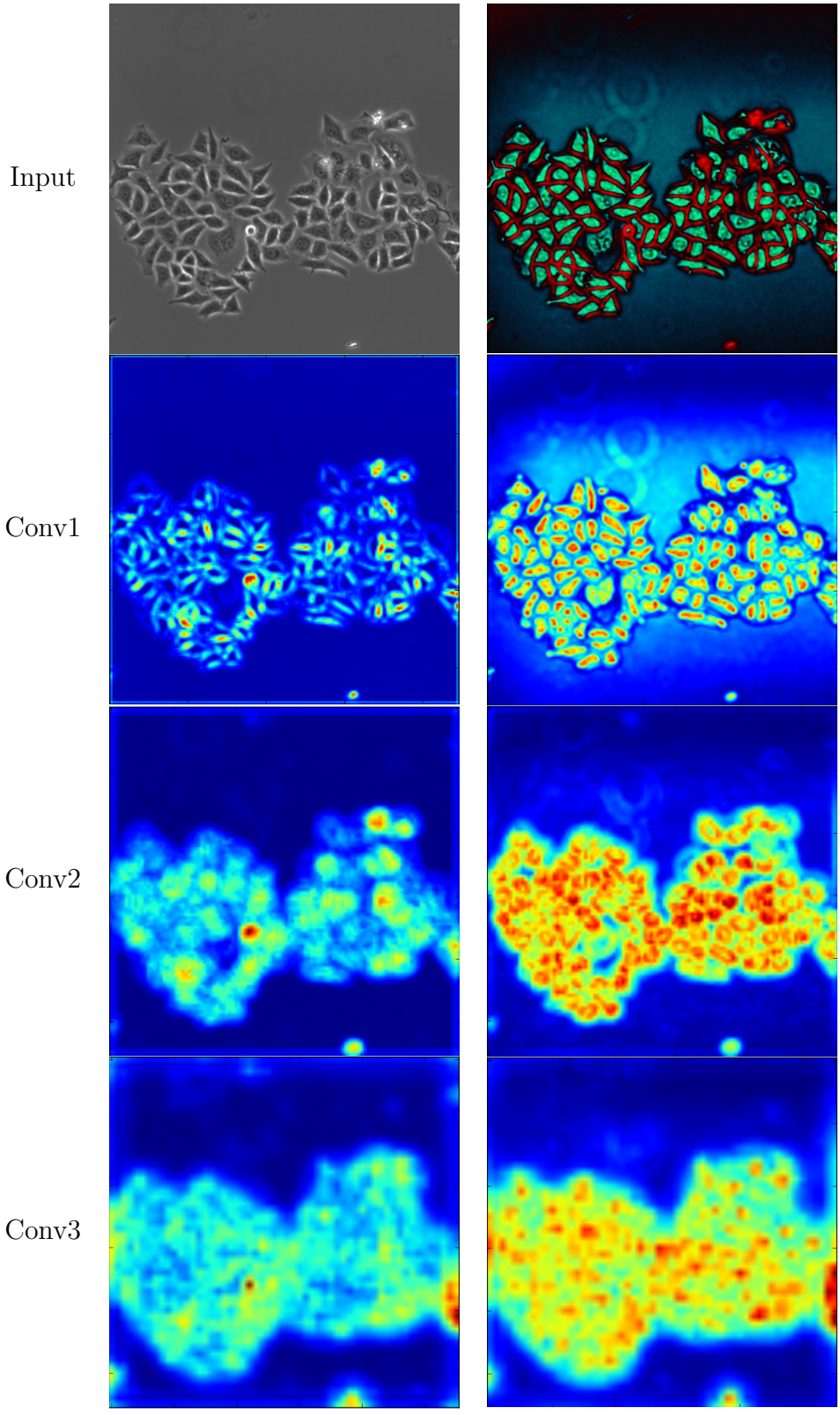
To explore the effect of preconditioning phase contrast microscopy images for deep learning, the average feature map activation after each set of convolution layer for grayscale phase contrast image and preconditioned phase retardation image are compared. Table 3.2 shows the avaerage feature maps after convolution layers of VGGNet16 for the grayscale and preconditioned RGB for a sample image.

From the feature maps shown in Table 3.2, it is observed that the average activations at each layer is stronger in the cell regions when the phase retardation image is used compared to the raw pixel values. The preconditioning into phase retardation representation therefore enhances the features learned by the CNN. The phase retardation feature modelling can therefore provide an efficient data augmentation step for deep learning architectures such as VGGNet16 as well as other new networks that would be constructed and specifically trained on phase contrast cell image data.

3.5 Conclusion

The work here demonstrated a method for cell segmentation by utilizing phase retardation features that makes use of weak and fewer annotations. While annotations are typically a very good way for improving algorithm performance, it is typically hard to obtain these in practice. Further work is needed to limit the number of parameter adjustments that are necessary to account for different imaging conditions, especially in case intrinsic values

Table 3.2: Feature activation maps of convolution layers



of the microscope are not known.

This method is used in the segmentation algorithm of other phase contrast data used in the work of this thesis. The only annotation we will require is to obtain the cell location at the time of initialization. But it would also be possible to obtain this information using some type of purpose build object detector. For tracking cells in any consecutive frame, the intrinsic location information from the tracking process could be exploited to increase the amount of prior location knowledge and hence improve the robustness of the overall segmentation. Cell state parameters based on the cellular morphology from such segmentation and tracking in time-lapse could then be obtained efficiently.

The chapter also explored the option of using phase retardation feature extraction as a data augmentation process to transform input raw greyscale images into RGB images that preserve the original structure of the input image. It was also shown that the phase retardation RGB image representation produced stronger activations at the first three convolution layers of a pre-trained deep CNN model VGGNet16.

Chapter 4

Boundary Estimation in Overlapping Cells

4.1 Introduction

Image-based measurements that capture shape and appearance patterns aid in quantifying phenotypic traits that capture the functional responses of the cells to chemical or environmental conditions. Using image-based computational tools as outlined in previous chapters, metrics that quantify cell health and function can be extracted from large datasets resulting from high throughput *in vitro* screens accurately and efficiently. Recent work demonstrates the potential to quantify cell state and function from nuclear and cellular morphology in fluorescence images [9][108].

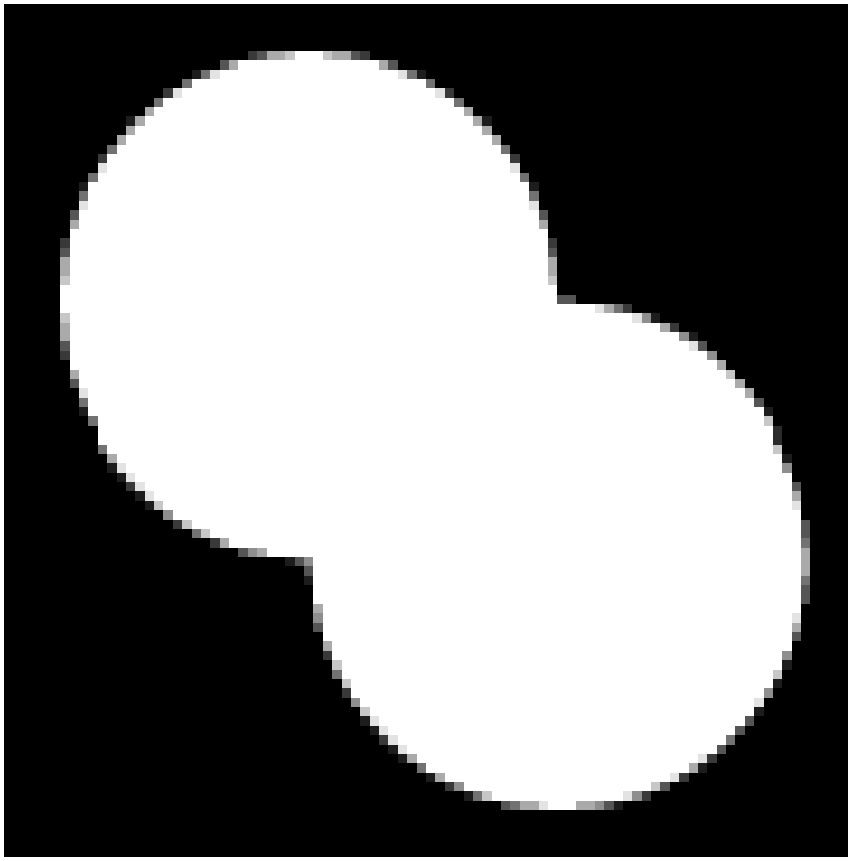
To achieve this at the single level, individual cells need to be segmented from the background. As reviewed in **Section** 2.2.1, segmentation methods that delineate the boundaries of each single cell in an image have been developed. For each segmented cell and corresponding nucleus, a dictionary of morphological and texture features that capture biologically relevant events can be computed. Extracting accurate morphometric information for each individual cell requires reliable detection and segmentation of object boundaries. With accurate segmentation, variations and trends in the population distribution of a given metric, such as size or texture, can be studied to enable the

quantification of cell morphology or the expression levels of molecular markers.

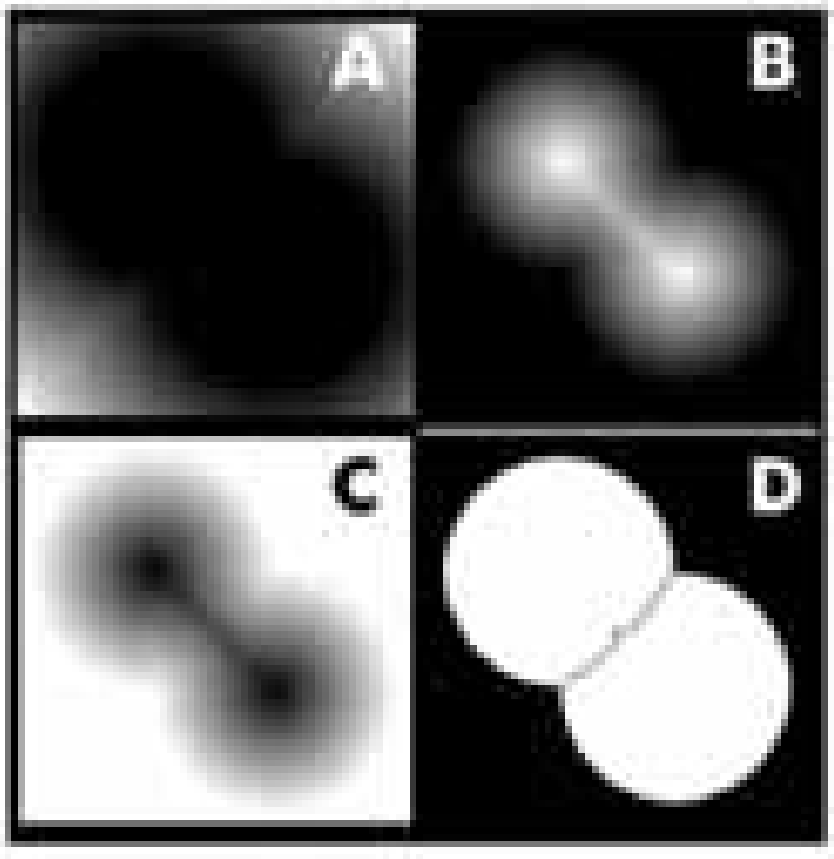
However, depending on the choice of cell line, cell condition or type of biological experiment, the resulting image presents cells crowding and showing varying levels of overlap and touching at the boundary. Separating occluded cells and touching boundaries is challenging in a conventional cell segmentation where accurate delineation of individual cell boundaries is required. In many segmentation tasks, separating touching or overlapping structures is achieved through a post-processing step using the watershed transform [109]. Watershed segmentation involves a series of mathematical morphological operations that results in an estimated boundary that separates the overlapping structures. Figure 4.1 shows a sample binary image and the corresponding operations that generated the estimated boundary. Though the conventional watershed segmentation approach has seen extensions and improvements, to incorporate prior structural knowledge [110] for instance, other approaches to detect overlap and estimate occluded boundaries have also been developed [111][112].

Considering the *true* boundaries are occluded and hence largely unknown, such boundary estimation is essentially an approximation to enable further analysis of cell morphology at the single cell level. In shape and appearance analysis following segmentation, there is an option to either ignore cells with overlapping regions. Another option is to in all regions including that obtained by boundary estimation in the analysis of shape and appearance. Both options present challenges to the accuracy of the downstream analysis as they could lead to erroneous distributions for the shape metrics used to describe cell population state or behaviour.

Ignoring occluded or overlapping structures eliminates the need to account for the approximation in estimating boundaries but could actually lead to errors in understanding the overall morphology metrics for a population of cells. For instance, in co-culturing of cells, at least two different types of cells are cultured together to evaluate cell-cell interactions as could be evident in a heterogenous cancer tumour [113]. In this case one cell type in such culture shows significantly more overlap and varied morphology than others as simulated in Figure 4.2(b) where only the larger cells cluster and overlap.

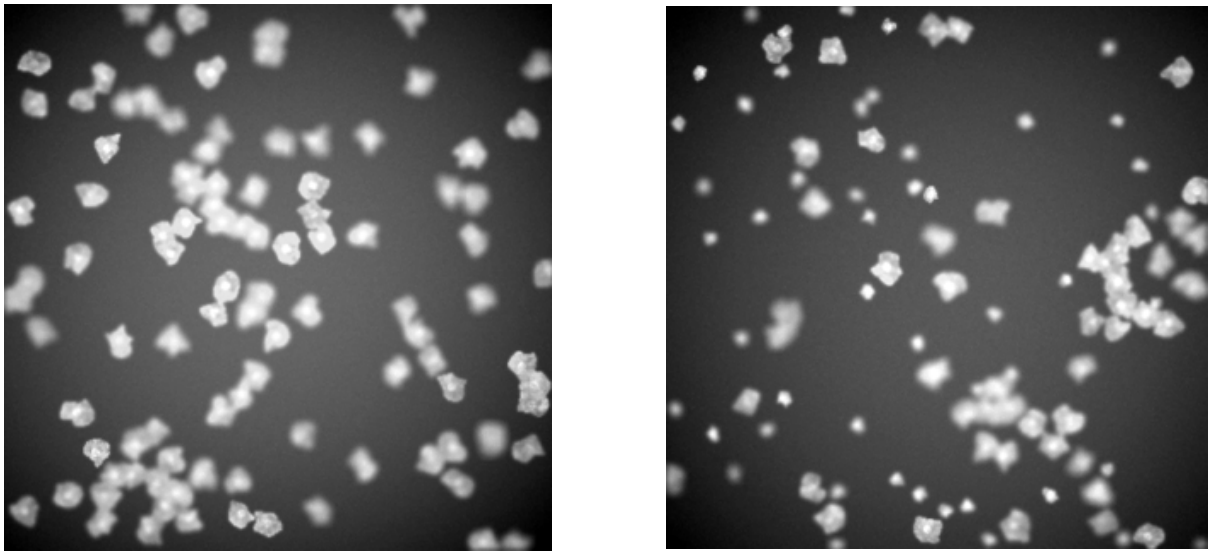


(a) Sample binary image



(b) Watershed transform

Figure 4.1: Sample binary image with two cells touching and occluding sections of their boundaries in (a) alongside transform operations A-D leading to the estimated boundary in (b). *Image courtesy: MathWorks*



(a) (b)

Figure 4.2: Sample simulated cell images showing (a) clustered and overlapping cells in a conventional mono-culture and (b) a co-culture of two cell types where the larger cell type shows clustering and overlapping whereas smaller cell type are evenly dispersed.

Ignoring all overlapping cells will amount to eliminating the contribution of that cell line in the overall population morphology statistic.

In a cell segmentation task, cells that do no overlap are segmented with high confidence whereas overlapping regions that require estimation methods are of low confidence due to the approximation required. Though low confidence regions are often included in the population statistics, their effect on the overall distribution of a metric or feature is often ignored. The work detailed here demonstrates how the different levels of confidence expressed in estimation of cell boundaries of overlapping cells can be incorporated into a population distribution that is suitable for phenotypic screening applications.

4.2 Method

This section provides details of the proposed approach aimed at demonstrating the effect of boundary estimation on morphometric population statistics. First, a conventional cell segmentation method, dynamic thresholding, is applied to segment cell regions from background. Dynamic thresholding is used here as this proves sufficient for the dataset used. A pipeline to detect and estimate overlapping boundaries and hence assign

confidence weights based on the level of estimation required is then developed. We then compute shape metric values for individual segmented cells and group the individual values into sub-populations based on the assigned confidence weight. For each group we estimate the distribution of different measurements using kernel density estimation. The various distributions are then combined into an overall weighted mixture distribution of the entire cell population. The mixture distribution is obtained in two ways: equally weighting the component distributions and variably weighting the components based on the segmentation confidence weights. We then assess the similarity of the variable weighted distribution and the equally weighted distribution from the underlining true distribution obtained from regions requiring no boundary approximation.

The approach can be summarised in the following three steps:

1. Detect/Separate the segmented regions into non-overlapping regions(requiring no estimation) and overlapping regions where boundaries have been approximated.

2. Compute the distribution of a structure/shape metric (e.g. area) for each region in each set.

3. Compute an overall mixture distribution for the metric from the individual distributions obtained from each set by weighting each distribution by a *confidence score*

By comparing the overall weighted mixture distribution to a reference distribution in a simulated dataset, it is demonstrated that accounting for the lower confidence in estimated boundaries in overlapping regions results in a distribution that approximates the true distribution better compared with treating all regions equally without weighting.

4.2.1 Datasets

Showing the impact of applying confidence level weights on the population distribution of cell features requires knowledge of the reference true distribution of the feature. Such reference distribution is obtained from an *ideal* dataset where no cells overlap and hence no estimation of occluded boundaries is required. Alongside the reference dataset is an

identical dataset where regions show varying extents of overlap. Here, the SIMCEP cell simulation tool developed by Lehmussola *et al.*[114] adapted to simulate datasets with the required overlap characteristics.

The SIMCEP tool offers a computational framework for simulating florescence cell images using a variety of user-defined parameters. Population level parameters control population properties such as the number of cells per image, extent of cell overlap allowed, and probability of cells to be in a cluster. Cell level parameters, on the other hand, control the structure and appearance of individual the cells in the image. Individual cell parameters include the size of a cell and its nucleus, cell texture, and presence of sub-cellular objects. Other parameters simulate the microscope camera effects.

4.2.1.1 Experimental datasets

For the experiments performed here, SIMCEP is used to generate two forms of simulated datasets; the mono-culture set and the co-culture set, samples which are shown in Figure 4.2(a) and (b) respectively. To generate the co-culture samples, the original SIMCEP was modified to allow *mixing* of two sets of cell structure morphologies in a single image such that only one cell type in the mixture is allowed to cluster with the other evenly dispersed. For each form, 6 datasets are generated each with similar cell density and morphology but with different number of overlapping cells. Each dataset consists of 10 images with 100 cells in each image. The datasets vary in the percentage of overlapping cells present, with each having either 30%, 40%, 50%, 60%, 70% or 80% of overlapping cells. As cells are placed randomly in each image, the variety of cell clusters overlapping in different ways is obtained.

4.2.1.2 Reference datasets

In addition to the datasets described above, reference datasets are created for both mono culture and co-culture cases. The reference dataset is similar to the other datasets except there are no overlaps between cells. All cells are dispersed evenly within the image. The reference datasets provide the *ground truth* distribution of the cell features that is used

as a reference in assessing the impact of estimating the boundaries in overlapping cells.

4.2.2 Segmentation and boundary estimation

In most microscopy modalities, cells show as bright or dark blobs in a particular intensity channel. In most fluorescence imaging, the background is usually uniform and there exist a good contrast between cell regions and the background. The simulated cell images in the datasets show similar good contrast. Cells are segmented from the background using dynamic thresholding where the threshold value is the weighted mean of the local neighbourhood of a pixel. Here, a gaussian filter is applied for weighting the neighbourhood pixels.

Considering the crowding and overlap of cells, some multiple cells often remain segmented as individual cells. Using such delineated region of such undersegmented cells to compute shape metrics could lead to erroneous morphometric distribution for a population of cells. To address this, we obtain the ground truth of all nuclei instances from the nuclear channel. For each segmented cell region, detecting two or more nuclei instances indicates an undersegmentation that has resulted in multiple cells segmented as one cell.

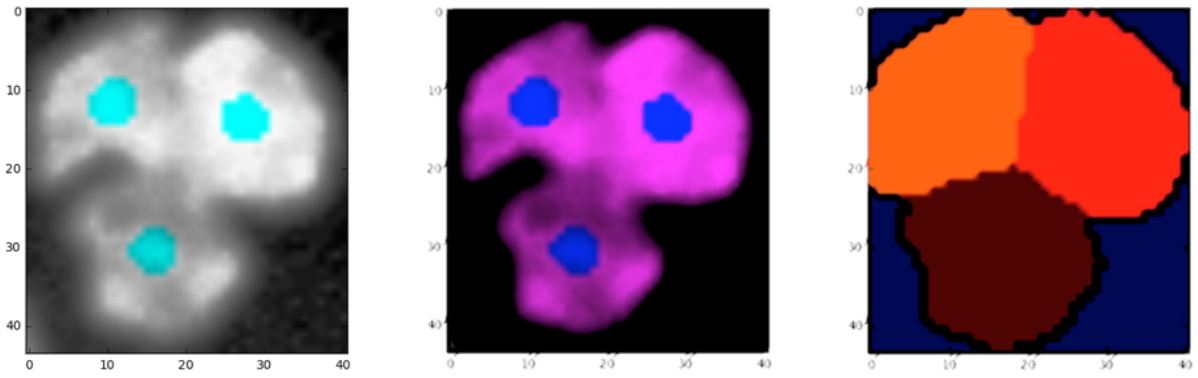


Figure 4.3: Separating overlapped cells in (a) using Voronoi diagram over seed nuclei detections in (b) to obtain the estimated boundaries shown in (c).

As the boundaries are occluded, a simple tessellation approach via a general Voronoi diagram is adopted to sub-divide the space of the initial segmentation into the number of nuclei instances within the region as shown in Figure 4.3. The tessellation is seeded

by the nuclei centres. This ensures that each new estimated region contains only one nucleus. Since the estimated boundaries may not be *true* cell boundaries, a weighting value is assigned to the region to distinguish our confidence in such segmentation from boundaries obtained without tessellation. A more sophisticated method as in [111] may be required for boundary separation in other datasets depending on the cell line and specific requirements of the segmentation task.

4.2.3 Feature density estimation and weighting

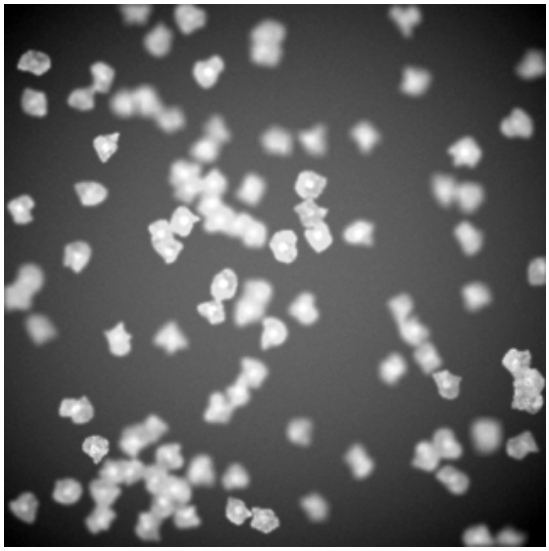
To phenotypically assess cell state and function, a variety of morphometric measurements are computed for each individual cell region. For the purposes for our analysis here, we adopt three of the most common of these metrics: *eccentricity*, *area* and *length of the major axis*. Eccentricity of a region is computed as the ratio of its minor to major axis length resulting in a value is between 0 and 1.

To compare each metric across different cell populations, we compute a distribution of the metric from the individual measurements of the the cells in the population. We obtain the distribution, f of the individual cell metrics $(x_1, x_2, ..., x_n)$ by computing the kernel density estimate as:

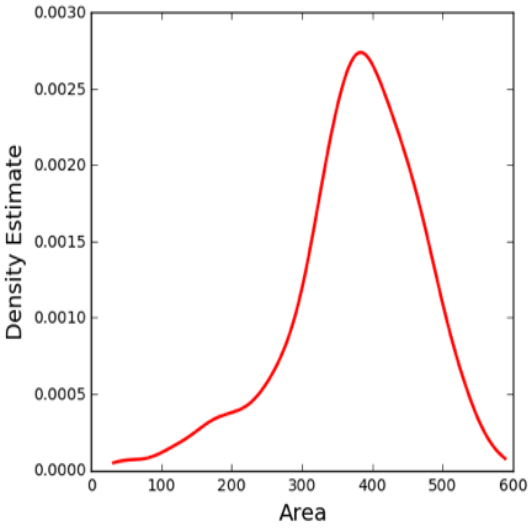
$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{4.1}$$

where n is the number of samples, the kernel K is a Gaussian and h is the bandwidth selected automatically using the Silverman rule [115].

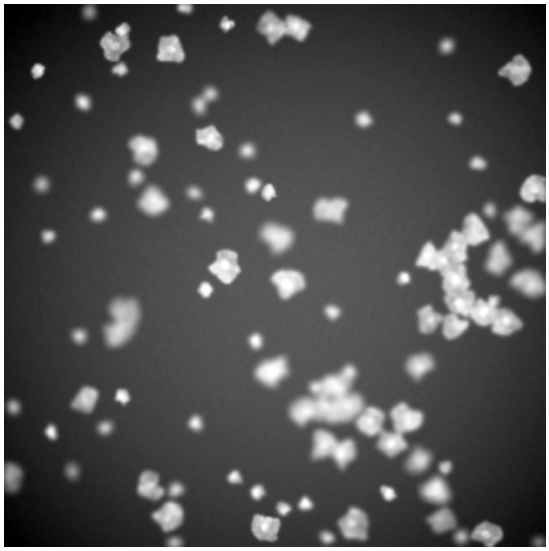
Figure 4.4 shows the density estimate for the reference mono-culture and co-culture datasets. It is observed that the area distribution for the mono-cultured (as in Figure 4.4b), cells is unimodal with one peak representing the average area whereas that of the co-cultured cells (Figure 4.4b) are bimodal, with two peaks representing the two average area values. This is consistent with what is expected given the nature of cells in the image, mono-culture is largely from only on cell type and co-culture is of two cell types of different sizes.



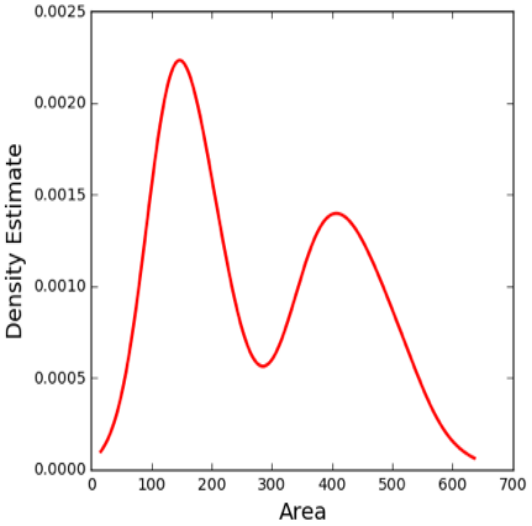
(a) Simulated mono-culture image



(b) Density estimate of area in co-culture



(c) Simulated co-culture image



(d) Density estimate of area in co-culture

Figure 4.4: Kernel density estimates for area shown in (b) and (d) of the reference datasets for mono-culture cells in (a) and co-culture cells in (c) respectively.

4.2.4 Segmentation confidence and weight assignment

The density estimate in Equation 4.1 is computed for sub-populations of cells obtained by grouping cells according to the extent of boundary estimation required to recover each individual. Such grouping relates to cells which can be segmented as individual cells without further processing, cells that only overlap with one other, and cells that overlap with multiple other cells requiring extensive processing to demarcate their borders from others. In this approach, a distribution is obtained each for cells requiring no split, cells that need to be separated from another, cells that need to be split from three or more other cells in a cluster, and so on. Hence, a density estimate for each metric for every segmentation confidence weight group is obtained. The densities $(f_1, f_2, ..., f_m)$ corresponding to each of the confidence groups with corresponding weights $(w_1, w_2, ...w_m)$ is combined into a single weighted mixture density for a given cell feature as:

$$F(x) = \sum_{j=1}^{m} w_j f_j(x) \tag{4.2}$$

where $w_j > 0$. In this way, the overall distribution $F(x)$ of a given metric, say area, in a cell population is the finite mixture of the individual metric distributions, f_j of the sub populations with assigned weights, w_j. The weights, w_j are obtained from the probability density function of a Gaussian distribution such that as the number of cells that is required to be split increases the weight assigned decrease exponentially along the normal density function. Figure 4.5 shows the plot of the function that determines the weight assigned to each confidence group depending on how many cells had to split to recover individual cell regions.

4.2.5 Assessing the impact of weighting

To explore the impact of estimating a weighted distribution, we also compute a distribution obtained from the same metric data without weighting. In this distribution, all measurements are treated equally without considering the confidence in the boundary estimation at the segmentation step. By computing the Kullback-Leibler (KL) divergence

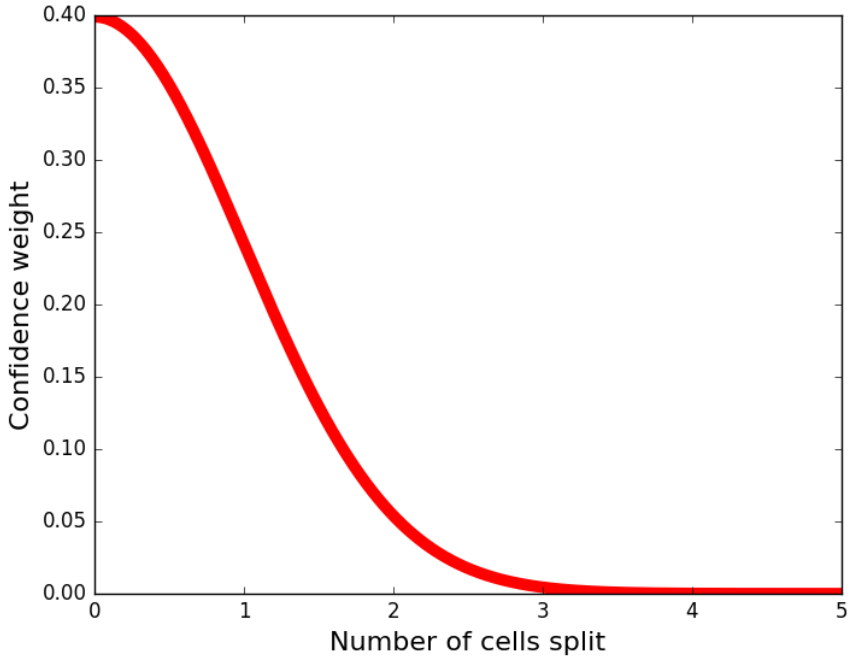


Figure 4.5: Confidence weights

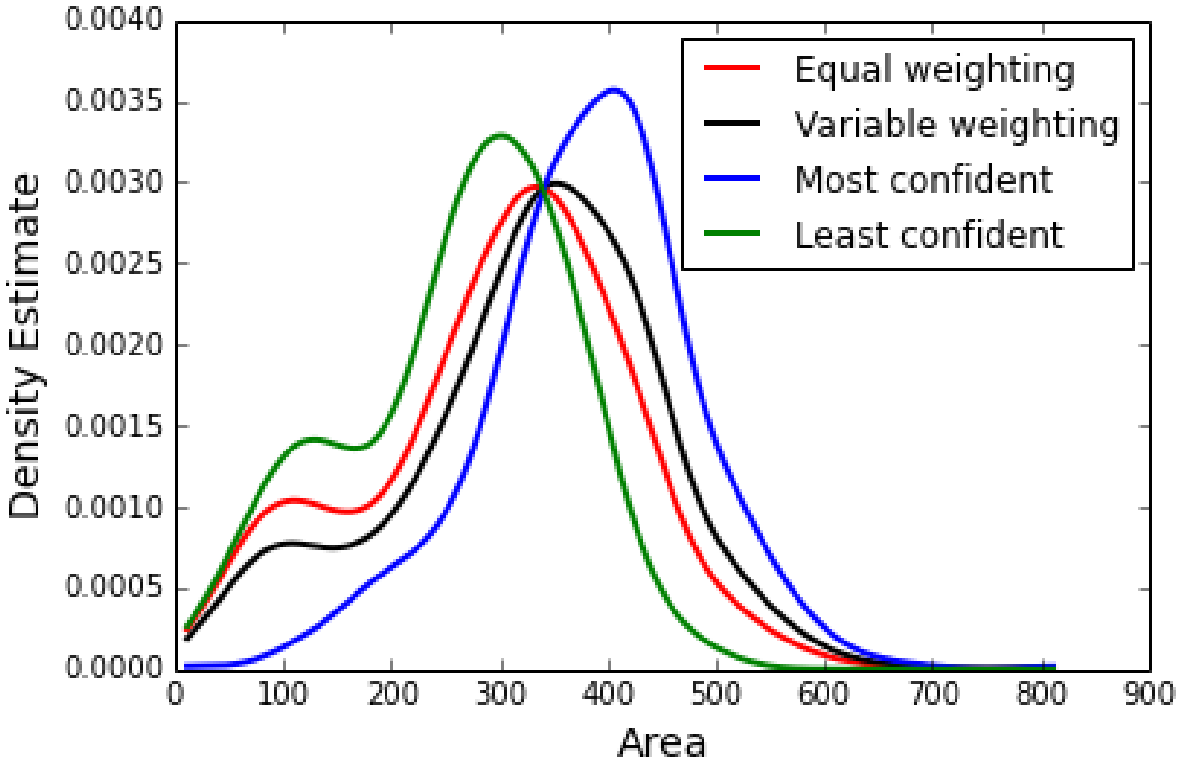
from the overall distribution obtained to that of the most confident segmentation, we quantify the information lost by approximating the *true* metric distribution with another that includes instances with boundary estimation. The higher the KL divergence, the higher the error incurred by the distribution used to approximate the true underlining distribution. We compute KL divergence, D_{KL} from a discrete distribution P to another Q as:

$$D_{KL}(P\|Q) = \sum_i P(i) ln \frac{P(i)}{Q(i)} \tag{4.3}$$

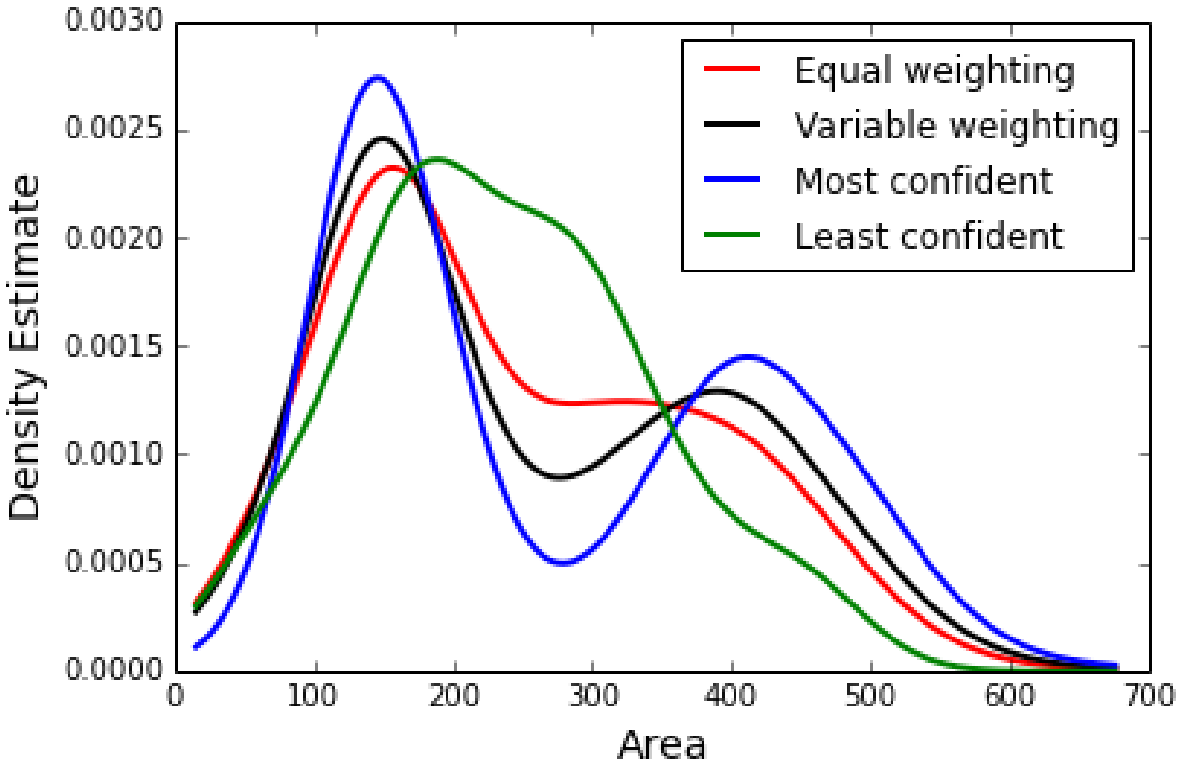
where $P(i)$ represents our most confident distribution whereas $Q(i)$ is the approximate equally or variable weighted distribution.

4.3 Results

First, we compute a reference distribution, f_o for each metric from a dataset where cells have no overlap. As we compute the reference distribution from a set of non-overlapping



(a) Mono-culture cells



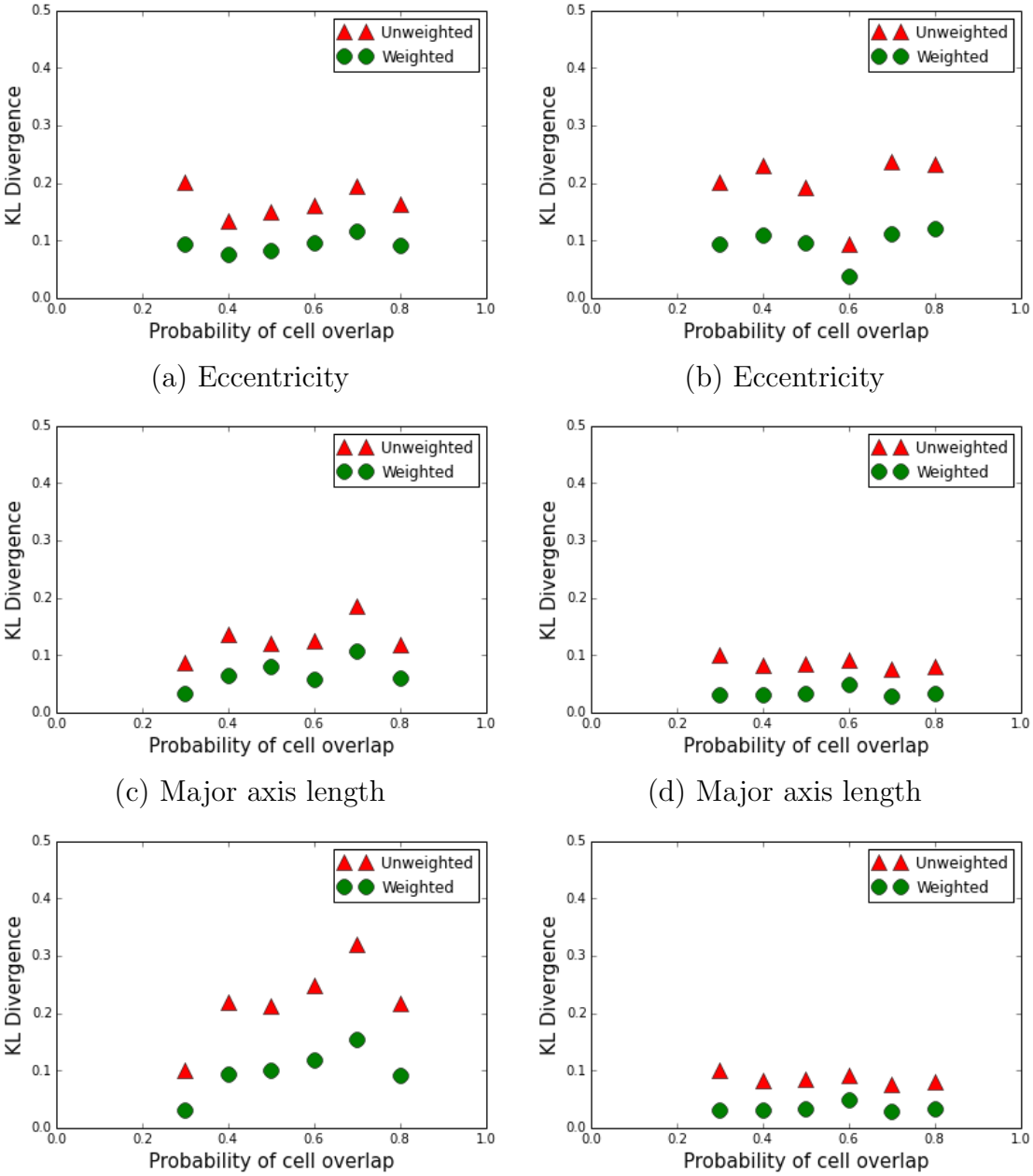
(b) Co-culture cells

Figure 4.6: Sample distributions of area metric showing components obtained from the least confident and most confident segmentations as well as the mixture distributions

cells and then only modify the relative overlap of this set of cells, it can be viewed as the true distribution for the given metric. Then for each dataset (having a given fraction of overlapping cells), we compute two mixture distributions, F_e and F_v as in (4.2) with equal and variable weights respectively. For each metric, we evaluate the effectiveness of our approach by comparing the the KL divergence between both the, F_e and F_v, and the reference distribution f_o. By comparing the KL divergences which are presented in Figure 4.7, we observe a lower divergence when the confidence weights are applied and the components in the distribution are weighted by the confidence in the segmentation. This indicates that applying the confidence weights results in a distribution that deviates less from the true distribution and is more stable as the fraction of overlapping cells changes.

4.4 Conclusion

This chapter proposed a framework to incorporate approximations in segmentation of cell regions in morphometric analysis for quantitative phenotypic screening assessment. The variable weighting scheme introduced here shows how the amount of overlapping cells in a cell population affects the population distribution and the extent to which the proposed weighting scheme could be applied. By incorporating the confidence level weights, an improvement in the estimation of the true underlying distribution is achieved. This approach is not limited to estimation of overlapping boundaries only, but also methods that result in different confidence levels in segmentation. For instance, probabilistic models for segmentation provide uncertainty values for each segmented area that express the confidence of the model in the accuracy of the segmented region. Such probability values could be used as the weights in obtaining the overall distribution for a given cell feature.



(a) Eccentricity

(b) Eccentricity

(c) Major axis length

(d) Major axis length

(e) Area

(f) Major axis length

Figure 4.7: Results showing the KL divergence from the true distribution of the equally weighted and confidence weighted for different fraction of overlapping cells. (a),(c), (e) and (b),(d),(f) shows the divergence values for mono-cultured and co-culture scenarios respectively. It is observed that in both cases the distance measured by KL of the unweighted from the true distribution is higher compared to the weighted distribution

Chapter 5

CMCFtrack: Single cell tracking

5.1 Introduction

Cell tracking is essential in understanding the temporal dynamics of cell behaviour in time-lapse sequences. The ability to follow the progress of individual cells through time from frame to frame allows a more detailed study of transient cell events and associated phenotypes. Section 2.2.2 provided a review of a variety of cell tracking methods in recent literature. This chapter however details the implementation of an extensible modular implementation of the coupled minimum-cost flow approach originally proposed by Padfileld *et al.*[116, 16] with an easy-to-use interface in the high-level Python language. The implementation was developed in collaboration with Alan Aberdeen, a fourth year engineering undergraduate who contributed to this as part of his fourth year final engineering project. Alan's internal technical report, available at the Department of Engineering Science, University of Oxford provides details of his contribution [117]. This open-source implementation, CMCFtrack, provides a base code that can be extended, modified and redistributed to improve cell tracking in video analysis of live cell imaging experiments.

Compared to conventional object tracking in natural images, cell tracking accuracy is generally challenged by densely-packed cell populations, low temporal resolution, cells leaving or re-entering frame, and cell events such as onset of division[118, 119, 120, 121].

The coupled minimum-cost flow (CMCF) tracking is based on the frame-to-frame object association paradigm where segmented objects in each frame are associated with objects in consecutive frames to form a track. CMCF tracking provides routines for peculiar cellular events to obtain an overall tracking framework that works well given a good segmentation of individual cells. Section 5.2 describes the CMCF model followed by the software implementation and usage details in Section 5.3.

5.2 Coupled minimum-cost flow tracking

A number of cell behaviours can be observed when tracking a cell in pairs of consecutive frames of a video or time lapse sequence. The cell may move, divide/split or multiple cells appear to merge. Also, cells near the frame border may also disappear in the next frame or cells not seen in a frame may appear in the next frame [41, 122]. The aim of object association tracking of cells is to provide an efficient framework to link individual objects in consecutive frames in a mapping that describes an event such as move, division, appear or disappear. Cells sometimes occlude each others boundaries in consecutive frames and hence seem to merge. This requires a *merge* event also to be accounted for. Linking objects in consecutive frames results in a track for each cell. The approach by Padfield *et al.* [16] formulates the object association problem in a graph framework that incorporates all the cell behaviours into a *flow network*.

A flow network is a directed graph. A directed graph, also referred to as a network, is composed of *vertices* connected together by *edges*. In a flow network, each edge has a capacity and each edge receives a flow. The amount of flow on an edge cannot exceed the capacity of the edge. A flow must satisfy the restriction that the amount of flow into a node equals the amount of flow out of it, unless it is a source, which has only outgoing flow, or sink, which has only incoming flow [123, 124]. Flow networks are used in modelling for a variety of applications including cardiovascular medicine and research, electrical systems, road traffic, fluids in pipes, and very widely in operations research to optimise systems [125, 126, 127, 128]. Details of the flow network as applied to cell

tracking here is provided later in Section 5.2.1.

To apply an object association tracking model, a segmentation of all cells in all frames should have been obtained [41][46][47][16][48][49][129]. The tracking approach is independent of the segmentation method used but its effectiveness depends largely on the accuracy of the segmentation obtained. This therefore allows the choice of segmentation to be any appropriate method that works best for the frames under consideration. For each segmented cell, features describing the position and morphological structure are computed. Each cell in a given frame is compared to other cells in the next frame by computing a compatibility score between their features. Details of features and how they are incorporated in the flow network is provided in Section 5.3

5.2.1 The flow network

Tracking a cell involves matching objects in pairs of consecutive frames. A bipartite graph is an appropriate representation for such objects that are to be matched between two adjacent sets [130, 131, 132, 133]. A bipartite graph $G = (V, E)$ is a graph whose vertices can be divided into two disjoint sets L and R such that every edge E connects a vertex in L to a vertex in R. Hence, a bipartite graph is a graph that does not contain any odd-length cycles[134]. In the specific case of cell tracking, the L vertices represent the cells in the previous frame, the R vertices represent the cells in the current frame, and the edges connecting the vertices represent the matching of cells across frames. This representation makes the bipartite graph the basic unit of the *flow network* that encodes the cell behaviours that make a track of a cell. Figure 5.1 shows a bipartite graph with two vertices/objects in the L set and three in the R set with connected edges.

Considering the bipartite graph shown, a matching, M between vertices L and R is the subset of all the edges, hence $M \subset E$. A vertex $v \in V$ is said to be matched by the matching M if some edge in M is incident on v, otherwise v is unmatched. A perfect matching is obtained when all vertices in the graph are matched. In the case of cell tracking, the aim is a perfect matching, otherwise unmatched vertices suggest cells that were unaccounted for. To match a cell across consecutive frames, edge weights can

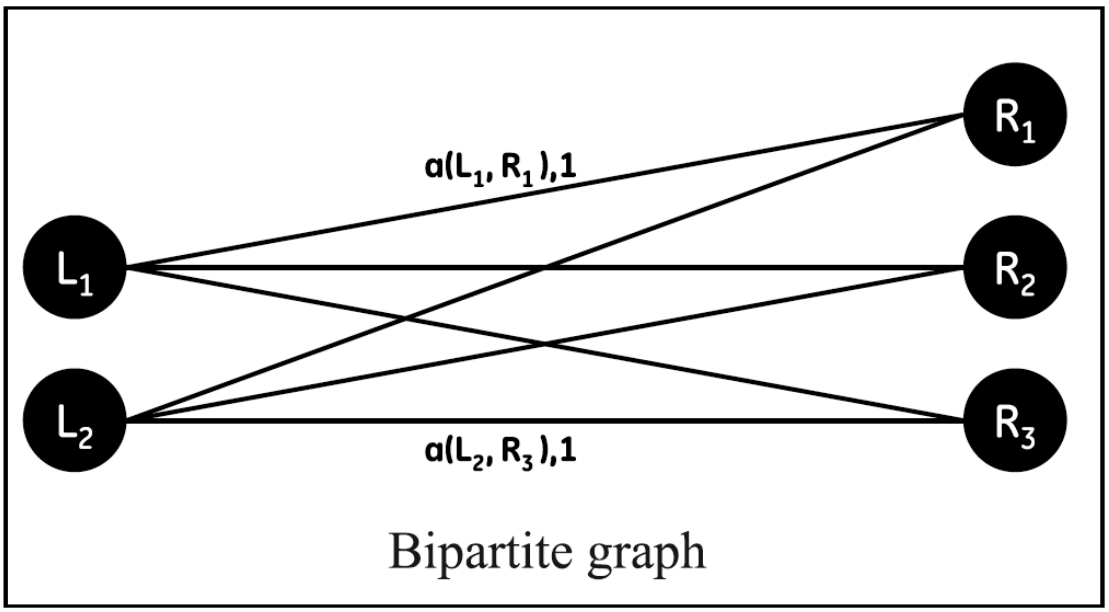


Figure 5.1: Simple bipartite graph. The black vertices on the left represent the cells on the previous frame, and the black vertices on the right represent the cells on the current frame. The left vertices are fully connected to the right vertices. Image courtesy: [16]

be defined for each pair of cells based on a computed metric such as distance between the cells or some structural similarity. In such instance, cells that are far from each other and show minor similarity across two frames will be less likely to be matched. This results in a weighted bipartite graph where each edge has an associated cost. A minimum cost matching is then the matching where the sum of the edge costs is minimum. The *assignment problem* is finding such a matching.

The assignment problem can then be modelled as a minimum cost flow problem [123]. This can be achieved by converting the the bipartite matching problem to a flow network by letting $G = (V, E)$ be a finite directed graph where each edge $(u, v) \in E$ has a non-negative, real-valued capacity $c(u, v) \geq 0$, and introducing a source node $(T+)$ and a sink node $(T-)$ as shown in Figure 5.2. The source node $T+$ has edges to all vertices in L and the sink node has edges to all vertices in R such that for any vertex $v \in V$ there is some path from source to sink, $T+ \rightarrow v \rightarrow T-$. Formulating the cell tracking problem as a minimum cost flow problem was also proposed by Al-Kofahi *et al.*[135]. Zhang *et al.* had also applied a similar approach for natural images[129]. However, Padfield and

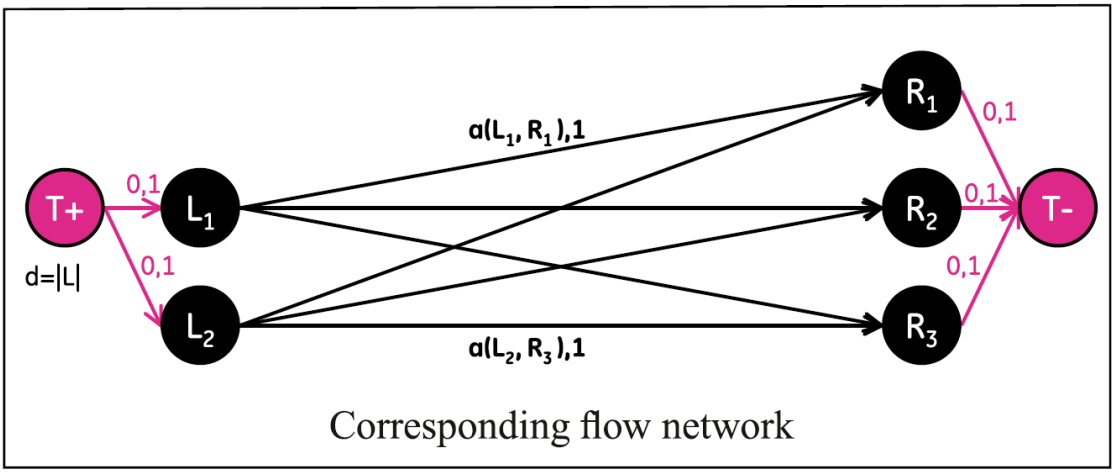


Figure 5.2: The flow network showing the the source ($T+$) and target ($T-$) vertices added to convert the bipartite graph to a minimum-cost flow problem. Image courtesy: [16]

colleagues introduced the coupled minimum cost flow formulation of the problem. Details of significance of the coupling are discussed later in Section 5.3.2.2.

To formalise the definition of the flow network, let each edge (u, v) have a capacity $c(u, v)$ and a real-valued cost $a(u, v)$. if we send $f(u, v)$ units of flow over the edge (u, v), a cost of $a(u, v)f(u, v)$ is incurred. For a minimum cost flow problem, the goal is to send d units of flow from $T+$ to $T-$ such that the the total cost is minimised. This presents an optimisation problem subject to constraints. *Capacity constraint*: ensures the total flow, f on an edge cannot exceed the capacity, c of that edge. *Skew symmetry*: imposes the net flow from u to v must be the opposite of the net flow from v to u. *Flow conservation*: ensures that the amount of flow into a vertex equals the amount of flow out of it, with the exception of the source and sink nodes. *Flow requirement*: dictates that the overall flow if from the source, $(T+)$ to the sink, $(T-)$. The minimum-cost flow problem can then be formalised as:

$$
\begin{array}{rll}
\text{minimise the flow cost:} & \sum_{(u,v)\in E} a(u,v)f(u,v) & \\
\text{subject to:} & & \\
\text{Capacity constraint:} & f(u,v) \leq c(u,v) & \forall u, v \in V \\
\text{Skew symmetry:} & f(u,v) = -f(v,u) & \forall u, v \in V \\
\text{Flow conservation:} & \sum_{v\in V} f(u,v) = 0 & \forall u, v \in V - \{T+, T-\} \\
\text{Flow requirement} & \sum_{v\in V} f(T+,v) = d &
\end{array}
$$

The minimum-cost flow problem is hence an optimisation of a linear function subject to linear controls and can therefore be solved by linear programming [136, 137, 138]. The solution of the optimisation problem is presented in detail later in Section 5.3.2.3.

5.2.2 Extending the graph for modelling cell behaviours

Imaging live cell experiments at sufficiently high temporal resolution to ensure transient morphological events are captured is challenging. There is always a high risk of damage to the cells or interfering with the normal behaviour of cells by the heat and light from the imaging system [139, 140, 141]. To minimise such effects, cells are imaged at lower temporal resolutions compared to traditional videos. For very motile cells, this results in cells seeming to *jump* from frame-to-frame. Cell divisions also tend to be sudden across frames. Also, the field of view of microscopes at a sufficient magnification to observe cells is usually smaller than the well in which the experiments are performed. Cells can therefore move into or out of the field of view creating the phenomenon of cells appearing and disappearing across frames. There is also the merging of cells close to each other from one frame to the next when one cell partially occludes the other. All the cases mentioned above can not be accurately modelled by the flow network described in Section 5.2.1 as each case will result in unmatched vertices in the network. To account for such cases, additional vertices and edges are added to the flow network.

5.2.2.1 Appearing and disappearing cells

To account for appearing and disappearing cells, two additional vertices, A and D for appearance and disappearance are added to the network as shown in Figure 5.3. The additional edges to connect the new vertices to existing vertices are also introduced. Edges from A to all vertices in R encode the case where all objects in the current frame appeared. Similarly, edges from all objects in L to D encode the case where all objects from the previous frame disappear in the current frame. For conservation of flow, an edge from A to the source, $T+$ and an edge from D to the sink, $T-$ are included. The introduction of A and D requires an adjustment to the flow requirement to ensure that

each vertex is accounted for with exactly one edge entering and one edge leaving. The new required flow is hence, $d = |L| + |R|$, where $|\cdot|$ is the cardinality of the set.

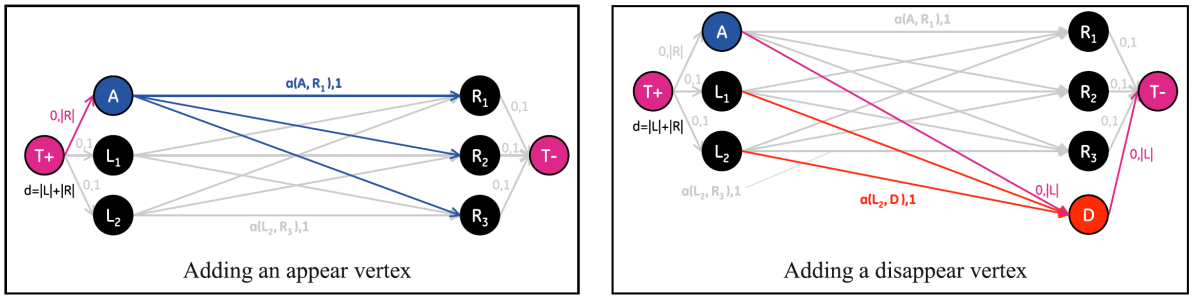


Figure 5.3: Flow network with additional vertices A and D for appearing and disappearing cells. Image courtesy: [16]

5.2.2.2 Splitting and merging cells

Cells naturally undergo mitosis; a process of division of the cell into two new daughter cells. When a cell splits into two between two consecutive frames, an additional object is introduced in the current frame that needs to be accounted for in the network. Also, cells grow in a three-dimensional environment but are often imaged in two dimensions. Cells could occlude portions of others in the image plane creating the an observation as a merge of cells. Merging can also be caused by a failure in segmentation where multiple close cells are detected as one cell object. In a similar approach to dealing with appearing and disappearing cells, a new vertices, S and M are introduced in the flow network to address such cases as shown in Figure 5.4.

Unlike the case of appearing and disappearing cells where adding new vertices connected to source and sink accounts for all cells, new split/merge vertices would not have the same number of entering and leaving edges, hence violating the flow conservation constraint. However, by including an edge from an appear vertex, A to each of the splitting vertices, S the inequality of the edges entering and leaving a split vertex can be fixed. To obtain a consistent feasible cell behaviour model, a constraint is required to ensure that the flow units through split/merge vertices is exactly zero when the vertex is not part of the graph solution for the network or exactly 2 when part of the solution.

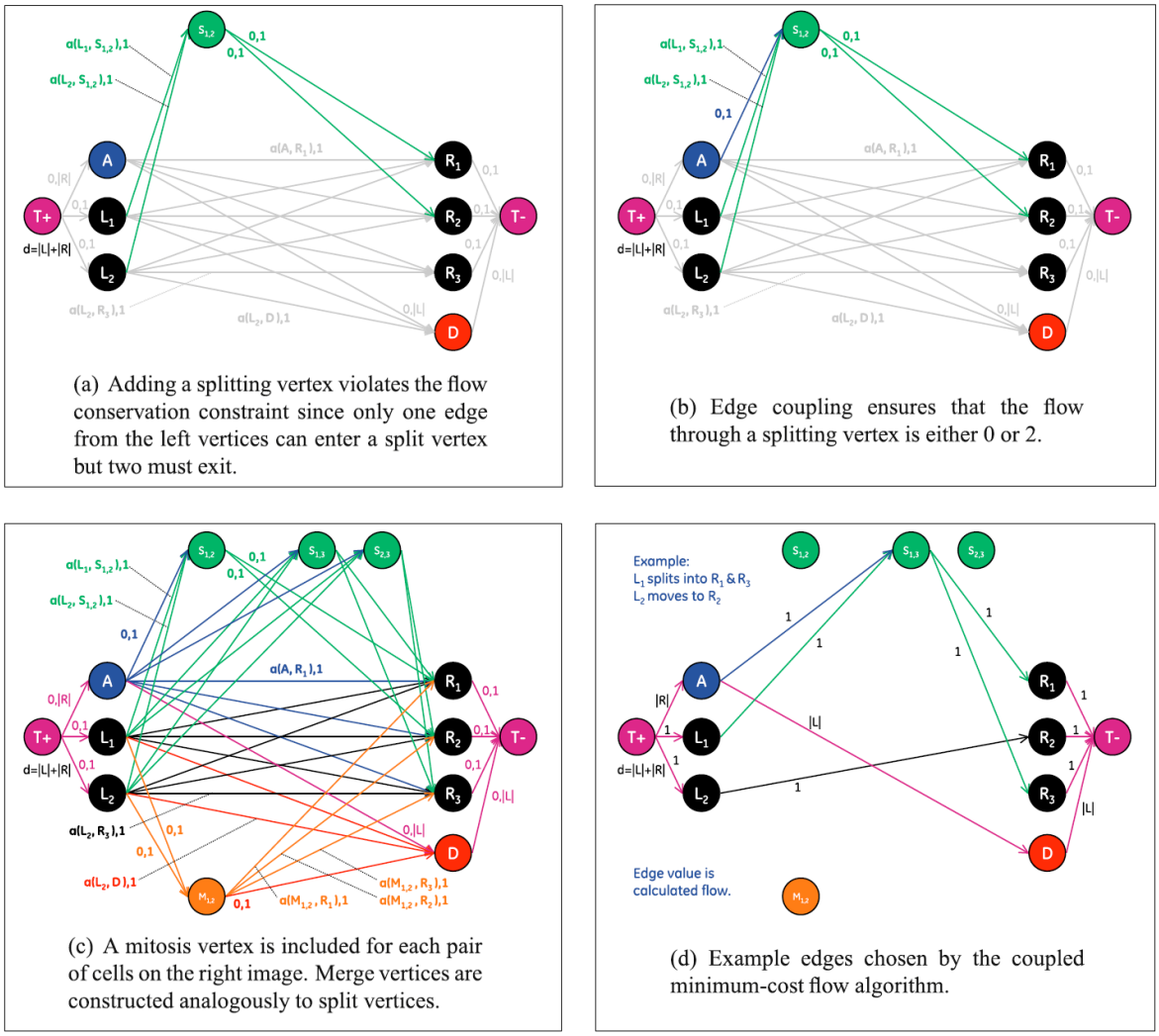


Figure 5.4: Flow network with additional vertices M and S split/merge cells.

To enforce this constraint, the concept of *edge coupling* is applied. With edge coupling, if an edge from an L vertex to a split vertex S is chosen then the corresponding edge from the appear vertex, A to the same split vertex must also be chosen. Edge coupling is complimentary in the case of merge events. For merge, if an edge from a right vertex R to the merge vertex is selected, then the coincident edge from the the merge vertex to the disappear vertex is also included in the solution. By edge coupling the minimum-cost flow network, the couple minimum-cost flow algorithm is obtained. Figure 5.4(d) shows an example matching network.

5.3 Software implementation

CMCFtrack is implemented in the popular open source cross-platform Python language and makes use of other open source libraries and packages widely-used in the scientific computing community. Python is a dynamic programming language that offers an extensive variety of data structures and flexibility that adapts well to the aim of delivering an extensible modular-designed implementation. As an open source project, the code is hosted on the online code-sharing platform GitHub. The base code is hosted on GitHub here: `https://github.com/alanaberdeen/coupled-minimum-cost-flow-track.git` and `https://github.com/akwesinketia/coupled-minimum-cost-flow-track`. GitHub is a web-based repository hosting service that offers distributed revision control and source code management features. The platform allows code to be shared and modified by multiple users working collaboratively while maintaining an efficient change log of approved versions.

Section 5.3.1 provides details of the Python data structures used. The optimisation engine and framework for solving the assignment problem is presented in Section 5.3.2 and installation and usage guidelines are outlined in Section 5.3.4. CMCFtrack relies on the base packages for scientific computing including `Numpy`, `SciPy` and `Matplotlib` [142, 143, 144].

5.3.1 Data structures

The flow network described in Section 5.2 is primarily vertices representing cells and weighted edges connecting the vertices formulating the specific assignment problem under consideration. The implementation requires appropriate data structures in Python that handles the vertices, edges and weights effectively. The graph structure for the flow network is built with the `NetworkX` package that provides a flexible framework to represent vertices by any hashable Python object such as `lists` and `dictionaries` [145].

5.3.1.1 Vertices

As indicated earlier, each vertex represents a cell and its features whereas the edges describe the association of the cells between frames. A vertex consists of all the computed features describing the cell. All cells have the same set of computed features. Features used to compute similarity between cells are usually morphology-based and can be computed from the binary mask image following segmentation. From the mask, pixel intensity and texture features for each cell can be extracted as well. For example, each cell will have a centroid, area, average intensity and diameter. In principle, the tracking algorithm does not dictate the features used, hence allowing any choice of appropriate morphology features to be used [146].

Here, a vertex is represented by the `dictionary` data structure. A `dictionary` in Python is an unordered set of {`key:value` } pairs where the keys must be unique identifiers within a dictionary. The dictionary values could hold any valid Python data type or object representing cell features. The features used in this implementation are computed from segmented frames using the `scikit-image` package [147]. Figure 5.5 shows the structure of a vertex representing the features of one cell, L_n.

5.3.1.2 Edges and costs

`NetworkX` is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. The vertices, structured as dictionaries, from adjacent frames are linked by edges in a `networkx` graph. Each edge is associated

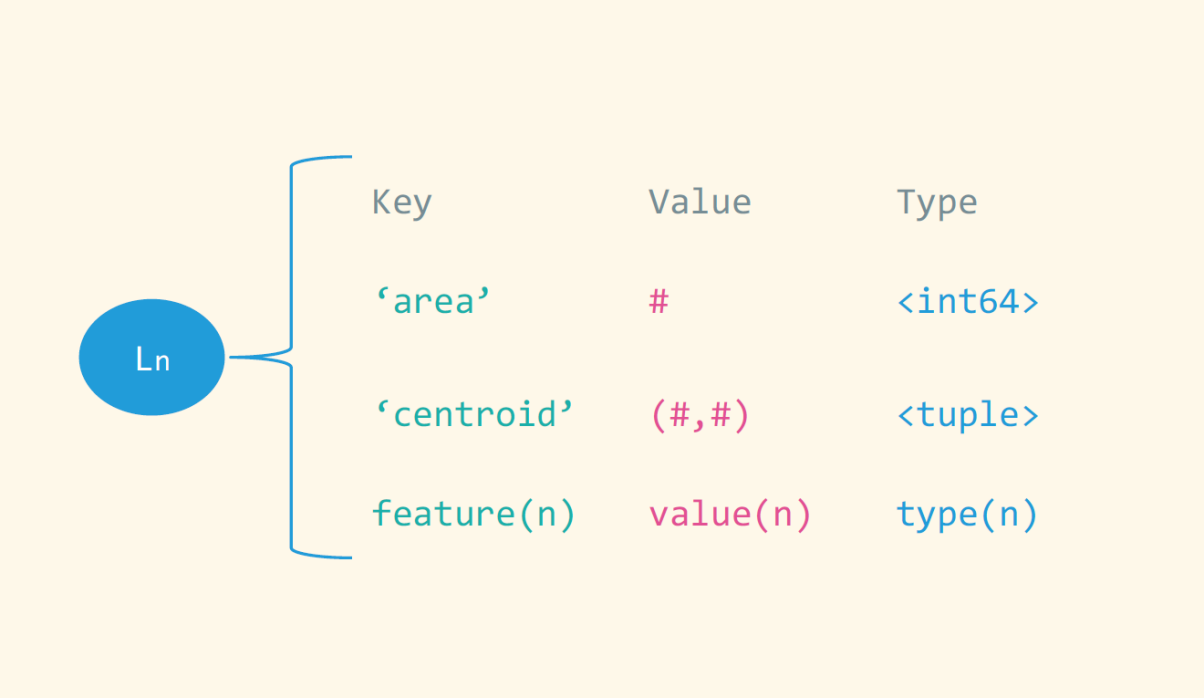


Figure 5.5: Structure of vertex `dictionary` showing `key` and `value` pairs with associated data types.

with a cost that depicts the dissimilarity between the two vertices/cells. Hence an edge for a correct association has lower edge cost. For two vertices, u and v, each composed of K features, the total cost metric for dissimilarity between adjacent frames is computed as:

$$c(u, v) = \sum_{i=1}^{K} w_i \vartheta_i(u, v) \tag{5.1}$$

where $\vartheta_i(u, v) = |\zeta_i(u) - \zeta_i(v)|$, the feature difference. ζ_i is the normalised component feature in the vertex dictionary. $c(u, v)$ is a the cost metric and w_i is a weight parameter introduced to control the relative importance of each feature component in computing dissimilarity.

For other cell behaviours aside *move*, the feature difference need to be adapted to reflect that one of the frames in the pair has a missing cell (for *appearance* and *disappearance*) or that two cells need to be accounted for instead of a one-to-one matching (for *split* and *merge*). The value $\zeta(\emptyset)$ is introduced for the missing cell. Hence in the case of appearance and disappearance, the feature difference vector is defined as

$\vartheta(\emptyset, v) = |\zeta(\emptyset) - \zeta(v)|$ and $\vartheta(v, \emptyset) = |\zeta(v) - \zeta(\emptyset)|$ respectively.

For *split* one cell in the current frame is needs to be matched with two cells in the next frame. Conversely, for *merge* two cells in the current frame are associated with one cell in the next frame. The difference vector for these behaviours will need to be adapted as well. The feature vector for the two cells involved in a merge or split in a given frame, t is computed as one, $\zeta(u_1, u_2)$, the combination of the cells u_1 and u_2. The difference vectors for split and merge is then expressed as $\vartheta(u, (v_1, v_2)) = |\zeta(u) - \zeta(v_1, v_2)|$ and $\vartheta((u_1, u_2), u) = |\zeta(u_1, u_2) - \zeta(v)|$ respectively.

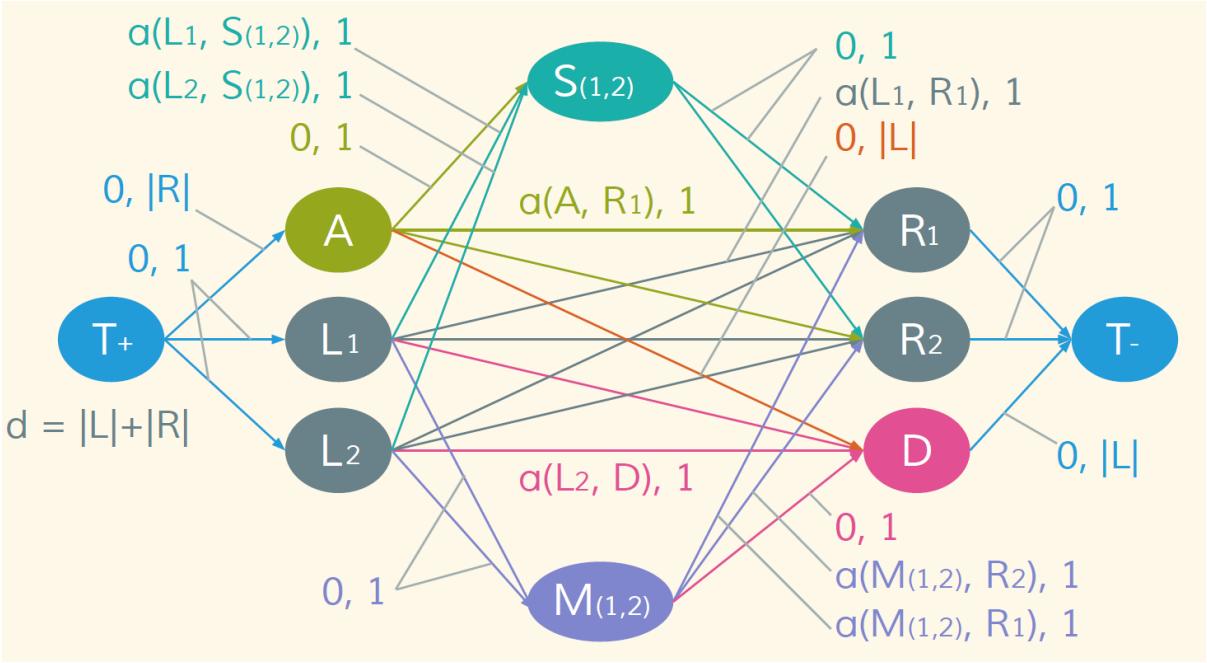
5.3.2 Solving the coupled minimum-cost flow problem

The aim of the tracking algorithm is to find the matching of cells between two frames such that the total edge costs are minimum. The solution for the network can be obtained through an optimisation framework via as detailed in Section 5.2.1. To construct the optimisation model, the flow network is transformed to an incidence matrix representation which shows the relationship between the two sets of cells in the separate frames. The sparse incidence matrix representation for the flow network is shown in Figure 5.6. In the matrix, rows and columns represent vertices and edges respectively. The entries, $[+]$ and $[-]$ show the origin and end point of edges in the directed flow network.

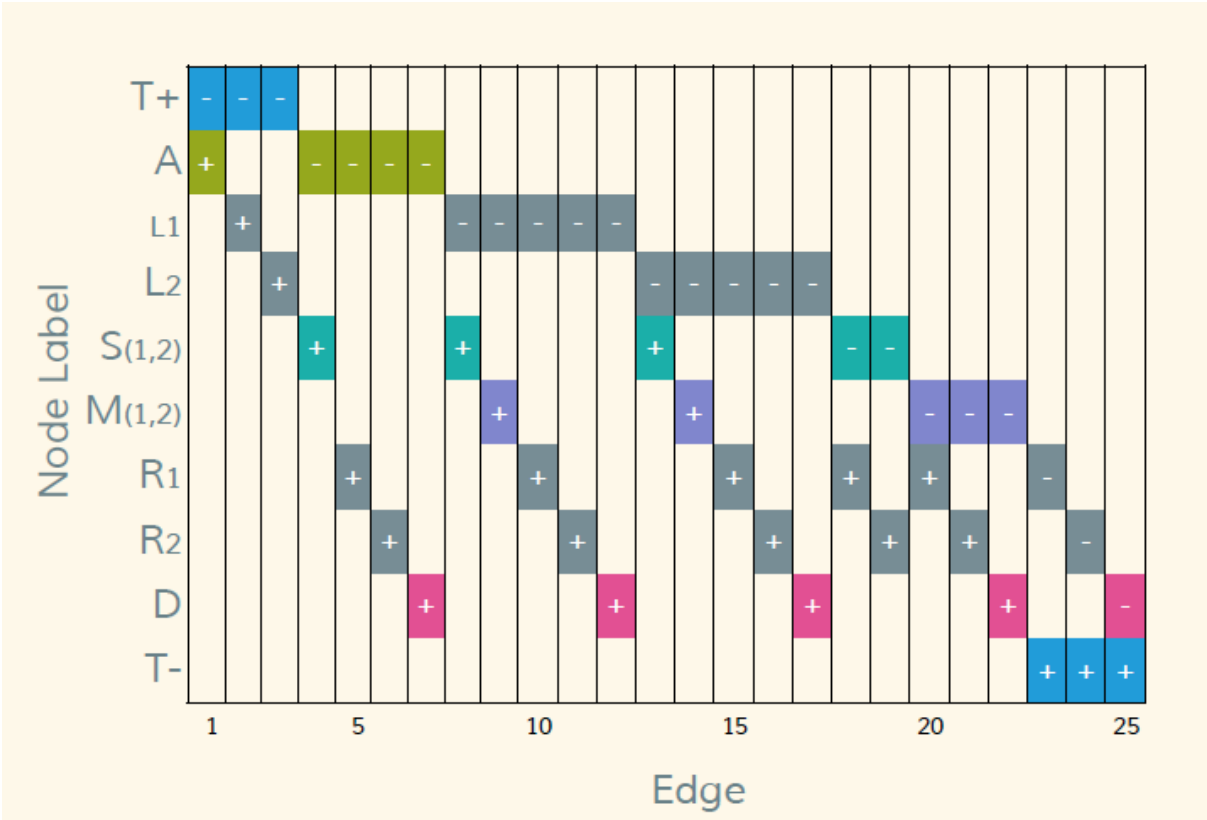
5.3.2.1 Pruning the network

In this Python implementation, the optimisation model is expressed in the `Pyomo` modelling framework. The speed of such a modelling framework is limited by the size of the graph to be instantiated. The size of the graph and hence the optimisation problem can be reduced to improve performance by removing nodes and edges with the highest cost. The set of edges connecting a vertex L to another in the next frame R in a *move* matching is pruned by only retaining a proportion $(\alpha|R|)$ of the lowest cost edges. An identical pruning method is applied to edges connecting appear and disappear vertices.

Additionally, the graph size can be reduced by retaining only a fraction, β of the total number of split vertices, S. This implies that only an estimated $\beta|S|$ cells will be



(a)



(b)

Figure 5.6: Flow network and corresponding incidence matrix. Image courtesy: [16]

splitting at any given time. An analogous procedure is applied to merge vertices as well. A similar approach to pruning the edges is applied by retaining the α fraction of edges. The values of the parameters α and β are user-defined.

A further improvement to performance is to modify the flow network by removing the explicit source ($T+$) and sink ($T-$) vertices and all the corresponding edges. The properties of the logical flow network is maintained by changing the requirements for the left vertex, L and appear vertex, A to act as sources. The right, R and disappear, D act as the sink.

5.3.2.2 Edge coupling

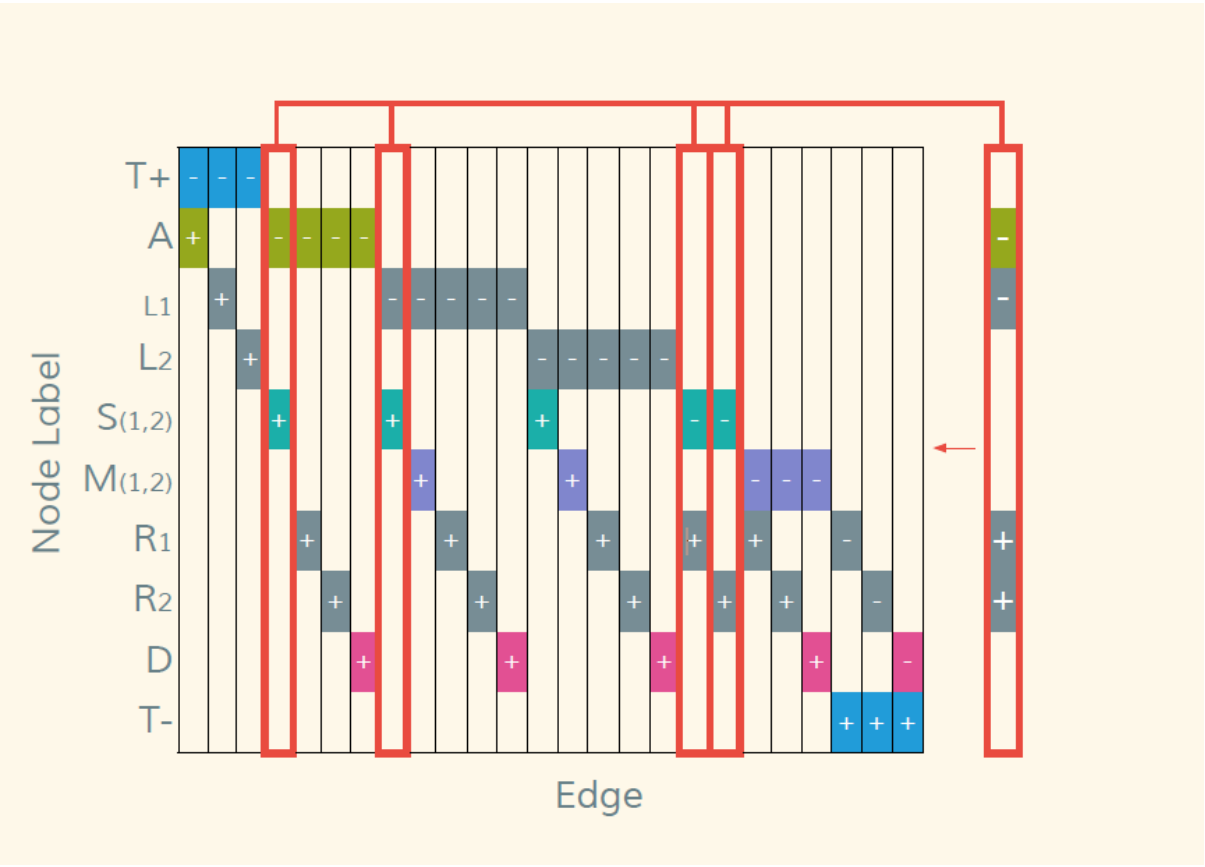
Following the introduction of the additional vertices to account for the merge and split cells, edge coupling is introduced to ensure the conservation of flow constraint is met as described in Section 5.2.1. In the incidence matrx representation, each split/merge vertex required four columns, two for entering edges and two for exiting edges. To implement the coupling in the incidence matrix the four edges are combined together to form one new column and added to the matrix as shown in Figure 5.7a. Figure 5.7b shows the resulting matrix following coupling.

5.3.2.3 Optimisation

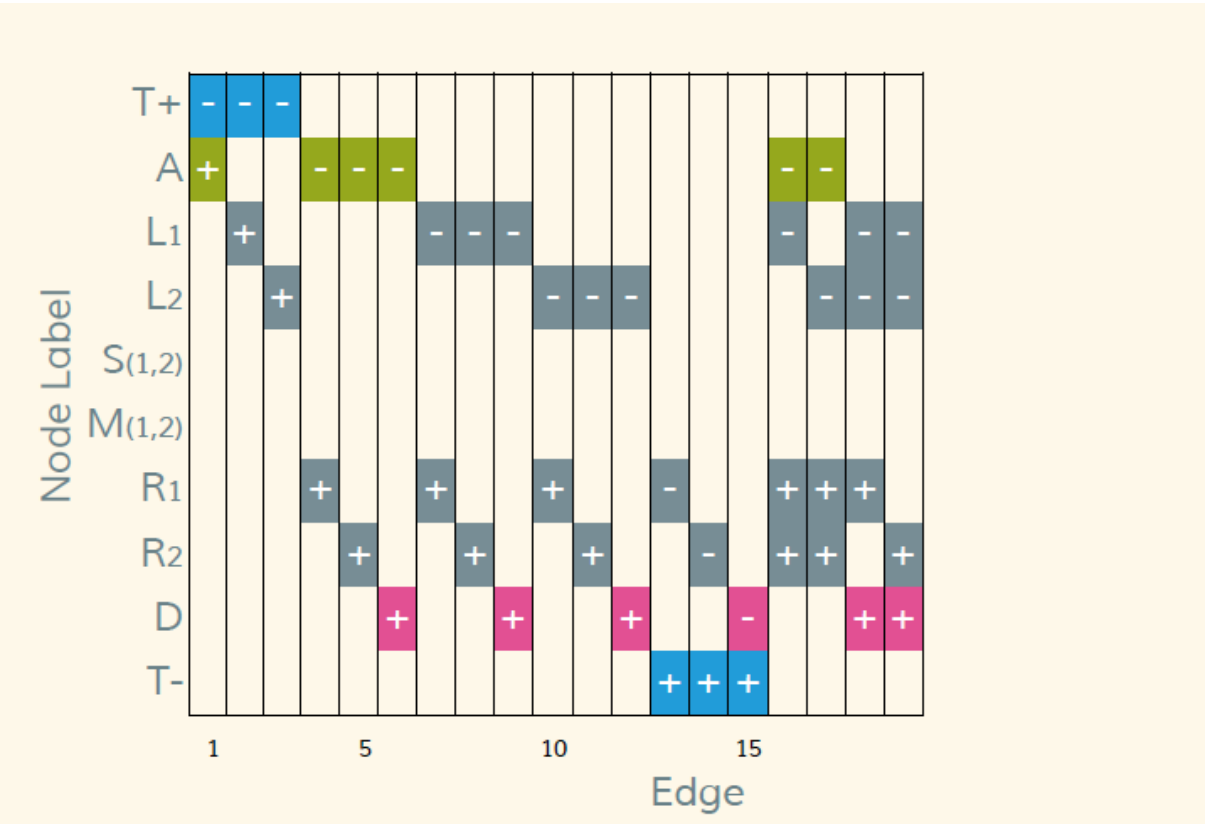
Solving the coupled minimum cost flow involves finding the subset of columns in the coupled incidence matrix where the total cost of those columns is minimised subject to the constraint that no two columns share common non-zero entries. For a set of rows, V and the set of colums, E, the problem is formalised as:

$$\min_{x}(c^T x) \quad \text{subject to} : Ax \leq b \tag{5.2}$$

where c is the vector of edge costs, A is the couple incidence matrix, and b is the vector of size $|V| \times 1$ detailing the sum of the flow through each vertex. x is the binary solution vector of size $1 \times |E|$ where $x_i = 1$ indicates edge E_i is included in the solution and $x_i = 0$



(a)



(b)

Figure 5.7: The coupling process and the corresponding coupled matrix

indicates it is not included. The problem can then by solved via linear programming [137].

To compute the numerical solution to the optimisation problem, we use the GNU Linear Programming Kit(GLPK) [148] that returns the solution vector, x. GLPK package is intended for solving large-scale linear programming (LP), mixed integer programming (MIP), and related problems. It consists of a set of routines written in `ANSI C` and organized in the form of a callable library that provides a fast and robust optimisation engine.

Considering the base code of our implementation is in Python, we formulate the optimisation problem instead in `Pyomo` [149, 150, 151]. Pyomo is a Python-based open-source software package that supports a diverse set of optimization capabilities for formulating, solving, and analyzing optimization models. It provides an easy-to-use interface to the GLPK engine as well as other popular optimisation modules. Using Pyomo, we achieve the speed of running optimisation in `C` code with an accessible Python interface that integrates well with the code of the other modules in one Python environment.

5.3.3 Output and visualisation

The tracking algorithm provides an association of cells for consecutive pairs of frames. From the frame pair associations, a complete trajectory of each cell can be constructed over the entire set of frames in the video or timelapse. To annotate unique tracks, each cell is given a number ID and its progression across frames labelled. As indicated earlier, each cell in a given frame is encoded in a Python `dictionary` type that allowed features of varying data types to be handled and compared as one Python object.

Considering a cell track will involve ordering matched dictionaries from frame-to-frame, the output object is `OrderedDict`; an ordered collection of the Python `dict` type. Figure 5.8 shows the output data structure for a cell track. The output tracks can then be exported and stored in compact cross-platform formats as a JavaScript Object Notation (JSON) file or as a comma-separated value (CSV) file. CSV stores tabular data

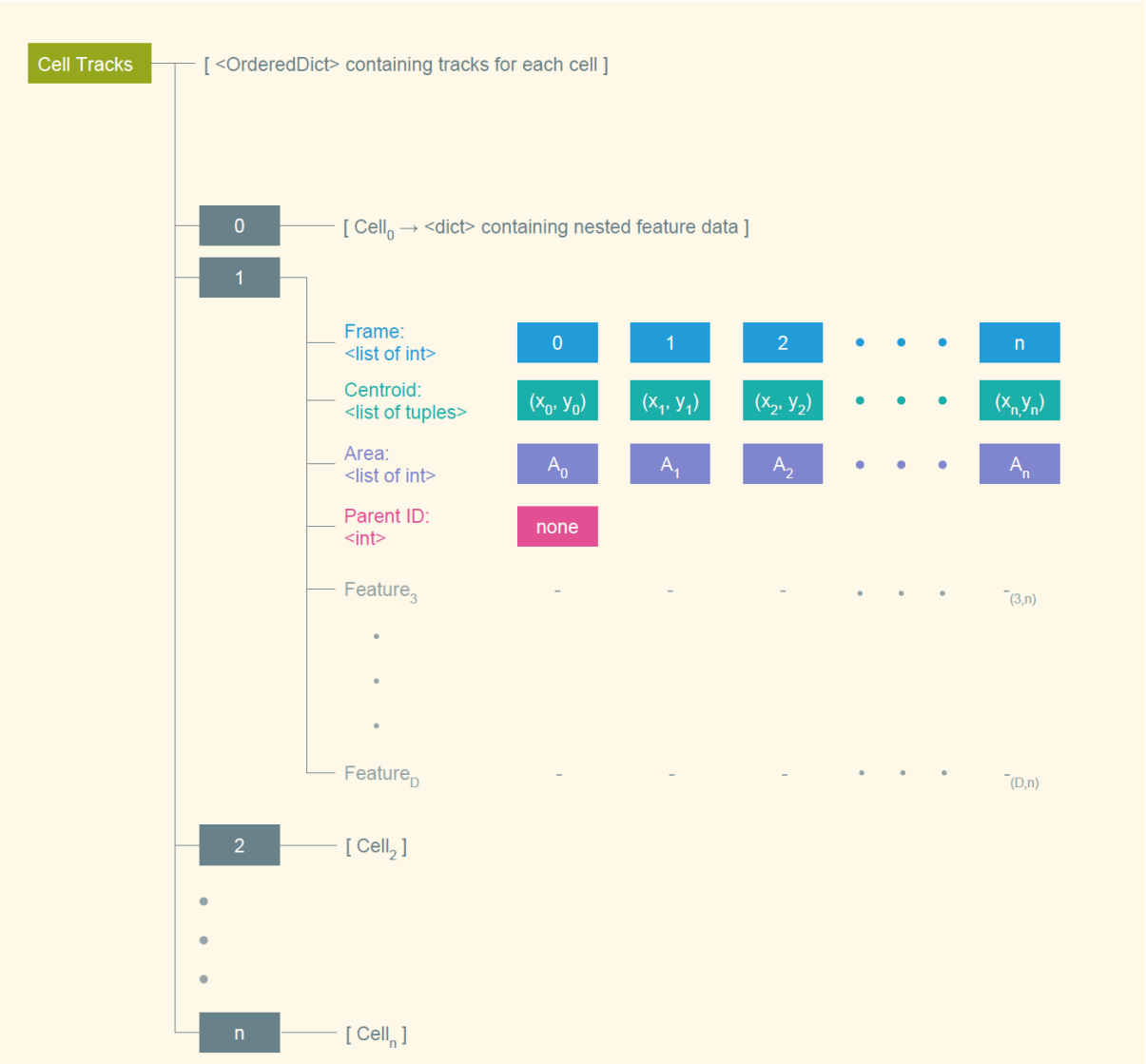


Figure 5.8: Data structure of cell tracks output.

in plain text. Such file data interchange formats allow extend the portability of the cell tracks obtained from the algorithm.

In addition to the portable files of the tracks, a web-based visual interface built in JavaScript is designed to enable viewing the cell track as segmented regions from across frames. The interface also allows the segmented cells to be stacked and aligned to along the track to observe the morphological changes over time. Figure 5.9a shows an older version with tracked cell trajectories. Figure 5.9b shows the current new interface.

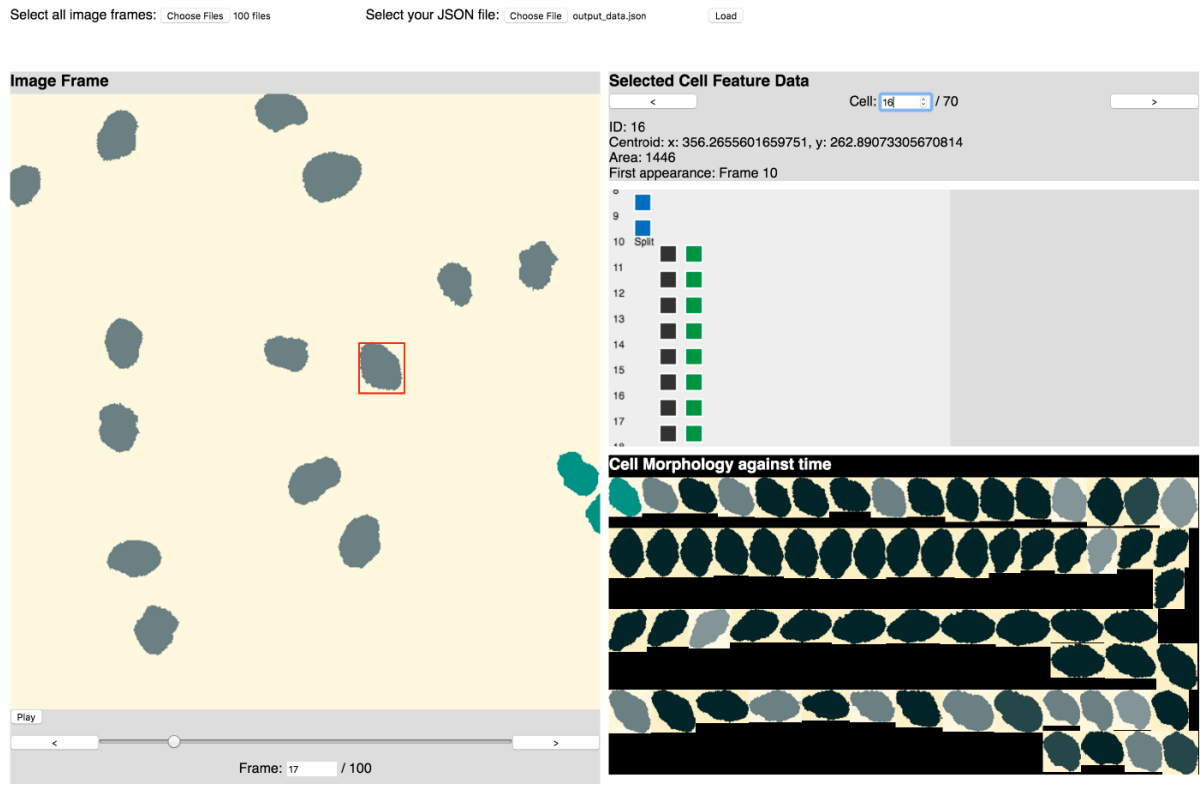
5.3.4 Installation and usage

The web interface demo shown in Figure 5.9 could provide an interface for cell biology researchers to visualise the time course of individual cells as well as changes in morphology for small scale experiments. For large scale further analysis, the data dump from `JSON` or `CSV` described in Section 5.3.3 could be used.

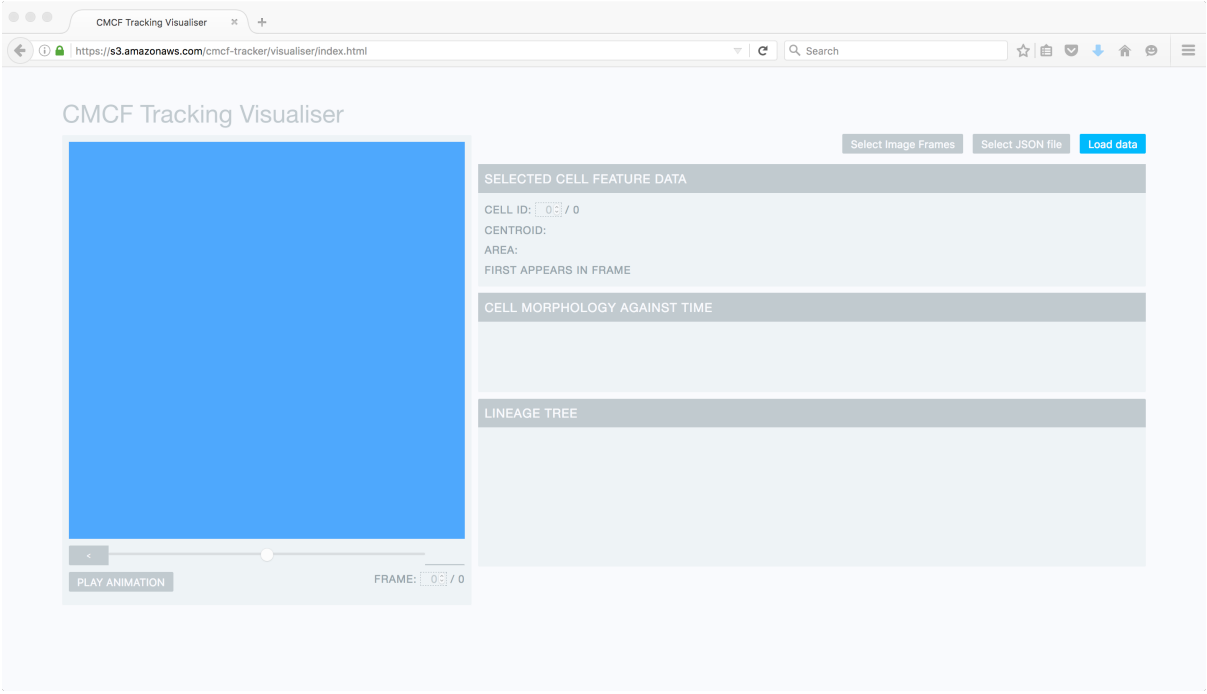
For developers, the Python source code can be cloned or downloaded from the GitHub repositories. To run the source code requires `Python 2.7` to be installed and the required add-on packages listed in the `requirements.txt` file installed. Developers can also provide additional functionality and bug fixes via pull requests.

5.4 Conclusion

This chapter presented CMCFtrack, an open source Python implementation of the coupled minimum-cost flow tracking algorithm. The algorithm offers a flow network modelling approach to frame-to-frame multiple object association association framework. To the best of our knowledge, this is the only open Python implementation of the algorithm although it is a well-cited algorithm for cell tracking [152, 41, 153]. The implementation also provides a web-based JavaScript visualisation tool to view cell tracks and corresponding cell morphology along each cell trajectory. The overall implementation is extensible and can be improved by other developers as well.



(a)



(b)

Figure 5.9: The web-based visualisation tool is shown running from an Amazon S3 server in browser. The layout allows different aspects of the user interface to be separated into panels to improve usability. **(a)** shows a fully functional initial implementation. The tool can be accessed at `https://s3.amazonaws.com/cmcf-tracker/visualiser/index.html`. **(b)** shows the current new interface.

Chapter 6

Cell State Labelling

6.1 Introduction

Changes in cell shape and appearance allow biologists to phenotypically monitor the progression of individual or groups of single cells over time. Such monitoring is usually aimed at understanding the mechanisms associated with time-based phenomena including cell cycle and the mode of progression towards death. Other live cell experiments are also directed at discovery new temporal patterns in specific cell lines as well as effects of candidate drugs or agents.

Temporal patterns associated with cell cycle, cell death or other phenomena can be observed as a sequence of cellular events. In label-free phenotypic screens, giving the appropriate magnification, such events can be visually observed in the shape and appearance of cells. Common cellular effects associated with cell death include nuclear condensation, nuclear fragmentation, membrane blebbing as well as binucleation [154]. Using fluorescence labels or stains other events such as caspace activation, Annexin V and Propidium Iodide (PI) incorporation can also be observed.

High throughput live cell experiments generate large volumes of multi-parameter imaging data that renders efficient manual visual analysis challenging, subjective or even infeasible. To obtain accurate computational models of the patterns in cell progression, each cell instance in a sequence of frames obtained from imaging need to be detected

and tracked over time. The shape and appearance features can then be computed and well-defined in a compact data structure. By fitting appropriate temporal models that group the shape and appearance features encoded in the data structure of each cell into states corresponding to cellular events, a computational representation of cell state transition over time can be obtained. The required transition models can be obtained from live cell imaging using sophisticated yet well-developed image analysis, statistical modelling, and computer vision techniques. Also, the order and nature of the events can be analysed using sequential models with parameters inferred from the multichannel imaging data. This has the potential to deliver efficient unbiased data-driven analysis of the large volume of time-lapse data produced.

A good example of a high throughput experiment that will benefit from such analysis is a recent multi-parameter study of pancreatic β-cells [18]. The authors of the study demonstrated a number of cell death mechanisms that deviate from known definitions can be defined. Indeed, that study hence identified several cell death modes where the order of events that usually define classical apoptosis were not observed. At the core of the conclusions of the study is the order of cellular events observed. Consequently, based on the order of events, the study concluded apoptosis is not the primary mode of adult primary β-cell death. Such studies provide good insight into the design of therapy aimed at regulating cell death. A drawback of the study however is that all individual cells had to be monitored manually by a human observer. This results in a labour-intensive and time-consuming endeavour that may be prone to human error.

The aim of this chapter is to provide a quantitative framework to define and model the transition of morphological cellular events in single cell sequences as well as approaches to compare and group sequences via optimal matching and hierarchical clustering. The tools proposed in this work provide the necessary quantitative analysis that could support the design of new therapies for regulating cell behaviour. These could then be used to explore the heterogeneity of the modes of death in high throughput studies.

6.2 Transforming time lapse to state sequences

This section presents the methods and approaches for transforming input timelapse sequences of images into a pattern of states corresponding to morphological cell events. The framework involves three main components: *segmentation and tracking*, *feature extraction*, and *state transition modelling*. The framework involves delineating cell structures from background in each frame followed by associating instances frame by frame to obtain a trajectory for each cell. Shape and appearance features are then computed for each frame for all trajectories. The section also provides details regarding the transition models used and the associated parameters.

6.2.1 Segmentation and tracking

To allow the analysis of the activity of single cells from a dataset, the video of the multichannel data from the screening is decomposed into a sequence of frames. In each frame, individual cell boundaries are detected and tracked over subsequent frames. This results in a sequence of frames for each single cell. For cases where there is a good contrast between the cell body and background such as in analysis of labelled nuclei (as in Figure 6.1 and Figure 6.2) segmentation of can be achieved via commonly known segmentation algorithms including watershed and adaptive thresholding. Watershed segmentation is applied to separate touching nuclei. Considering fluorescence-labelled nuclei shows good contrast between regions of interest and the background, this simple

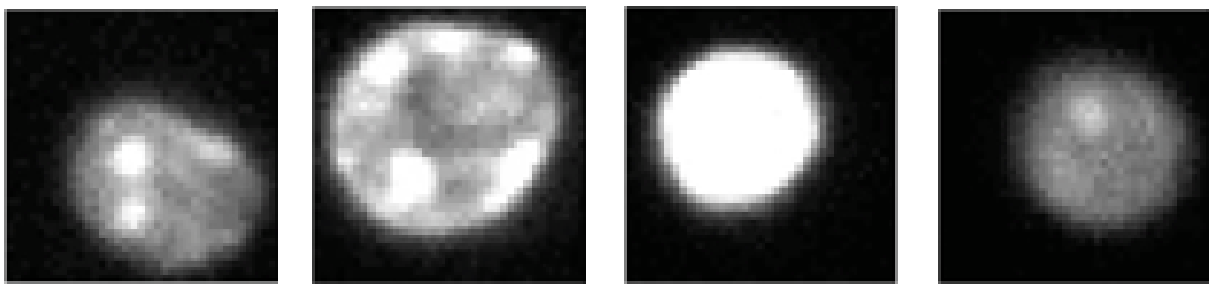
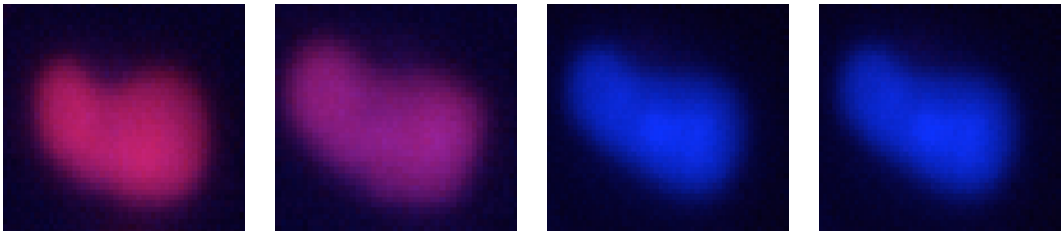


Figure 6.1: Frames from a single nucleus sequence showing the changes in morphology and texture over time towards death in 375 frames. From left to right, Frame 9, 18, 72, 216



(a) (b) (c) (d)

Figure 6.2: Frames from a single nucleus sequence showing the changes in intensity of applied labels. From left to right, Frame 1, 3, 10, 18

segmentation approached proved sufficient to obtain a good segmentation. Whole cell bodies or other nuclei imaged by other modalities such as phase contrast may however require more sophisticated methods as presented in Chapter 2 and 3. Figure 6.3 shows a sample segmentation of nuclei in a frame of a sequence. In other cases where contrast is poor and depending on the imaging modality used, more sophisticated methods as described in Chapter 3 are applied.

Tracking is achieved by frame-to-frame naive nearest neighbour association of features of segmented cells. Post processing to eliminate gaps and inconsistent trajectories is also applied. This tracking approach is sufficient for the datasets where cells are mostly stationary and do not undergo a great deal of motion from frame-to-frame. For each frame in a single nucleus sequence, a feature vector is computed based on the shape, appearance, texture and temporal evolution from previous frame. Feature vectors for each frame provide the input observations to the transition models that are subsequently fit to the entire sequence.

6.2.2 Feature extraction and formulation

The progression of a cell in a live imaging experiment is observed in shape and intensity changes over time. To capture the progression computationally requires obtaining features that account for the geometry of the segmented cell as well as local intensity and texture features.

With the boundary of a single nucleus delineated in all frames of a given sequence, the shape (e.g. area, compactness, eccentricity, solidity) and texture features are computed.

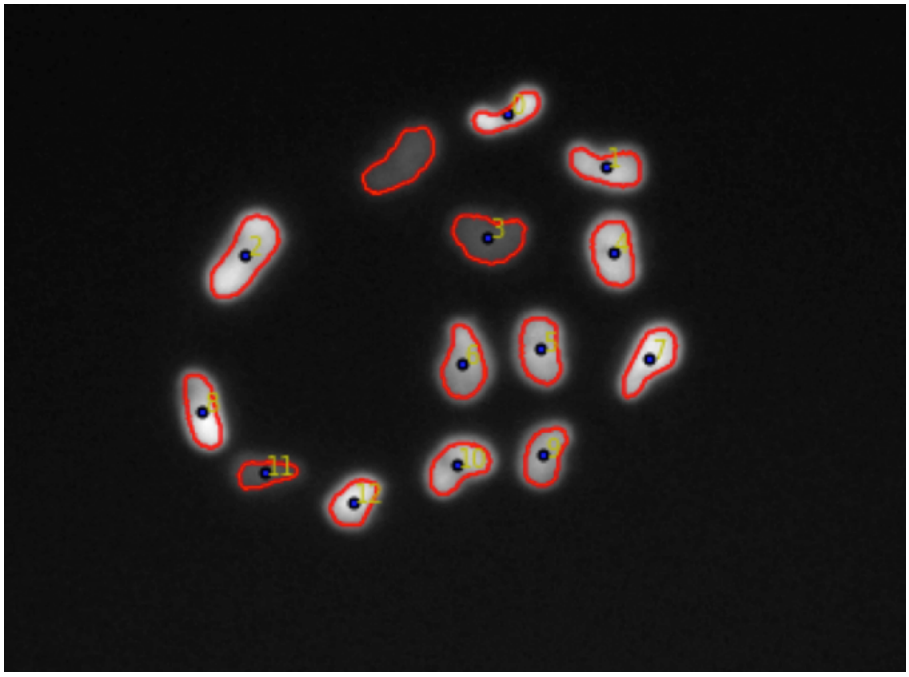


Figure 6.3: Sample segmentation using simple adaptive thresholding of nuclei in selected frame

Thus, in addition to shape features for each cell in a frame, Haralick's, texture and histogram of oriented intensity gradients (HOG) features are computed [155, 156]. In addition to the spatial features, the histogram of optical flow magnitudes (HOF) is computed to account for the temporal progression from frame-to-frame [157, 158]. A concatenation of these vectors results in a feature vector $[x_1, x_2, ..., x_n]$ for each frame in the m-frame sequence. Consequently, for all frames we obtain the normalised [n x m] matrix that represents the morphological signature of the path of the given cell over the time course of the experiment.

6.2.3 State transition modelling

Understanding the pattern of phenotypic changes in cells require temporal models to partition cell trajectory of features into distinct states. Each state then represents a morphology profile where similar morphological features belong to the same state. Hence frames where cells show similar appearance and encoded in their computed features will be more likely to be assigned to the same state. Cell processes, including mitotic division, onset and progression of death show a pattern of morphological states.

As reviewed in Section 2.2.3, the partition of trajectories into states here uses the *sticky* hierarchical Dirichlet process hidden Markov model (HDP-HMM). For our analysis, the implementation of the sticky HDP-HMM by [159] is adopted. To obtain states for the sequence of frames for each cell object, a generative model can be obtained by fitting the the frame features with the sticky HDP-HMM. As a motivating example, using the average intensity features of sequences similar to that shown in Figure 6.2, the state sequences shown in Fig. 6.4 are obtained.

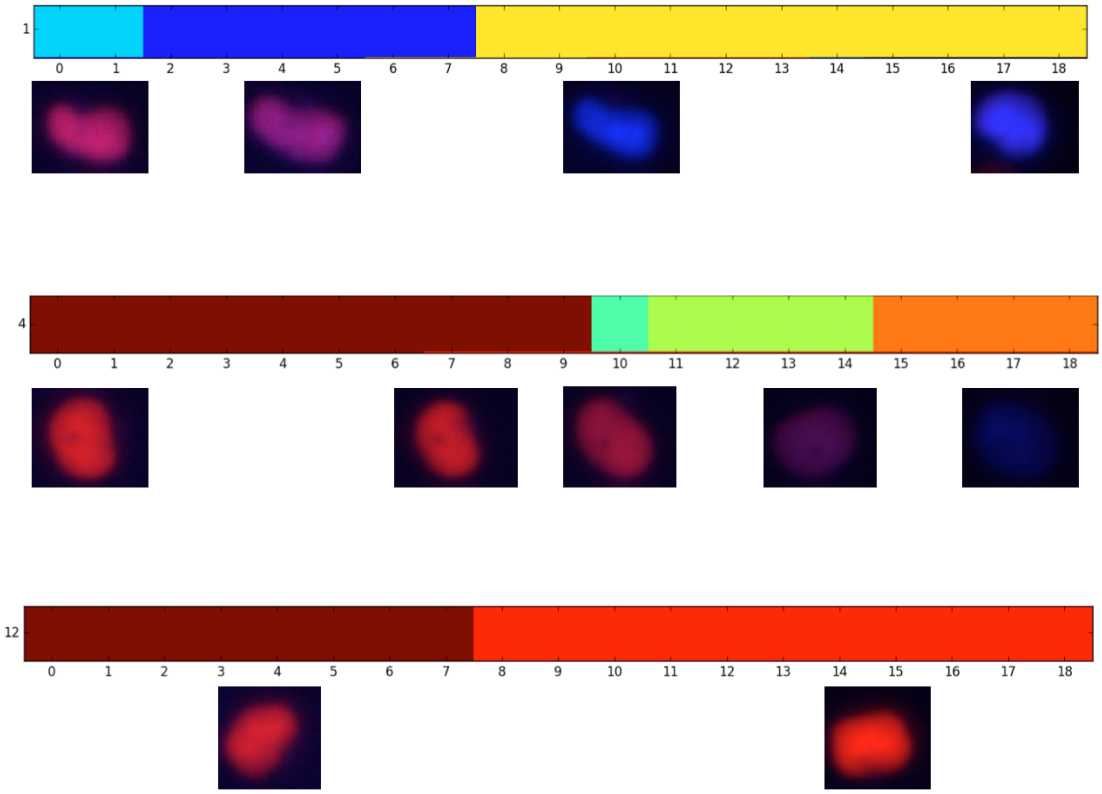


Figure 6.4: State sequences obtained after fitting sticky HDP-HMM. Each colour block represents a state.

6.3 Analysing state sequences

Transforming time lapse cell videos to state sequences provides a compact representation of each cell object. The state sequence representation highlights similarities in morphology over time as well as points in time where cell morphology features change.

The state sequence structure hence provides a basis to compute various statistical characteristics that describe and provide insights into the nature and heterogeneity of the morphological changes and events in cells over time.

6.3.1 State sequence characteristics

The multi-parameter study conducted by Yang *et al.* [18] investigated the forms of cell death in pancreatic β-cell death by observing changes in a number of parameters including nuclear condensation, Caspase-3 activity, Annexin V, and Propidium Iodide. All parameters were measured primarily by intensity in the given channel. Fig. 6.5 shows the summary of the manual analysis [18].

Transforming the timelapse image data sequence to a state sequence enables such analysis to be conducted on a larger scale computationally. Characteristics of the state sequences can then be computed to reveal insights about the image sequences. From sequences of the nuclear channel of the β-cells as shown in the sample frames of a sequence in Figure 6.1, morphological features are computed for each frame. The generative sticky HDP-HMM model fit to the sequences to obtain state sequences for each cell object. Considering the posterior model from which the state sequences are determined is sampled from a base distribution given the data, obtaining the model could also be viewed as a *novel* state discovery process where new states can be observed as new sequences are added and the model resampled. Figure 6.6 shows a grid of the state sequences for 100 cell objects (375 frames each) from the dataset. Each row represents a cell trajectory over time. Each colour block represents a different state and each frame of the sequence is assigned a state. From the state sequences, statistics that describe the characteristics of the morphology evolution of the cells is computed.

6.3.1.1 Distribution of states

Obtaining the state sequences for the time lapse data as shown in Figure 6.6 provides a good first visualisation of how the morphology states of individual cell structures progress over time. At every frame, each cell takes on a given state. To understand

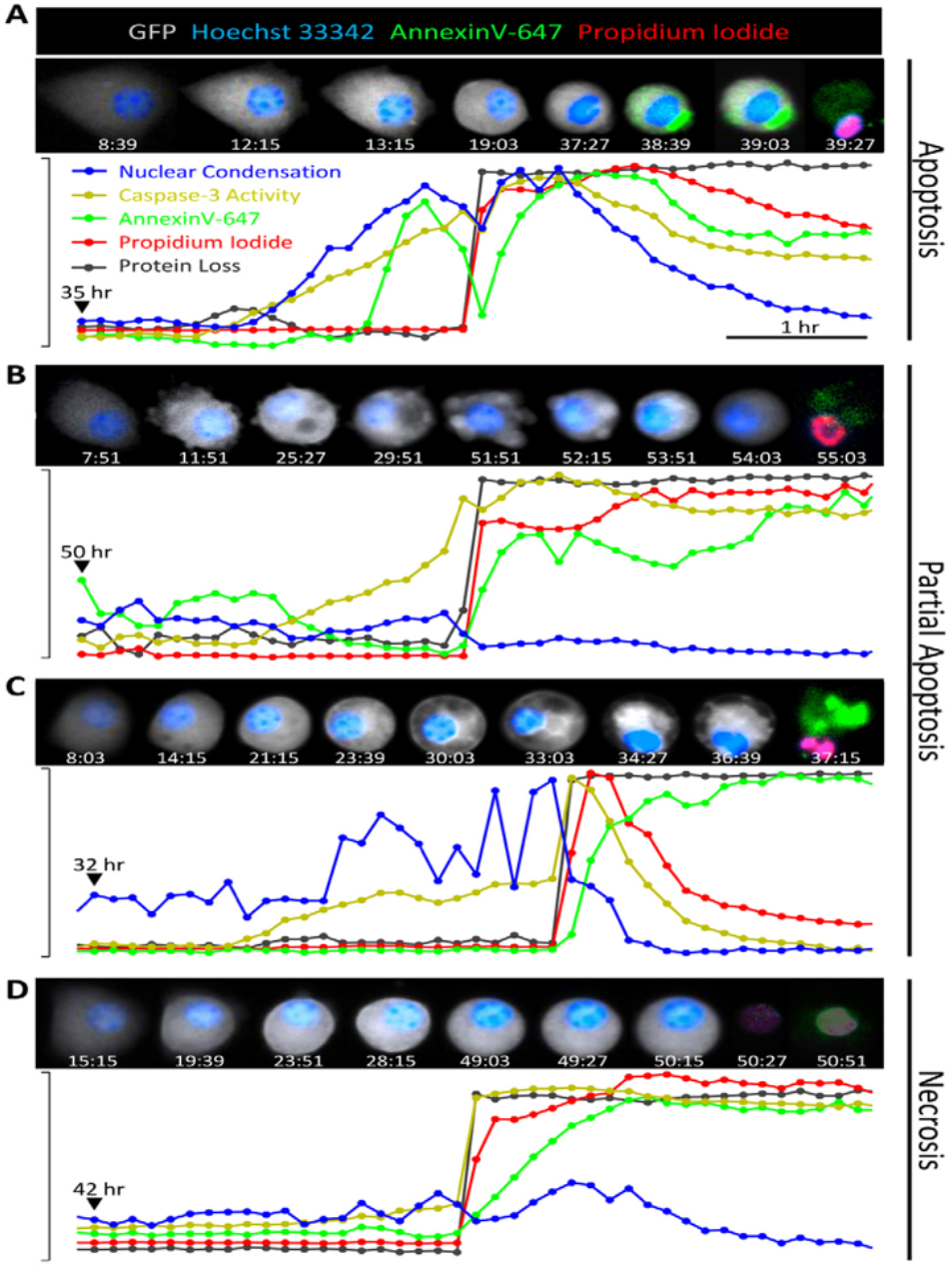


Figure 6.5: Manual multi-parameter analysis of the forms of cell death in pancreatic β-cells. As adopted from [18].

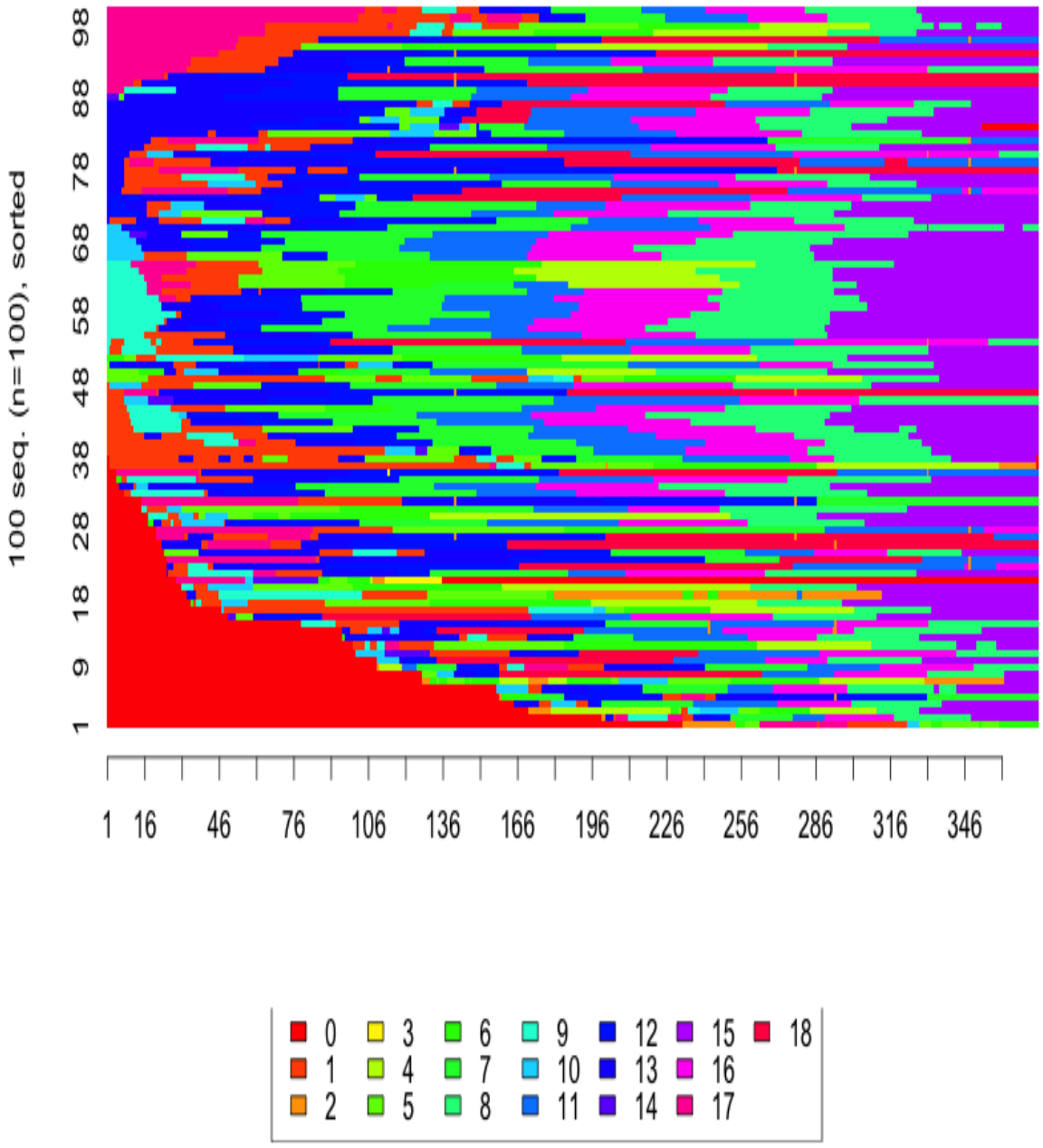


Figure 6.6: State sequences of 100 cell sequences. Each number/color block represents a state

which states are present and their relative proportions at each frame of the timelapse, a plot of the distribution of states over all the group of cells is shown in Figure 6.7. From the distribution plot, the predominant states at any frame gives a good indication of which morphological events are more likely and those that are rare at different times in a timelapse experiment.

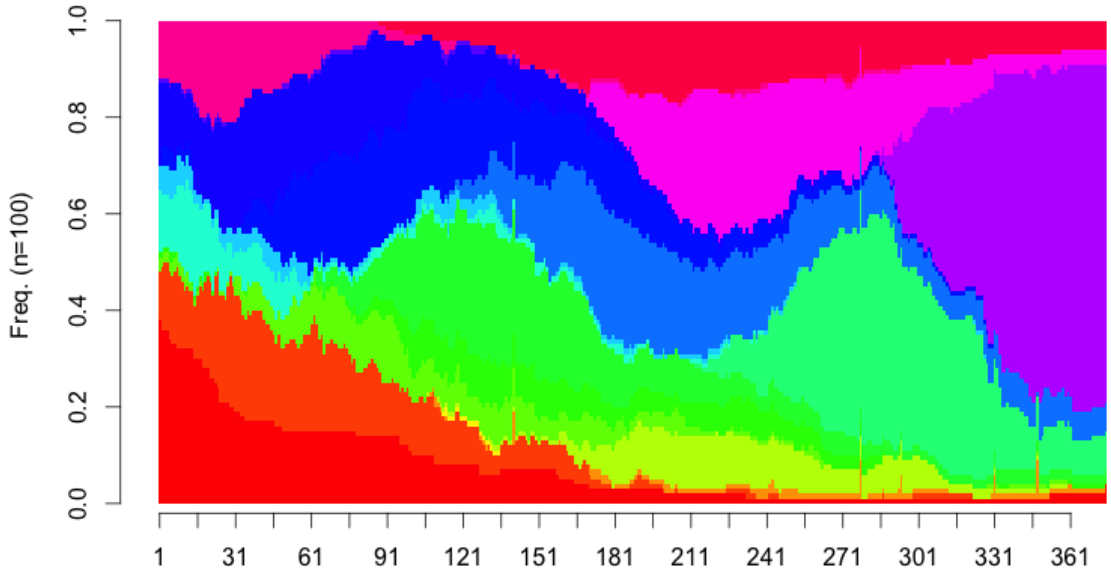


Figure 6.7: Distribution of states at each frame in the sequence.

6.3.1.2 Mean duration of states

Morphological events in cells can be transient or persist over extended periods of time over the course of an experiment. Whether an event lasts longer over time depends on the intrinsic nature of the event itself. For instance, nuclear condensation, where the nucleus of the cell appears denser, tends to progress slowly and often observed over a longer time than the point of nuclear division which occurs relative quicker. Other considerations that affect the duration of an event include cell type and drug treatment.

The relative duration of states is analysed by computing the mean time (in frames) spent in each state from the state sequence. In terms of cell behaviour, the mean time

provides insight into how long each state lasts across all sequences considered. Hence this provides an indicator of short-lived or long-lived cell processes. Figure 6.8 shows that in the 375-frame 100 sequences assessed, the average continuous time spent in a state could be as high as 45 or as low as 2.

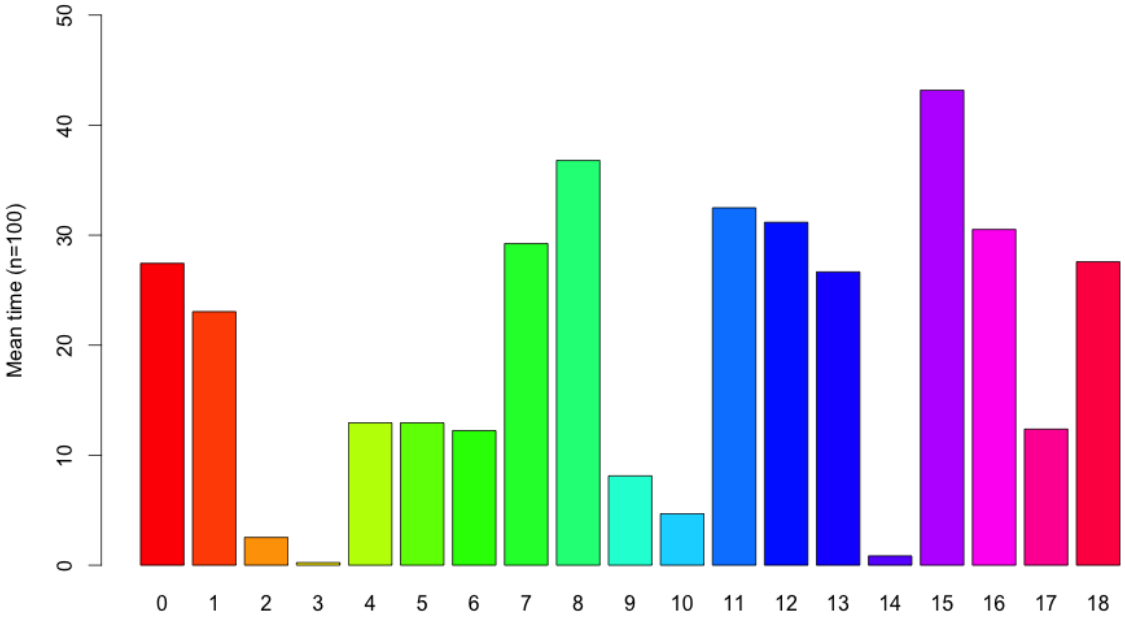


Figure 6.8: Relative duration of states.

6.3.1.3 Entropy of states

Another interesting insight that could be obtained from the state sequences is to determine the dissimilarity in states of the individual cells at each frame. This could be measured by the *entropy* of states at each frame as shown in Figure 6.9. It is observed that the entropy is highest around the mid-section of the sequence and plummets towards the end stages. Considering the sequences were obtained from dying cells, this is expected as cells tend to show similar morphology at the beginning of experiments and then progress at different rates showing a variety of morphologies toward death. Towards the final frame, most cells are already dead and show all tend to depict similar appearance. This is shown in low entropy in early frames, higher in mid frames and the lowest in the final frames.

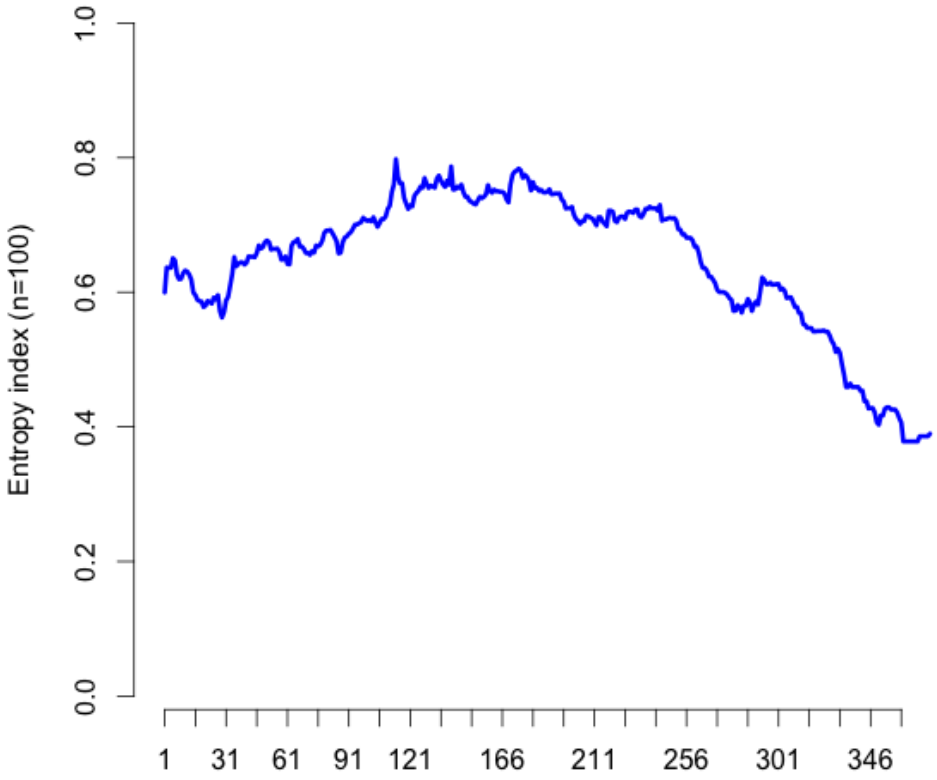


Figure 6.9: Entropy of states at each frame.

6.3.2 State transitions

One key insight required when studying live cells is the understanding of the various phases, states or events that are exhibited and whether there are specific patterns. A state transition occurs when there is a change from one state to another. In the context of cell morphology analysis, this provides insight into which morphological states occur together or which states are precursors for others. By analysing all transition pairs, Figure 6.10 shows the 5 most frequent state transitions in the cell sequences under investigation. Transition state pairs are denoted as $s_i > s_{i+1}$; where $S = \{s_1, s_2, ..., s_N\}$ is the set of all possible states. From the analysis, it is observed that a cell in state labelled **8** is most likely to transition to state **15**. This then provides a simple predicting tool for the next state of cells observed in any given state. Further approaches to understand and cluster overall patterns will be presented in later sections.

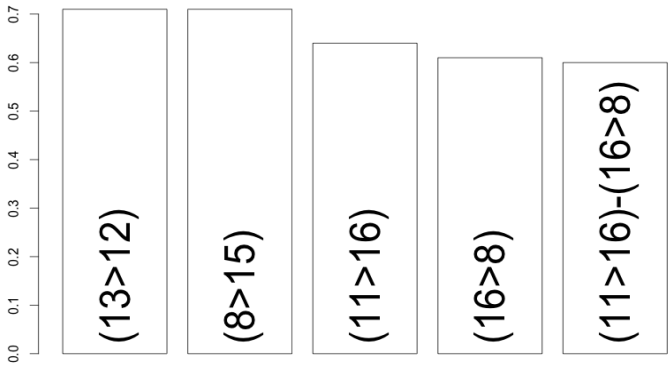


Figure 6.10: Top 5 most frequent event transitions with transition state pairs are denoted as $s_i > s_{i+1}$

6.4 Clustering sequences

Discovering and understanding overall behavioural patterns present in a group of cells is a key goal in timelapse experiments. Having obtained the state model, it is interesting to investigate if there are clusters of identical sequences and the nature of the representative sequence(s) for each cluster. In terms of cell death, the clusters could represent modes of cell death whereas the representative sequences show the typical states and path taken by a dying cell via the given mode.

Using sequence analysis methods originally designed to analyse DNA, RNA and peptide sequences [160], the state sequence that results for each cell sequence can be grouped via clustering and insightful patterns in the sequences can be obtained. To discover clusters, a pairwise comparison of all state sequences resulting in a dissimilarity matrix is required. The dissimilarity between state sequences is computed via optimal matching as in [161]. This is then used in Ward's hierarchical clustering to group the sequences into clusters. An extension of optimal matching method applied to social science research sequences [162], and as implemented by [161], is adopted.

By extracting nearest neighbour sequences with the the least dissimilarity from the cluster centre, candidate representative sequences can be obtained. Redundant sequences

are then eliminated to obtain a unique set of representative sequences for the cluster. These representative sequences show the typical state sequence in each cluster and a good indication of the states and transitions that characterise any observed cluster.

6.4.1 Clustering state sequences

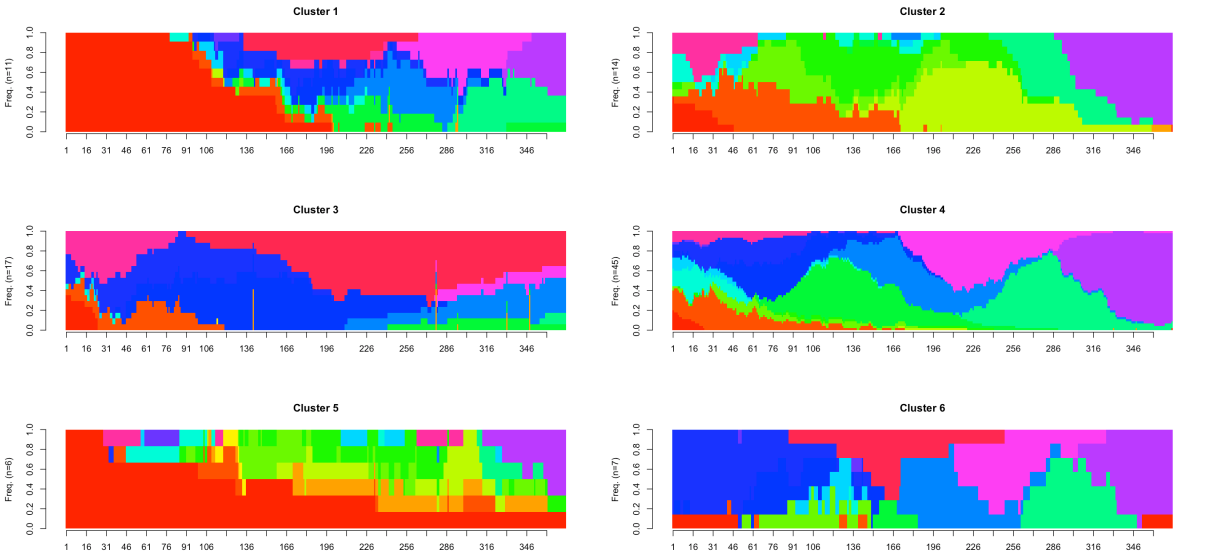


Figure 6.11: State distribution within each cluster

To investigate the clusters within the cell sequences, hierarchical clustering is applied to the state sequences shown in Figure 6.6. To visualise the inter-cluster variability in the cell sequences, Figure 6.11 shows the state distribution of sequences in each of 6 clusters. It is observed that each cluster presents a unique distribution of states. Figure 6.12 shows the representative sequences from the 6 clusters.

6.5 Extension to cell population dynamics

In certain live cell imaging experiments, the progression of population of states is of particular interest rather than individual cells. In such experiments, biologists seek to understand the effect of chemical or environmental agents on the overall morphology dynamics of a group of cells. The methods described above can be applied to such experiments to gain similar insights as at the single cell level. To obtain states for a

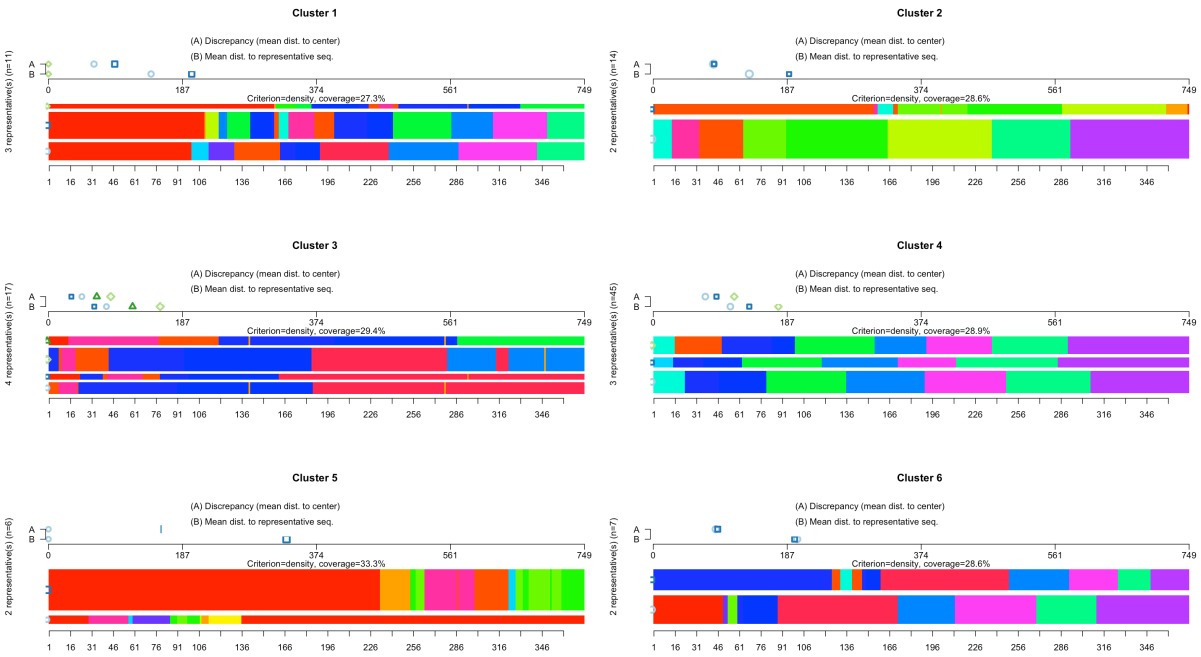


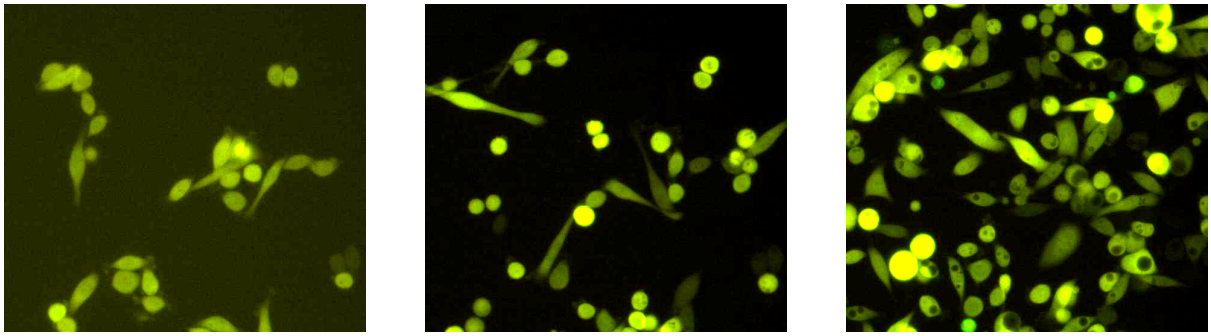
Figure 6.12: Representative sequence(s) of each of 6 clusters

group of cells implies labelling the frames in the video sequence based on the overall average feature characteristics of each frame. With state labels obtained for each video sequence, similar methods of state sequence analysis and clustering are applied.

6.5.1 Population state labelling

Figure 6.13 shows sample frames from an experiment where the cell population shows changes in cell count, intensity of a fluorescent label as well as in morphology. The aim is to label the various states in the video over time to obtain a state sequence. State sequences of multiple experiments with different treatments are then compared, analysed and grouped.

By applying the sticky HDP-HMM to an aggregate feature of each frame sequence of 30 videos from multiple experiments, the state sequences shown in Figure 6.14 are obtained. The state sequences reveal similar patterns in early frames and more diverse patterns as frames progress in time.



(a) Frame 2 (b) Frame 50 (c) Frame 193

Figure 6.13: Sample frames from a 193-frame video sequence

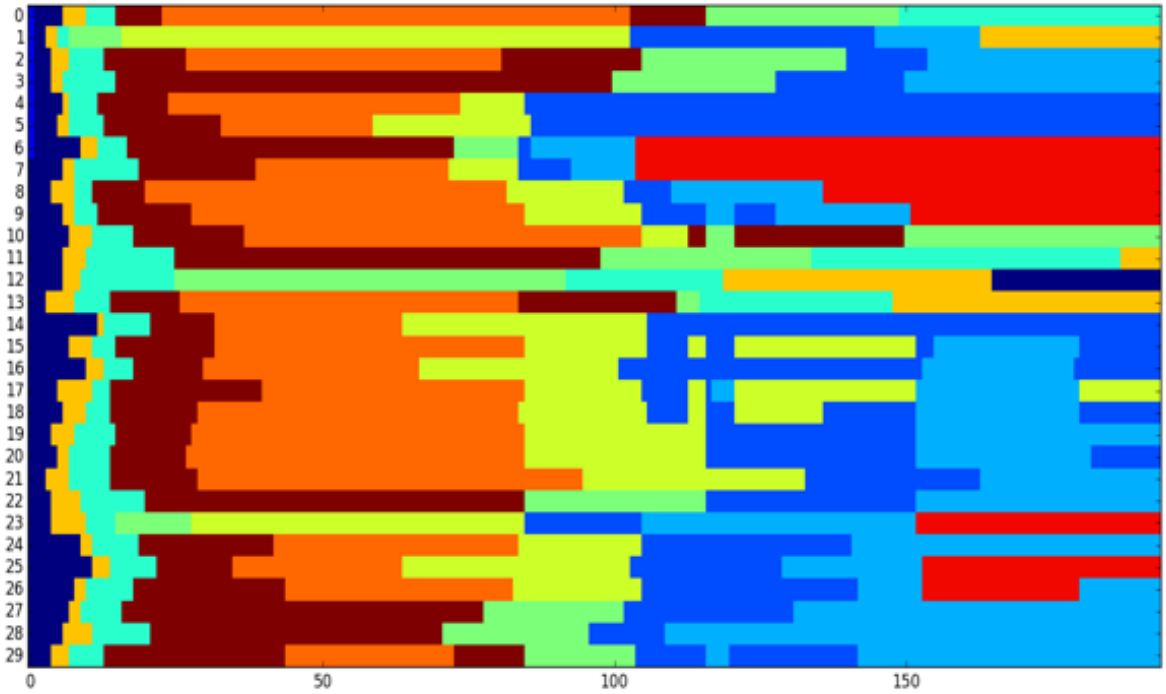


Figure 6.14: State sequence of 30 videos

6.5.2 Population state sequence analysis

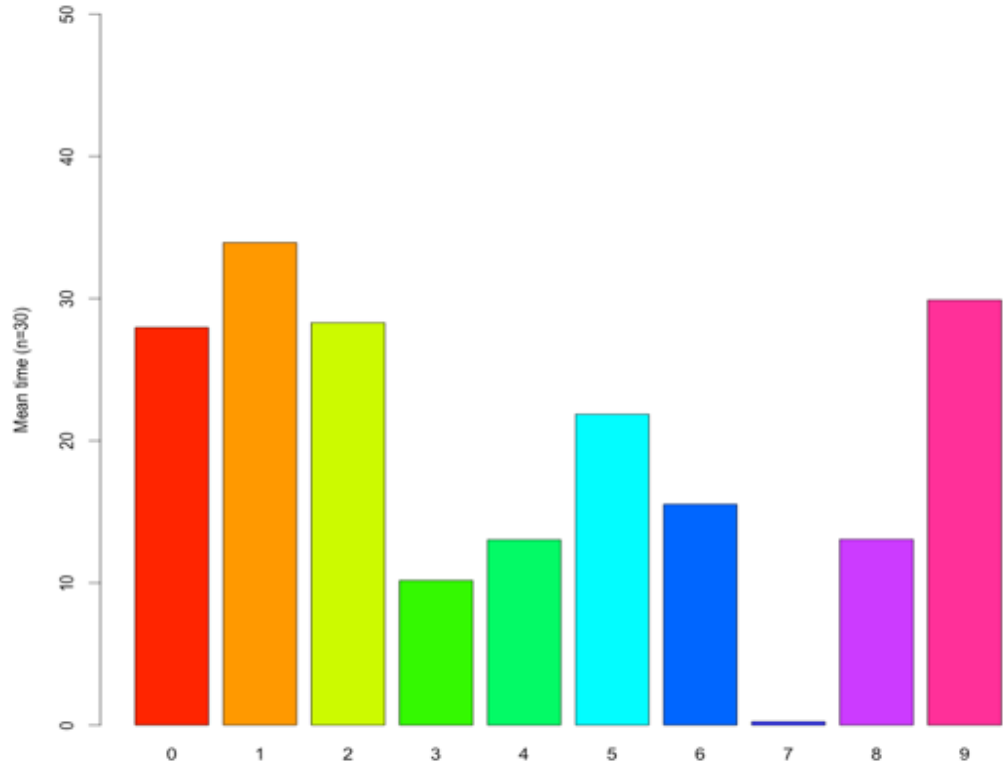
From the state sequences, similar analysis obtained from single cells is performed. Figure 6.15a shows the relative duration of states whereas Figure 6.15b shows the entropy of states. Here, the duration of states reveal which population state dominates the frames as well as which states are only observed in few frames.

6.5.3 Clustering population state sequences

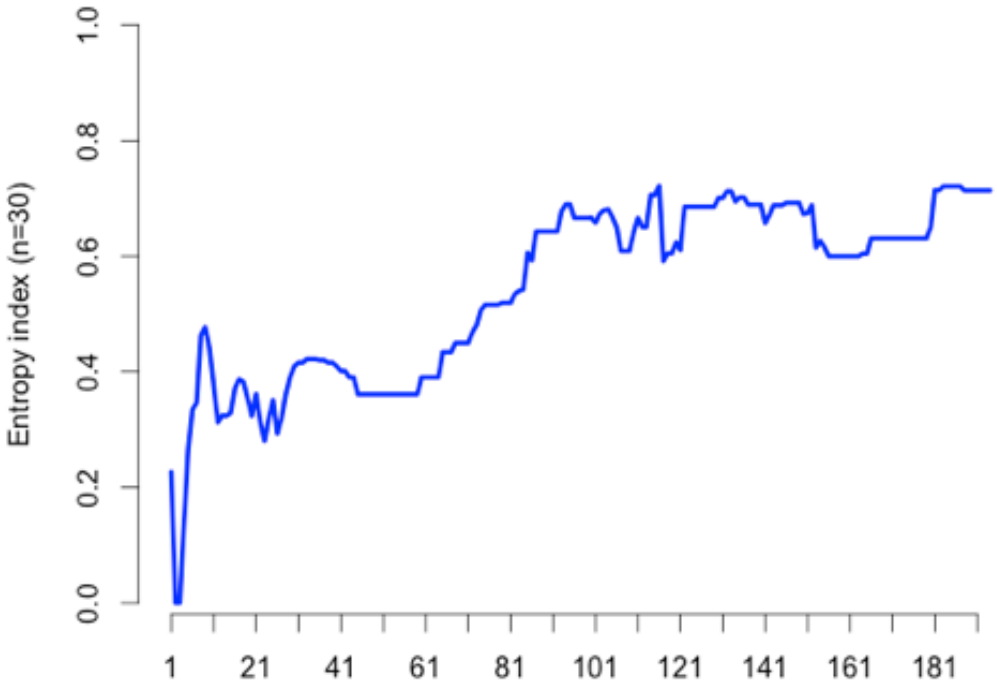
Similar to single cell state sequences, population state sequences can be grouped into clusters as well. Figure 6.16 shows the clusters obtained by agglomerative clustering of the population state sequences as well as the representative sequences in each cluster.

6.6 Conclusion

Manual analysis of high throughput live single cell experiments is challenging. Developing computational analysis methods offers unbiased insights and improves efficiency. The methods presented in this chapter demonstrated approaches to exploring and understanding patterns in live cells. The model that is based on a sticky HDP-HMM leads to the identification of 17 cellular states. The average duration in a given state ranges from 2-45 frames. By initial inspection this model results in annotations that could effectively discriminate between different forms of cell death. The approach outlined also demonstrates an unsupervised framework to group cell sequences obtained from high throughput phenotypic screens based on morphological evolution over time. Also, as demonstrated in populations, the approach could also be applied to other live cell imaging studies where only the overall aggregate change in cell morphology is of primary interest.

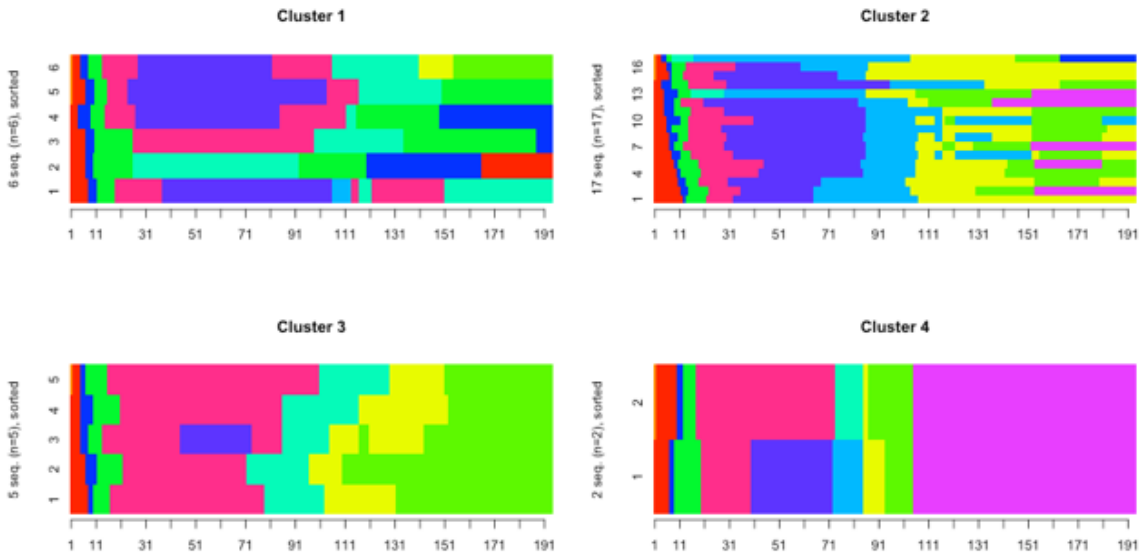


(a) Relative duration of states

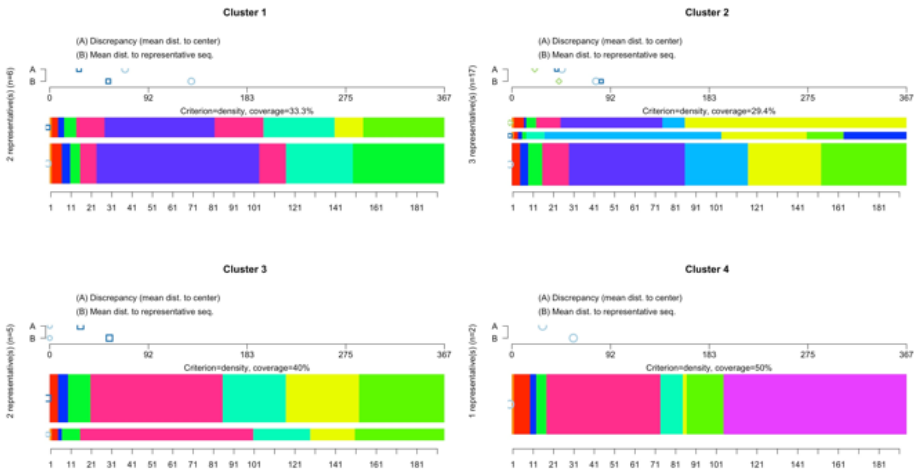


(b) Entropy of states

Figure 6.15: State sequence characteristics



(a) Cluster state sequences



(b) Representative sequences in each cluster sequence

Figure 6.16: Clusters in population state sequences

Chapter 7

Summary and future work

This thesis focused on developing computational tools to enable large scale analysis of image data obtained from phenotypic screening of cells. The work presented in the thesis propose pipelines and methods aimed at accelerating analysis of live cell imaging data in a simple and efficient way. The chapter summarises the work and contributions of this thesis as well as provides an outlook for related future work. The summary organised under the main themes covered in the contributing chapters of the thesis: (i) Cell segmentation; (ii) Cell boundary estimation; (iii) Cell tracking; and (iv) Cell sequence modelling.

7.1 Cell segmentation

Cell segmentation is identified as a key step in enabling the study of a population of cells at a single cell level. Segmentation methods provide tools to isolate cells from background and image artefacts that may interfere with analysis. The contributing work in cell segmentation as detailed in Chapter 3 proposed a pipeline to segment cells in phase contrast images using the image formation model that extracts physically inspired features known as phase retardation features. The work also proposed and showed that the phase retardation features could be used to precondition phase contrast images to provide a richer input image for newly developed deep convolutional neural networks (CNN) for cell segmentation.

To apply the phase retardation feature extraction appropriate, an open source Python

implementation of the method is developed. Section 3.2 presented a weakly-supervised learning segmentation approach detailing the phase retardation feature extraction, phase retardation pixel aggregation into superpixels, and clustering of superpixels cell regions. The approach shows good segmentation accuracy compared to related work using similar annotation.

Deep learning models have state-or-the-art performance on image classification and object detection tasks. The input to deep learning models is usually raw pixels of natural images resulting in a learned model given the appropriate output labels. The work in this thesis explored the option of preprocessing the input image based on the known image formation process model. Chapter 3 proposed the use of phase retardation features as a preconditioning step for input images in deep CNN models. Section 3.4 showed how preconditioned images could help improve learning of relevant features in an example pre-trained CNN model, VGGNet16. The approach showed that stronger activations can be achieved at convolution layers in preconditioned images compared to raw greyscale pixel values.

7.2 Cell boundary estimation

Cell segmentation methods generally attract significant attention when developing tools for cell image analysis considering it is a key step that impacts all downstream analysis. Biological cells however tend to form clumps, touch and overlap at the boundary. Post-processing approaches are available to separate touching and overlapping boundaries but directly accounting for the estimation of such boundaries is rare. Chapter 4 proposed and demonstrated the effect of a variable feature weighting approach to account for approximations in segmentation.

The approach showed that by assigning higher weights to features from cells where no approximate boundary is obtained and lower weights when approximation is required to separate cells delivers a representation of population statistics closer to the true statistics. The work here shows that incorporating weights in the feature distribution leads to a more

accurate estimation of the true distribution of the given feature in a population of cells.

7.3 Cell tracking

Single cell tracking to obtain individual cell trajectories over time is important in studying dynamic cell behaviour at the single cell level. With all cells in each frame segmented Chapter 5 describes the Python implementation and extension of the minimum cost flow network tracking algorithm originally proposed by Padfield *et al.* [16]. The tracking method implemented accounts for known natural cell processes such as division as well as occlusion and overlap that occasionally occurs in consecutive image frames. The implementation also includes a visualisation interface for viewing cell morphologies in each trajectory. The output is formatted in a compact JSON data format that can be exported and analyzed on other computing programs or platforms.

7.4 Cell sequence labelling

To understand the dynamic behaviour of individual cells as well as population of cells, Chapter 6 presented non-parametric Bayesian clustering methods to fit temporal models to cell trajectories. The work outlined a quantitative framework to define and model the transition of morphological cellular events in single cell sequences as well as approaches to compare and group sequences via optimal matching and hierarchical clustering.

With a set a segmented cell trajectories, a biologist or experimentalist can apply the sequence modelling approach and associated tools outlined in Chapter 6 to obtain a summary visualisation of morphological states present and their relative occurrence in time. A biologist can also cluster the state sequences into groups that provide insight into the existing patterns and the representative structure of signature sequences of each pattern group.

The tools proposed provides the necessary quantitative analysis that could support the design of new therapies for regulating cell behaviour by studying cell morphology transitions. This also enables the analysis of the heterogeneity of the modes of death in

phenotypic studies.

7.5 Future work

The work presented in this thesis provides a foundation to extend to a unifying pipeline to apply such analysis to large scale high throughput screens. Future work in each of the themes can be furthered to achieve an phenotypic analysis platform that can be used biologists and analysts to summarise and understand dynamic morphological changes in cells.

As briefly explored in Chapter 3, deep learning models can be applied on large image data from high throughput screens to achieve robust detection and segmentation performance. Further sequence clustering methods can also be pursued to examine additional sequence dynamics that may reveal further insights into how sequences evolve in different forms of cell death.

In modelling temporal dynamics for live cells, unsupervised deep learning, including recurrent neural network models that have been successful in modelling time series data such as audio and speech [163, 164] could provide better alternative modelling approach to classical state models.

The key bottleneck in adopting transition modelling of cell phenotypes from exploratory analysis into mainstream deterministic biological analysis is the translation of computationally resolved temporal states to biologically expected or relevant states. This could be addressed by manual labelling of states to enable appropriate statistical learning. Manual labelling is however burdensome for large datasets and limits the extent of the variety of states that could be discovered in exploratory analysis. Developing a pipeline that combines the intuition of the knowledge of biology alongside the data-driven approach to resolve discovered states would be immensely valuable. The exploratory techniques presented in this can therefore be augmented with partial labelling approaches that take advantage of human labelling of a small sample and learns a model to resolve phenotypic states.

Bibliography

[1] Andrew C Good, Stanley R Krystek, and Jonathan S Mason. High-throughput and virtual screening: core lead discovery technologies move towards integration. *Drug Discovery Today*, 5(12):61–69, 2000.

[2] Lorenz M Mayr and Dejan Bojanic. Novel trends in high-throughput screening. *Current opinion in pharmacology*, 9(5):580–588, 2009.

[3] Robert P Hertzberg and Andrew J Pope. High-throughput screening: new technology for the 21st century. *Current opinion in chemical biology*, 4(4):445–451, 2000.

[4] Paul Lang, Karen Yeow, Anthony Nichols, and Alexander Scheer. Cellular imaging in drug discovery. *Nature Reviews Drug Discovery*, 5(4):343–356, 2006.

[5] David C Swinney and Jason Anthony. How were new medicines discovered? *Nature reviews Drug discovery*, 10(7):507–519, 2011.

[6] Mark A Lindsay. Target discovery. *Nature Reviews Drug Discovery*, 2(10):831–838, 2003.

[7] Rainer Pepperkok and Jan Ellenberg. High-throughput fluorescence microscopy for systems biology. *Nature Reviews Molecular Cell Biology*, 7(September):690–696, 2006.

[8] David J Stephens and Victoria J Allan. Light microscopy techniques for live cell imaging. *Science (New York, N.Y.)*, 300(5616):82–6, April 2003.

[9] Siwon Choi, Wei Wang, Alexandrew J S Ribeiro, Agnieszka Kalinowski, Siobhan Q Gregg, Patricia L Opresko, Laura J Niedernhofer, Gustavo K Rohde, and Kris Noel Dahl. Computational image analysis of nuclear morphology associated with various nuclear-specific aging disorders. *Nucleus*, 2(6):570–579, 2011.

[10] Guido Kroemer, L Galluzzi, Peter Vandenabeele, J Abrams, Emad S Alnemri, EH Baehrecke, MV Blagosklonny, WS El-Deiry, P Golstein, DR Green, et al. Classification of cell death: recommendations of the nomenclature committee on cell death 2009. *Cell Death & Differentiation*, 16(1):3–11, 2008.

[11] Kenneth Chu, Noella Teele, Michael W. Dewey, Norman Albright, and William C. Dewey. Computerized video time lapse study of cell cycle delay and arrest, mitotic catastrophe, apoptosis and clonogenic survival in irradiated 14-3-3 and cdkn1a (p21) knockout cell lines. *Radiation Research*, 162(3):270–286, 2004.

[12] Thomas Nketia, Jens Rittscher, and Alison Noble. Utilizing phase retardation features to segment cells in phase contrast microscopy images. In *Conference in Medical Image Understanding and Analysis, MIUA 2014*, July 2014.

[13] Thomas A Nketia, J Alison Noble, and Jens Rittscher. Towards quantifying the impact of cell boundary estimation on morphometric analysis for phenotypic screening. In *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, pages 781–784. IEEE, 2015.

[14] Zhaozheng Yin, Takeo Kanade, and Mei Chen. Understanding the phase contrast optics to restore artifact-free microscopy images for segmentation. *Medical image analysis*, 16(5):1047–62, July 2012.

[15] Hang Su, Zhaozheng Yin, Seungil Huh, and Takeo Kanade. Cell segmentation in phase contrast microscopy images via semi-supervised classification over optics-related features. *Medical image analysis*, 17(7):746–65, October 2013.

[16] Dirk Padfield, Jens Rittscher, and Badrinath Roysam. Coupled minimum-cost flow

cell tracking for high-throughput quantitative analysis. *Medical image analysis*, 15(4):650–68, August 2011.

[17] Elayne M Chan, Sutheera Ratanasirintrawoot, In-Hyun Park, Philip D Manos, Yuin-Han Loh, Hongguang Huo, Justine D Miller, Odelya Hartung, Junsung Rho, Tan A Ince, et al. Live cell imaging distinguishes bona fide human ips cells from partially reprogrammed cells. *Nature biotechnology*, 27(11):1033–1037, 2009.

[18] Yu Hsuan Carol Yang and James D Johnson. Multi-parameter single-cell kinetic analysis reveals multiple modes of cell death in primary pancreatic β-cells. *J Cell Sci*, 126(18):4286–4295, 2013.

[19] Yu Hsuan Carol Yang, Yury Y Vilin, Michel Roberge, Harley T Kurata, and James D Johnson. Multiparameter screening reveals a role for na+ channels in cytokine-induced β-cell death. *Molecular Endocrinology*, 28(3):406–417, 2014.

[20] Jyoti K Jaiswal, Ellen R Goldman, Hedi Mattoussi, and Sanford M Simon. Use of quantum dots for live cell imaging. *Nature methods*, 1(1):73–78, 2004.

[21] II Shemiakina, GV Ermakova, PJ Cranfill, MA Baird, RA Evans, EA Souslova, DB Staroverov, AY Gorokhovatsky, EV Putintseva, TV Gorodnicheva, et al. A monomeric red fluorescent protein with low cytotoxicity. *Nature communications*, 3:1204, 2012.

[22] Jonas Kühn, Etienne Shaffer, Julien Mena, Billy Breton, Jérôme Parent, Benjamin Rappaz, Marc Chambon, Yves Emery, Pierre Magistretti, Christian Depeursinge, Pierre Marquet, and Gerardo Turcatti. Label-free cytotoxicity screening assay by digital holographic microscopy. *Assay and drug development technologies*, 11(2):101–7, March 2013.

[23] Carlos Arteta, Victor Lempitsky, J Alison Noble, and Andrew Zisserman. Learning to detect cells using non-overlapping extremal regions. *MICCAI*, 15:348–56, January 2012.

[24] Rehan Ali, Mark Gooding, Tünde Szilágyi, Borivoj Vojnovic, Martin Christlieb, and Michael Brady. Automatic segmentation of adherent biological cell boundaries and nuclei from brightfield microscopy images. *Machine Vision and Applications*, 23(4):607–621, May 2011.

[25] Andrew A Hill, Peter LaPan, Yizheng Li, and Steve Haney. Impact of image segmentation on high-content screening data quality for sk-br-3 cells. *BMC bioinformatics*, 8(1):340, 2007.

[26] Jan-Philip Bergeest and Karl Rohr. Fast globally optimal segmentation of cells in fluorescence microscopy images. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 2011*, volume 6891 of *Lecture Notes in Computer Science*, pages 645–652. Springer Berlin Heidelberg, 2011.

[27] Cheng Chen, Wei Wang, John A Ozolek, and Gustavo K Rohde. A flexible and robust approach for segmenting cell nuclei from 2d microscopy images using supervised learning and template matching. *Cytometry Part A*, 83(5):495–507, 2013.

[28] Xiaoxu Wang, Weijun He, Dimitris Metaxas, Robin Mathew, and Eileen White. Cell segmentation for division rate estimation in computerized video time-lapse microscopy. In *Proceedings of the SPIE*, 2007.

[29] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[30] Michael Felsberg and Gerald Sommer. The monogenic signal. *IEEE Transactions on Signal Processing*, 49(12):3136–3144, 2001.

[31] ME Ambühl, C Brepsant, J-J MEISTER, AB Verkhovsky, and IF Sbalzarini. High-resolution cell outline segmentation and tracking from phase-contrast microscopy images. *Journal of microscopy*, 245(2):161–170, 2012.

[32] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[33] Weidi Xie, J Alison Noble, and Andrew Zisserman. Microscopy cell counting and detection with fully convolutional regression networks. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, pages 1–10, 2016.

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[35] Marc Antonini, Michel Barlaud, Pierre Mathieu, and Ingrid Daubechies. Image coding using wavelet transform. *IEEE Transactions on image processing*, 1(2):205–220, 1992.

[36] M.A.A. Dewan, M.O. Ahmad, and M.N.S. Swamy. Tracking biological cells in time-lapse microscopy: An adaptive technique combining motion and topological features. *Biomedical Engineering, IEEE Transactions on*, 58(6):1637–1647, June 2011.

[37] K. Thirusittampalam, M.J. Hossain, O. Ghita, and P.F. Whelan. A novel framework for cellular tracking and mitosis detection in dense phase contrast microscopy images. *Biomedical and Health Informatics, IEEE Journal of*, 17(3):642–653, May 2013.

[38] Fuxing Yang, MichaelA. Mackey, Fiorenza Ianzini, Greg Gallardo, and Milan Sonka. Cell segmentation, tracking, and mitosis detection using temporal context. *Medical Image Computing and Computer Assisted Intervention MICCAI*, 3749:302–309, 2005.

[39] Kang Li, Eric D. Miller, Mei Chen, Takeo Kanade, Lee E. Weiss, and Phil G. Campbell. Cell population tracking and lineage construction with spatiotemporal context. *Medical Image Analysis*, 12(5):546 – 566, 2008.

[40] O. Dzyubachyk, W.A. Van Cappellen, J. Essers, W.J. Niessen, and E. Meijering. Advanced level-set-based cell tracking in time-lapse fluorescence microscopy. *Medical Imaging, IEEE Transactions on*, 29(3):852–867, March 2010.

[41] Xinghua Lou and FA Hamprecht. Structured Learning for Cell Tracking. *NIPS*, pages 1–9, 2011.

[42] Erik Meijering, Oleh Dzyubachyk, Ihor Smal, and Wiggert A. van Cappellen. Tracking in cell and developmental biology. *Seminars in Cell And Developmental Biology*, 20(8):894 – 902, 2009. Imaging in Cell and Developmental Biology Planar Cell Polarity.

[43] W. J. Godinez, M. Lampe, R. Eils, B. Muller, and K. Rohr. Tracking multiple particles in fluorescence microscopy images via probabilistic data association. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 1925–1928, March 2011.

[44] Tien-Chuan Ku, Yun-Nih Huang, Chien-Chang Huang, De-Ming Yang, Lung-Sen Kao, Tai-Yu Chiu, Chung-Fan Hsieh, Pei-Yo Wu, Yuh-Show Tsai, and Chung-Chih Lin. An automated tracking system to measure the dynamic properties of vesicles in living cells. *Microscopy research and technique*, 70(2):119–134, 2007.

[45] William J Godinez and Karl Rohr. Tracking multiple particles in fluorescence time-lapse microscopy images via probabilistic data association. *IEEE transactions on medical imaging*, 34(2):415–432, 2015.

[46] X. Lou, F. O. Kaster, M. S. Lindner, B. X. Kausler, U. Kothe, B. Hockendorf, J. Wittbrodt, H. Janicke, and F.A. Hamprecht. Deltr: Digital embryo lineage tree reconstructor. *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 1557–1560, March 2011.

[47] Fuhai Li, Xiaobo Zhou, Jinwen Ma, and S.T.C. Wong. Multiple nuclei tracking using integer programming for quantitative cancer cell cycle analysis. *Medical Imaging, IEEE Transactions on*, 29(1):96–105, Jan 2010.

[48] Ryoma Bise, Zhaozheng Yin, and Takeo Kanade. Reliable cell tracking by global data association. *Biomedical Imaging: From Nano to Macro, IEEE International Symposium on*, pages 1004–1010, 2011.

[49] T. Kanade, Zhaozheng Yin, R. Bise, Seungil Huh, Sungeun Eom, M.F. Sandbothe, and Mei Chen. Cell image analysis: Algorithms, system and applications. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 374–381, Jan 2011.

[50] A.J. Hand, T. Sun, D.C. Barber, D.R. Hose, and S. Macneil. Automated tracking of migrating cells in phase-contrast video microscopy sequences using image registration. *Journal of Microscopy*, 234(1):62–79, 2009.

[51] P. Matula, P. Matula, M. Kozubek, and V. Dvorak. Fast point-based 3-d alignment of live cells. *Image Processing, IEEE Transactions on*, 15(8):2388–2396, Aug 2006.

[52] C.A. Wilson and J.A. Theriot. A correlation-based approach to calculate rotation and translation of moving cells. *Image Processing, IEEE Transactions on*, 15(7):1939–1951, July 2006.

[53] Bo Yang and R. Nevatia. An online learned crf model for multi-target tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2034–2041, June 2012.

[54] Yuan Li, Chang Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2953–2960, June 2009.

[55] Xinghua Lou, M. Schiegg, and F.A. Hamprecht. Active structured learning for cell tracking: Algorithm, framework, and usability. *Medical Imaging, IEEE Transactions on*, 33(4):849–860, April 2014.

[56] Chang Huang, Bo Wu, and Ramakant Nevatia. Robust object tracking by hierarchical association of detection responses. In *Computer Vision–ECCV 2008*, pages 788–801. Springer, 2008.

[57] Chang Huang, Yuan Li, and R. Nevatia. Multiple target tracking by learning-based hierarchical association of detection responses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(4):898–910, April 2013.

[58] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[59] S. Bonneau, M. Dahan, and L.D. Cohen. Single quantum dot tracking based on perceptual grouping using minimal paths in a spatiotemporal volume. *Image Processing, IEEE Transactions on*, 14(9):1384–1395, Sept 2005.

[60] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[61] Michael Held, Michael HA Schmitz, Bernd Fischer, Thomas Walter, Beate Neumann, Michael H Olma, Matthias Peter, Jan Ellenberg, and Daniel W Gerlich. Cellcognition: time-resolved phenotype annotation in high-throughput live cell imaging. *Nature methods*, 7(9):747–754, 2010.

[62] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

[63] Nathalie Harder, Felipe Mora-Bermúdez, William J Godinez, Jan Ellenberg, Roland Eils, and Karl Rohr. Determination of mitotic delays in 3d fluorescence microscopy images of human cells using an error-correcting finite state machine. In *Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on*, pages 1044–1047. IEEE, 2007.

[64] Nathalie Harder, Felipe Mora-Bermúdez, William J Godinez, Annelie Wünsche, Roland Eils, Jan Ellenberg, and Karl Rohr. 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.

[65] Qing Zhong, Alberto Giovanni Busetto, Juan P Fededa, Joachim M Buhmann, and Daniel W Gerlich. Unsupervised modeling of cell morphology dynamics for time-lapse microscopy. *nature methods*, 9(7):711–713, 2012.

[66] Xiaobo Zhou, Jun Yang, Meng Wang, et al. A novel cell tracking algorithm and continuous hidden markov model for cell phase identification. In *2006 IEEE/NLM Life Science Systems and Applications Workshop*, pages 1–2. IEEE, 2006.

[67] Amr El-Labban, Andrew Zisserman, Yusuke Toyoda, Alexander W Bird, and A Hyman. Temporal models for mitotic phase labelling. *Medical image analysis*, 18(7):977–988, 2014.

[68] Matthew James Beal. *Variational algorithms for approximate Bayesian inference*. University of London London, 2003.

[69] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the american statistical association*, 2012.

[70] Yee Whye Teh. Dirichlet process. In *Encyclopedia of machine learning*, pages 280–287. Springer, 2011.

[71] Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

[72] Emily B Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. A sticky hdp-hmm with application to speaker diarization. *The Annals of Applied Statistics*, pages 1020–1056, 2011.

[73] J Sethuraman. A constructive definition of dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[74] Kevin W Eliceiri, Michael R Berthold, Ilya G Goldberg, Luis Ibáñez, B S Manjunath, Maryann E Martone, Robert F Murphy, Hanchuan Peng, Anne L

Plant, Badrinath Roysam, Nico Stuurman, Nico Stuurmann, Jason R Swedlow, Pavel Tomancak, and Anne E Carpenter. Biological imaging software tools. *Nature methods*, 9(7):697–710, jul 2012.

[75] Chris Allan, Jean-Marie Burel, Josh Moore, Colin Blackburn, Melissa Linkert, Scott Loynton, Donald Macdonald, William J Moore, Carlos Neves, Andrew Patterson, Michael Porter, Aleksandra Tarkowska, Brian Loranger, Jerome Avondo, Ingvar Lagerstedt, Luca Lianas, Simone Leo, Katherine Hands, Ron T Hay, Ardan Patwardhan, Christoph Best, Gerard J Kleywegt, Gianluigi Zanetti, and Jason R Swedlow. OMERO: flexible, model-driven data management for experimental biology. *Nature methods*, 9(3):245–53, mar 2012.

[76] Kristian Kvilekval, Dmitry Fedorov, Boguslaw Obara, Ambuj Singh, and B S Manjunath. Bisque: a platform for bioimage analysis and management. *Bioinformatics (Oxford, England)*, 26(4):544–52, feb 2010.

[77] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, Jean-Yves Tinevez, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak, and Albert Cardona. Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7):676–82, jul 2012.

[78] Anne E Carpenter, Thouis R Jones, Michael R Lamprecht, Colin Clarke, In Han Kang, Ola Friman, David a Guertin, Joo Han Chang, Robert a Lindquist, Jason Moffat, Polina Golland, and David M Sabatini. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7(10):R100, January 2006.

[79] Fabrice de Chaumont, Stéphane Dallongeville, Nicolas Chenouard, Nicolas Hervé, Sorin Pop, Thomas Provoost, Vannary Meas-Yedid, Praveen Pankajakshan, Timothée Lecomte, Yoann Le Montagner, Thibault Lagache, Alexandre Dufour, and Jean-Christophe Olivo-Marin. Icy: an open bioimage informatics platform for extended reproducible research. *Nature methods*, 9(7):690–6, jul 2012.

[80] Carsten Haubold, Martin Schiegg, Anna Kreshuk, Stuart Berg, Ullrich Koethe, and Fred A Hamprecht. Segmenting and Tracking Multiple Dividing Targets Using ilastik. *Advances in anatomy, embryology, and cell biology*, 219:199–229, 2016.

[81] T S Yoo, M J Ackerman, W E Lorensen, W Schroeder, V Chalana, S Aylward, D Metaxas, and R Whitaker. Engineering and algorithm design for an image processing API: A technical report on ITK-the Insight Toolkit. *Studies in health technology and informatics*, pages 586–592, 2002.

[82] Ken Martin, Luis Ibáñez, Lisa Avila, Sébastien Barré, and Jon H Kaspersen. Integrating segmentation methods from the Insight Toolkit into a visualization application. *Medical image analysis*, 9(6):579–93, dec 2005.

[83] Bradley C Lowekamp, David T Chen, Luis Ibáñez, and Daniel Blezek. The Design of SimpleITK. *Frontiers in neuroinformatics*, 7:45, 2013.

[84] B Roysam, G Lin, C Bjornsson, and Others. The FARSIGHT project: associative multi-dimensional image analysis methods for optical microscopy. *Rittscher, J. et al*, 2008.

[85] Jonathan Luisi, Arunachalam Narayanaswamy, Zachary Galbreath, and Badrinath Roysam. The FARSIGHT trace editor: An open source tool for 3-D inspection and efficient pattern analysis aided editing of automated neuronal reconstructions. *Neuroinformatics*, 9(2-3):305–315, sep 2011.

[86] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library.* " O'Reilly Media, Inc.", 2008.

[87] Stéfan van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and Scikit image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 2014.

[88] David J. Crossman, Yufeng Hou, Isuru Jayasinghe, David Baddeley, and Christian Soeller. Combining confocal and single molecule localisation microscopy: A correlative approach to multi-scale tissue imaging. *Methods*, 88:98–108, 2015.

[89] Anne E Carpenter, Thouis R Jones, Michael R Lamprecht, Colin Clarke, In H Kang, Ola Friman, David A Guertin, Joo H Chang, Robert A Lindquist, Jason Moffat, et al. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7(10):R100, 2006.

[90] Thouis R Jones, In Han Kang, Douglas B Wheeler, Robert A Lindquist, Adam Papallo, David M Sabatini, Polina Golland, and Anne E Carpenter. Cellprofiler analyst: data exploration and analysis software for complex image-based screens. *BMC bioinformatics*, 9(1):1, 2008.

[91] Fabrice De Chaumont, Stéphane Dallongeville, Nicolas Chenouard, Nicolas Hervé, Sorin Pop, Thomas Provoost, Vannary Meas-Yedid, Praveen Pankajakshan, Timothée Lecomte, Yoann Le Montagner, et al. Icy: an open bioimage informatics platform for extended reproducible research. *Nature methods*, 9(7):690–696, 2012.

[92] Michael D Abràmoff, Paulo J Magalhães, and Sunanda J Ram. Image processing with imagej. *Biophotonics international*, 11(7):36–42, 2004.

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

[94] Beate Neumann, Thomas Walter, Jean-Karim Hériché, Jutta Bulkescher, Holger Erfle, Christian Conrad, Phill Rogers, Ina Poser, Michael Held, Urban Liebel, et al. Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. *Nature*, 464(7289):721–727, 2010.

[95] Dale Muzzey and Alexander van Oudenaarden. Quantitative time-lapse fluorescence

microscopy in single cells. *Annual review of cell and developmental biology*, 25:301, 2009.

[96] Amr El-Labban, Carlos Arteta, Andrew Zisserman, Alexander W Bird, and A Hyman. Mitotic phase based detection of chromosome segregation errors in embryonic stem cells. In *2013 IEEE 10th International Symposium on Biomedical Imaging*, pages 892–895. IEEE, 2013.

[97] Xiaoxu Wang, Weijun He, Dimitris Metaxas, Robin Mathew, and Eileen White. Cell segmentation for division rate estimation in computerized video time-lapse microscopy. In *Proceedings of the SPIE*, 2007.

[98] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[99] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[100] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Fua Pascal, and Susstrunk Sabine. Slic superpixels. *EPFL Technical Report 149300*, (June), 2010.

[101] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, NY, USA, 1991.

[102] W.R. Crum, O. Camara, and D. L G Hill. Generalized overlap measures for evaluation and validation in medical image analysis. *Medical Imaging, IEEE Transactions on*, 25(11):1451–1461, Nov 2006.

[103] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollar. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pages 1990–1998, 2015.

[104] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[105] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[106] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[107] Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2740–2748, 2015.

[108] Thouis R Jones, Anne E Carpenter, Michael R Lamprecht, Jason Moffat, Serena J Silver, Jennifer K Grenier, Adam B Castoreno, Ulrike S Eggert, David E Root, Polina Golland, et al. Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning. *Proceedings of the National Academy of Sciences*, 106(6):1826–1831, 2009.

[109] Jos BTM Roerdink and Arnold Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta informaticae*, 41(1, 2):187–228, 2000.

[110] Vicente Grau, AUJ Mewes, M Alcaniz, Ron Kikinis, and Simon K Warfield. Improved watershed transform for medical image segmentation using prior information. *IEEE transactions on medical imaging*, 23(4):447–458, 2004.

[111] Xin Qi, Fuyong Xing, D.J. Foran, and Lin Yang. Robust segmentation of overlapping cells in histopathology specimens using parallel seed detection and

repulsive level set. *Biomedical Engineering, IEEE Transactions on*, 59(3):754–765, March 2012.

[112] Thouis R Jones, Anne Carpenter, and Polina Golland. Voronoi-based segmentation of cells on image manifolds. In *Computer Vision for Biomedical Image Applications*, pages 535–543. Springer, 2005.

[113] Yasuhiro Miki, Katsuhiko Ono, Shuko Hata, Takashi Suzuki, Hiroyuki Kumamoto, and Hironobu Sasano. The advantages of co-culture over mono cell culture in simulating *in vivo* environment. *The Journal of steroid biochemistry and molecular biology*, 131(3):68–75, 2012.

[114] Antti Lehmussola, Pekka Ruusuvuori, Jyrki Selinummi, Heikki Huttunen, and Olli Yli-Harja. Computational framework for simulating fluorescence microscope images with cell populations. *Medical Imaging, IEEE Transactions on*, 26(7):1010–1016, 2007.

[115] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[116] Dirk Padfield, Jens Rittscher, and Badrinath Roysam. Coupled minimum-cost flow cell tracking. In *International Conference on Information Processing in Medical Imaging*, pages 374–385. Springer, 2009.

[117] Alan Aberdeen. Automated cell-tracking and visualization. Technical report, Department of Engineering Science, University of Oxford, 2016.

[118] Kang Li, Eric D Miller, Lee E Weiss, Phil G Campbell, and Takeo Kanade. Online tracking of migrating and proliferating cells imaged with phase-contrast microscopy. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 65–65. IEEE, 2006.

[119] AJ Hand, T Sun, DC Barber, DR Hose, and S MacNeil. Automated tracking of migrating cells in phase-contrast video microscopy sequences using image registration. *Journal of microscopy*, 234(1):62–79, 2009.

[120] P Vallotton, A Ponti, CM Waterman-Storer, Edward D Salmon, and G Danuser. Recovery, visualization, and analysis of actin and tubulin polymer flow in live cells: a fluorescent speckle microscopy study. *Biophysical journal*, 85(2):1289–1306, 2003.

[121] Ivo F Sbalzarini and Petros Koumoutsakos. Feature point tracking and trajectory analysis for video imaging in cell biology. *Journal of structural biology*, 151(2):182–195, 2005.

[122] Takeo Kanade, Zhaozheng Yin, Ryoma Bise, Seungil Huh, Sungeun Eom, Michael F Sandbothe, and Mei Chen. Cell image analysis: Algorithms, system and applications. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 374–381. IEEE, 2011.

[123] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. Network flows: theory, algorithms, and applications. 1993.

[124] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

[125] Steven R McDougall, ARA Anderson, MAJ Chaplain, and JA Sherratt. Mathematical modelling of flow through vascular networks: implications for tumour-induced angiogenesis and chemotherapy strategies. *Bulletin of mathematical biology*, 64(4):673–702, 2002.

[126] Narsingh Deo. *Graph theory with applications to engineering and computer science*. Courier Dover Publications, 2016.

[127] Gerta Rücker. Network meta-analysis, electrical networks and graph theory. *Research synthesis methods*, 3(4):312–324, 2012.

[128] Bela Bollobas. *Graph theory: an introductory course*, volume 63. Springer Science & Business Media, 2012.

[129] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object

tracking using network flows. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[130] Umesh Bellur and Roshan Kulkarni. Improved matchmaking algorithm for semantic web services based on bipartite graph matching. In *IEEE International Conference on Web Services (ICWS 2007)*, pages 86–93. IEEE, 2007.

[131] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 117–128. IEEE, 2002.

[132] Kaspar Riesen, Michel Neuhaus, and Horst Bunke. Bipartite graph matching for computing the edit distance of graphs. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 1–12. Springer, 2007.

[133] Yuhang Wang, Fillia Makedon, James Ford, and Heng Huang. A bipartite graph matching framework for finding correspondences between structural elements in two proteins. In *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, volume 2, pages 2972–2975. IEEE, 2004.

[134] Armen S Asratian, Tristan MJ Denley, and Roland Häggkvist. *Bipartite graphs and their applications*, volume 131. Cambridge University Press, 1998.

[135] Omar Al-Kofahi, Richard J Radke, Susan K Goderie, Qin Shen, Sally Temple, and Badrinath Roysam. Automated cell lineage construction: a rapid method to analyze clonal development established with murine neural progenitor cells. *Cell cycle*, 5(3):327–335, 2006.

[136] David G Luenberger. *Introduction to linear and nonlinear programming*, volume 28. Addison-Wesley Reading, MA, 1973.

[137] Thomas H Cormen. *Introduction to algorithms*. MIT press, 2009.

[138] Mokhtar S Bazaraa, John J Jarvis, and Hanif D Sherali. *Linear programming and network flows*. John Wiley & Sons, 2011.

[139] Keir C Neuman, Edmund H Chadd, Grace F Liou, Keren Bergman, and Steven M Block. Characterization of photodamage to escherichia coli in optical traps. *Biophysical journal*, 77(5):2856–2863, 1999.

[140] Yang Li, Hao He, Huanyu Song, Bowen Liu, Minglie Hu, and Chingyue Wang. Investigation of photodamage by femtosecond laser to cells via gold nanorods. *Journal of Innovative Optical Health Sciences*, page 1650034, 2016.

[141] Rui-Jun Fan, Qiang Sun, Ling Zhang, Yan Zhang, and An-Hui Lu. Photoluminescent carbon dots directly derived from polyethylene glycol and their application for cellular imaging. *Carbon*, 71:87–93, 2014.

[142] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[143] Eric Jones, Travis Oliphant, P Peterson, et al. Open source scientific tools for python, 2001.

[144] John D Hunter et al. Matplotlib: A 2d graphics environment. *Computing in science and engineering*, 9(3):90–95, 2007.

[145] Daniel A Schult and P Swart. Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conferences (SciPy 2008)*, volume 2008, pages 11–16, 2008.

[146] Dirk Padfield and James Miller. A label geometry image filter for multiple object measurement. *Insight Journal*, 301, 2008.

[147] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.

[148] Andrew Makhorin. Glpk (gnu linear programming kit), 2008.

[149] William E Hart. Python optimization modeling objects (pyomo). In *Operations Research and Cyber-Infrastructure*, pages 3–19. Springer, 2009.

[150] William E Hart, Jean-Paul Watson, and David L Woodruff. Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation*, 3(3):219–260, 2011.

[151] William E Hart, Carl Laird, Jean-Paul Watson, and David L Woodruff. *Pyomo–optimization modeling in python*, volume 67. Springer Science & Business Media, 2012.

[152] Mitko Veta, Paul J Van Diest, Stefan M Willems, Haibo Wang, Anant Madabhushi, Angel Cruz-Roa, Fabio Gonzalez, Anders BL Larsen, Jacob S Vestergaard, Anders B Dahl, et al. Assessment of algorithms for mitosis detection in breast cancer histopathology images. *Medical image analysis*, 20(1):237–248, 2015.

[153] Martin Maška, Vladimír Ulman, David Svoboda, Pavel Matula, Petr Matula, Cristina Ederra, Ainhoa Urbiola, Tomás España, Subramanian Venkatesan, Deepak MW Balak, et al. A benchmark for comparison of cell tracking algorithms. *Bioinformatics*, 30(11):1609–1617, 2014.

[154] U Ziegler and P Groscurth. Morphological features of cell death. *Physiology*, 19(3):124–128, 2004.

[155] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[156] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and Shai Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498. IEEE, 2006.

[157] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):500–513, 2011.

[158] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011.

[159] M J. Johnson and A S. Willsky. Bayesian nonparametric hidden semi-markov models. *J. of Machine Learning Res.*, 14:673–701, 2013.

[160] A Abbott. Sequence analysis: new methods for old ideas. *Annual review of sociology*, pages 93–113, 1995.

[161] A Gabadinho, G Ritschard, M Studer, and N S Müller. Extracting and rendering representative sequences. In *Knowledge Discovery, Knowlege Engineering and Knowledge Management*, pages 94–106. Springer, 2011.

[162] S Aisenbrey and A E Fasang. New life for old ideas: The second wave of sequence analysis bringing the course back into the life course. *Sociological Methods & Research*, 38(3):420–462, 2010.

[163] Martin Längkvist, Lars Karlsson, and Amy Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42:11–24, 2014.

[164] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.