



Technische Universität München

Department of Mathematics

Chair of Mathematical Modeling of Biological Systems

Technische Universität München

Master's Thesis in Bioinformatics

Single-cell analysis of cancer drug response using computer vision and learning algorithms on time-lapse micro-trench data

Pandu Raharja



Technische Universität München

Department of Mathematics
Chair of Mathematical Modeling of Biological Systems

Technische Universität München

Master's Thesis in Bioinformatics

**Single-cell analysis of cancer drug response using
computer vision and learning algorithms on time-lapse
micro-trench data**

**Wirkungsanalyse von Krebsmedikamenten in Einzeller
Auflösung durch die Anwendung von Computer-Vision-
und Machine-Learning-Algorithmen auf Microtrench-
Videoaufnahme**

Author: Pandu Raharja

Supervisor: Prof. Dr. Fabian Theis, Dr. Carsten Marr

Advisor: Prof. Dr. Fabian Theis

Prof. Dr. Dmitrij Frishman

Submitted: 15.12.2017

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe. Weiterhin erkläre ich, eine Diplomarbeit in diesem Studiengebiet erstmalig einzureichen.

München, den February 1, 2018

Statement of authorship

I declare that I completed this thesis on my own and that information which has been directly or indirectly taken from other sources has been noted as such. Neither this nor a similar work has been presented to an examination committee.

Munich, February 1, 2018

Acknowledgement

First and foremost, I would like to thank mom and dad, for the support even from faraway lands. Even though we are continents apart, your pray and hope will be always with me.

To Jennifer Carissa, whose support has made it possible to go through the hard parts of my life in the last years. This thesis is only possible because of you.

To Julian, Ganesh and other best friends which I resort to for personal assistance. Friends really are hard to find.

And finally, to Prof. Fabian Theis, Prof. Joachim Rädler, Elisavet Chatzopoulou, Felix Buggenthal and Alexandra Murschhauser for tremendous assistant in doing this project. Never did I know a project could be done in this maner.

Finally for my best supervisor, Carste Marr. Your boundless patience, deep insight and attention to detail have made this project impossible without your selfless assistance. I hope we could do more things in the future.

Terminology and abbreviations

Following terms are used frequently in this thesis and therefore merit special explanation:

| Name | Explanation |
|--------------|---|
| micro-trench | Single unit of micro-trench inside a well |
| wafer | The plate where the micro-trench is manufactured on |
| well | Macro-containment where wafer is located, usually part of larger wells installation (2 by 4 wells in our case). |
| sticky slide | The containment box where wells are located in |
| slice | An image |
| stack | An ordered sequence of images (usually coming from the same channel) |
| channel | A connection between data source (camera) and storage. Images coming from the same channel have same capture properties formatting. |

Note that in this thesis, following things are used interchangeably:

- image and slice, while referring to an image particularly a capture image of the trenches and its processed versions.
- image and figure, while referring to an image shown in this paper in general.

Symbol Explanation

| | |
|----------------------|--|
| B_t | A 1-bit monochrome image taken at time t (t -th image) |
| G_t | Grayscale image at time t |
| $\mathbb{G}_{x,y}$ | Sobel gradient of pixel $M_{t,x,y}$ |
| $\mathfrak{g}_{x,y}$ | Quadratic Sobel gradient of pixel $M_{t,x,y}$ |
| $G(x)$ | One dimensional Gaussian function |
| $G(x, y)$ | Two dimensional Gaussian function |
| $I_t(x, y)$ | Intensity function for pixel (x, y) of RGB image taken at time t |
| $J_t(x, y)$ | Brightness and contrast transforming function for pixel (x, y) of grayscale image taken at t |
| M | A stack of images |
| M_t | An RGB image captured at time t |
| $M_{t,x,y}$ | Pixel at (x, y) in M_t |
| $M_{t,x,y,1}$ | First channel of pixel $M_{t,x,y}$ |
| M_t^b | A b -bit encoded RGB image captured at time t |
| $M_{t,x,y,i}^m$ | Pixel-wise mean-subtracted pixel $M_{t,x,y}$ |
| $\langle M \rangle$ | Pixel-wise mean image of stack M . |
| T_R | Gradient threshold value in a quadtree (used in RATS algorithm). |

Abstract

Quantitative measurement of cancer drug response is essential to objectively gauge the efficacy of cancer drugs. So far, there has been no method to track and quantitatively measure single-cell response of cancer drug treatment. A novel pipeline is presented in this thesis. First, a quasi-high-throughput method to track cells and quantitatively analyze single-cell response to drugs. We investigate the response of model cancer cell lineages, MOLM and Jurkat, to known anti-cancer drugs Vincristine and Doxorubicine. While the method enabled relatively easy and quasi-high-throughput analysis of cancer treatment *in vitro*, our pipeline could also be adapted in various contexts involving single-cell analysis with reasonable amount of modifications necessary.

Contents

| | |
|--|-----------|
| Chapter 1 Introduction | 5 |
| 1.1 Variability of cell-to-cell response towards cancer treatment | 6 |
| 1.2 Hypothesis | 7 |
| 1.3 Advances in microfluidics, image processing and machine learning | 7 |
| 1.3.1 Microfluidics | 8 |
| 1.3.2 Image Processing and Computer Vision | 8 |
| 1.3.3 Machine Learning | 9 |
| 1.4 Comparison with other methods | 9 |
| 1.4.1 Cell death determination | 9 |
| 1.4.2 Other model cell lines | 9 |
| 1.5 The structure of the thesis | 9 |
| Chapter 2 Data and Methods | 11 |
| 2.1 Experimental setting and data | 11 |
| 2.1.1 Cell culture | 11 |
| 2.1.2 Microfluidics | 12 |
| 2.1.3 Cancer treatment regimes | 15 |
| 2.1.4 Cell death signal | 18 |
| 2.1.5 Experiment setup and image capturing | 20 |
| 2.2 Definitions | 22 |
| 2.2.1 Image channels and encoding | 22 |
| 2.2.2 Intensity, brightness and contrast | 26 |
| 2.2.3 Robust automatic threshold selection (RATS) | 28 |
| 2.2.4 Holes filling | 29 |
| 2.2.5 Gaussian blur | 30 |
| 2.2.6 Contrast limited adaptive histogram equalization (CLAHE) | 31 |
| 2.2.7 Subtraction by pixel-wise mean intensity | 34 |
| 2.2.8 Cell recognition | 35 |
| 2.2.9 Cell tracking | 36 |
| 2.2.10 Shift correction | 38 |
| 2.2.11 Support vector machine (SVM) | 41 |
| 2.2.12 Pearson's correlation coefficient (PCC) | 45 |
| 2.2.13 F-Test | 45 |
| Chapter 3 Analytic Pipeline | 47 |
| 3.1 Image computing | 47 |
| 3.2 Micro-trench masking | 47 |
| 3.2.1 Brightness and contrast adjustment of in-focus image | 48 |
| 3.2.2 RATS denoising | 49 |
| 3.2.3 Micro-trench mask filling | 49 |

| | | |
|---------------------------------------|---|-----------|
| 3.3 | Single cell tracking in the brightfield channel | 49 |
| 3.3.1 | Gaussian blurring | 50 |
| 3.3.2 | Local contrast normalization | 50 |
| 3.3.3 | Mean intensity correction | 50 |
| 3.3.4 | Cells recognition | 50 |
| 3.3.5 | Cells tracking | 51 |
| 3.4 | Single cell tracking in the fluorescent channel and cell death signal determination | 51 |
| 3.4.1 | Brightness and contrast adjustment | 52 |
| 3.4.2 | Mean intensity correction | 52 |
| 3.4.3 | Cells recognition and tracking | 52 |
| 3.5 | Pipeline visualization | 54 |
| 3.6 | Implementation and availability | 59 |
| Chapter 4 Results | | 60 |
| 4.1 | Distribution of cell cycle duration times | 60 |
| 4.2 | Variability of time-to-death between sister cells | 61 |
| 4.3 | Variability of time-to-death by time-in-cycle | 61 |
| 4.4 | Machine learning models of cell death | 62 |
| Chapter 5 Summary and Outlook | | 63 |
| 5.1 | Fully automated microfluidics pipeline | 64 |
| 5.2 | Fully automated pipeline for single-cell analysis | 64 |
| 5.2.1 | Automated micro-trench masking | 65 |
| Chapter A Tables | | 78 |
| A.1 | Inferred shift in wells | 78 |
| Chapter B Experiment Protocols | | 79 |
| B.1 | Images acquisition | 79 |
| B.2 | Placeholder | 79 |

List of Figures

| | | |
|------|--|----|
| 2.1 | ibidi® sticky-Slide 8 Well | 13 |
| 2.2 | The structure of micro-trenches | 14 |
| 2.3 | The typical view of micro-trench setting | 15 |
| 2.4 | The structure and mechanism of Vincristine | 16 |
| 2.5 | The structure and mechanism of Daunorubicin | 18 |
| 2.6 | The structure and mechanism of Caspase 3/7 | 19 |
| 2.7 | The structure and mechanism of Propium Iodide (PI) | 20 |
| 2.8 | Effect on focal plane on out-of-focus image | 23 |
| 2.9 | Spectral sensitivity of human | 24 |
| 2.10 | The in-focus image position 41 from unsynchronized experiment. | 25 |
| 2.11 | Normalized distribution of intensity of Figure 2.10. | 25 |
| 2.12 | Brightness and contrast transformation | 27 |
| 2.13 | Example of quadtree | 29 |
| 2.14 | Example of image contours | 30 |
| 2.15 | Visualizatoin of CLAHE algorithm | 33 |
| 2.16 | Example of application of CLAHE algorithm | 33 |
| 2.17 | A LoG operator. Notice the lack of dimension in an operator as it transforms an image (in one space) into another image (still in the same space). | 35 |
| 2.18 | Example of application of the LoG operator | 36 |
| 2.19 | Example of search grid from the shift inference algorithm | 40 |
| 2.20 | Slice-wise sum of pixel difference in a stack | 41 |
| 2.21 | Illustration of SVM in 2D | 42 |
| 3.1 | Example of intensity values along the vertical axis in an out-of-focus image | 48 |
| 3.2 | Example of images from green and red fluorescent channels | 53 |
| 3.3 | Illustration of annotation of PI death signal | 54 |
| 3.4 | Brightfield channel pipeline | 56 |
| 3.5 | Fluoroscence channel pipeline | 58 |
| 3.6 | Combination of tracking data with cell death information creating cell death signal-corrected tracks data | 59 |

List of Tables

| | | |
|-----|---|----|
| 2.1 | Configuration table of treatments for synchronized experiment | 21 |
| 2.2 | Image capture frequency of the experiment | 21 |
| 2.3 | Configuration table of drug treatments for synchronized experiment | 21 |
| A.1 | The inferred shift (in pixels) for each image position in unsynchronized experiment. The shifts are inferred using the shift correction algorithm explained in Subsection 2.2.10. | 78 |

Chapter 1

Introduction

Cancer is among the deadliest diseases ever known to human being. It is a leading cause of death in 2009, second only to cardiovascular diseases [1]. The numbers are disconcerting, especially in the developed world. In the United States alone, half of men and a third of women are expected to develop some kind of cancer. According to US government, in 2016 alone, an estimated 1,685,210 people will be diagnosed with cancer, while 595,690 more will die from it [2].

Worldwide, the International Agency for Research on Cancer's GLOBOCAN series report that, in 2014, [3].

I.

- readable to people without background in the fields
- non technical at all

II.

- what have the researches done
- biologics
- technicals

III.

thesis overview

4~8 pages

Generally, following questions are to be investigated in the experiment:

- We would like to investigate the dynamics of the AML-M5a MOLM-13 cell line's response towards chemotherapeutic treatment regimes in various drugs' concentration levels (for both Vincristine and Daunorubicin).
- We also would like to look into the effect of cell cycle on the AML-M5a MOLM-13 cell line's response to drugs. Specifically, following questions need to be answered: does the response of cancer cells depends on cell cycle?

To answer aforementioned questions, we designed the experiments as follow.

1.1 Variability of cell-to-cell response towards cancer treatment

Cell-to-cell variability in response to external stimuli is a pervasive trait in cellular systems that prevails even in isogenic cell populations. Here, heterogeneity might be caused by epigenetic modifications, differences in the cell cycle phase, or by intrinsic stochastic fluctuations in gene expression and biochemical regulation. The implications of heterogeneity for cancer progression and treatment are poorly understood. In some cases, heterogeneity is dominated by intrinsic fluctuations in the stochastic expression of key regulators that randomly alter the sensitiveness of individual cancer cells. A raw model of this scenario has been put forward in recent work on TRAIL-induced apoptosis [1, 2]. Experiments and simulations show that variability in cell fate is sensitive to small stochastic increases in the levels of Bcl-2 and are transiently heritable to siblings [3, 4]. The study on TRAIL-induced apoptosis led to a novel interpretation of fractional killing and predicts reversible resistance to chemotherapy. The profound consequences for cancer treatment have been subject to theoretical studies on the stochastic origins of cell-to-cell variability in cancer cell death decisions [5-7]. Cancer is an intrinsically highly diverse disease; tumors of any different histological type not only exhibit genetic diversity but also display their variation when exposed to all forms of chemotherapy [8]. Most state of the art chemotherapeutic drugs in clinical use, target rapidly dividing cells and trigger apoptosis. Vincristine (VCR), an antitumor vinca alkaloid, binds to tubulin and stops the dividing cell from separating its chromosomes during metaphase in M-phase. It is thus considered an M-phase dependent drug [9]. In contrast, daunorubicin, an anthracycline aminoglycoside antineoplastic, intercalates on DNA and inhibits the function of the enzyme topoisomerase II during transcription and replication. Daunorubicin is thus expected to act throughout the whole cell cycle and but especially strong during the DNA replication in the S-phase. Both drugs are used to fight hematopoietic disorders such as Acute Myeloid Leukemia (AML)[10, 11] among other neoplasms. In the literature, a drug that interferes with the cell cycle is in general considered cell cycle dependent, but in practice it is not clear whether it only acts on a specific cell cycle phase, due to side effect toxicities. Based on this, it is hypothesized that both VCR and daunorubicin are cell cycle dependent. Since it is assumed that VCR acts only during M-phase in their cycling, we can expect that cells that are closer to M-phase will die earlier [12, 13]. It is essential to understand sources of heterogeneous response of cancer to therapy and , in order to design novel therapeutic strategies and potent agents, that not only targets key signaling pathways with high specificity but also address the contextual role of cell cycle timing in cancer therapy. The response of cells to chemotherapeutic drugs has to be investigated. In this context, time-lapse imaging, which allows for recording accurate histories of individual cancer cell fates and cancer subpopulations, received increasing attention [12, 14, 15]. However, the to study the effect of particular cell cycle phases on chemo sensitiveness, single cells have to be observed continuously throughout division, drug addition, and death. has not been explored at the single cell level. The typical bottleneck for a high-throughput analysis for such is cell tracking. Tracking single cells in time-lapse microscopy movies is a challenging problem. Different automatic image analysis tracking tools have been proposed [19,20] and compared [16,17] [16-20] but to achieve tracks for fast moving cells movements, high cell densities, challenging cell identification, and long-term observations, also time-intensive manual tracking becomes necessary to achieve accurate tracksis often applied

[18,21] to generate tracks with maximal accuracy. For many approaches, the workload of manual tracking has to be compared to correcting erroneous track from tracking algorithms [schrÃP_der review]. [21] . Confining cells spatially obviously reduces possible tracking errors and facilitates the application of tracking algorithms. In particular, non-adherent cells are painstaking to track since diffusive and convective drift in long-term measurements complicate cell assignments [20]. Among the techniques to capture non-adherent cells for long-term observation microfluidic devices [22] as well as micro-well platforms [23-28] were have been developed. Alternatively, individual cell cycle phases were imaged using fluorescent cell cycle reporters [29]. The confinement of single cells into well-defined spatial structures provides a straightforward implementation to facilitate automatic tracking since long-term crossing of individual tracks is avoided. PConsequently, platforms that confine single starting cells and thus lead to spatially separated cell families (also called clones) are an especially derived from a single mother cell are a useful tool to investigate cell cycle length, sister cell correlations, or the effect of cell cycle phases to to enable the automatization of the image analysis and to yield in a faster and effortless way to collect time traces that can address questions regarding the sources of cell-to-cell variability in a high-throughput manner. Here we introduce a platform that enables the continuous observation of cell families derived from individual non-adherent cells of the leukemia cell line MOLM-13. The platform employs arrays of micro-trenches optimized to observe cells for two consecutive generations. We demonstrate that automated image analysis is feasible and allows for precisely determination determiningof the cell division cycle time distribution and sister cell correlations. A key feature of the platform is the direct and parallel observation of many hundreds of cells with individualin different cell cycle statess. We show that the time-to-death after induction of apoptosis of the leukemia cell line MOLM-13 using the anti-mitotic agent vincristine (VCR) and daunorubicin (dauno) has a small dependence on cell cycle . The results were found consistent with experiments using cells that were synchronized using standard thymidine cell cycle arrest. Using arrays of micro-trenchesOur approach also enabled a time correlation analysis which showed that the time-to-death of daughter cells correlates with the time spent in cell cycle, while this effect is not detectable when cells are synchronized.

1.2 Hypothesis

In this part, the hypothesis underlying the experiment is presented.

1.3 Advances in microfluidics, image processing and machine learning

In the recent decades, as in many other fields, there are numerous groundbreaking advances in the fields of microfluidics, image processing and machine learning.

The advances, coupled with general technological advances in computing power and energy efficiency, have made it possible for us to design almost completely automated analytical pipeline for single-cell microfluidic system.

While not all methods and algorithms in this project are the most recent, many of them are state-of-the-art and/or good enough for the pipeline to work seamlessly.

1.3.1 Microfluidics

As the name suggests, 'microfluidics' concerns the manipulation of fluids in a small working dimension, typically starting from nanometers to lower millimeters [4]. In modern context, the entire fields tries to find application of micro-sized devices which hold and control the state of liquid [4] including cell culture medium. There are two categories of microfluidics: active and passive microfluidics devices. The separation is based on the device's ability to actively manipulate the flow and control of devices [5]. Active devices use micro-valves to perform sophisticated chemical processes [6]. This goes as far as reactions at individual cell level [7]. A passive device, which our micro-array system is, exploits on the other hand its physical property to provide rapid controlled environment for micro-sized experiments.

Active microfluidic methods for analysis and manipulation of biological cells have been done in various way and form. In 2003, Wheeler *et al* developed a novel microfluidic device from poly-dimethylsiloxane using multilayer soft lithography technology for the analysis of single cells [8]. The microfluidic setup facilitates the passive and gentle separation of a single cell from the bulk cell suspension. This in turn enables the precise delivery of reagents as little as one nanoliter to the cell. In other use cases, the optical-based microfluidic methods have been used to sort cell with very high accuracy [9]. This family of method utilizes the fact that different dielectric particles respond differently to an applied light field [10]. Combined with the minuscule spatial setting, the methods are compatible for single-cell resolution analysis. For example, optical-based microfluidic methods have been used to sort cells with very high accuracy [9, 11, 12].

As a method, passive macrofluidic methods are mostly used to provide specialized environment in cell-size resoultion. For example, microfluidic settings have been used to keep spatio-temporal identity of single cell for the analysis of the underlying biological dynamics of the isolated cells [13, 14], which form the methods the design and synthesis of our micro-trench system are based on.

In the last decades, the recent advances in both passive and active microfluidics have created an entire field [4] with use cases ranging from *in vivo* imaging [15], single-cell analysis [8] to cellular biophysics [16]. This project leverages in these advances coupled with equally outstanding advances in image acquisition, image processing and machine learning methods which will be described in the subsequent subsections.

The use of microfluidics enables us firstly to study the distribution of division times among single cells and also to correlate the division times between sister cells, which are genetically very similar. Secondly, the array of micro-trenches enabled the study of the response dynamics of single-cells to doxorubicin, a widely used chemotherapeutic drug, and the comparison of the response to this agent between a chemically synchronized and a non-synchronized population

1.3.2 Image Processing and Computer Vision

Before the runaway advance of deep learning methods in the last years, object and area detection methods generally rely on discriminating certain patterns and features in the picturek – with and without utilizing machine learning methods. In 2001, Paul Viola and Michael Jones proposed a method that was able to recognize face [17], and later, objects in almost real-time fasion [18].

1.3.3 Machine Learning

1.4 Comparison with other methods

1.4.1 Cell death determination

Measuring cell death is a crucial part of the experiment, as the reliable determination of it is the basis of most analysis in this thesis. There are several way to measure cell deaths with varying complexity and accuracy. Each method contains certain assumptions of cell death.

For example, determining cell death by cell movement assumes death of a cell if no movement beyond random flux is observed in certain amount of time. This obviously has certain drawbacks, such as when the observation is done in non-static environment. Moreover, defining the limit of the random flux, above which a given cell is assumed to actively move, is not a trivial task. Some kind of gold standard for a given cell line and environment has to be established manaully, which is very time consuming. This fact is again made even more complicated by the fact that many cells show different movement pattern upon introduction of treatment. It is well known that some cells tend to move faster or slower under stress, the situation many cancerous cells in our experiment will experience upon addition of cancer drug treatment [19, 20, 21].

The second method is using cell size. During apoptosis, the cells would shrink. Given It is known that cell size

1.4.2 Other model cell lines

Some other cell lines were examined as potential probe cell line in this experiment. One of them is Jurkat Cell, a model cell commonly used to study T Cell Leukimia, T cell signalling mechanism and the expression of various HIV-related chemokines [22]. The cell line was a considered since it is well-studied [23, 24]. This is especially true if we consider apoptotic mechanism of the cell line, a problem this project and other related projects by our and partner labs are trying to investigate. There are several seminal publications about the dynamics of apoptotic mechanism of Jurkat cells we could well compare our results to [25]. Samali *et al* [26] even studies the dynamics of Caspase expression in Jurkat cells, a topic dealt a lot in this project as the chapters progress (see for example Subsection 2.1.3) while Kasai *et al* considers the aspect of spindle checkpoint in the context of apoptotic cell death [27]. However, we figured out early on that the cell motility of the Jurkat cell line was increased dramatically (a phenomenon observed by others before us [28]) upon the introduction of chemotherapeutic treatment – the increase dramatic enough that the cells managed to escape the micro-trench it initially landed in.

1.5 The structure of the thesis

The thesis is presented as two closely-related things. First, we present high throughput system that enables analysis in single-cell resolution of cells' response towards certain chemical reaction and a software suite that processes, analyzes and visualizes the data.

This is an end-to-end solution that possibly can help researcher in their research. Second, we apply this method on the main question of the project: the response dynamics of cancer cell towards chemotherapeutic treatment. Here several questions are posed and addressed using the system and software suit.

Chapter 2 contains

Chapter 2

Data and Methods

As mentioned in previous parts of this thesis: this project consists of three part – the problem statement regarding the dynamics of cancerous single-cells under pressure of treatment, the microfluidics which enables the single-cell protocol and the software implementation used to process and analyze the time-lapse data coming out of the experiment.

This chapter considers two aspects of the project: the experimental setting and the theoretical aspects behind the data analysis pipeline. In the first half of the chapter, we deal mostly with the experimental background and the underlying questions of single cell dynamics of cancer cell under stress with focus on chemotherapeutic pressure. In this part, the highlight of the experiment, the microfluidic system for cell containment is elucidated. In conjunction with the system, some biomedical and biochemical aspects of the experiment are also mentioned. This includes the drug, the auxiliary chemicals used in the experiment and the cell lines probed. The second part deals mostly with the quantitative methods and algorithms used to process data into meaningful observations. This part is opened with definitions used in the methods section.

2.1 Experimental setting and data

2.1.1 Cell culture

To enable cross reference and comparability of experiment results, a model cell line is used: the acute monocytic leukemia (AML) cell line **AML-M5a MOLM-13**. The line used in our experiment was derived from the cell line initially described by Matsuo et al in 1997 [29]. In the paper, the authors developed the line from the peripheral blood of a relapse patient with AML of subtype FAB M5a, which is characterized by predominantly monoblastic leukemia cells visible in pap smear [30]. Due to extensive research done on the cell line and the well-explained mechanism of the cell line dynamics and response towards cancer medications, the cell line is an ideal candidate for *in vitro* study of monocytic differentiation, leukemogenesis and treatment dynamics [29, 31, 32].

For our experiment, the AML-M5a MOLM-13 cell line was cultured in Gibco® RPMI 1640 GlutaMAX medium, produced by Life Technologies [33]. The medium is popular choice in human cell biology for both experiments and biological syntheses using human cells and their derivatives [34, 35]. It is pre-supplemented with stable form of L-glutamine to prevent ammonia buildup, a common and serious problem in cell culture due to its cell toxicity[36]. The medium is further supplemented with Gibco® Fetal Bovine Serum

(FBS), also offered by Life Technologies [37], as supplement for the AML-M5a MOLM-13 cell culture.

2.1.2 Microfluidics

In our cases, the microfluidics is designed by the Soft Condensed Matter Group at the Faculty of Physics at the Ludwig-Maximilians-Universität München. There are several papers related to the system. For example, in 2013, Marel et al proposed the method of creating micro-wells for single-cell containment based on three-dimensional poly(ethylene glycol)-dimethacrylate (PEG-DMA) microstructures [38]. Later on in 2015, Sekhavati et al publishes the first design of micro-trench arrays that are used in our experiment [14, 5].

In order to track non-adherent cells in a label-free manner over several generations, a set of micro-trenches ($30 \times 120 \mu\text{m}$) out of PEG-DA (Polyethylene(glycol) Diacrylate), which can accommodate four to six cells are designed and fabricated. The platform facilitates cell tracking leading to the observation of hundreds of families of cells, derived from one single mother at each case.

First, the SU-8 (MicroChem Corp, USA) wafer was fabricated in a cleanroom facility using a ProtoLaser LDI system (LPKF Laser & Electronika, Naklo, Slovenia), with a 375 nm wavelength laser and $1 \frac{1}{4}\text{m}$ spot diameter. Polydimethylsiloxane (PDMS) prepolymer solution is mixed with the crosslinker in a 10:1 ratio (w/w) (Sylgard 184, Dow Corning, USA) and then degassed under vacuum. PDMS is then purred on the SU-8 wafer, degassed and cured in 50 °C. The resulting PDMS stamp is peeled off the wafer and cut into appropriate shapes. The PDMS pieces, with 25 1/4 pillars in height, are activated with argon plasma and then immediately placed upside down on a silanized with TMSPMA (3-(Trimethoxysilyl)propyl methacrylate, Sigma-Aldrich) glass coverslip. A solution of PEG-DA (Mn=258) containing 2% v/v of the 2-hydroxy-2methylpropiophenone (both from Sigma-Aldrich, Germany) is freshly prepared and then a drop is deposited at the edge of the PDMS stamp. The PDMS stamp is filled by capillary force induced flow. PEG-DA is then polymerized in an UV-ozone cleaning system (UVOH 150 LAB, FHR, Ottendorf, Germany). Next, the PDMS stamps are peeled off and the resulting micro-trenches of cross-linked PEG-DA are dried in an oven (Binder GmbH, Tuttlingen, Germany) overnight at 50 °C. Finally, the slides are sonicated with 70% ethanol and distilled water before a sticky slide is attached on top (8-well sticky slide, ibidi GmbH, Munich, Germany).

The design of the micro-trench and the schematic representative of cell tracking are seen in Figure 2.2. After the manufacturing process the structure is divided into two main parts:

- The micro-trenches are The smallest structure of the setting, measuring about 120 microns by 30 microns. The base of the trench is made of Polyethylene (glycol) Diacrylate (PEGDA), an inert substance commonly found as construction material in microfluidic system [14]. Each well contains about 2400 micro-trenches contained in one containment box.
- The sticky slide contain 8 wells where the wafer containing micro-trenches is located. The sticky slide chosen for containing the wafers holding the micro-trenches is ibidi® sticky-Slide 8 Well (see Figure 2.1). In the project, each cell treatment is isolated in one containment box. This ensure the separations of different chemicals used in each treatment.

Due to limitations in the image capturing coverage area, each well is further divided into eight image positions. Each image position represents an area the camera captures the full image of. In each well there are 7 to 8 image positions and in one experiment with eight wells there are 63 image positions in total. Tables 2.1 and 2.3 show the list of all image positions in the experiment and their drugs concentration.

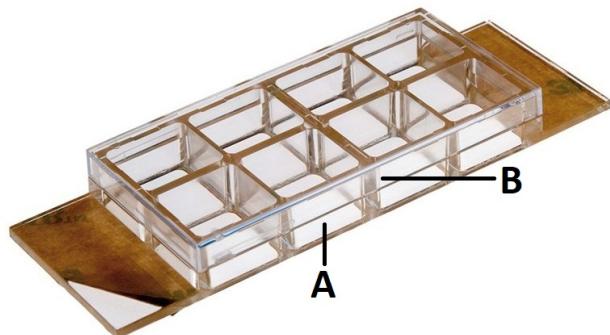


Figure 2.1: ibidi® sticky-Slide 8 Well. The base SU-8 wafer is located in each of the containment box (**A**). The SU-8 wafer is then fabricated in the surface of each containment box using nano photolithographic printing. The microfluidic system is then poured and stamped on top of the wafer. Note that each containment box is upside-open. The cap ((**B**) is used to prevent the ingress of foreign materials into the medium. *Image taken and modified from ibidi GmbH's website.*

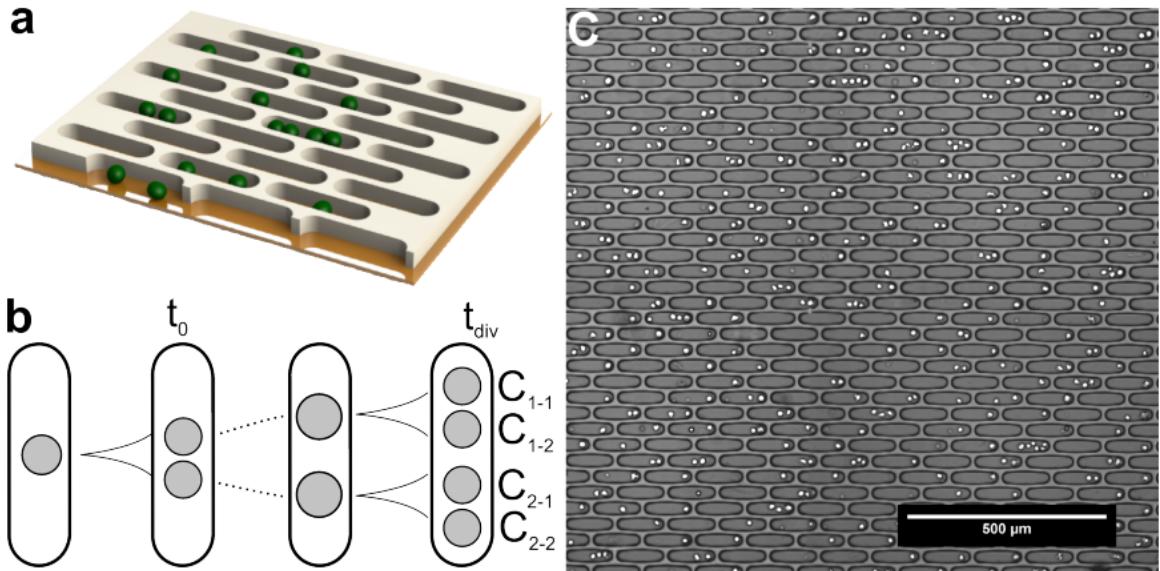


Figure 2.2: The structure of micro-trenches: **(A)** 3D model of micro-trench on surface with cells inside. **(B)** The schematic representation of a time-lapse in a trench. First, a singly-placed cell is tracked in a micro-trench. At time t_0 , the cell divides into two daughter cells. The two cells will be kept tracked until at one point each of the daughter cells will divide at the same time at time t_{div} . Note the simplification of the sample. First, not every cell is singly-placed inside a trench. Indeed, not every trench is occupied by cells. Second, not every cell divides. Third, not every cell line observed has three generations in it. And finally, not every children's division times are at the same time. Indeed, this special case almost never happens in real life. **(C)** The sample view into the environment with cells occupying some micro-trenches. Here, the micro-trenches have dimension of 120 μm long and 30 μm wide. Note also the pointish characteristic of the cells taken in out-of-focus fashion. This improves the performance of the tracking algorithms. *Figure taken from (Sekhavati, 2015) [5].*

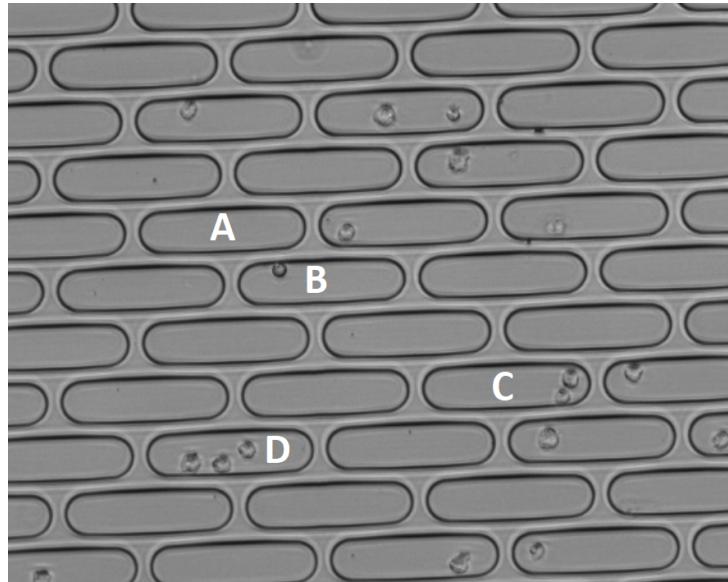


Figure 2.3: The typical view of micro-trench setting. Some trenches contain no cell at all (**A**). Several trenches contain exactly one cell (**B**). A few more trenches contain two cells (**C**) while in rare cases the trench may contain more than two cells (**D**).

2.1.3 Cancer treatment regimes

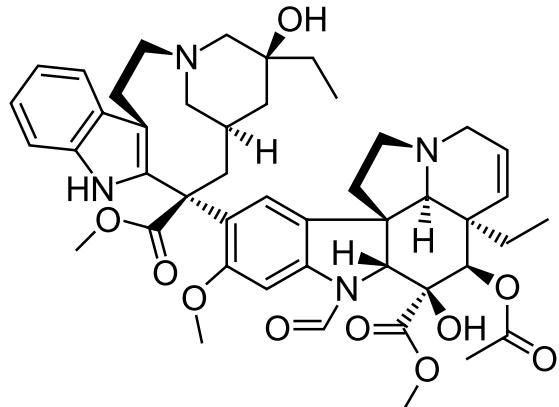
As mentioned in previous chapters, the objective of the project pertains mostly the dynamics of single cancer cell under treatment. After mentioning the cell lines of interest (Chapter 2.1.1) and the microfluidic setup used to contain cancer cells in single-cell setting (Chapter 2.1.2), we arrived at the last aspect of the experimental setting: the chemical treatment used on the cells.

For cell cultures mentioned in the previous subsection, two drug treatment regimes are developed: Vincristine and Daunorubicin – both chemotherapeutic compounds well known for treating various kinds of cancer [39, 40, 41, 42].

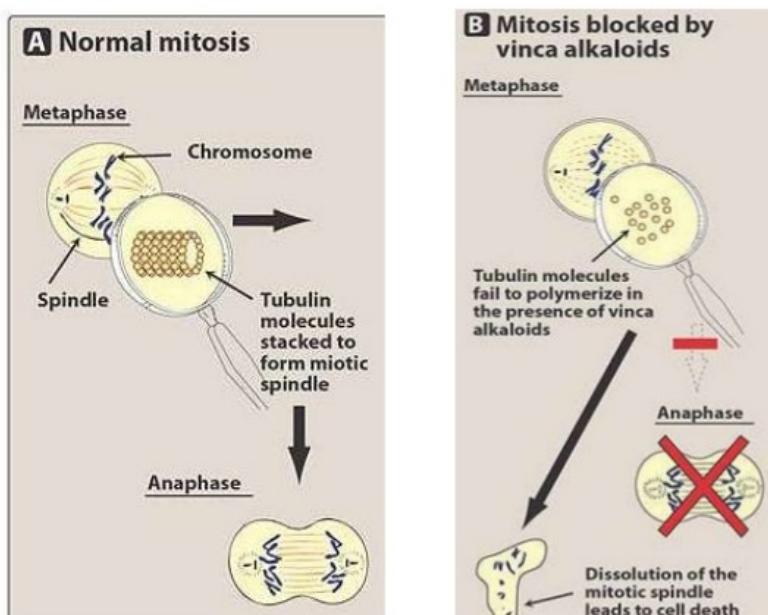
Vincristine

Vincristine, a vinca alkaloid, is initially isolated from the Madagascar periwinkle *Catharanthus roseus* (basionym *Vinca roseus*, hence the name) [43]. It is mainly known as Tubulin polymerase inhibitor [44], a subclass of the mitotic inhibitor family of drugs [45]. Mechanistically, it prevents Tubulin polymerization in two ways. First, it binds elongating Tubulin polymer and reduces the affinity of the elongating polymer [46] towards Tubulin monomers therefore preventing the monomers to bind on the elongating polymer. Meanwhile, further elongation by the polymers is also prevented via allosteric inhibition. It has also been shown that Vincristine depolymerizes stable microtubuli in the axonal part of rats' neuron cells [45]. Due to these mechanistic actions, the effect of Vincristine is most emphasized during the time of high Tubulin synthesis, e.g. during the separation of chromosomes in the Metaphase by means of tearing them with the simultaneous pulling and pushing mechanism of Tubulin poly- and depolymerization [47]. In the context of chemotherapy, Vincristine is often used as combination in CHOP (cyclophosphamide, doxorubicin, vincristine, and prednisone) regime [48] against non-Hodgkin's lymphoma; MOPP [49], COPP [50] and BEACOPP [51] regimes against Hodgkin's lymphoma; and

Stanford V regimes against acute lymphoblastic leukemia [52]. It is also used to certain degree as immunosupresant due to its mitotic inhibiting characteristics [53].



(a)



(b)

Figure 2.4: (a) The molecular structure of Vincristine. (b) Visualization of Vincristine's mechanism of action. During normal metaphase, microtubuli elongate from centrosome towards cell equator while pulling apart fully replicated chromosomes. Vincristine and other vinca alkaloids prevent this from happening by competitive inhibition, allosteric inhibition and active depolymerization of extending microtubuli. Unsuccessful exit from the metaphase forces the cell to undergo programmed cell death.

Daunorubicin

Daunorubicin is initially isolated from bacterium *Streptomyces peucetius* [?, ?]. It is part of the anthracycline class of drug [42] extracted mainly from *Streptomyces* bacte-

ria. Some well-known members of this class are Doxorubicin [54], Epirubicin [55] and Idarubicin [56]. Together, they are among the most effective cancer drugs ever deployed and are used towards more kinds of cancer than other group of chemotherapeutic agents [57, 58, 59]. Like many chemotherapeutic agents including Vincristine, anthracyclines attack cancerous tissues and cells by preventing their division [39]. Unlike vinca alkaloids however, anthracyclines prevent the division by disrupting another mechanistic part of cell division: the DNA polymerization [42]. There are four ways anthracyclines disrupt DNA polymerization:

- Anthracyclines intercalate with base pairs involved in DNA polymerization thus preventing the extension of DNA strands [60].
- Anthracyclines covalently inhibit type II topoisomerase which is responsible for RNA and DNA supercoil relaxation [61]. Inhibition of type II topoisomerase causes supercoiled RNA and DNA to be inaccessible for initiation of duplication thus breaking the DNA [62].
- Anthracyclines induce oxidative stress on cancer cell organelles by generating free oxygen radicals. These radicals in turn damage DNA, proteins and cell membranes and initiate Caspase induced apoptosis [63, 64].
- Anthracyclines disrupt epigenetic, transcriptomic and DNA upstream regulations by removing histones from DNA strands [65]. This also exposes DNA strands to degradation factors such as DNA methylase [66] and oxidative damage [67]
- In the presence of formaldehyde, anthracyclines cross-link with the DNA covalently, creating cytotoxin that will disrupt DNA from functioning properly [68].

In normal chemotherapeutic regimes, both Vincristine and Daunorubicin are prescribed intravenously to patients [69]. Needless to say, both drugs will disrupt both cancerous and healthy cells. The effect is however mostly felt in actively dividing cells and organs such as blood cells and hair follicles due to chemotherapeutic agents' highly disruptive effect during cell division as mentioned above [69, 70].

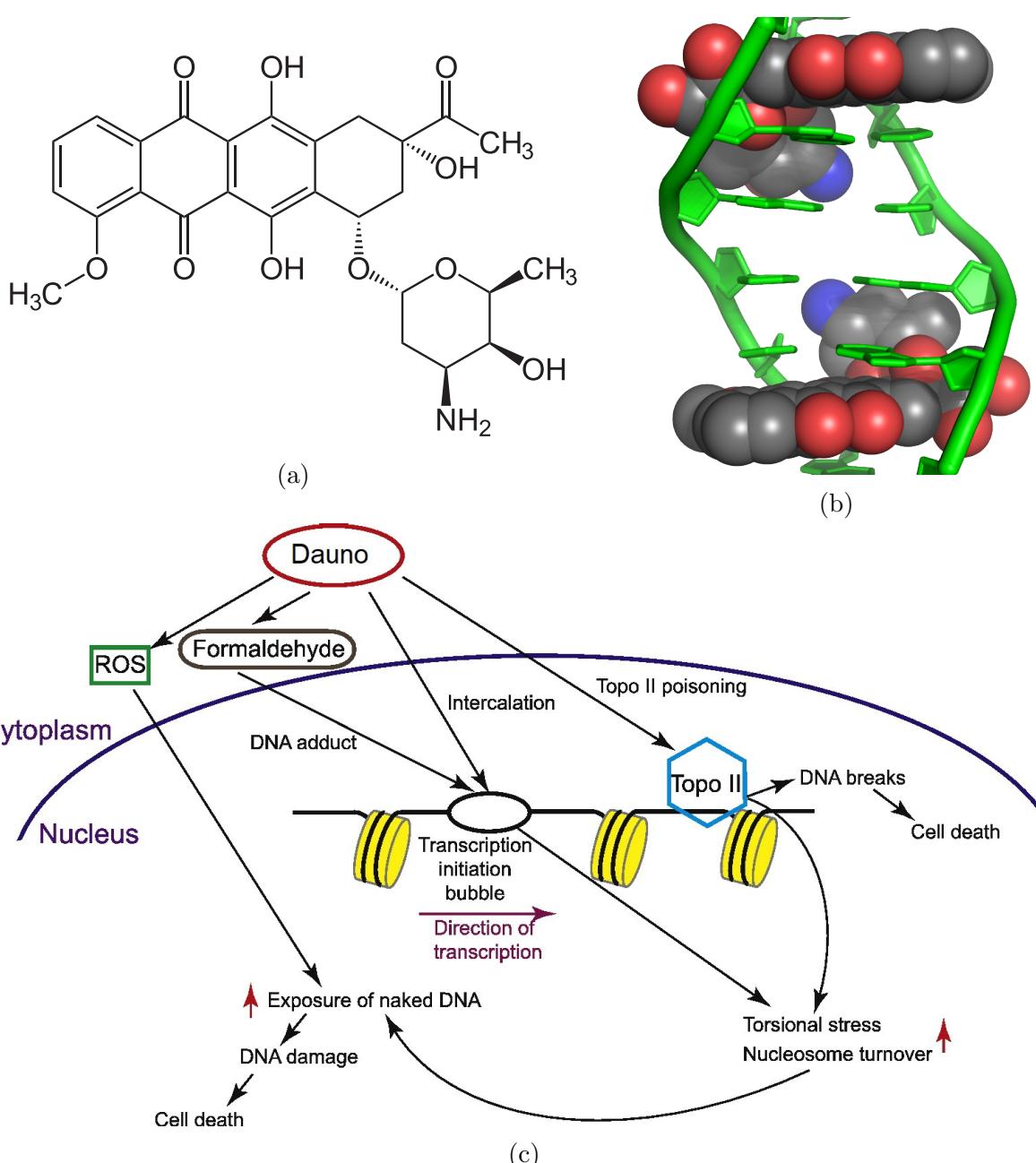


Figure 2.5: (a) The molecular structure of Daunorubicin. (b) Two anthracyclines intercalating with DNA double helix. (c) Schematic diagram of Daunorubicin’s mechanism of action. From left to right: daunorubicin induces reactive oxygen species (ROS) which damage the DNA. Daunorubicin interferes with the DNA by (1) covalently crosslinking with the DNA mediated by formaldehyde and (2) intercalating DNA double helix. Daunorubicin also inhibits Topoisomerase II which prevents DNA supercoil’s relaxation. Not shown in image: daunorubicin removes histone (shown in yellow in image) disrupting epigenetic regulations. Modified from (Yang et al, 2014 [71])

2.1.4 Cell death signal

As mentioned in Subsection 2.1.3, the introduction of cancer treatment disrupts mitotic cell activities which in turn activates cell death pathways. Two biochemical signals are

selected to observe cell death caused by the disruption of mitotic process:

Caspase 3/7

Caspase 3 and 7 are both known as executioner Caspases in programmed cell death pathways [72]. For the project CellEvent Caspase3/7 vial from ThermoFisher Scientific is used [73]. Besides Caspase 3/7, the treatment consists also of a four amino acid peptide (DEVD) conjugated to a nucleic acid binding dye. This cell-permeable substrate is intrinsically non-fluorescent, because the DEVD peptide inhibits the ability of the dye to bind to DNA. After activation of Caspase 3 or Caspase 7 in apoptotic cells, the DEVD peptide is cleaved, enabling the dye to bind to DNA and produce a bright, fluorogenic response. It has excitation/emission maxima of 502/530 nm (see Figure 2.6c).

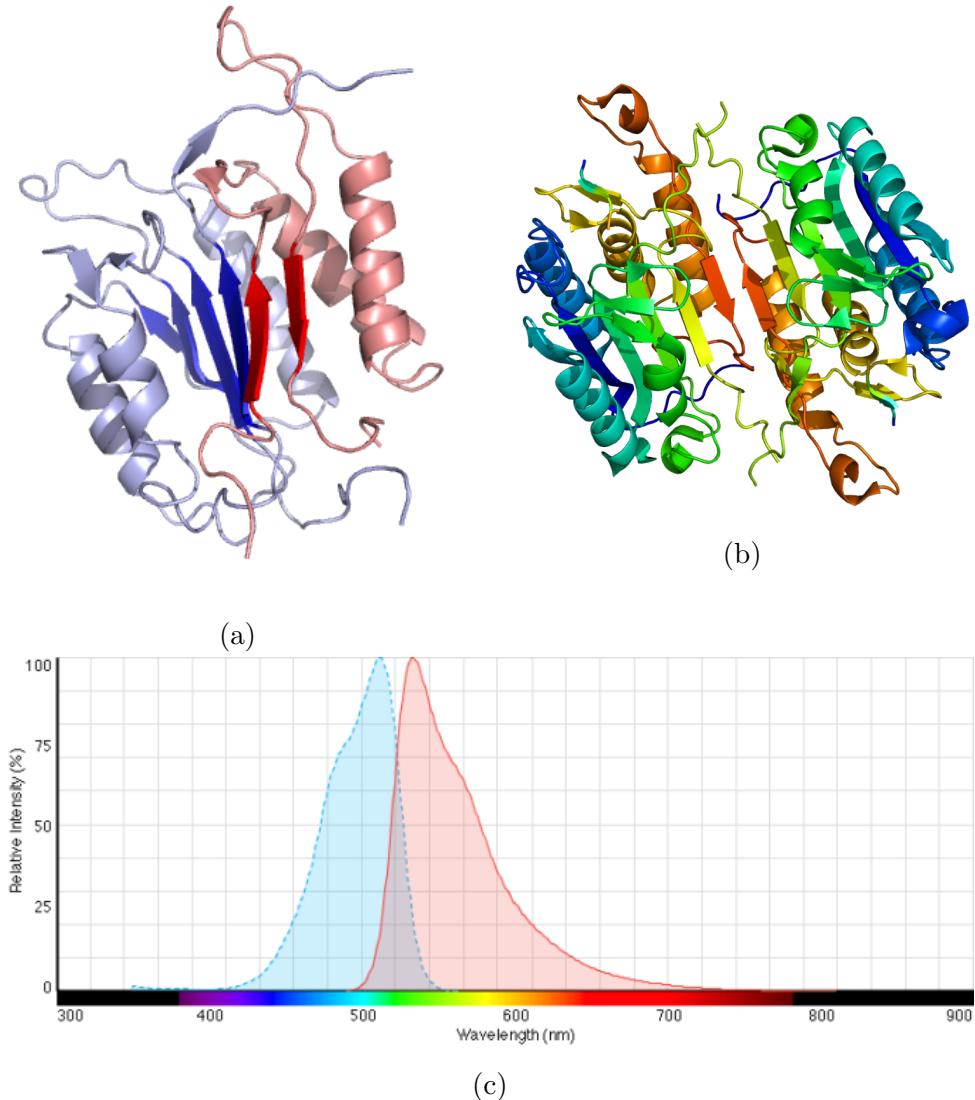


Figure 2.6: (a) The molecular structure of Caspase 3 heterodimer (b) The molecular structure of Caspase 7 homodimer. The p17 (light blue) and p12 (pink) subunits are shown. (c) The excitation (blue) and emission (red) spectra of Caspase3/7 vial used in the experiment. Notice the emission peaking around the 525-535 nm range in green channel. Image taken from ThermoFisher Scientific product specification page [73].

Propium Iodide (PI)

PI is membrane impermeable and generally excluded from viable cells. Therefore it is commonly used for identifying dead cells in a population and as a counterstain in multicolor fluorescent techniques. PI binds to DNA by intercalating between the bases with little or no sequence preference and with a stoichiometry of one dye per 4-5 base pairs of DNA [74]. In aqueous solution, the dye has excitation/emission maxima of 493/636 nm (see Figure 2.7b). PI also binds to RNA, necessitating treatment with nucleases to distinguish between RNA and DNA staining. Once the dye is bound to nucleic acids, its fluorescence is enhanced 20- to 30-fold. For the project PI vial from ThermoFisher Scientific is used [75].

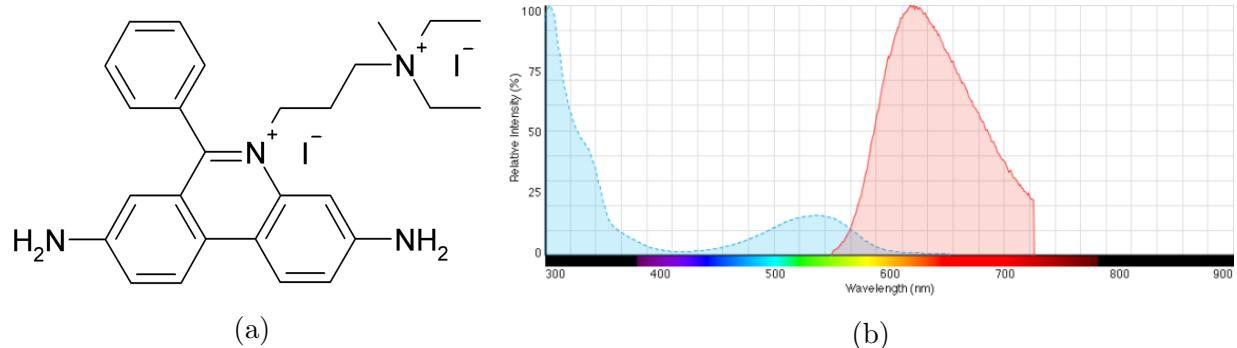


Figure 2.7: a) The molecular structure of Propium Iodide (PI) (b) The excitation (blue) and emission (red) spectra of PI. Notice the peaking around the 630-650 nm range in red channel. Image taken from ThermoFisher Scientific product specification page [75].

2.1.5 Experiment setup and image capturing

In this project, two experiments are done:

Unsynchronized experiment.

In the unsynchronized experiment, all 8 wells of the ibidi® sticky-Slide 8 Well ('slide') are used with one micro-trench wafer in each well (see Subsection 2.1.2 for the description of the slide). Every well is poured with different concentrations of drugs. To cover a large concentration range, a logarithmic scale is used to select concentration starting with 1 nanomolar (nM) all the way to 1000 nM (see Table 2.1). The use of a logarithmic scale is also in line with the logarithmic nature of many enzymatic and molecular processes in cell biology [76, 77]. Following concentrations are used: 1 nM, 5 nM, 10 nM, 100 nM and 1000 nM of Vincristine; 10 nM and 100 nM of Daunorubicin; and no drug as control. The unsynchronized experiment concentration configuration can be seen in Table 2.1. The recording lasts for 45 hours. After 21 hours of recording, the drug treatment is introduced into each well. Table 2.2 shows the image capturing frequency for each image type.

| Condition | Positions |
|--------------------|-----------|
| A1: 1000 nM - VCR | 1-8 |
| A2: 100 nM - VCR | 9-15 |
| A3: 10 nM - VCR | 16-23 |
| A4: 1 nM - VCR | 24-31 |
| B4: 10 nM - Dauno | 32-39 |
| B3: 100 nM - Dauno | 40-47 |
| B2: 0 nM - VCR | 48-55 |
| B1: 5 nM - VCR | 56-63 |

Table 2.1: Configuration table of treatments for synchronized experiment

| What | What for | t |
|-------------------------------------|------------------------------------|------------|
| In-focus brightfield (BF) image | Sanity check, micro-trench masking | 30 minutes |
| Out-of-focus BF image | Image tracking, image analysis | 15 minutes |
| Red fluorescence channel (PI) | Image tracking, image analysis | 15 minutes |
| Green fluorescence channel (Csp3/7) | Image tracking, image analysis | 15 minutes |

Table 2.2: Image capture frequency of the experiment

Synchronized experiment

The main difference between unsynchronized and synchronized experiments is the synchronization step. This process involves arresting cell cycle with the so-called "double thymidine block" protocol [78], which arrests cell development at G_1/S -phase by stopping DNA synthesis using double thymidine block, a well known DNA synthesis inhibitor [79].

Like in the unsynchronized experiment, several concentration levels are tested in this experiment (see Table 2.3). The image capturing starts right after the cells are synchronized and lasts 24 hours. The image capturing frequency is exactly the same as in the unsynchronized experiment (see Table 2.2).

| Condition | Positions |
|--------------------|-----------|
| A1: 1000 nM - VCR | 1-8 |
| A2: 100 nM - VCR | 9-15 |
| A3: 10 nM - VCR | 16-23 |
| A4: 1 nM - VCR | 24-31 |
| B4: 0 nM | 32-39 |
| B3: 5 nM - VCR | 40-47 |
| B2: 10 nM - Dauno | 48-55 |
| B1: 100 nM - Dauno | 56-63 |

Table 2.3: Configuration table of drug treatments for synchronized experiment

Image capturing

Imaging was performed under an inverted Nikon Ti Eclipse microscope with a motorized stage (Tango XY Stage Controller, Märzhäuser Wetzlar GmbH & Co. KG, Germany), a

CFI Plan Fluor DL 10X objective, a pco.edge 4.2 Camera (PCO AG, Kelheim, Germany) and a Lumencor Sprectra LED fluorescence lamp. For detection of the Caspase-3/7 and the PI marker, the following filters were used respectively, 474/27 nm, 554/23 (excitation) and 515/35 nm, 595/35 nm (emission). Brightfield out of focus (-20 μm) images were taken every 10 minutes and in-focus brightfield and fluorescence images every 30 minutes for 48 hours. Vincristine or Daunorubicin were added after 20 hours from the beginning of the imaging. During the recording samples were kept at a constant temperature of 37°C and CO₂ using an Okolab heating and CO₂ 2 box (OKOLAB S.R.L., NA, Italy). For the synchronized population, the double thymidine block protocol was followed. Briefly, MOLM-13s cells at the exponentially growing phase were incubated in blocking medium (culture medium supplemented with 2 mM Thymidine (CAS 50-89-5, Calbiochem®, Germany)) for 24 hours. Cells were then released and incubated in culture medium for 8 hours and finally were incubated in blocking medium for 12 hours. After 2 hours, the synchronized population was seeded in the slide bearing the micro-trenches together with the markers and drugs at the same conditions as the unsynchronized population, and imaged for 24 hours.

2.2 Definitions

This section contains formal definitions and methods used in this thesis.

2.2.1 Image channels and encoding

Several image standards and encodings are being processed in our pipeline. We start from the raw image captured by the camera going down to processed images (See Subsubsection 2.1.5 for detailed protocols on image acquisition). There are two brightfield channels (in-focus and out-of-focus channel) and two fluorescent channels (red and green) in this project. Out-of-focus brightfield images are used to track cells as their unsharpness can be exploited through image processing creating robust patch of cells, as shown by Buggenthin [80]. The red and fluorescent channels capture fluorescence emission in 620-750 nm and 495-570 nm respectively (see Subsection 2.1.4).

- In-focus brightfield image, the focal distance of the camera d_f is exactly the same as the distance of the camera to the center of the cell d_m , i.e. $d_f = d_m$.
- In out-of-focus brightfield image, the difference of focal plane and center of the cells, $d_f - d_m$, influences the quality of the image in several ways (see Figure 2.8). It was determined that the ideal image for segmentation came from the setting with $d_f - d_m = -20\mu\text{m}$.
- The red fluorescent channel is used to capture emission from PI activation due to PI's DNA binding emission (see Subsection 2.1.5).
- The green fluorescent channel is used to capture emission from caspase3/7 activation due to DAVD peptide cleavage (see Subsection 2.1.5).

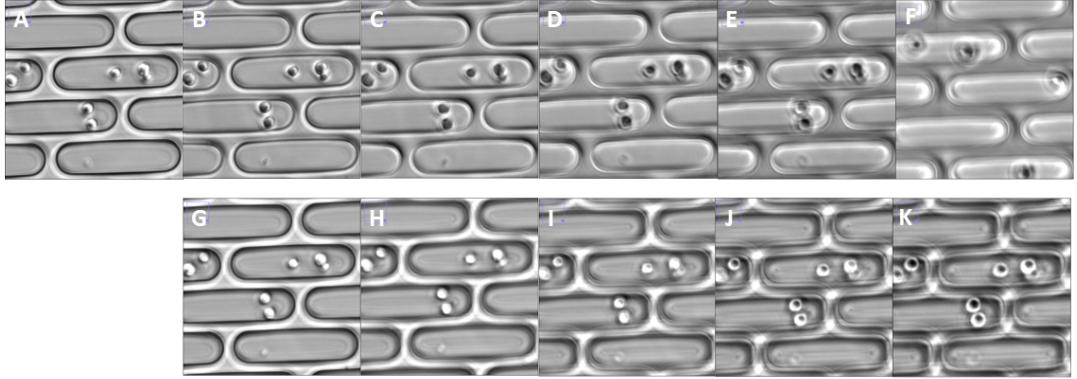


Figure 2.8: In-focus image and out-of-focus brightfield images taken at various focus levels. In the first row are images focused behind the microfluidics system while in the second row are images focused before the system. A: in focus image. B: $+10\mu m$. C: $+20\mu m$. D: $+30\mu m$. E: $+40\mu m$. F: $+50\mu m$. G: $-10\mu m$. H: $-20\mu m$. I: $-30\mu m$. J: $-40\mu m$. K: $-50\mu m$.

The camera records TIFF images (see Appendix B.1 for the detailed protocols) [81]. Each recorded stack consists of slice with one slice representing a single image capture at a time point. Each stack is encoded using RGB color model. This is also the case for the fluorescent image in which each slice shows the intensity in corresponding color channel (red, green or blue) as a monochromatic RGB image.

RGB color model represents the pixel as a combination of red, green and blue color. This encoding is able to represent any human visible color and is useful enough for most use cases [82, 83]. The most commonly used RGB encoding is the 8 bit encoding. Here, each pixel is represented as an RGB pixel having red, green and blue color values ranging from 0 to 255. Mathematically, this means that each pixel $M_{t,x,y}$ of a slice captured at time t can be represented as a triple, i.e.,

$$M_{t,x,y} := (M_{t,x,y,R}, M_{t,x,y,G}, M_{t,x,y,B}) \in [0 : 255]^3 \quad (2.1)$$

for 8-bit RGB encoding. Consequently, a slice M_t captured at time t of width w and height h is a 3-dimensional matrix of dimension $w \times h$, i.e.

$$M_t \in p_c^{w \times h \times 3}$$

with $p_c \in [0 : 255]$. A stack (a sequence of slices) M of n slices is in turn a 4-dimensional matrix:

$$M \in p_c^{n \times w \times h \times 3}$$

To convert the RGB value of a pixel $M_{x,y}^a$ from one bit-encoding to another $M_{x,y}^b$, linear conversion is used:

$$M_{x,y}^b = (\lceil M_{x,y,1}^a \frac{2^b}{2^a} \rceil, \lceil M_{x,y,2}^a \frac{2^b}{2^a} \rceil, \lceil M_{x,y,3}^a \frac{2^b}{2^a} \rceil) \quad (2.2)$$

where a and b refer to the bit length of the source and target encoding respectively (commonly known as bid depth). Commonly used depths are 8, 16 (high color format), 24 (true color format) and 48 bits (deep color format) [84, 85, 86].

Sometimes, other type of encodings like grayscale and 1-bit monochrome are used. A grayscale image essentially shows the intensity of the pixels in an image. A grayscale slice G can be represented as a matrix of integer, i.e. $G \in p_c^{w \times h}$. Like RGB image, linear scaling can be applied to transform grayscale images across bit depth:

$$G_{x,y}^b = \lceil G_{x,y}^a \frac{2^b}{2^a} \rceil \quad (2.3)$$

An RGB image can be transformed to a grayscale image by combining the intensity from every channel:

$$G_{x,y} = \frac{M_{x,y,1} + M_{x,y,2} + M_{x,y,3}}{3}$$

This however does not reflect human perception of light, as human eyes' spectral sensitivity is not uniform across the color spectrum [87]. Indeed, as Osorio and Vorobyev shows in 2005, each species has its own specific spectral sensitivity distribution [88], as can be seen in following figure:

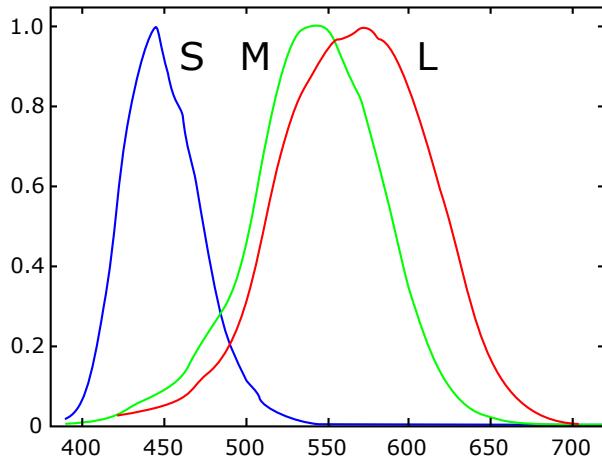


Figure 2.9: Spectral sensitivity of human. Note that the sensitivity distribution is maximum normalized. Each curve shows the relative sensitivity of each type of human cone cells (S, M and L types). The curve color represents the dominating absorbed color range of each cell type.

There are some conversions published on transforming RGB value to human intensity perception based on the measured spectral sensitivity [89, 90, 91]. Among the most commonly used is the *BT.601* standard from International Telecommunication Union (ITU). It recommends the following luminosity weight for RGB to grayscale conversion [90]:

$$G_{x,y} = \frac{0.299M_{x,y,R} + 0.587M_{x,y,G} + 0.114M_{x,y,B}}{3} \quad (2.4)$$

For following image:

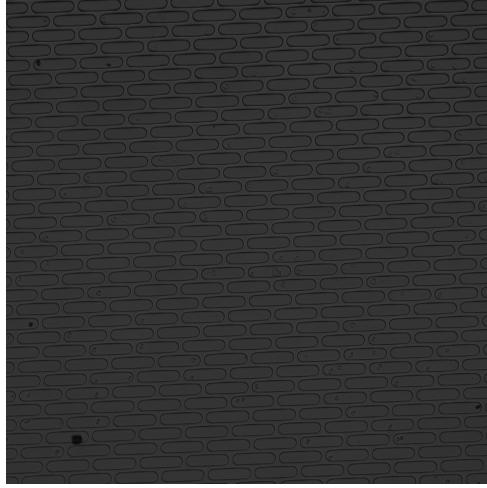


Figure 2.10: The in-focus image position 41 from unsynchronized experiment.

using Equation 2.4 to convert RGB to intensity , we can convert the image into grayscale image representing the intensity. The grayscale image has the intensity distribution as shown in following histogram:

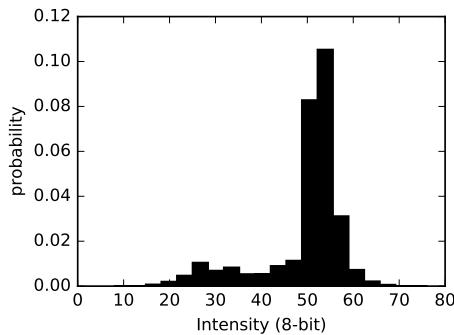


Figure 2.11: Normalized distribution of intensity of Figure 2.10.

Reverting back a grayscale image to an RGB image in turn only consists of applying grayscale value to each color component:

$$M_{x,y} := (G_{x,y}, G_{x,y}, G_{x,y})$$

A 1-bit monochrome image is encoded as a binary matrix:

$$B \in p_b^{w \times h} \wedge p_b \in \{0, 1\}$$

This encoding is useful when not using not the complete information of the image but rather separation of interesting parts in the image, since since it requires less memory (1 bit per pixel vs 24 bits per pixel of normal RGB image) and enables bitwise operation of the CPU which takes constant time [92]. Some use cases are for example region of interest (ROI) bounding and contour and boundary visualization [93].

The conversion from RGB to binary images can be done by defining a cutoff value c on one of the channels, e.g. for red channel, following conversion can be used:

$$B_{x,y} = \begin{cases} 1 & \text{if } M_{x,y,R} > c \\ 0 & \text{else} \end{cases}$$

Note that, a bit encoding of an image refers to unsigned encoding. Thus, an m -bit encoding allows value ranging from 0 to $2^m - 1$.

2.2.2 Intensity, brightness and contrast

Intensity, brightness and contrast can be modified to show and hide parts of an image. In this project, many image processing steps utilize the modification of them. Previously we defined intensity in image encoding. In this subsection we will define brightness and contrast and their adjustment.

Mathematically, we can represent brightness and contrast adjustment as a mapping between one domain to another. Let $l(M_{t,i,j})$ be a function that maps the RGB value of a pixel $M_{t,i,j}$ to its corresponding intensity I_t . Conveniently, we can use ITU's *BT.601* standard formula written in Subsection 2.2.1 to convert an RGB to grayscale image.

We then define the $I_t : [0 : w] \times [0 : h] \rightarrow \mathbb{N}_0$ function that maps the coordinate of a pixel to its corresponding intensity value i.e.

$$I_t(i, j) := l(M_{t,i,j})$$

Note that the codomain of the function I_t depends on the encoding used. For a 8-bit encoding this will be then $[0 : 255]$.

An intensity transformation J_t is then defined as a linear transformation of I_t with **gain** and **bias** parameters α and β respectively:

$$J_t(i, j) := \alpha \cdot I_t(i, j) + \beta \quad (2.5)$$

with $\alpha > 0$ [94]. The gain and bias parameters are also known as **contrast** and **brightness** parameters. In this regard, increasing/decreasing brightness is equivalent to increasing/decreasing β . The same thing also applies for contrast parameter, increasing α will increase the contrast of the image.

Predictably, doing transformation using transformation function J_t will inevitably cause the resulting intensity to be outside of the allowed value range $[0 : 2^m]$ for m -bit encoding. To understand this situation, we first have to consider the boundedness of eyes perception. The argumentation for boundedness can be shown by realizing that the excitation of a neuron follows a sigmoid function [95]. Thus given no impulsion the neuron will stay in ground state, while very large impulse is bounded due to biochemical constraint of a neuron. Mapped in the context of a sigmoid function, a steady state corresponds to 0 while asymptotically unlimited excitation corresponds to 1. In our context, a non-excited state corresponds to 0 intensity while full-excitation corresponds to $2^m - 1$ intensity. Thus, the Equation 2.5 can be bounded by introducing upper and lower bounds of 0 and $2^m - 1$, i.e.

$$J_t(i, j) := \begin{cases} 2^m - 1 & \text{if } \alpha \cdot I_t(i, j) + \beta > 2^m - 1 \\ 0 & \text{if } \alpha \cdot I_t(i, j) + \beta < 0 \\ \alpha \cdot I_t(i, j) + \beta & \text{else} \end{cases}$$

We can for example, transform Figure 2.10 using $J_t(i, j)$ with $\alpha = 1$ and $\beta = 100$. The transformed image can be seen in Subfigure 2.12a with Subfigure 2.12c shows the intensity distribution of the transformed image. We can also try to modify α . Changing α to a value larger than 1 will scale the image's intensity and emphasize/de-emphasize image contrast. Subfigure 2.12c shows the transformed image for $\alpha = 3$ and $\beta = 0$.

See for example following image:

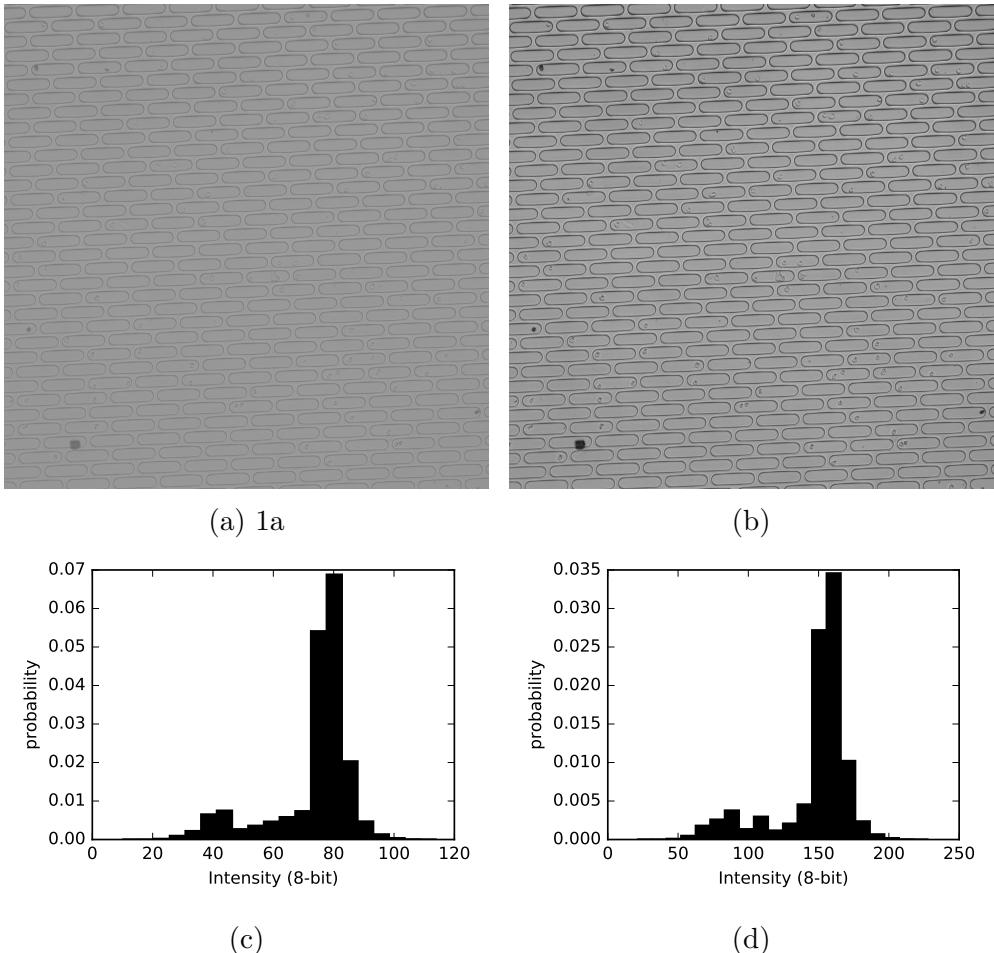


Figure 2.12: (a) The in-focus image position 41 from unsynchronized experiment transformed with $\alpha = 1$ and $\beta = 100$. (b) The in-focus image of position 41 from unsynchronized experiment transformed with $\alpha = 3$ and $\beta = 0$. Notice the contrast has increased significantly, especially around area with high intensity differential such us the margin of a micro-trench and the dark spot in the lower left part of the slice. (c) Normalized distribution of intensity of (a). (d) Normalized distribution of intensity of the (b). Note that the difference in height across bins in two histograms is mainly caused by the definition of bins in the histogram function.

Geometrically, brightness describes the level of visibility of a pixel. Adjusting β shifts the entire intensity distribution to the right by β ¹. On the other hand, contrast can be understood as the distance of closely resembling pixel [93]. Two pixels which are very similar will look very different given a high contrast.

¹Note the wording. For negative β , the shift is negative to the right, i.e. to the left

2.2.3 Robust automatic threshold selection (RATS)

Robust Automated Threshold Selection (RATS), based on description by Wilkinson and Schut[96], computes a threshold map for an image based on two criteria: pixel values, and pixel gradients [97].

The gradient is computed using the so-called Sobel kernel [98] a commonly used in computer vision for edge detection as it particularly emphasizes edges in an image upon transformation. It is a discrete differential operator which computes an approximation of the gradient of the intensity function J .

Originally, the Sobel gradient $\mathbb{G}_{x,y}$ of coordinate (x, y) is defined as

$$\mathbb{G}_{x,y} := \sqrt{\Delta_x^2 + \Delta_y^2}$$

Whereas Δ_x and Δ_y are defined as

$$\Delta_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{M}_t$$

and

$$\Delta_y = \begin{bmatrix} +1 & +2 & +1 \\ +0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{M}_t$$

with \mathbf{M}_t denoting the image at time t and asterisk symbol $*$ denotes a two dimensional signal processing convolution [99]. For example, for a 3×3 matrix,

$$X_{3 \times 3} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

and Δ_x , the value of the middle pixel $(2, 2)$ after evaluating the convolution $\Delta_x * X_{3 \times 3}$ is,

$$\left(\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \right) [2, 2] = i \cdot 1 + 0 \cdot h - 1 \cdot g + 2 \cdot f + 0 \cdot e - 2 \cdot d - 1 \cdot g + 0 \cdot h - 1 \cdot a$$

It has been shown, however, that only taking the quadratic gradient yields comparable result [96]. The Sobel operator is thus defined as quadratic gradient in Fiji RATS plugin, i.e.

$$\mathbb{g}_{x,y} = \mathbb{G}_{x,y}^2$$

Pixels having gradients smaller than threshold $\lambda\sigma$ are then removed, where σ is the noise (generally, standard deviation of the expected background is used, e.g. standard deviation of the whole image) and λ is the scaling factor (Wilkinson determined $\lambda = 3$ to be a good approximation [96]). Visually, this step can be understood as removal of background noise from the image.

To refine the filtering further, iterative correction is applied. First, a quadtree construction is applied in an image. A quadtree is a tree-like data structure in which each

node has exactly four children (Finkel and Bentley, 1974 [100]). In the context of an image, an image is recursively separated into smaller quadtree until the standard deviation of the intensity in every quadtree is lower than a pre-defined limit (usually the standard deviation of the intensity of the whole image σ). Figure 2.13 gives an example on how a quadtree is constructed from an image.



Figure 2.13: Example of quadtree. The left side is the quadtree representation of original image on the right. *Creative Commons courtesy of anonymous.*

In the second step, we attempt to remove the filter locally. To do that, we first define a local threshold T_R within the smallest square. For a quadrant, the threshold is computed in following way:

$$T_R = \frac{\sum g_{x,y} \cdot M_{t,x,y}}{\sum g_{x,y}} \quad (2.6)$$

Where $\sum g_{x,y}$ the sum of all children quadrants' gradient. For the smallest quadrant, only $g_{x,y}$ is used instead, i.e.

$$T_R = \frac{g_{x,y} \cdot M_{t,x,y}}{g_{x,y}} \quad (2.7)$$

This threshold is used to filter pixels within a quadrant. Pixels with gradient value smaller than T_R of a quadrant are then filtered out. If T_R is very low that no point in a quadrant is filtered out, the T_R of the parent is taken instead. This step is iterated until a parent square with at least one filtered pixel is reached. Visually, the refining step is equal to filtering out the area of an image in which the pixels are mostly noise. In the initial global filtering process, many noisy pixels might not be filtered since they happen to have gradients larger than $\lambda\sigma$. Using this bottom-up approach these pixels will be eventually filtered out.

2.2.4 Holes filling

Filling micro-trench holes to create the contour is done in two steps:

1. Recognition of the outer part of micro-trench contours, followed by

2. Filling of such contour with a new neutral value (black).

Following image explains the definition of contours and their hierarchy:

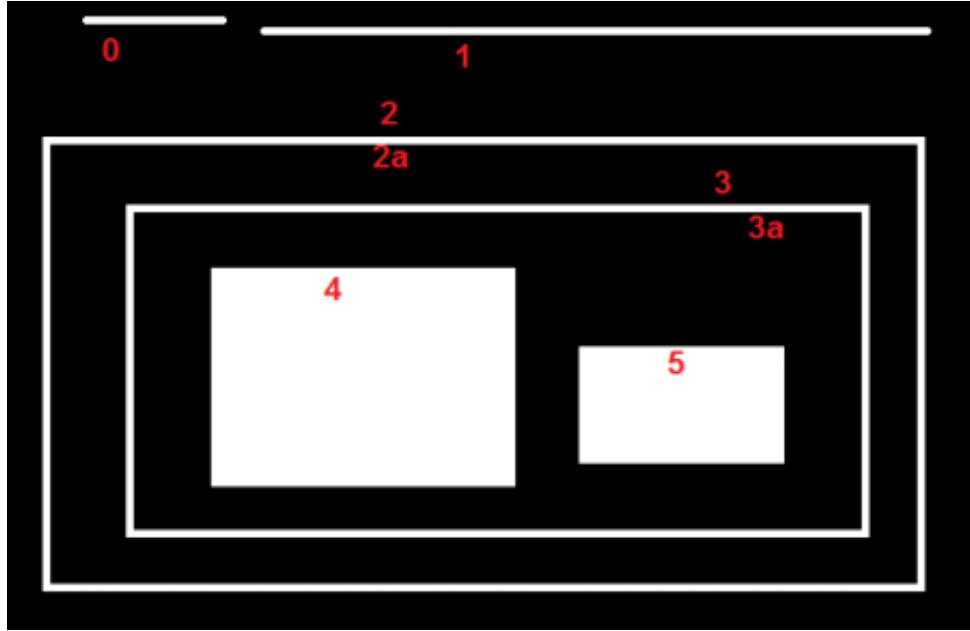


Figure 2.14: Example of contours and their hierarchy. **Contours 0, 1 and 2** are **outermost contours** and of **hierarchy-0**. **Contour 2a** is **child contour** and of **hierarchy-1**. **Contour 3** is child contour of **contour 2a** and of **hierarchy-2**. **Contour 3a** is child contour of **contour 3** and of **hierarchy-3**. Both **contours 4** and **5** are children of **contour 3a** and of **hierarchy-4**. All contours with same hierarchy are topologically equal. Contour Adopted from OpenCV tutorials by Bradski and Kaehler [101].

There are several well known algorithms for hierarchical contours recognition. In our case, the algorithm of Suzuki and Abe [102] is used. The algorithm works by walking along the border between two regions of different color. The border walking process is done hierarchically creating tree-like dependency structure of contours. We then take the outermost (hierarchically the highest) contours which are then filled. The filling in turn is simply done by assigning neutral value in every pixel located inside every outermost contour.

2.2.5 Gaussian blur

Gaussian blurring in the context of image processing is the process of applying a Gaussian noise on an image. The Gaussian function $G : \mathbb{R} \rightarrow \mathbb{R}$ is a function defined as

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

In 2 dimensional setting, we extend the function to $G : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ defined as

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.8)$$

Applying this function on a 2D image creates concentric circles centered around the center of the distribution with values around the center distributed normally. This means that, for an arbitrary σ and a uniform image with the intensity of n in every pixel, convolving the image with Equation 2.8 centered at point (i, j) yields a transformed image with the transformed intensity I^G of each point distributed normally around (i, j) with scaling of n and standard deviation σ :

$$I^G(x, y) = n \cdot \frac{1}{2\pi\sigma^2} e^{-\frac{(x-i)^2+(y-j)^2}{2\sigma^2}} \quad (2.9)$$

The idea of transforming an image with one selected center point is however not very practical as the pixels far away from image will have intensity of almost zero. We can however, use the fact that the intensity of pixels around the center point fall exponentially the further the pixels are away to introduce a dependency of each pixel on its surrounding environment. This will introduce a blur due to pull-down effect of the neighboring pixels. Hence the name **Gaussian blurring**.

Mathematically, we can express the Gaussian blurring process as convolving the pixels around a pixel (x, y) and take the average of the convoluted image as the new intensity value in (x, y) . Using the continuous Gaussian intensity transformation defined in Equation 2.9, the Gaussian convolution function is defined as following:

$$I^{conv_G}(x, y) = \frac{\sum_{x-\lfloor d/2 \rfloor < i < x+\lfloor d/2 \rfloor} \sum_{x-\lfloor d/2 \rfloor < j < x+\lfloor d/2 \rfloor} I^G(i, j)}{(\lfloor d/2 \rfloor)^2}$$

with d denoting the L_1 distance of the blur. We can furthermore refine the convolution further by using L_2 distance (i.e. radius) instead:

$$I^{conv_G}(x, y) = \frac{\sum_{(i,j) \in C} I^G(i, j)}{\|C\|} \quad (2.10)$$

for $C := \{(i, j) \text{ where } d_{L_2}((x, y), (i, j)) \leq d\}$. It can be seen from the equation that increasing the radius d increases the pull-down effect on the pixel and hence increases the smoothness of in the image. We exploit this property to create regional gradient which will be used to normalize an image by its background gradient. By dividing the value of each pixel by the convoluted value of it we can correct the image by its global background noise. The Gaussian blur-corrected intensity I^{conv_G} is thus defined as

$$I^{c_G}(x, y) = \frac{I(x, y)}{I^{conv_G}(x, y)} \quad (2.11)$$

Visually, the method can be understood as low pass filter. It removes the higher frequency 2D signals, i.e. the value with high local variance, resulting with an image with lower fidelity and local variance.

2.2.6 Contrast limited adaptive histogram equalization (CLAHE)

CLAHE, initially developed by Karel Zuiderveld in 1994 [103], is an instance of the class of algorithms called adaptive histogram equalization (AHE) commonly used to improve contrast in an image. An AHE algorithm generally works by transforming each pixel with a transformation function derived from the neighboring region instead of the whole image.

By doing this, an AHE algorithm can count for the variation of brightness and contrasts in areas of an image [104].

The distinction of CLAHE among other AHE algorithms is the contrast limiting property of the algorithm. Given a contrast distribution of an area around a given point, CLAHE conducts following steps:

1. First, the algorithm takes a predefined clipping value of the intensity histogram of the area around a center point.
2. The algorithm then calculates the intensity histogram of the area.
3. For some intensity values there will be more pixels than allowed with the intensity values. We call these intensity values the clipping intensity area.
4. Adapt the intensity histogram by reassigning the intensity of some pixels in the clipping intensity area uniformly across the range of intensity in the image (see upper part of Figure 2.15).
5. Repeat until for randomly selected areas in the image the clipping value constraint is satisfied, i.e. for every randomly chosen pixel, the intensity histogram contains no clipping intensity area.

Figure 2.15 visualizes the step 2, 3 and 4 of the algorithm description above. As can be seen in the area intensity CDF in the lower part of Figure 2.15, the algorithm results in a more equalized distribution of intensity. Visually, this increases the contrast and in turn sharpens the image. Figure 2.16 shows an area of the well before and after the application of CLAHE. Note the increased brightness and emphasized convolution effect around the micro-trench margin area.

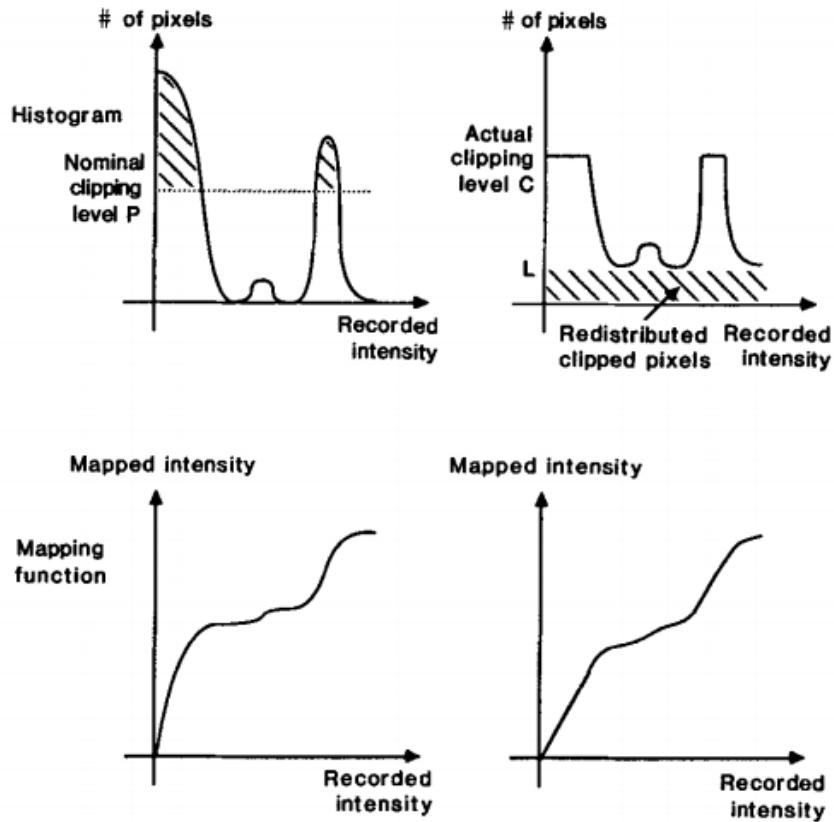


Figure 2.15: Description of CLAHE algorithm. Taken from Pizer, Zuiderfeld et al, 1987 [104] Note the smoothen intensity CDF function (lower part of the figure) after reassigning some pixels with pixel values above the clipping value.

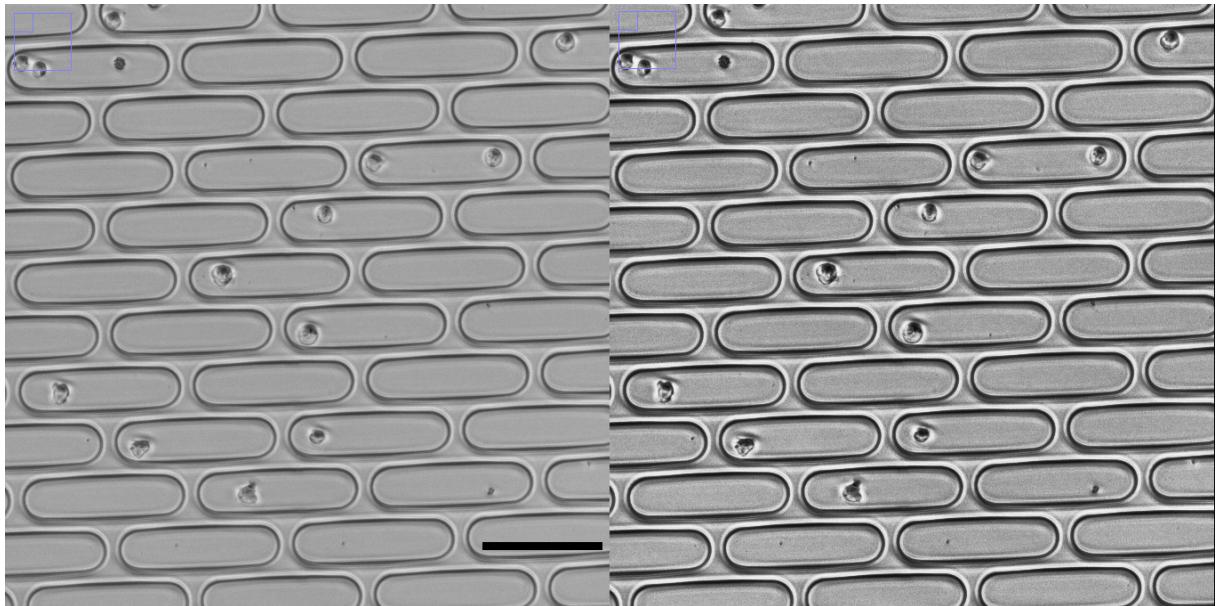


Figure 2.16: The upper left corner of the in-focus brightfield channel of the image position 43 at time $t = 0$: before (left) and after (right) the application of CLAHE on the image. Scale bar on the left figure is $100\mu M$ long.

2.2.7 Subtraction by pixel-wise mean intensity

To subtract an RGB image at time M_t by its pixel-wise mean intensity $\langle M \rangle$, the average intensity for each pixel across all time points is made for a pixel. For a stack with n images, we can simply define the pixel-wise mean of the R channel for pixel (x, y) as following:

$$\langle M \rangle_{x,y,R} := \frac{\sum_{1 \leq t \leq n} M_{t,x,y,R}}{n} \quad (2.12)$$

The same applies to $\langle M \rangle_{x,y,G}$ and $\langle M \rangle_{x,y,B}$. For a grayscale stack we define analogously:

$$\langle G \rangle_{x,y} := \frac{\sum_{1 \leq t \leq n} G_{t,x,y,i}}{n} \quad (2.13)$$

The pixel-wise subtracted value of R channel is thus defined as point-wise operation for each pixel,

$$M_{t,x,y,R}^m := M_{t,x,y,R} - \langle M \rangle_{x,y,R} \quad (2.14)$$

The same applies to $\langle M \rangle_{x,y,G}$ and $\langle M \rangle_{x,y,B}$. The transformed RGB value is thus defined as

$$M_{t,x,y}^m := [M_{t,x,y,1}^m; M_{t,x,y,2}^m; M_{t,x,y,3}^m] \quad (2.15)$$

As for a grayscale image we define following:

$$G_{t,x,y}^m := G_{t,x,y} - \langle G \rangle_{x,y} \quad (2.16)$$

In the context of the single-cell enclosure on a wafer, this means that in the parts of images in which only dynamic movement of cells are observed, the pixel-wise mean intensity value of dynamic parts will be very low. Subtracting every pixel the area from every time point with the average will barely affect the original intensity value. On the other hand, applying the method on areas with static objects such as micro-trench wall and its surrounding reduces each pixel of the area with exactly the same intensity value as it barely changes during the experiment. This will nullify the static parts to large extent.

This method can be improved by iteratively repeating the calculation of $\langle M \rangle$. This will remove periodically static parts or static parts that abruptly moved (due to sudden shift in the well with regard to the camera for example). We call this algorithm **k-subtraction by pixel-mean intensity**:

```

Data:  $M_t$ 
Result:  $M_t^m$ 
Parameters:  $k$ 
for  $i = 0$  to  $k - 1$  do
    | calculates  $\langle M \rangle$  from  $M_t$  ;
    |  $M_t^m = M_t - \langle M \rangle$  ;
    |  $M_t = M_t^m$  ;
end
```

Output: M_t^m

Algorithm 1: The k-subtraction by pixel-mean intensity algorithm

2.2.8 Cell recognition

Due to the cells' rotund shape, the **blob detection** family of algorithm is well-suited for recognizing cells. In our pipeline, we use the Laplacian of Gaussian (LoG) detector method.

Laplacian of Gaussian (LoG) detector

Laplacian of Gaussian (LoG), also known as Marr-Hildreth-Opreator [105], is among the first and still the most popular method for detecting blob [82]. It is characterized by applying the second derivative of the σ -scaled Gaussian to detect a blob in an image.

To derive the method we first consider a 2-dimensional Gaussian kernel function:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.17)$$

As the name suggests, the representation of LoG operator is obtained by applying the Laplace-Operator Δ on the Gaussian,

$$L(x, y) = \Delta G(x, y)$$

This can be expanded into,

$$\begin{aligned} L(x, y) &= \frac{\partial^2 G(x, y)}{\partial x^2} + \frac{\partial^2 G(x, y)}{\partial y^2} \\ L(x, y) &= -\frac{1}{\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \left(1 - \frac{x^2+y^2}{2\sigma^2}\right) \end{aligned}$$

Figure 2.20 shows the representation of the LoG operator in 2D. Notice the inverse hat characteristic giving it the nickname **Mexican hat operator**.

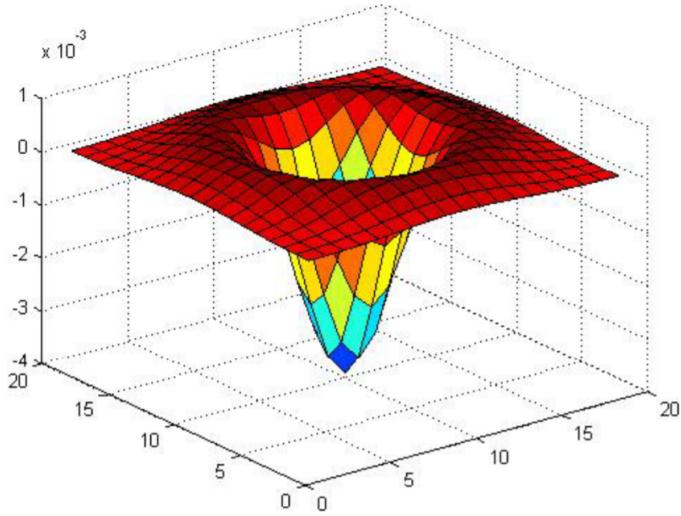


Figure 2.17: A LoG operator. Notice the lack of dimension in an operator as it transforms an image (in one space) into another image (still in the same space).

The operator is then applied on two dimensional matrix of a grayscale image. To get maximum response from a blob, the zeros of the operator have to be aligned with

the circle to be detected [105]. Figure 2.18 shows an example of idealized circle with the diameter r and the corresponding response from the LoG operator on the surface of the circle. Applied to pre-processed out-of-focus image, LoG will return response with maxima in the center of cell sizes in similar fashion to the idealized circle.

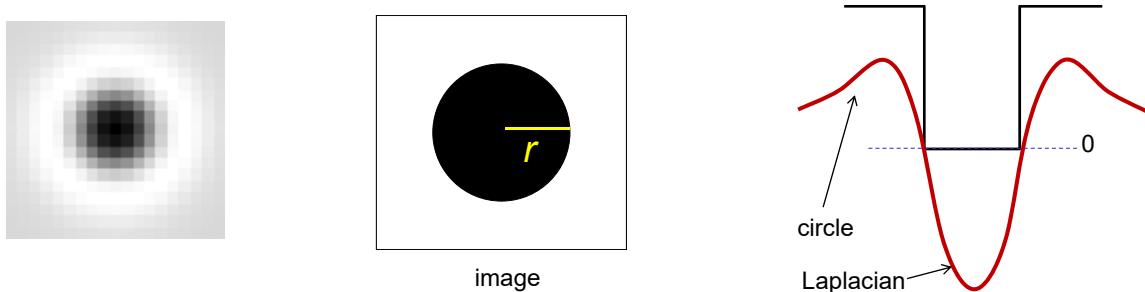


Figure 2.18: Example of applying the LoG operator on a 2D circle. The minimum will be achieved if the diameter of the LoG operator is roughly equal to the diameter of the cell. This however does not have to be the case. In case of non-perfect blob or blob with different diameter, local minimum can still be detected by the cell recognition algorithm [106].

2.2.9 Cell tracking

To track detected cells, the Linear Assignment Problem (LAP) tracker, by Jaqaman et al, 2018, is used. The method builds upon the cell recognition method explained in Subsection 2.2.8.

Linear Assignment Problem (LAP) framework of cell tracking

The algorithm follows closely the case of linear assignment problem in bipartite graph. Given a bipartite graph $\mathcal{G}_b := \{\mathcal{G}_1, \mathcal{G}_2\}$ and the assignment cost function $c(v_i, v_j)$ for $v_i \in \mathcal{G}_1$ and $v_i \in \mathcal{G}_2$, find a bijection $f_{\mathcal{G}_b} : \mathcal{G}_1 \rightarrow \mathcal{G}_2$ so that the total cost function:

$$\sum_{v_i \in \mathcal{G}_1} c(v_i, f_{\mathcal{G}_b}(v_i)) \quad (2.18)$$

is minimized. As seen in Equation 2.18, the cost function and its corresponding constraint (bipartite assignment) is linear. Hence, the "linear" part of the assignment problem.

We can frame our cell tracking problem as a variant of linear assignment problem. Recall that, for each slice, a set of blobs representing cells are detected. To track cells, each blob in a slice has to be connected with a blob in proceeding and succeeding slices (except in the case where a cell died).

Revisiting Subsection 2.2.8, it is obvious that upon cell detection the next step is to track single cell movements in a stack is to assign detected blobs \mathcal{G}_t at time t with the blobs \mathcal{G}_{t+1} detected at time $t+1$, for $t \in [1 : n - 1]$. To do that, first we define particle-to-particle cost function c . Jaqaman and his colleagues argued that [107], due to Brownian nature of cell movement, the square distance of blobs or its derivation should be used as cost function:

$$d(v_i, v_j) = \|v_i - v_j\|_2^2 \quad (2.19)$$

The algorithm is roughly divided into two main steps:

- Creation of track segments through frame-to-frame blobs linking.
- Gap closing and cell division inference to achieve the closing of the track segments.

Both steps are framed as a linear assigned problem. In the first step, two consecutive slices M_t and M_{t+1} are optimized for the links. To do that, a $(m+n) \times (n+m)$ matrix \mathcal{C} is created, where m and n refer to the number of detected blobs in M_t and M_{t+1} respectively. The matrix contains four quadrants:

- The upper left quadrant ($m \times n$ elements) contains the costs of linking blobs in M_t to those in M_{t+1} , also known as the segment linking cost.
- The upper right quadrant ($m \times m$ elements) contains the costs of not linking blobs in M_t to any blobs in M_{t+1} , also known as the segment stop cost.
- The lower left quadrant ($n \times n$ elements) contains the costs of not linking blobs in M_{t+1} to any blobs in M_t , also known as the segment start cost.
- The lower right quadrant ($n \times m$ elements) is the auxiliary matrix used by the LAP framework as formalism for its algorithm. The matrix is created by transposing the upper right left quadrant and replacing all non-infinity cost with the minimal cost (by default 0).

The segment linking cost (upper left quadrant) is calculated as following:

$$c(v_k, v_l) = \begin{cases} (d(v_k, v_l) \cdot (1 + \sum_f (3W_f \frac{f_{f1} - f_{f2}}{f_{f1} + f_{f2}}))^2 & \text{if } d(v_k, v_l) \leq d_{max} \\ \infty & \text{else} \end{cases}$$

where f refers to each feature penalty defined for the optimization and W_f , f_{f1} and f_{f2} refer to the feature penalty factor, the value of feature f of v_k and the value of feature f of v_l respectively. Generally, only blob related features are used in the algorithm, e.g. total blob intensity, average blob intensity and minimum/maximum blob intensity. Note that:

- If the distance is larger than the pre-defined maximum distance d_{max} , then the link is forbidden. A blocking cost (∞) is assigned between two blobs.
- If no penalty feature is introduced then the cost is simply the squared distance of two blobs.

The non-linking cost (the upper right and lower left quadrants) is calculated as follows:

$$c(v_k, v_l) = \begin{cases} 1.05 \cdot C & \text{if } v_k = v_l \\ \infty & \text{else} \end{cases}$$

where C is the maximum value of the upper left quadrant of the matrix.

The optimization over the matrix is then solved with the Munkers & Kuhn algorithm [108], which solves the problem in cubic time $\mathcal{O}(n^3)$. The algorithm returns assignment minimizing the sum of the assignment costs.

The interpretation of the cost functions is following: without any penalty, the optimization problem favors the solution which minimizes the sum of squared distance between two blobs. This is in line with previously mentioned assumption of the Brownian motion of cells. By adding feature penalties, we aim at favoring linking blobs that are more similar to each other. In brute single particle linking problems, spots are generally all the same, and they only differ by the coordinate. However, there is a variety of problems for which these feature penalties can add robustness to the tracking process. In our case for example, the cells might pose several features that change over time depending on the treatment and cell-dependent characteristics (the phenomenon we further investigate using machine learning methods, see Subsections 2.2.11 and ?? and Section 4.4).

Upon the linking of blobs into track segments, more refinement is then done to achieve globally optimal cell tracks configuration. To do this, three events are considered in our model:

- In case of *gap closing* events, the end of a track segment is linked to the start of another track segment.
- In case of *splitting* events, the stat of one track segment is linked to non-terminal part of another track segment.

The matrix is created in similar fashion to the cost matrix in the first step, with the details best referenced directly to the article by Jaqaman et al [107]. Also similar to the first part is the use of penalty features during segment creation. Unlike the the first part however, the segment merging cost is blocking (∞) if the segments in question are separated by frames larger than pre-defined maximum number of frame gaps.

The optimal solution for the problem is again found using the Munkers & Kuhn algorithm [108].

2.2.10 Shift correction

Shift correction is done to correct the slice alignment within stacks. As mentioned in Subsection 2.1.5, the images were captured discretely at the frequency of $\frac{1}{10\text{min}}$ and $\frac{1}{30\text{min}}$ depending on the channel. During this time, the setup might move albeit very slightly causing very small shift. The bulk of the shift, however, happened during the introduction of the drug treatment. At this time, the drug is introduced to the medium using a pipette. The liquid released pushes the wafer somewhat causing noticeable shift in camera's field of view (see Table A.1 of Appendix A for the inferred shift in every image position).

Now, consider the case in which the images are shifted in a time-lapsed movie during the introduction of drug treatment. No rotation of camera is assumed, hence there are only two degree of freedoms (vertical and horizontal). Thus, a shift can be defined as a vector movement \vec{v} of all points $x_{i,j} \in M_t$ in the time-lapse from time t to $t + 1$. Given two degrees of freedom and discreteness of the problem due to pixel representation, the task is reduced to finding difference in x- and y-axis (δ_x and δ_y), so that the difference of transformed pixels at t and t_{i+1} are minimized, i.e.:

$$\arg \min_{\delta_x, \delta_y} \{d(M_t, M_{t+1}^{\delta_x, \delta_y})\}$$

Where $M_{t+1}^{\delta_x, \delta_y}$ is the entries of matrix M_{t+1} after applying the shift $\vec{v} := (\delta_x, \delta_y)^T$, i.e.

$$M_{t+1, x, y}^{\delta_x, \delta_y} = M_{t+1, x-\delta_x, y-\delta_y}$$

and the distance function d is defined as all-channel all-pixel sum of differences between two image:

$$d_{RGB}(M_i, M_j) = \sum_{c \in \{R, B, G\}} \sum_x \sum_y |M_{i,x,y,c} - M_{j,x,y,c}|$$

Since some pixels are lost from the field of view during a shift, only a subset of subsequent images is used to determine the shift, preferably those around the center point. This will allow the largest search space possible, since the distance to all four margins of the image is maximized at the center point. The search for the optimal (δ_x, δ_y) pair is implemented as a grid search along the x- and y-axis. An example of the search grid is shown in Figure 2.19. Algorithm 2 shows the pseudocode of shift inference algorithm for RGB images.

Data: M_t, M_{t+1}

Result: M_t^m

Parameters: d, l

D distance matrix for various shifting configurations ;

$c := (c_x, c_y)$ coordinate of center pixel of M_t ;

$M'_t := M_t[c_x - l : c_x + l][c_y - l : c_y + l]$ sub-image of M_t centered around (c_x, c_y) ;

for $i = -d$ **to** d **do**

for $j = -d$ **to** d **do**

$c' := (c_x - i, c_y - j)$;

$M'_{t+1} := M_{t+1}[c'_x - l : c'_x + l][c'_y - l : c'_y + l]$;

$D[i, j] := d_{RGB}(M'_t, M'_{t+1})$

end

end

Output: $\arg \min_{i,j} \{D\}$

Algorithm 2: Shift inference algorithm for RGB images

Since the time-lapsed data consists mainly of grayscale image, the RGB encoding could be directly transformed to grayscale encoding (see Equation 2.4 of Subsection 2.2.1). Using the transformed method also speeds up the calculation process since the distance function only computes the difference of grayscale channel's values:

$$d_G(G_i, G_j) = \sum_x \sum_y |G_{i,x,y} - G_{j,x,y}|$$

For this case, the shift inference algorithm can simply be modified by replacing d_{RGB} with d_G .

Due to lost pixels around the margin of before and after images, only the overlapping part of both slides are included after the correction. Thus, for an inferred shift of (δ_x, δ_y) , the new dimension of the images is then $(m - \delta_x) \times (n - \delta_y)$. This change would then propagate to the other time-lapse images to maintain consistency of the images.

Ideally, the shift correction should be done for each time point to reduce the track dropout rate caused by image shifts. This is however computationally very expensive. Moreover, inferring the shift for every recording time is not really necessary since the biggest shift, as mentioned before, only happens right before and after the treatment. The difference difference between consecutive images of the image position 26 can be seen in 2.20. Here we can see that the major spike of difference is only observed upon the introduction of the drug treatment.

As described in Subsection 2.2.9, the tracking algorithm allows certain amount of tolerance represented as maximum distance d_{max} . In this regard, the frame shifts happening not during the drug treatment introduction are way within the tolerance of our tracking algorithm. As shown in (TODO: add dropout rate figure), the dropouts caused by frame shifts in the other time points are basically noisy dropout caused by random noises in the time-lapse movie being tracked as cells [107].

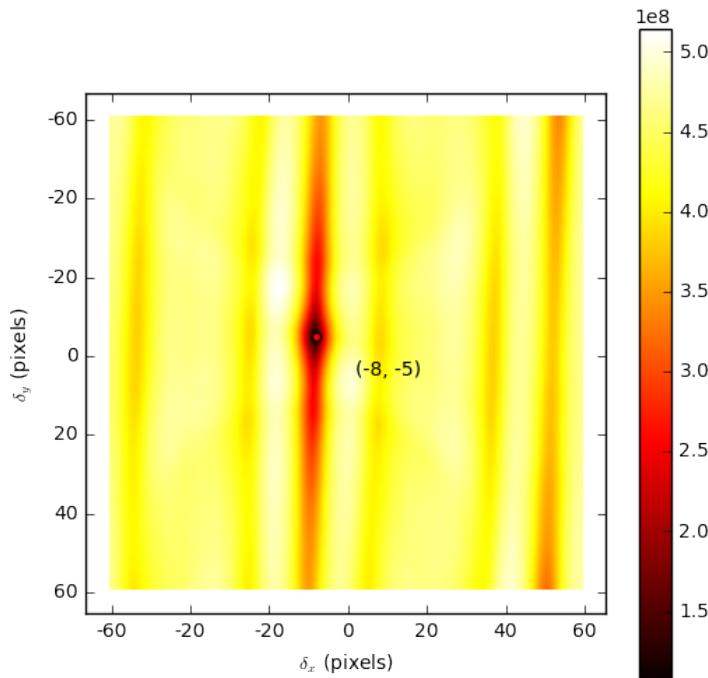


Figure 2.19: Search grid shift for the image position 26. The search was conducted for shift between the last time point before and the first time point after the drug treatment. The minimum is marked with thick black dot, which is returned after every grid-search call as inferred shift. In the position, the shift was inferred to be 8 pixels upwards and 5 pixels leftwards. Note the repeating pattern of relatively favorable configurations after approximately 50 horizontal and 100 vertical pixels caused by lattice nature of the micro-trenches.

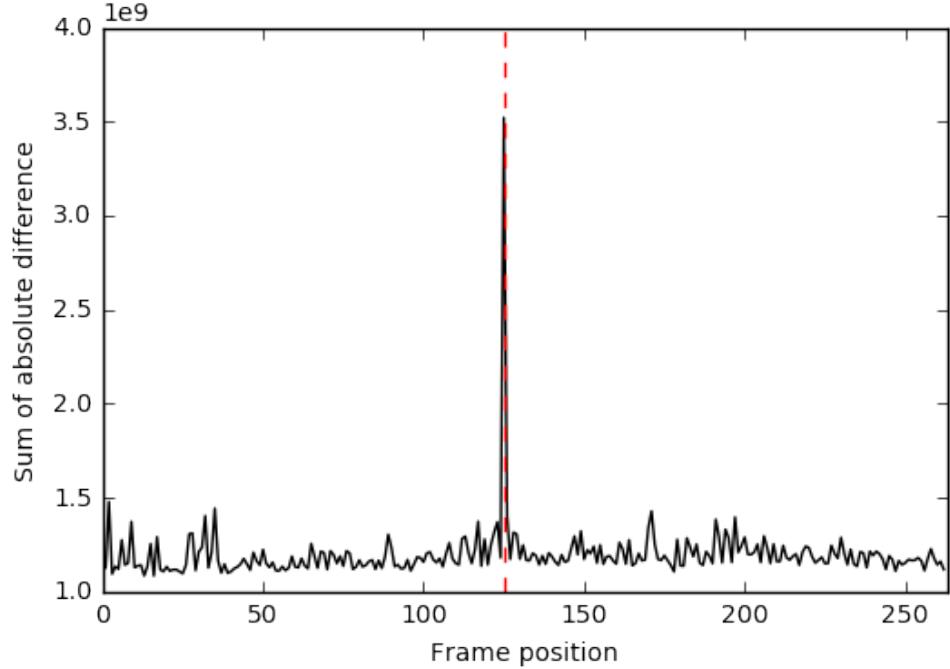


Figure 2.20: Pixel difference between consecutive frames in the image position 26. In most cases, the pixel difference between the frames is mainly caused by moving cells. The difference during the introduction of the drug treatment (red dashed line), on the other hand, is caused by physical shift of the frame. While moving cells mostly caused minimum noise-like pixel difference, the physical shift of field of view distorts the physical alignment and evokes immense pixel difference.

2.2.11 Support vector machine (SVM)

In machine learning, an SVM is a construct which, given the training set $\mathbf{S} \subset \mathbf{D}$ with,

$$\mathbf{S} = \{\mathbf{x}_1, \dots, \mathbf{x}_{|\mathbf{S}|}\}$$

and corresponding target class,

$$\mathbf{T} = \{y_1, \dots, y_{|\mathbf{S}|}\}$$

finds following things,

- A hyperplane that separates the input by its class, so that every point belonging to one class is located on one side of the hyperplane. This hyperplane is, in turn, defined by,
- support vectors.

A hyperplane is defined as a set of points \mathbf{x} in \mathbf{S} satisfying following criteria,

$$\mathbf{w} \cdot \mathbf{x} - b = 0 \tag{2.20}$$

(See Figure 2.21) where \mathbf{w} and $\frac{b}{\|\mathbf{w}\|}$ denote the normal vector to the hyperplane and the distance of the hyperplane from the origin along the normal vector \mathbf{w} . For every class c_i , the set of data points satisfying criteria,

$$\mathbf{w} \cdot \mathbf{x} - b = c_i \quad (2.21)$$

are called support vectors. The distance from the hyperplane to support vectors is thus,

$$\frac{1}{\|\mathbf{w}\|}$$

For two-classes classification, the classes are conventionally annotated as -1 and 1 . As the Equations suggests, for every point beyond (seen from the perspective of hyperplane) the support vectors of the class $c = -1$, following unequality applies,

$$\mathbf{w} \cdot \mathbf{x} - b < -1 \quad (2.22)$$

The analogous applies to the class $c = 1$,

$$\mathbf{w} \cdot \mathbf{x} - b > 1 \quad (2.23)$$

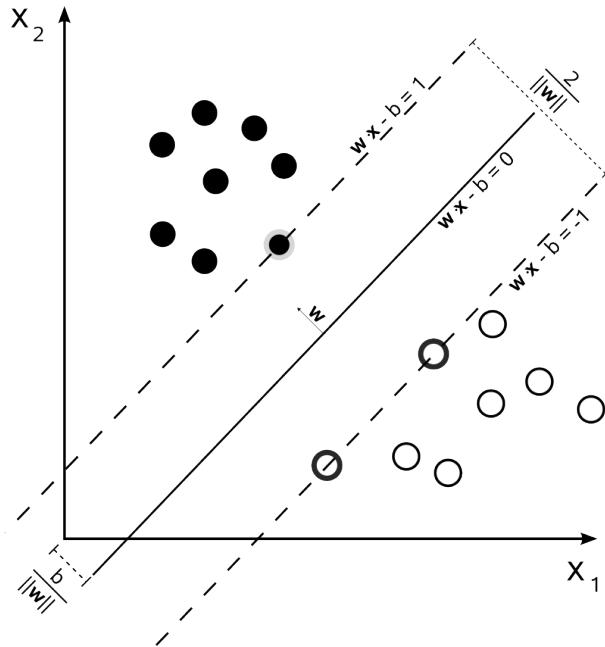


Figure 2.21: Illustration of a support vector machine in 2D. For black class there is one hyperplane-defining point (marked with grey margin) while for white class there are two (marked with bold black margin). The support vector \mathbf{w} is maximized for each class.

Following large margin principle [109, 110], an SVM tries to find support vectors that maximize $\|\mathbf{w}\|$. Given separability of the training data, the support vector \mathbf{w} can then be solved by solving following optimizing problem,

$$\begin{aligned} & \underset{\mathbf{w}}{\text{minimize}} \quad \|\mathbf{w}\| \\ & \text{subject to } y_i(\mathbf{w} \cdot \mathbf{x}_i - b \leq 1) \text{ for } i = 1, \dots, \|S\|. \end{aligned}$$

This hard-margin only converges only when the data are linearly separable in mapped space (also known as **feature space**). This is especially bad since many problems are

not linearly separable in their original space. There are two fundamental ways of relaxing this problem to enable classification using SVM:

- Relaxation of the definition of the SVM by allowing data points to be misclassified.
- Blowing up the input space into sufficiently high dimensional features using the kernel trick.

Relaxation of SVM

A relaxation of above explained problem is known as soft margin SVM or ξ -SVM [111]. The problem allows misclassification of some training data. In SVM, misclassification occurs when a data point belonging to a certain class c_j is located **not** in the area defined by the margin $\mathbf{w} \cdot \mathbf{x}_{c_j} = c_j$ for class c_j .

The problem is thus reduced to minimizing following term,

$$\left[\frac{1}{n} \sum_{i=1}^{|S|} \max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i - b)) \right] + \lambda \|\mathbf{w}\|^2 \quad (2.24)$$

The term inside of summation is called **classification error**. For correctly classified class we have $\mathbf{w} \cdot \mathbf{x}_i - b \leq -1$ and $\mathbf{w} \cdot \mathbf{x}_i - b \geq 1$ for $c_i = -1$ and $c_i = 1$ respectively, i.e. the summation term is 0 for every correctly classified data point. The coefficient λ is regularization coefficient which penalizes the magnitude of normal vector \mathbf{w} . Note that the higher the dimension of \mathbf{w} the larger the penalty is. This is important for next part on **kernel methods**.

Minimizing Term 2.24 is equal to optimizable with differentiable objective function [112]. We can for example introduce a variable ζ_i defined as,

$$\zeta_i = \max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i - b)) \quad (2.25)$$

This can be written as $y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 - \zeta_i$. Geometrically this is the maximum distance of a wrongly classified data point from the support vector. Using Definition 2.25 we can reduce optimization posed in Term 2.24 to,

$$\begin{aligned} & \underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{|S|} \sum_{i=1}^{|S|} \zeta_i + \lambda \|\mathbf{w}\|^2 \\ & \text{subject to} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \leq 1 - \zeta_i \\ & \text{and} \quad \zeta_i \geq 0 \text{ for all } i. \end{aligned}$$

Kernel method

As mentioned before, the optimization problem posed in Subsection 2.2.11 converges only in the case of linear separability of training data. While this mostly is not the case, Vapnik and Cortes [111] proposed the so-called **kernel trick**. It utilizes a kernel function Φ which maps the training set into high dimensional space representation. Essentially, a kernel function $\Phi : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+$ is a symmetric and non-negative function following the criteria by Mercer [113] defining it as, among others, general measure of similarity between two vectors.

In our case, we focus on two very well-known examples of kernel function, **the polynomial kernel function of order n** , defined as,

$$\Phi_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + r)^n \text{ with } r > 0 \quad (2.26)$$

and the **radial basis function (RBF)**, defined as,

$$\Phi_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \quad (2.27)$$

In both examples we can observe the assumed dimensionality of target feature space. A polynomial kernel function of order n maps the training data to n -dimensional feature space. Thus, blowing up the training data into higher dimensional requires the explicit assignment of a very high n value. The radial basis function on the other hand does not assume any dimensionality (or rather, it assumes unbounded dimensionality) as the Gaussian function used only assumes the input vectors \mathbf{x}_i and \mathbf{x}_j to be of the same dimensionality.

While assuming very high/unbounded dimensionality seems convenient at the start, this is not very straightforward, since:

- Given a non-powerful kernel function Φ that only maps the input into bounded number of dimension, the optimization problem may not converge if there is no representation that is mapable using the kernel function.
- Given a powerful kernel function Φ capable of blowing up the dimension to very high dimensions kernel might settle for an unnecessarily sparse dimensional reducing the generalizability of the problem.

Regularization

One solution is to introduce regularization. As already shown in Term 2.24, we can add a regularization term such as,

- L_1 -regularization: $\lambda \|\mathbf{w}\|^2$ [114], and
- L_2 -regularization (Tikhonov regularization): $\lambda \|\mathbf{w}\|$ [115].

Adding regularization will control against overfitted model by penalizing higher dimensional hyperplane \mathbf{w} as it the error will get blown up the higher dimensional, and its error minimizing hyperplane, are chosen.

Regressive SVM

SVM can be extended to enable regression. The method, initially proposed by Smola, Vapnik et al [116] and commonly known as Support Vector Regression (SVR), modifies the optimization problem of standard SVM to:

$$\begin{aligned} & \underset{\mathbf{w}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{w}\| \\ & \text{subject to} \quad y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - \mathbf{b} \leq \epsilon \\ & \text{and} \quad \langle \mathbf{w}, \mathbf{x}_i \rangle + \mathbf{b} - y_i \leq \epsilon \end{aligned}$$

where \mathbf{b} and $\langle \bullet, \bullet \rangle$ refer to the intercept of a linear model and the inner product operator. Note the term

$$\mathbf{y} = \langle \mathbf{w}, \mathbf{x}_i \rangle + \mathbf{b} + \sigma$$

being the term for standard linear model with intercept \mathbf{b} and error term σ .

2.2.12 Pearson's correlation coefficient (PCC)

In statistics, the PCC is a measure of the linear correlation between two random variables X and Y [117]. It has a value ranging from -1 to 1 . A correlation value of -1 denotes a perfectly inverse correlation, 0 denotes no linear correlation and 1 denotes a perfect positive correlation.

For two random variables X and Y , the PCC is defined as follows:

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \quad (2.28)$$

where $\text{cov}(X, Y)$ is the covariance of random variables X and Y , σ_X is the standard deviation of X and σ_Y is the standard deviation of Y .

The covariance of two random variables is in turn defined as

$$\text{cov}(X, Y) = E[(X - \mu_X)(Y - \mu_Y)] \quad (2.29)$$

with $E[X]$ denoting expected value of a random variable X . Combining Equations 2.28 and 2.29 and using definition of mean ($\mu_X = E[X]$) and standard deviation ($\sigma_X^2 = E[(X - E[X])^2]$) we can derive the formula of $\rho_{X,Y}$ to:

$$\rho_{X,Y} = \frac{E[XY] - E[X]E[Y]}{\sqrt{E[X^2] - [E[X]]^2} \sqrt{E[Y^2] - [E[Y]]^2}} \quad (2.30)$$

To calculate p-value, we first calculate t^* -value, defined as

$$t^* = \frac{\rho_{X,Y} \sqrt{n-2}}{\sqrt{1 - \rho_{X,Y}^2}} \quad (2.31)$$

where n denotes the number of observation. The p-value is then defined as probability of Student's t distribution T with $n - 2$ degrees of freedom having value x smaller than t , i.e.

$$p = P(Z \leq t^* \mid Z \sim T) \quad (2.32)$$

2.2.13 F-Test

In statistics, F-Test is statistical test in which the statistic $F_{X,Y}$ is assumed to be F-distributed under null Hypothesis, i.e. $F_{X,Y} \sim F(n_1, n_2) \mid H_0$ [117]. The distribution arise from the ratio of the χ^2 -variance of two normally distributed random variables, i.e. for two random variables $U \sim \chi^2(n_1)$ and $V \sim \chi^2(n_2)$, the ratio

$$F = \frac{U/n_1}{V/n_2} \quad (2.33)$$

is then F-distributed. The symbols n_1 and n_2 denote the degree of freedom of U and V . Conducting an F-test is thus reduced to checking whether, given two distributions X and Y , the F statistic of two distributions is F distributed. To do that, we first compute the F statistic of both distributions,

$$F_{X,Y} = \frac{\text{var}(X)}{\text{var}(Y)} \quad (2.34)$$

with **var** denoting the variance of the distribution $\text{var}(X) = E[(X - \mu)^2]$ The degree of freedom of both distributions is defined as,

$$n_1 = |X| - 1 \quad (2.35)$$

$$n_2 = |Y| - 1 \quad (2.36)$$

The cumulative density function (cdf) F_{cdf} of the F-distribution is given by

$$F_{cdf}(x; n_1, n_2) = \mathbf{I}_{\frac{n_1 x}{n_1 x + n_2}} \left(\frac{n_1}{2}, \frac{n_2}{2} \right) \quad (2.37)$$

where \mathbf{I}_x is the regularized beta function

$$\mathbf{I}_x(a, b) = \frac{\mathbf{B}(x; a, b)}{\mathbf{B}(a, b)} \quad (2.38)$$

and **B** is the beta function

$$B(x; a, b) = \int_0^x t^{a-1} (1-t)^{b-1} dt. \quad (2.39)$$

Using the standard definition of one-sided right tail p-value we can derive for the p-value of the F statistic as following:

$$\begin{aligned} p_F &= Pr(Z \geq F_{X,Y} | H_0) \\ p_F &= 1 - F_{cdf}(F_{X,Y}; n_1, n_2) \end{aligned} \quad (2.40)$$

for $Z \sim F(n_1, n_2)$.

Chapter 3

Analytic Pipeline

In this chapter the image and data analytic pipeline is presented. Each pipeline of image processing analysis is elaborated with reference to publications and the definitions from Chapter 2. Every method developed/used in the pipeline is brought forward and explained with references to scientific literature and the definitions brought forward in Chapter 2.

3.1 Image computing

To assess single cell characteristics (like its lifetime, time-to-death, division time, daughter cells and other information relating to its time- and generation-dependent cell cycle information), we track all cells in the brightfield channel from the start of the movie, assign cells to individual micro-trenches to e.g. filter out micro-trenches with multiple starting cells, and determine cell death via marker onset in the fluorescent channels.

Figure 3.5a shows how the image is sequentially processed from out-of-focus image to cell trees information.

3.2 Micro-trench masking

Several techniques could be applied to highlight certain areas in the image. In Subsection 1.3.2, several advances in computer vision methods are described. Furthermore, general advances in the field of machine learning is chronicled in Subsection 1.3.3. While the collection of advanced methods for region and image detection abound [118, 119, 120], some simple interpretable methods could be used best to detect and mask the micro-trenches. In particular, we can see that the area around a micro-trench exposes strong intensity gradient (see Figure 3.1): the area around the margin of a micro-trench is much darker than the other parts of the well. This can be explained by the fact that the light is reflected less around the wall area and thus the intensity decreases. Additionally, the light beam coming out of the laser is not perfectly perpendicular to the well and thus the non-perpendicular reflection is not reflected back to the camera sensor.

Before us, there are several methods that exploit this kind of phenomenon. Cheng et al [121] for example shows it is possible to recognize salient objects in image by using contrast and brightness adjustment. Our method on the other hand, goes further by doing robustness improvement by doing noise cancellation step (See 2.2.3).

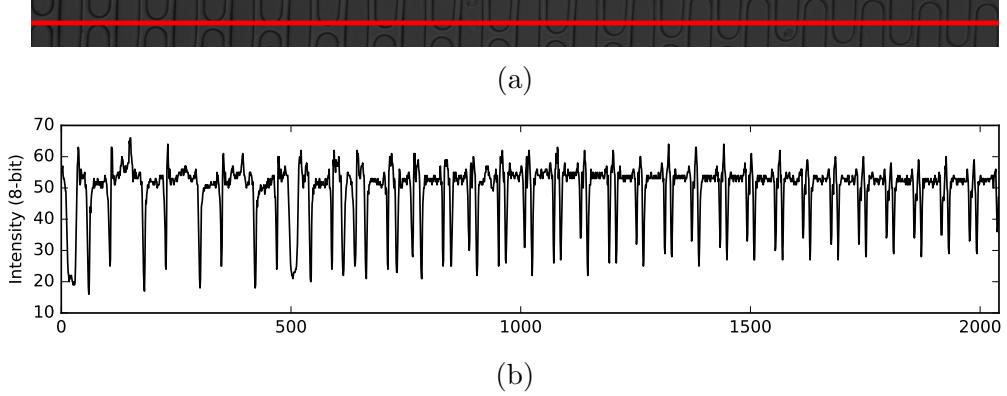


Figure 3.1: (a) The highlighted area around the 500-th column of the first slice of the image position 41. The range starts from the 450-th until the 550-th column. (b) The intensity of the red-marked area in (a), the region around micro-trench margin is indicated by the sudden drop of intensity. Note also the slow drift of the intensity as the pixel goes south (right side of the plot). This phenomenon is called *intensity gradient* (see Subsection 2.2.5 for more).

We could thus exploit this knowledge by designing micro-trench masking algorithm as follows:

3.2.1 Brightness and contrast adjustment of in-focus image

First, we adjust the brightness and contrast so that the area far away from the margin is encoded as maximum intensity, while the area around the margin is encoded as minimum intensity. One simple and robust way to do this is to reduce the intensity of each position with the maximum intensity value of the region around the margin.

Mathematically, we can define two sets of points, \mathfrak{M} and \mathfrak{S} . \mathfrak{M} refers to the set of points around the trench margin while \mathfrak{S} denotes the set of points far away from it. This brings us to the following transformation,

$$J_i(i, j) := I_t(i, j) + \operatorname{argmax}_{(x,y) \in \mathfrak{M}} I_t(x, y)$$

creating transformed intensity I^t .

As can be seen in Figures 2.11 and 3.1, while sets \mathfrak{M} and \mathfrak{S} are locally separable, across the board this does not seem so clear. We can thus improve the transformation by also adjusting the contrast parameter α by increasing it so that $\alpha > 1$. This brings the pixels with similar intensity values around the decision boundary (somewhere between the two distributions in Figure 2.11) apart and thus ameliorates the determination of decision boundary by the user. Applying this, we have now the new transformation:

$$J_i(i, j) := \alpha I_t(i, j) + \operatorname{argmax}_{(x,y) \in \mathfrak{M}} I_t(x, y) \text{ with } \alpha > 1$$

creating transformed intensity I^t .

Note that this is still not a perfect transformation, as there are some pixels in \mathfrak{S} with intensity lower than the minimum intensity of pixels in \mathfrak{M} . Besides doing this manually, we also improve this by refining the transformed intensity I^t further with the next step: Robust Automatic Threshold Selection.

Subfigures 3.5j and 3.5k show images of micro-trenches before and after brightness and contrast adjustment in the image processing pipeline. The step is done manually for each image position using the **Bright/Contrast Adjustment** function in Fiji.

3.2.2 RATS denoising

After brightness and contrast adjustment, some parts inside the micro-trench still have pixels that were not transformed to complete white color (maximum intensity). This can be removed by the global and local noise correction through Robust Automatic Threshold Selection (RATS, see Subsection 2.2.3). The algorithm is applied on each adjusted brightfield image, which creates a binary image of the micro-trench margins.

Subfigures 3.5k and 3.5l show the input and output of RATS in the image processing pipeline. The noise pertaining in a micro-trench is removed upon application of gradient-based filtering. We can thus say that,

- the first part of RATS (g -filtering by $\lambda\sigma$) is designed to move global level noise which in this case means white noises coming from non-perfect reflection of light from the well.
- the second part of RATS (recursive g -filtering by T_r) is designed to remove local noise cause by local distortion due to location-specific artifacts (direction of incoming light not perfectly perpendicular with regard to micro-trench's base etc).

The step is done automatically using the **Robust Automatic Threshold Selection (RATS)** plugin in Fiji with following parameters:

| Parameter | Value |
|-----------------------------------|-------|
| <code>noise_threshold</code> | 25 |
| <code>lambda_factor</code> | 3 |
| <code>min_leaf_size_pixels</code> | 408 |

3.2.3 Micro-trench mask filling

After applying brightness and contrast adjustment and RATS, the image now contains only the margin of micro-trenches. We can then fill these holes to create micro-trench masks. Subsection 2.2.4 describes the holes filling algorithm. The step is done automatically using the **Fill Holes** function in Fiji. Subfigures 3.5l and 3.5m show the input and output of the holes filling algorithm in the image processing pipeline. Upon holes filling, Each mask (appearing as a black rod-like form) is assigned a unique identity, which is then used as an identifier for tracks in both the brightfield and the fluorescent channel for filtering of cells and clones (see Sections 3.3 and 3.4).

3.3 Single cell tracking in the brightfield channel

To identify and track single cells from movie start, each out-of-focus brightfield image is processed in the following way (see Figure 3.5a for pipeline visualization):

3.3.1 Gaussian blurring

First, a Gaussian blur with a large radius of 50 pixel is applied to each brightfield image. The algorithm is explained in Subsection 2.2.5. Equation 2.10 of Subsection 2.2.5 is used to convolve every pixel in an image. The method results in the estimation of the image's background and global gradient. We now control the input image against the background and gradient by dividing every pixel of the original image by the pixel in the same position in the convoluted image (see Equation 2.11 of Subsection 2.2.5).

This step frees the out-of-focus image from global background and gradient which may have been arisen during image acquisition process caused by non-homogeneous lightning and unsynchronized lightning for example. This step is done using **Gaussian blur** filter implementation in Fiji with $\sigma = 50$ as parameter.

Subfigures 3.5b, 3.5c and 3.5d depict the input of the algorithm, computed background and output of the algorithm in the image processing pipeline.

3.3.2 Local contrast normalization

After removing global background and gradient, we normalize the local contrast of each image(Zuiderveld, 1994 [103]). Subsection 2.2.6 explains the details of the algorithm. This separates foreground from background. Upon running the algorithms the cells will be visible as negative while other parts of the image (micro-trench, noises etc) disappear. Figures 3.5d and 3.5e depict the input and output of the algorithm in the image processing pipeline. This step is done using **Normalize Local Contrast** function in Fiji with following parameters:

| Parameter | Value |
|--------------------|-------|
| block_radius_x | 100 |
| block_radius_y | 100 |
| standard_deviation | 1 |

3.3.3 Mean intensity correction

To reduce noise from micro-trench margins in the foreground, we calculate the pixel-wise intensity average from all slices in one image position and subtract this from each image to generate a binary image with mainly cells in the foreground. The details of this step is explained in Subsection 2.2.7. Equatoin This creates a mask of the cells. Equation 2.13 is used to calculate the mean while the correction is done using Equation 2.16.

Subfigures 3.5e, 3.5f and 3.5g depict the input of the algorithm, computed pixel-wise mean intensity and output of the algorithm in the image processing pipeline.

3.3.4 Cells recognition

After the image pre-processing, the image now consists of only the negative of the cells (see Subfigure 3.5g). Subsection 2.2.8 explains the Laplacian of Gaussian (LoG) detector in depth. The cells are recognized using the **Downsampled LoG Detector** in the Fiji TrackMate plugin [106] with following parameters:

| Parameter | Value |
|----------------------------------|-------|
| <code>pixel_width</code> | 0.647 |
| <code>pixel_height</code> | 0.647 |
| <code>voxel_depth_pixel</code> | 1.000 |
| <code>downsample_factor</code> | 2 |
| <code>blob_diameter_pixel</code> | 15 |
| <code>threshold</code> | 0 |

Subfigure 3.5h shows the recognized cells from input image shown in Subfigure 3.5g.

3.3.5 Cells tracking

Upon detection, the cells are assigned to tracks and concatenated to cellular trees. The details of the method used, the Linear Assignment Problem (LAP) Tracker [107], is elaborated in Subsection 2.2.9. The tracking is done using the **LAP Tracker** in the Fiji TrackMate plugin [106] with following parameters:

| Parameter | Value |
|--|-------|
| <code>pixel_width</code> | 0.647 |
| <code>pixel_height</code> | 0.647 |
| <code>voxel_depth_pixel</code> | 1.000 |
| <code>time_interval_second</code> | 600 |
| <code>frame_to_frame_linking_pixel</code> | 25 |
| <code>track_segment_gap_closing_pixel</code> | 25 |
| <code>track_segment_gap_closing_max_frame</code> | 6 |
| <code>track_segment_splitting_pixel</code> | 25 |

Figure 3.5i shows the tracked trajectory of the cells from Figure 3.5h.

3.4 Single cell tracking in the fluorescent channel and cell death signal determination

The main issue of fully tracking the cells using only brightfield images is the fact that the reliability of the tracking decreases as the cells are put under stress. Dying cells stop moving, become small and unstructured, and loose a distinctive bright signal in the brightfield channel, which makes them hard to track. The longer the experiment proceeds after treatment, the more likely that the cells will be lost from the tracking algorithm due to aforementioned factors.

We thus track dying cells in the fluorescent channels and concatenate the tracks with earlier tracks in the brightfield to determine time-to-death. Unlike in brightfield images, dying cells start emitting light under fluorescent light beam as they begin dying. This continues as the programmed cell death advances and the emitted light becomes even brighter (the progression of the emission follows roughly the step function, see Figure 3.3).

Knowing at one point the disruption by the drug treatments induces cell death and thus emission of PI or Caspase fluorescent signals (see t_{PMP} in Figure 3.3), finding the

cell death time is reduced to simply finding the time point in which the blob of the cell in fluorescent channels is detected. Take for example the Figure 3.2a. Here we see that on the upper right side of the image there are at least two blobs visible in the naked eye. We can thus do similar pre-processing step as that of brightfield images followed by image detection to detect at which time the onset happened (see Figure 3.5).

3.4.1 Brightness and contrast adjustment

This step follows closely similar method used in micro-trench masking in Subsection 3.2.2. The brightness and contrast are adjusted to remove the artefacts on the wafer. Subfigures 3.5b and 3.5c show the input and output of the brightness and contrast adjustment step.

3.4.2 Mean intensity correction

Again, the method follows closely the same method described in Subsection 3.3.3. The resulting image can be seen in Subfigure 3.5d.

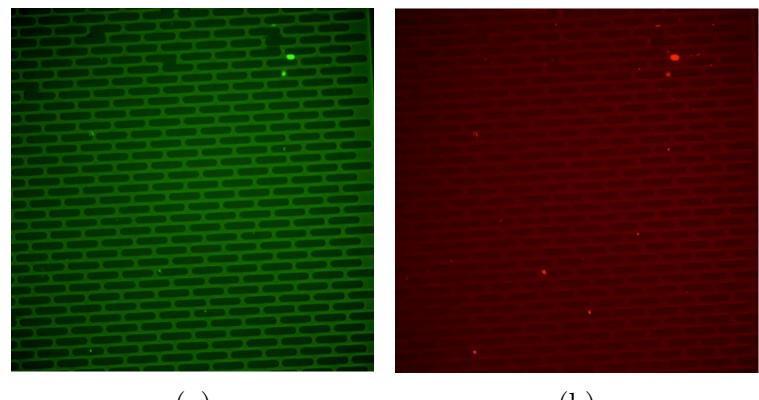
3.4.3 Cells recognition and tracking

After the pre-processing steps, the images undergo the detection and tracking steps in the similar fashion as the brightfield images. Following parameters are used to detect the cells:

| Parameter | Value |
|----------------------------------|-------|
| <code>pixel_width</code> | 0.647 |
| <code>pixel_height</code> | 0.647 |
| <code>voxel_depth_pixel</code> | 1.000 |
| <code>downsample_factor</code> | 2 |
| <code>blob_diameter_pixel</code> | 30 |
| <code>threshold</code> | 0 |

The assumed diameter is larger since the emission spreads around the cell and thus makes the cell negative larger than the cell body itself. Following parameters are used to track the cells:

| Parameter | Value |
|--|-------|
| <code>pixel_width</code> | 0.647 |
| <code>pixel_height</code> | 0.647 |
| <code>voxel_depth_pixel</code> | 1.000 |
| <code>time_interval_second</code> | 1800 |
| <code>frame_to_frame_linking_pixel</code> | 35 |
| <code>track_segment_gap_closing_pixel</code> | 35 |
| <code>track_segment_gap_closing_max_frame</code> | 2 |
| <code>track_segment_splitting_pixel</code> | 35 |



(a)

(b)

Figure 3.2: Example of images captured in (a) green and (b) red fluorescent channels. Upon programmed cell death, the bright emission is detected in corresponding channels coming from the activation of PI and Caspase 3/7 expression paths.

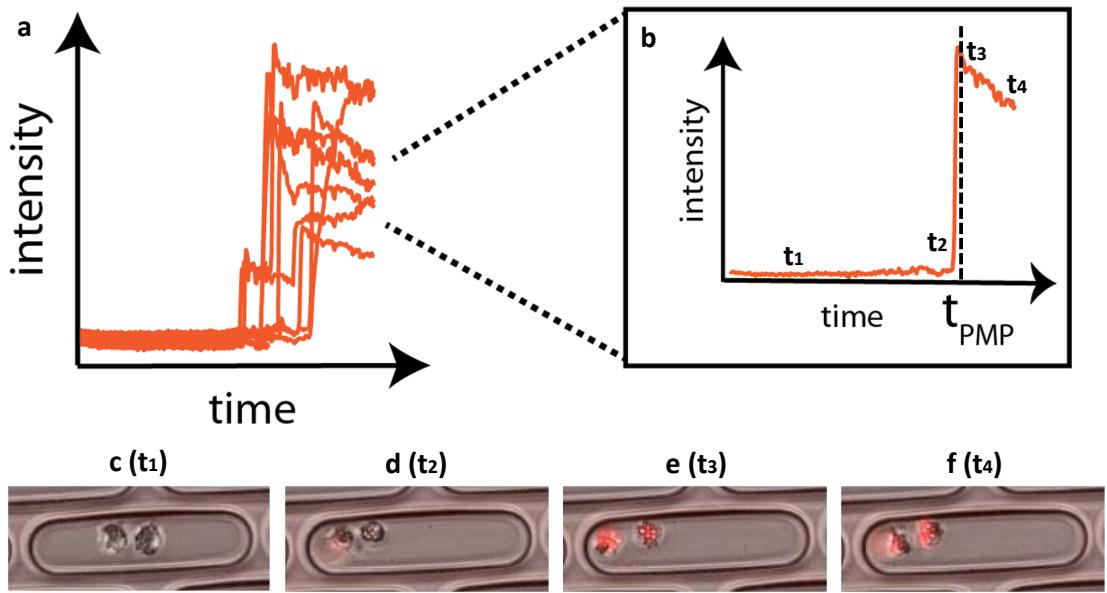


Figure 3.3: Illustration of annotation of PI death signal emission as cell death time (t_{PMP}). (a) shows how the total emission intensity will look like if it's being tracked (image taken for one related experiment of Radler's lab). (b) The hypothetical progression of total intensity of *the cell on the left* upon treatment with some approximate time points representing various cell phases marked with t_1 , t_2 , t_3 and t_4 . Initially, the left cell behaves normally upon treatment (c/ t_1). Only upon reaching the mitotic phase, the addition of Vincristine/Daunorubicin disrupts the process. Failure of entering the next phase sets the cell onto programmed cell death (d/ t_2). As the programmed cell death process advances, the PI emission became stronger (e/ t_3). At this point, the cel recognition will recognize the cell death signal as cell and thus represents cell death annotation. After a while, the emission will keep going on after a while (f/ t_4). We are, however, only interested with the onset time (t_{PMP}) to determine the cell death time. Note especially the difference of onset time between the left and right cells. At t_2 , the left cell has begun programmed cell death process while the right cell has not. This difference in time-to-death between sisters is one of the focuses of the project.

3.5 Pipeline visualization

The image processing pipelines for both brightfield and fluorescence channels can be seen in following figures:

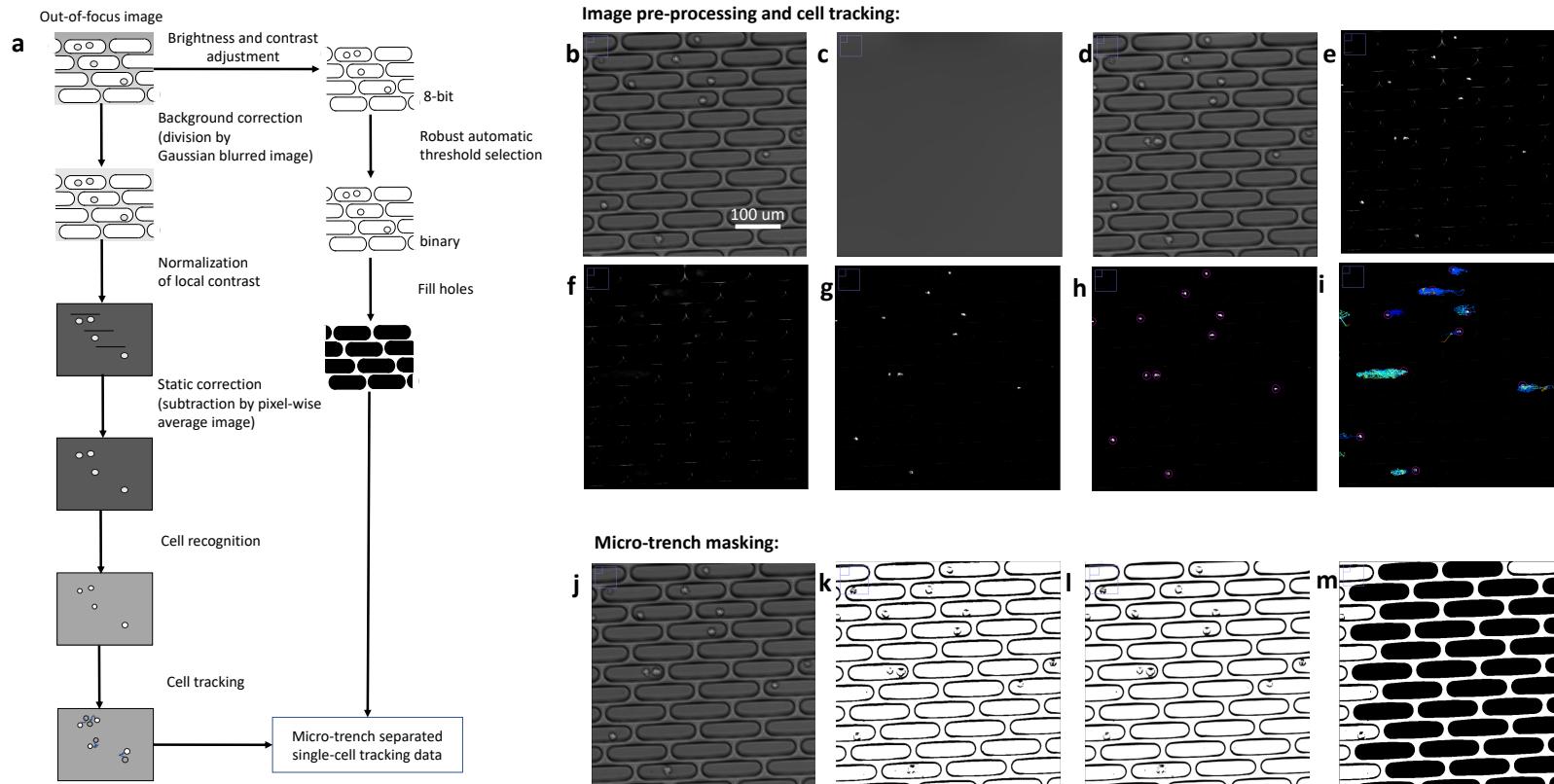


Figure 3.4: Brightfield channel pipeline. (j) The original out-of-focus image of the upper left corner of the image position 41. (k) The view of the same area as (l) after applying brightness and contrast adjustment. (m) The view of the same area after applying RATS on image in (n) showing the contour of micro-trenches in the area. (d) The contours filled with neutral color (black). Note that some micro-trenches are not filled as they are open and thus topologically not the outermost contours. Scale bar on the uppermost figure is $100\mu M$ long

57

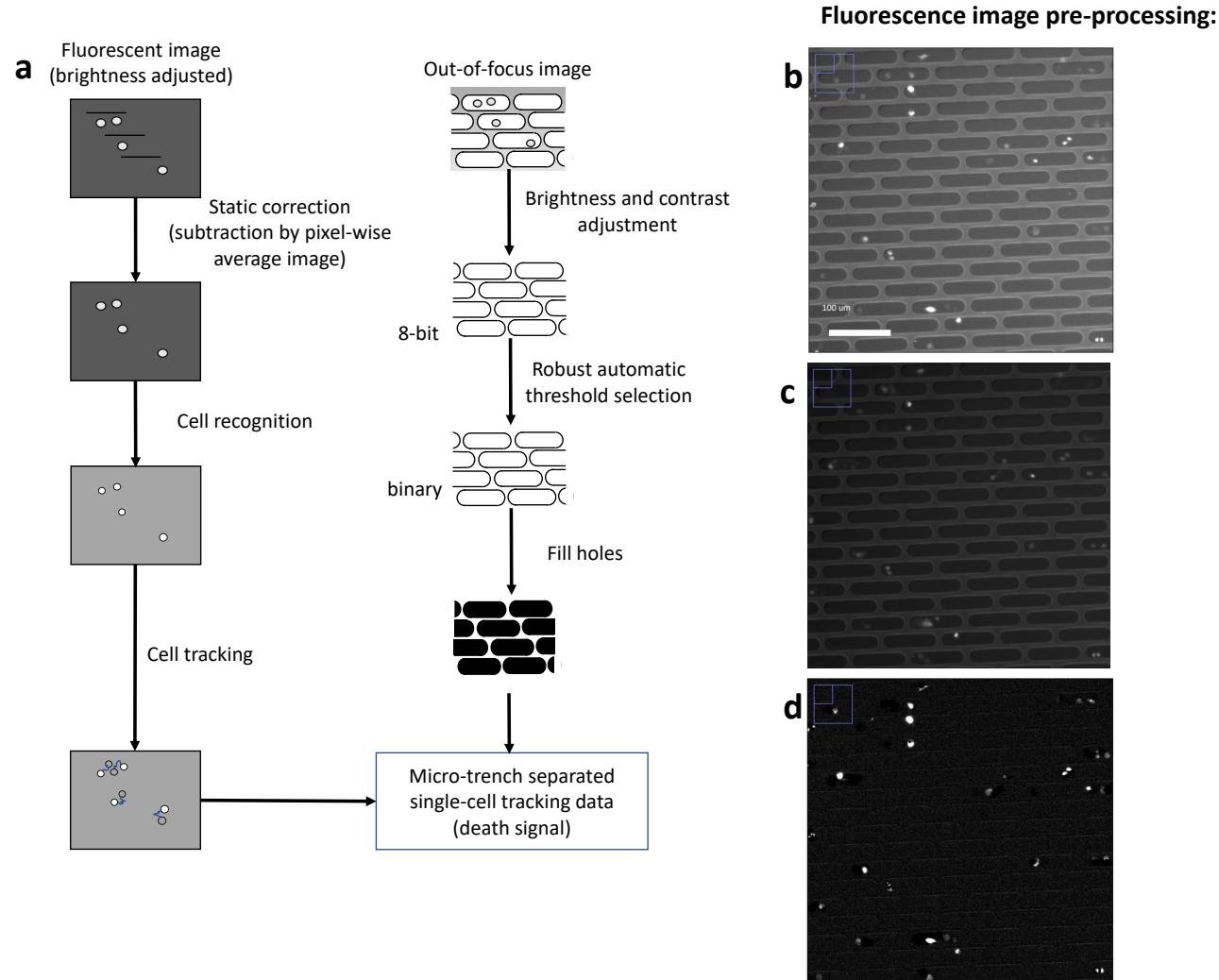


Figure 3.5: Fluorescence channel pipeline

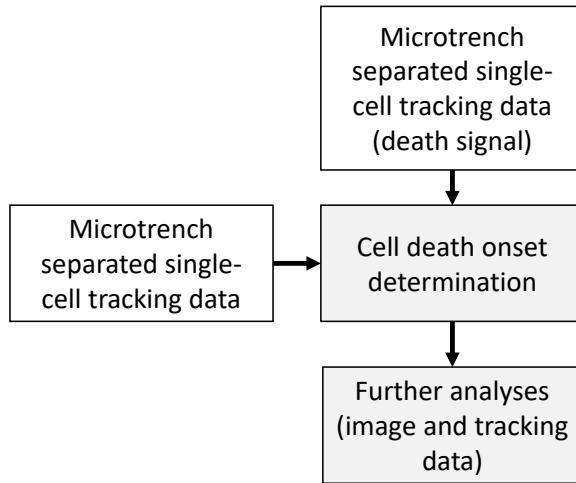


Figure 3.6: Combination of tracking data with cell death information creating cell death signal-corrected tracks data.

3.6 Implementation and availability

The image and tracking data is combined and post-processed using Python scripts utilizing OpenCV[101], NumPy [122], pandas [123] and Matplotlib [124], and the Jython and Trackmate [106, 125] plugin in Fiji [126]. The Python package scikit-learn is used for training and testing both support vector machine and random forest [127]. Our code is available at:

<https://github.com/raharjaliu/MA/tree/master/source>.

Chapter 4

Results

In this chapter the findings of the project are presented. The application of the pipeline in the image yields actionable data which in turn are analyzed. The chapter is separated into various parts:

- Distribution of cell cycle duration times. Here we analyze how the cell cycle duration time varies by cells in the wells.
- Variability of time-to-death between sister cells.
- Variability of time-to-death by time-in-cycle.
- Machine learning models of cell death.

4.1 Distribution of cell cycle duration times

Daughter cells contained in the micro-trenches and being separately trackable (Figure 2a) allow to accurately determine the cell cycle length (Figure 2b) and the time difference in division time points between sister cells (Figure 2c). For each cell, we determine the first division time point t_0 , the second division of the first daughter cell at time t_1 , and the division second daughter cell at t_2 . Figure 2a also shows an exemplary image series of one trench filled with initially one cell which gives rise to four cells within the first 30 hours. In our experiments 320 MOLM-13 cells were observed for 40 hours to detect a minimum of two divisions. The cell cycle duration distribution with a mean of 19.7 ± 2.6 (mean \pm std) hours is well described by both a lognormal distribution (red dotted line, Figure 2b) and a gamma distribution (dashed blue line, Figure 2b). Figure 2c shows the distribution of the difference between the cell cycle duration times for sister cells. In this case, 160 pairs of daughter cells were analyzed. The dashed blue line is a gamma distribution fit with a mean value of 2 h. In the inset a correlation plot of the cell cycle duration times for sister cells (black dots) and for randomly paired cells (red dots) is presented. Daughter cells divide in a highly correlated manner with a Pearson correlation of $r=0.85$, while random paired cells show a much smaller correlation ($r=0.25$).

The observed lognormal cell cycle duration distribution is well documented in literature, with both the size of mammalian cells as well as cell doubling times following a lognormal distribution [26, 27]. The underlying mechanism of cell division timing is an active field of research. It is however assumed that the so-called added model accepted in

bacteria [28] is also valid in somatic cells. In general size regulation can arise from various types of coupling between cell size, cell growth and cell cycle progression [29]. It has also been proposed, based on a coupled mathematical model of mammalian cell cycle and circadian clock, that the circadian clock triggers critical size control in the mammalian cell cycle and that it is more readily observed in cell lines that contain circadian rhythms [30]. In this context it is important that the adder mechanism also predicts exponential time correlation between sister cells as observed in our experiments [28].

4.2 Variability of time-to-death between sister cells

4.3 Variability of time-to-death by time-in-cycle

MOLM-13 cells were seeded from the cell culture flask in the micro-trenches array and directly placed under the time-lapse microscope. Mother cell mitosis, i.e. the division time (t_0), is used as a reference point of the cell-phase indication. After 20 hours, when most cells had divided once based on the previously measured cell cycle duration distribution (Figure 2b), the drugs (Vincristine or Daunorubicin) were added (Figure 3a). Time-lapse movies were simultaneously taken in brightfield and fluorescence. To track the cells in an automated routine (see Methods) we acquired the brightfield images in a slightly defocused mode (focus at $-20 \frac{1}{4}m$), which resulted in images with slightly blurred but well peaked intensity distribution. Fluorescence images of both the propidium iodide (PI) and the Cell Event Caspase 3/7 marker for the Vincristine data and of only the Cell Event Caspase 3/7 marker for the Daunorubicin data, since Daunorubicin is autofluorescent in the red region, were acquired. In Figure 3a a timeline of the experimental procedure is shown together with exemplary brightfield and fluorescence (PI marker) images are shown in overlay.

The measured time-to-death distribution is shown in figures in 3b for increasing drug concentration. The red black curves represent the Kernel density estimation of the probability density function of each histogram. The plots in Figure 3c depict, for each drug treatment, a scatter plot of the time passed in the cell cycle, i.e. the time passed from the division until the drug was added, with the time-to-death i.e. the time passed from the drug administration time point until the death of the cell. The blue lines are best linear fits and show the dependence of the time-to-death on the time spent in the cell cycle. The colored areas indicate the cell cycle phase based on the division distribution in figure 2b, the duration for each phase is calculated based on the phase durations proposed in T. S. Weber et al. 2014 [32]. We observed a negative correlation between the time in the cell cycle and the death time, which increased for increasing VCR concentrations (1-100 nM), but there was no correlation for the highest VCR concentration (1000 nM), indicating that in this high concentration side effect toxicities are prominent. The Pearson product-moment correlation test results are presented in Table 1. In the case of Daunorubicin (100 nM), we observed an even stronger negative correlation between the time spent in the cell-cycle and the time-to-death in the case of Daunorubicin (100 nM). Especially the time-to-death of the cells that were in the S phase when the drug was added, the phase in which Daunorubicin should be more effective, had a larger deviation in comparison to the time-of-death of cells that were in the G1 or in the G2/M phase.

Comparison of synchronized and unsynchronized populations. We performed a cy-

totoxic test on the micro-trenches array using a synchronized cell population with the "double thymidine block" procedure [33] (Figure 4). Thymidine arrests cells at the G1/S border, which is after the division time point within the length of G1 phase. Cells were released 3 h before the start of imaging and seeded in micro-trenches. The drug was added just before starting imaging. In Figure 4a, we show a timeline of the experimental procedure. In Figure 4b, we plot the distributions of the time-to-death in each drug concentration. The blue black curves represent the Kernel density estimation of the probability density function of each histogram. For all drug concentrations the maximum of these distributions is near the 15th hour after starting imaging.

In Figure 5 we compare our results of the unsynchronized cell population (Figure 3) with a cell population synchronized with thymidine block (Figure 4). For both the unsynchronized and the synchronized population, the total number of dead cells as a function of drug concentration is shown in Figure 5a and shows the expected increase in death response with increasing dose. Figure 5b, shows the death times between the synchronized (blue) and the unsynchronized (red) population. The distributions plotted are the normalized number of cells in all drug treatments, for all cells tracked. Apart from a peak at the beginning of the measurement in the synchronized population, the shape of the two distributions is equivalent. In the synchronized population less cells have a short time-to-death, especially evident in the VCR 1000 nM concentration (Figure 3b and 4b), indicating that the synchronized population is more resistant to death. In Figure 5c, we show scatter plots of the time-to-death between sister cells in an ensemble of all drug treatments. The Pearson correlation coefficient (r) is 0.5 (medium correlation) for the sister cells of the unsynchronized population and 0.06 for the synchronized. The ellipses indicate the directionality of the correlation. Thus, the time-to-death between sister cells was positively correlated for the unsynchronized population but not for the synchronized.

4.4 Machine learning models of cell death

Chapter 5

Summary and Outlook

We showed that arrays of micro-structured trenches provide a platform that enables a label-free method for tracking cells and for approximating the cell cycle phase without the use of molecular markers. Our approach allows to set individual clocks in single cells using the first division for each occupied trench as a starting point. This is highly accurate and overcomes typical drawbacks of low detection efficiency of fluorescence-based indicators such as the FUCCI marker, which usually have low transfection efficiency round 20-40%, and generally short duration of staining, namely 15 hours, which is no sufficient for our long-time measurements. By compartmentalizing the cell population in small groups we enable an image-based cell tracking. Without the use of such an array, time-lapse observation of a cell population for such long hours, even beyond 48 hours, is impossible for both adherent and non-adherent cells since cells escape from the field of view very fast, within a few hours. In addition, compartmentalization of the cell population not only reduces the error of mixing the identities of adjacent cells but also the time and computational power needed for tracking.

In addition, we demonstrate the practicability of the micro-trench platform to determine the time-to-death after induction of cell death with vincristine and daunorubicin. At high concentration vincristine stimulates microtubule depolymerization and mitotic spindle destruction. At lower clinically relevant concentrations, it blocks mitotic progression. As a result, we expected a negative correlation between the time spent in the cell-cycle and the time-to-death in the case of VCR, since a cell that spent a long large amount of time in the cell-cycle should be closer to the M-phase so it should have a shorter time-to-death i.e. it should die earlier. Indeed here we observed that the time-to-death negatively correlates with the time spent in the cell cycle and that this correlation becomes more prominent with increasing VCR concentration up to 100 nM, but there is no correlation at the highest concentration (1000 nM). The reason for no correlation in the 1000 nM might be side effect toxicities that happen through the whole cell cycle and as it has been shown before [12], after exposure to antimitotic drugs cells display complex fate profiles, such as unequal cell division producing aneuploid daughter cells, exiting the cell cycle without undergoing cell division (mitotic slippage), or exiting G1 and undergoing apoptosis or senescence. On the other hand, the correlation of the time-to-death with the time spent in the cell cycle is more prominent in the case of Daunorubicin (100 nM) and the deviation of the time-to-death for the cells in the S-phase during drug addition, is larger compared to the cells in G1 or G2/M phase (Figure 3b).

We furthermore find that the sister cell correlations of the time-to-death for the unsyn-

chronized and the synchronized populations differ. We observed similar response between sister cells in the unsynchronized experiment as far as the time-to-death is concerned, during an observation period of 24 hours after the addition of the drugs. At the same time the distributions of the time-to-death (average of all different drug concentrations) of the unsynchronized and the synchronized populations are similar, which would suggest that synchronizing the cells with the double thymidine block did not affect their response to both tested drugs. However, a closer look into the data and especially when analyzing the response between sister cells of the synchronized population, we observed that there is no correlation of the time-to-death between sister cells, contrary to the results from the unsynchronized population. This observation suggests that the double thymidine block procedure has an effect on sister cell response heterogeneity to vincristine and daunorubicin. Sister cell response variability is a matter drawing attention since it is an indication whether different phenotypes stems from genetic differences or from differences in the protein state of the cells. The results available so far strongly depend on the cell process that is interrogated. For instance, it has been previously reported that sister cells undergo apoptosis synchronously [5, 34]. However, in response to antimitotic drugs the fate of one sister is independent of the fate of the other [12]. In another study, sister cell fate in response to TRAIL-induced apoptosis correlated, as did also the time-to-death between HeLa sister cells but this correlation decayed as a function of time since division, the time period tested was 8 hours. Based on this observation, the transient heritability in fate model was proposed, which states that protein synthesis promotes cell divergence so that sister cells soon become no more similar to each other than random paired cells[1].

In future work, the micro-trenches array can be used with more than one drug, namely for combination therapy, to determine the optimal administration timing of each drug. Microfluidic devices have facilitated single-cell studies and boosted the collection of quantitative experimental data, such as the measurement of single-cell mass with high accuracy[35]. Measuring the cell size or the volume added to the cell after its mother division are crucial quantitative data to elucidate the underlying mechanisms that drive cell division. The micro-trenches array coupled with time-lapse microscopy and automated image analysis is a step toward this direction.

5.1 Fully automated microfluidics pipeline

5.2 Fully automated pipeline for single-cell analysis

Improvements in analysis pipeline:

- better method for masking recognition
- possibly even machine learning based methods

I. Summary (and discussion?)

Connect Results with Background

II. Outlook

Improvemnt capability

5.2.1 Automated micro-trench masking

One positive thing about the design of our data analysis pipeline is the modularity of its component. As mentioned multiple times in this thesis (in particular in Subsection 3.2, our method can well be improved by automatizing the micro-trench masking process. Although itself not a bottleneck of the process, as the entire dataset of 63 image positions could be processed in less than an hour, automating this process will save time and more importantly, clicking and dragging efforts, thus lets scientists do the things that matter more for them.

There are several methods that came to mind in improving the micro-trench masking, which generally can be separated in two groups: ML-based methods and non-ML-based methods. ML-based methods generally require either a manual annotation of correctly recognized micro-trench (a class of methods called supervised learning methods) or certain prior knowledge regarding distinctive features of micro-trench that separate it from background image and/or the cells (a class of methods called unsupervised/semi-supervised learning methods).

In the following part of this subsection various potential implementations of both ML-based and non-ML-based methods are listed down.

Maximally interesting extremal region (MNER)

Introduced by Oliver Hilsenbeck in his master's thesis in 2014 [128]

Bibliography

- [1] Akulapalli Sudhakar. History of cancer, ancient and modern treatment methods. *Journal of cancer science & therapy*, 1(2):1, 2009.
- [2] The Centers for Disease Control National Cancer Institute and the North American Association of Central Cancer Registries Prevention, American Cancer Society. Cancer Statistics. <https://www.cancer.gov/about-cancer/understanding/statistics>, 2017. [Online; accessed 07-November-2017].
- [3] Jacques Ferlay, Isabelle Soerjomataram, Rajesh Dikshit, Sultan Eser, Colin Mathers, Marise Rebelo, Donald Maxwell Parkin, David Forman, and Freddie Bray. Cancer incidence and mortality worldwide: sources, methods and major patterns in globocan 2012. *International journal of cancer*, 136(5), 2015.
- [4] George M Whitesides. The origins and the future of microfluidics. *Nature*, 442(7101):368–373, 2006.
- [5] Farzad Sekhavati. *Dynamic response of individual cells in heterogeneous population*. PhD thesis, lmu, 2015.
- [6] SS Antman JE Marsden, L Sirovich S Wiggins, L Glass, RV Kohn, and SS Sastry. *Interdisciplinary Applied Mathematics*. Springer, 1993.
- [7] Klaus Eyer, Phillip Kuhn, Conni Hanke, and Petra S Dittrich. A microchamber array for single cell isolation and analysis of intracellular biomolecules. *Lab on a Chip*, 12(4):765–772, 2012.
- [8] Aaron R Wheeler, William R Throndset, Rebecca J Whelan, Andrew M Leach, Richard N Zare, Yish Hann Liao, Kevin Farrell, Ian D Manger, and Antoine Dari-don. Microfluidic device for single-cell analysis. *Analytical chemistry*, 75(14):3581–3586, 2003.
- [9] MP MacDonald, GC Spalding, and Kishan Dholakia. Microfluidic sorting in an optical lattice. *Nature*, 426(6965):421–424, 2003.
- [10] Svetlana A Tatarkova, Wilson Sibbett, and Kishan Dholakia. Brownian particle in an optical potential of the washboard type. *Physical review letters*, 91(3):038101, 2003.
- [11] Mark M Wang, Eugene Tu, Daniel E Raymond, Joon Mo Yang, Haichuan Zhang, Norbert Hagen, Bob Dees, Elinore M Mercer, Anita H Forster, Ilona Kariv, et al. Microfluidic sorting of mammalian cells by optical force switching. *Nature biotechnology*, 23(1):83–87, 2005.

- [12] Jean-Christophe Baret, Oliver J Miller, Valerie Taly, Michaël Ryckelynck, Abdeslam El-Harrak, Lucas Frenz, Christian Rick, Michael L Samuels, J Brian Hutchison, Jeremy J Agresti, et al. Fluorescence-activated droplet sorting (fads): efficient microfluidic cell sorting based on enzymatic activity. *Lab on a Chip*, 9(13):1850–1858, 2009.
- [13] Xuan Mu, Wenfu Zheng, Jiashu Sun, Wei Zhang, and Xingyu Jiang. Microfluidics for manipulating cells. *Small*, 9(1):9–21, 2013.
- [14] Farzad Sekhavati, Max Endele, Susanne Rappl, Anna-Kristina Marel, Timm Schroeder, and Joachim O Rädler. Marker-free detection of progenitor cell differentiation by analysis of brownian motion in micro-wells. *Integrative Biology*, 7(2):178–183, 2015.
- [15] Nikos Chronis, Manuel Zimmer, and Cornelia I Bargmann. Microfluidics for in vivo imaging of neuronal and behavioral activity in *caenorhabditis elegans*. *Nature methods*, 4(9):727–731, 2007.
- [16] R Di Leonardo, L Angelani, D Dellâ€™Arciprete, Giancarlo Ruocco, V Iebba, S Schippa, MP Conte, F Mecarini, F De Angelis, and E Di Fabrizio. Bacterial ratchet motors. *Proceedings of the National Academy of Sciences*, 107(21):9541–9545, 2010.
- [17] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.
- [18] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.
- [19] Kenneth J Pienta, William B Isaacs, Donald Vindivich, and Donald S Coffey. The effects of basic fibroblast growth factor and suramin on cell motility and growth of rat prostate cancer cells. *The Journal of urology*, 145(1):199–202, 1991.
- [20] Gabriel Fenteany and Shoutian Zhu. Small-molecule inhibitors of actin dynamics and cell motility. *Current topics in medicinal chemistry*, 3(6):593–616, 2003.
- [21] Maria Grazia Ruocco, Karsten A Pilones, Noriko Kawashima, Michael Cammer, Julie Huang, James S Babb, Mengling Liu, Silvia C Formenti, Michael L Dustin, and Sandra Demaria. Suppressing t cell motility induced by anti-ctla-4 monotherapy improves antitumor effects. *The Journal of clinical investigation*, 122(10):3718, 2012.
- [22] Ulrich Schneider, Hans-Ulrich Schwenk, and Georg Bornkamm. Characterization of ebv-genome negative null and t cell lines derived from children with acute lymphoblastic leukemia and leukemic transformed non-hodgkin lymphoma. *International journal of cancer*, 19(5):621–626, 1977.
- [23] David S Johnson, Ali Mortazavi, Richard M Myers, and Barbara Wold. Genome-wide mapping of in vivo protein-dna interactions. *Science*, 316(5830):1497–1502, 2007.

- [24] Mark Schena, Dari Shalon, Renu Heller, Andrew Chai, Patrick O Brown, and Ronald W Davis. Parallel human genome analysis: microarray-based expression monitoring of 1000 genes. *Proceedings of the National Academy of Sciences*, 93(20):10614–10619, 1996.
- [25] Roberta A Gottlieb, Judy Nordberg, Evan Skowronski, and Bernard M Babior. Apoptosis induced in jurkat cells by several agents is preceded by intracellular acidification. *Proceedings of the National Academy of Sciences*, 93(2):654–658, 1996.
- [26] Afshin Samali, Jiyang Cai, Boris Zhivotovsky, Dean P Jones, and Sten Orrenius. Presence of a pre-apoptotic complex of pro-caspase-3, hsp60 and hsp10 in the mitochondrial fraction of jurkat cells. *The EMBO journal*, 18(8):2040–2048, 1999.
- [27] Takefumi Kasai, Yoichi Iwanaga, Hidekatsu Iha, and Kuan-Teh Jeang. Prevalent loss of mitotic spindle checkpoint in adult t-cell leukemia confers resistance to microtubule inhibitors. *Journal of Biological Chemistry*, 277(7):5187–5193, 2002.
- [28] Bryan C Barnhart, Patrick Legembre, Eric Pietras, Concetta Bubici, Guido Franzoso, and Marcus E Peter. Cd95 ligand induces motility and invasiveness of apoptosis-resistant tumor cells. *The EMBO journal*, 23(15):3175–3185, 2004.
- [29] Y Matsuo, RAF MacLeod, CC Uphoff, HG Drexler, C Nishizaki, Y Katayama, G Kimura, N Fujii, E Omoto, M Harada, et al. Two acute monocytic leukemia (aml-m5a) cell lines (molm-13 and molm-14) with interclonal phenotypic heterogeneity showing mll-af9 fusion resulting from an occult chromosome insertion, ins (11; 9)(q23; p22p23). *Leukemia (08876924)*, 11(9), 1997.
- [30] Daniel A Arber, Attilio Orazi, Robert Hasserjian, Jürgen Thiele, Michael J Borowitz, Michelle M Le Beau, Clara D Bloomfield, Mario Cazzola, and James W Vardiman. The 2016 revision to the world health organization (who) classification of myeloid neoplasms and acute leukemia. *Blood*, pages blood–2016, 2016.
- [31] Louise M Kelly, Jin-Chen Yu, Christina L Boulton, Mutiah Apatira, Jason Li, Carol M Sullivan, Ifor Williams, Sonia M Amaral, David P Curley, Nicole Duclos, et al. Ct53518, a novel selective flt3 antagonist for the treatment of acute myelogenous leukemia (aml). *Cancer cell*, 1(5):421–432, 2002.
- [32] S Yokota, H Kiyo, M Nakao, T Iwai, S Misawa, T Okuda, Y Sonoda, T Abe, K Kahsim, Y Matsuo, et al. Internal tandem duplication of the flt3 gene is preferentially seen in acute myeloid leukemia and myelodysplastic syndrome among various hematological malignancies. a study on a large series of patients and cell lines. *Leukemia (08876924)*, 11(10), 1997.
- [33] Thermo Fischer Scientific Life Technologies. Gibco Cell Culture Media. <https://www.thermofisher.com/de/de/home/life-science/cell-culture/mammalian-cell-culture/classical-media.html>, 2017. [Online; accessed 12-November-2017].
- [34] Keril J Blight, Alexander A Kolykhalov, and Charles M Rice. Efficient initiation of hcv rna replication in cell culture. *Science*, 290(5498):1972–1974, 2000.

- [35] Tatsuya Shimizu, Masayuki Yamato, Yuki Isoi, Takumitsu Akutsu, Takeshi Setomaru, Kazuhiko Abe, Akihiko Kikuchi, Mitsuo Umezawa, and Teruo Okano. Fabrication of pulsatile cardiac tissue grafts using a novel 3-dimensional cell sheet manipulation technique and temperature-responsive cell culture surfaces. *Circulation research*, 90(3):e40–e48, 2002.
- [36] LD Satter and LL Slyter. Effect of ammonia concentration on rumen microbial protein production in vitro. *British journal of nutrition*, 32(2):199–208, 1974.
- [37] Thermo Fischer Scientific Life Technologies. Fetal Bovine Serum. <https://www.thermofisher.com/de/de/home/life-science/cell-culture/mammalian-cell-culture/fbs.html>, 2017. [Online; accessed 12-November-2017].
- [38] Anna-Kristina Marel, Susanne Rappl, Alicia Piera Alberola, and Joachim Oskar Rädler. Arraying cell cultures using peg-dma micromolding in standard culture dishes. *Macromolecular bioscience*, 13(5):595–602, 2013.
- [39] The American Society of Health-System Pharmacists. Daunorubicin hydrochloride – Monograph for Professionals. <https://www.drugs.com/monograph/daunorubicin-hydrochloride.html>, 2015. [Online; accessed 02-January-2018].
- [40] Enrique Raviña. *The evolution of drug discovery: from traditional medicines to modern drugs*. John Wiley & Sons, 2011.
- [41] Takashi Tsuruo, Harumi Iida, Shigeru Tsukagoshi, and Yoshio Sakurai. Overcoming of vincristine resistance in p388 leukemia in vivo and in vitro through enhanced cytotoxicity of vincristine and vinblastine by verapamil. *Cancer research*, 41(5):1967–1972, 1981.
- [42] David A Gewirtz. A critical evaluation of the mechanisms of action proposed for the antitumor effects of the anthracycline antibiotics adriamycin and daunorubicin. *Biochemical pharmacology*, 57(7):727–741, 1999.
- [43] National Cancer Institute. Definition of vincristine sulfate – NCI Drug Dictionary. <https://www.cancer.gov/publications/dictionaries/cancer-drug?CdrID=42251>, 2015. [Online; accessed 02-January-2018].
- [44] Takashi Tsuruo, Harumi Iida, Shigeru Tsukagoshi, and Yoshio Sakurai. Increased accumulation of vincristine and adriamycin in drug-resistant p388 tumor cells following incubation with calcium antagonists and calmodulin inhibitors. *Cancer research*, 42(11):4730–4733, 1982.
- [45] Allan Jordan, John A Hadfield, Nicholas J Lawrence, and Alan T McGown. Tubulin as a target for anticancer drugs: agents which interact with the mitotic spindle. *Medicinal research reviews*, 18(4):259–296, 1998.
- [46] Sharon Lobert, Bojana Vulevic, and John J Correia. Interaction of vinca alkaloids with tubulin: a comparison of vinblastine, vincristine, and vinorelbine. *Biochemistry*, 35(21):6806–6814, 1996.

- [47] Richard J Owellen, Albert H Owens, and Douglas W Donigian. The binding of vinchristine, vinblastine and colchicine to tubulin. *Biochemical and biophysical research communications*, 47(4):685–691, 1972.
- [48] Wolfgang Hiddemann, Michael Kneba, Martin Dreyling, Norbert Schmitz, Eva Lengfelder, Rudolf Schmits, Marcel Reiser, Bernd Metzner, Harriet Harder, Susanna Hegewisch-Becker, et al. Frontline therapy with rituximab added to the combination of cyclophosphamide, doxorubicin, vincristine, and prednisone (chop) significantly improves the outcome for patients with advanced-stage follicular lymphoma compared with therapy with chop alone: results of a prospective randomized study of the german low-grade lymphoma study group. *Blood*, 106(12):3725–3732, 2005.
- [49] BF Brandriff, ML Meistrich, LA Gordon, AV Carrano, and JC Liang. Chromosomal damage in sperm of patients surviving hodgkin’s disease following mopp (nitrogen mustard, vincristine, procarbazine, and prednisone) therapy with and without radiotherapy. *Human genetics*, 93(3):295–299, 1994.
- [50] MG Pfreundschuh, WD Schoppe, R Fuchs, KH Pflüger, M Loeffler, and V Diehl. Lomustine, etoposide, vindesine, and dexamethasone (cevd) in hodgkin’s lymphoma refractory to cyclophosphamide, vincristine, procarbazine, and prednisone (copp) and doxorubicin, bleomycin, vinblastine, and dacarbazine (abvd): a multicenter trial of the german hodgkin study group. *Cancer treatment reports*, 71(12):1203–1207, 1987.
- [51] Volker Diehl, Jeremy Franklin, Dirk Hasenclever, Hans Tesch, Michael Pfreundschuh, Bernd Lathan, Ursula Paulus, Markus Sieber, Jens-Ulrich Rueffer, Michael Sextro, et al. Beacopp, a new dose-escalated and accelerated regimen, is at least as effective as copp/abvd in patients with advanced-stage hodgkin’s lymphoma: interim report from a trial of the german hodgkin’s lymphoma study group. *Journal of Clinical Oncology*, 16(12):3810–3821, 1998.
- [52] Nancy L Bartlett, Saul A Rosenberg, Richard T Hoppe, Steven L Hancock, and Sandra J Horning. Brief chemotherapy, stanford v, and adjuvant radiotherapy for bulky or advanced-stage hodgkin’s disease: a preliminary report. *Journal of Clinical Oncology*, 13(5):1080–1088, 1995.
- [53] Yeon S Ahn, William J Harrington, Robert C Seelman, and Charles S Eytel. Vinchristine therapy of idiopathic and secondary thrombocytopenias. *New England Journal of Medicine*, 291(8):376–380, 1974.
- [54] National Cancer Institute. Definition of doxorubicin hydrochloride – NCI Drug Dictionary. <https://www.cancer.gov/publications/dictionaries/cancer-drug?CdrID=38860>, 2015. [Online; accessed 02-January-2018].
- [55] National Cancer Institute. Definition of epirubicin hydrochloride – NCI Drug Dictionary. <https://www.cancer.gov/publications/dictionaries/cancer-drug?cdrid=39232>, 2015. [Online; accessed 02-January-2018].

- [56] National Cancer Institute. Definition of idarubicin hydrochloride – NCI Drug Dictionary. <https://www.cancer.gov/publications/dictionaries/cancer-drug?cdrid=39753>, 2015. [Online; accessed 02-January-2018].
- [57] Raymond B Weiss. The anthracyclines: will we ever find a better doxorubicin? In *Seminars in oncology*, volume 19, pages 670–686. Elsevier, 1992.
- [58] Giorgio Minotti, Pierantonio Menna, Emanuela Salvatorelli, Gaetano Cairo, and Luca Gianni. Anthracyclines: molecular advances and pharmacologic developments in antitumor activity and cardiotoxicity. *Pharmacological reviews*, 56(2):185–229, 2004.
- [59] Xuyang Peng, Billy Chen, Chee Chew Lim, and Douglas B Sawyer. The cardiotoxicology of anthracycline therapeutics: translating molecular mechanism into preventative medicine. *Molecular interventions*, 5(3):163, 2005.
- [60] Chris H Takimoto and Emiliano Calvo. Principles of oncologic pharmacotherapy. *Cancer management: a multidisciplinary approach*, 11:1–9, 2008.
- [61] James C Wang. Cellular roles of dna topoisomerases: a molecular perspective. *Nature reviews Molecular cell biology*, 3(6):430–440, 2002.
- [62] KM Tewey, TC Rowe, L Yang, BD Halligan, and LF Liu. Adriamycin-induced dna damage mediated by mammalian dna topoisomerase ii. *Science*, 226(4673):466–468, 1984.
- [63] Tomas Simunek, Martin Sterba, Olga Popelova, Michaela Adamcova, Radomir Hrdina, and Vladimir Gersl. Anthracycline-induced cardiotoxicity: overview of studies examining the roles of oxidative stress and free cellular iron. *Pharmacological Reports*, 61(1):154–171, 2009.
- [64] Barry Halliwell. Free radicals, antioxidants, and human disease: curiosity, cause, or consequence? *The lancet*, 344(8924):721–724, 1994.
- [65] Baoxu Pang, Xiaohang Qiao, Lennert Janssen, Arno Velds, Tom Groothuis, Ron Kerkhoven, Marja Nieuwland, Huib Ovaa, Sven Rottenberg, Olaf Van Tellingen, et al. Drug-induced histone eviction from open chromatin contributes to the chemotherapeutic effects of doxorubicin. *Nature communications*, 4:1908, 2013.
- [66] Thomas Vaissière, Carla Sawan, and Zdenko Herceg. Epigenetic interplay between histone modifications and dna methylation in gene silencing. *Mutation Research/Reviews in Mutation Research*, 659(1):40–48, 2008.
- [67] Mats Ljungman and Philip C Hanawalt. Efficient protection against oxidative dna damage in chromatin. *Molecular carcinogenesis*, 5(4):264–269, 1992.
- [68] Andrew HJ Wang, Yi Gui Gao, Yen Chywan Liaw, and Yu Kun Li. Formaldehyde crosslinks daunorubicin and dna efficiently: Hplc and x-ray diffraction studies. *Biochemistry*, 30(16):3812–3815, 1991.
- [69] Roland T Skeel and Samir N Khleif. *Handbook of cancer chemotherapy*. Lippincott Williams & Wilkins, 2011.

- [70] Mayo Clinic. Chemotherapy and hair loss: what to expect during treatment. <https://www.mayoclinic.org/tests-procedures/chemotherapy/in-depth/hair-loss/art-20046920>, 2016. [Online; accessed 02-January-2018].
- [71] Fan Yang, Sheila S Teves, Christopher J Kemp, and Steven Henikoff. Doxorubicin, dna torsion, and chromatin dynamics. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, 1845(1):84–89, 2014.
- [72] Bruce Alberts. *Molecular biology of the cell*. Garland science, 2017.
- [73] ThermoFisher Scientific. CellEvent Caspase3/7 Green Detection Reagent – ThermoFisher Scientific. <https://www.thermofisher.com/order/catalog/product/C10423>, 2018. [Online; accessed 21-January-2018].
- [74] Takeshi Suzuki, Keiko Fujikura, Tetsuya Higashiyama, and Kuniaki Takata. Dna staining for fluorescence and laser confocal microscopy. *Journal of Histochemistry & Cytochemistry*, 45(1):49–53, 1997.
- [75] ThermoFisher Scientific. Propium Iodide – ThermoFisher Scientific. <https://www.thermofisher.com/order/catalog/product/P1304MP>, 2018. [Online; accessed 21-January-2018].
- [76] GN Wilkinson. Statistical estimations in enzyme kinetics. *Biochemical Journal*, 80(2):324, 1961.
- [77] Michael A Savageau. Biochemical systems analysis: I. some mathematical properties of the rate law for the component enzymatic reactions. *Journal of theoretical biology*, 25(3):365–369, 1969.
- [78] Jane V Harper. Synchronization of cell populations in g1/s and g 2/m phases of the cell cycle. *Cell Cycle Control: Mechanisms and Protocols*, pages 157–166, 2005.
- [79] CJ Bostock, DM Prescott, and JB Kirkpatrick. An evaluation of the double thymidine block for synchronizing mammalian cells at the g1-s border. *Experimental cell research*, 68(1):163–168, 1971.
- [80] Felix Buggenthin. Computational prediction of hematopoietic cell fates using single cell time lapse imaging. 2011.
- [81] The National Digital Information Infrastructure and Preservation Program at the Library of Congress. TIFF/IT for Image Technology. <https://www.loc.gov/preservation/digital/formats/fdd/fdd000072.shtml>, 2016. [Online; accessed 02-January-2018].
- [82] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.
- [83] Nikil Jayant, James Johnston, and Robert Safranek. Signal compression based on models of human perception. *Proceedings of the IEEE*, 81(10):1385–1422, 1993.
- [84] Jae S Lim. Two-dimensional signal and image processing. *Englewood Cliffs, NJ, Prentice Hall*, 1990, 710 p., 1990.

- [85] Gaurav Sharma and H Joel Trussell. Digital color imaging. *IEEE transactions on image processing*, 6(7):901–932, 1997.
- [86] Gary J Sullivan, Jens Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.
- [87] Gunter Wyszecki and Walter Stanley Stiles. *Color science*, volume 8. Wiley New York, 1982.
- [88] D Osorio and M Vorobyev. Photoreceptor spectral sensitivities in terrestrial animals: adaptations for luminance and colour vision. *Proceedings of the Royal Society of London B: Biological Sciences*, 272(1574):1745–1752, 2005.
- [89] Matthew Anderson, Ricardo Motta, Srinivasan Chandrasekar, and Michael Stokes. Proposal for a standard default color space for the internetâ€”srgb. In *Color and imaging conference*, volume 1996, pages 238–245. Society for Imaging Science and Technology, 1996.
- [90] International Telecommunication Union Radiocommunication Sector (ITU-R). BT.601: Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios. <http://www.itu.int/rec/R-REC-BT.601/>, 2007. [Online; accessed 02-January-2018].
- [91] International Telecommunication Union Radiocommunication Sector (ITU-R). BT.709 : Parameter values for the HDTV standards for production and international programme exchange. <http://www.itu.int/rec/R-REC-BT.709-6-201506-I/en/>, 2015. [Online; accessed 06-January-2018].
- [92] Brian W Kernighan, Dennis M Ritchie, Clovis L Tondo, and Scott E Gimpel. *The C programming language*, volume 2. prentice-Hall Englewood Cliffs, 1988.
- [93] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [94] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [95] Michael S Gazzaniga. *The cognitive neurosciences*. MIT press, 2004.
- [96] Michael H.F. Wilkinson and F. Schut. Digital image analysis of microbes. In Michael H.F. Wilkinson and F. Schut, editors, *Digital Image Analysis of Microbes: Imaging, Morphometry, Fluorometry and Motility Techniques and Applications*, chapter 6. John Wiley and Sons, New York, 1998.
- [97] Ben Tupper, Curtis Rueden, Johannes Schindelin, and Mark Hiner. RATS: Robust Automatic Threshold Selection. http://imagej.net/RATS:_Robust_Automatic_Threshold_Selection, 2017. [Online; accessed 17-January-2017].
- [98] Irvin Sobel. An isotropic 3×3 image gradient operator. *Machine vision for three-dimensional scenes*, pages 376–379, 1990.

- [99] Steven W Smith et al. The scientist and engineer’s guide to digital signal processing. 1997.
- [100] Raphael A Finkel and Jon Louis Bentley. Quad trees a data structure for retrieval on composite keys. *Acta informatica*, 4(1):1–9, 1974.
- [101] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library.* ” O’Reilly Media, Inc.”, 2008.
- [102] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.
- [103] Karel Zuiderveld. Contrast limited adaptive histogram equalization. In *Graphics gems IV*, pages 474–485. Academic Press Professional, Inc., 1994.
- [104] Stephen M Pizer, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987.
- [105] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London B: Biological Sciences*, 207(1167):187–217, 1980.
- [106] Jean-Yves Tinevez, Nick Perry, Johannes Schindelin, Genevieve M Hoopes, Gregory D Reynolds, Emmanuel Laplantine, Sebastian Y Bednarek, Spencer L Shorte, and Kevin W Eliceiri. Trackmate: An open and extensible platform for single-particle tracking. *Methods*, 115:80–90, 2017.
- [107] Khuloud Jaqaman, Dinah Loerke, Marcel Mettlen, Hirotaka Kuwata, Sergio Grinstein, Sandra L Schmid, and Gaudenz Danuser. Robust single-particle tracking in live-cell time-lapse sequences. *Nature methods*, 5(8):695–702, 2008.
- [108] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.
- [109] Vladimir Vapnik and Alexey Chervonenkis. A note on one class of perceptrons. *Automation and remote control*, 25(1):103, 1964.
- [110] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [111] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [112] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.
- [113] James Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209:415–446, 1909.

- [114] Mee Young Park and Trevor Hastie. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):659–677, 2007.
- [115] Andrey Nikolaevich Tikhonov, Vasiliy Iakovlevich Arsenin, and Fritz John. *Solutions of ill-posed problems*, volume 14. Winston Washington, DC, 1977.
- [116] Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161, 1997.
- [117] Ludwig Fahrmeir, Christian Heumann, Rita Künstler, Iris Pigeot, and Gerhard Tutz. *Statistik: Der Weg zur Datenanalyse*. Springer-Verlag, 2016.
- [118] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.
- [119] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [120] 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.
- [121] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2015.
- [122] Stéfan 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.
- [123] Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. SciPy Austin, TX, 2010.
- [124] John D Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.
- [125] Samuele Pedroni and Noel Rappin. *Jython Essentials: rapid scripting in Java.* ” O'Reilly Media, Inc.”, 2002.
- [126] 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.
- [127] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

- [128] Oliver Hilsenbeck. Maximally interesting extremal regions (miner) and a probabilistic model for cell tracking. 2014.

Appendices

Appendix A

Tables

A.1 Inferred shift in wells

| Position | δ_x | δ_y | Position | δ_x | δ_y | Position | δ_x | δ_y |
|----------|------------|------------|----------|------------|------------|----------|------------|------------|
| 01 | -2 | -4 | 22 | -7 | -4 | 43 | -7 | -3 |
| 02 | -3 | -4 | 23 | -7 | -4 | 44 | -7 | -3 |
| 03 | -3 | -4 | 24 | -9 | -4 | 45 | -7 | -3 |
| 04 | -3 | -4 | 25 | -9 | -5 | 46 | -7 | -3 |
| 05 | -3 | -3 | 26 | -8 | -5 | 47 | -7 | -2 |
| 06 | -3 | -4 | 27 | -8 | -4 | 48 | -4 | -2 |
| 07 | -3 | -4 | 28 | -8 | -4 | 49 | -4 | -2 |
| 08 | -4 | -3 | 29 | -8 | -4 | 50 | -4 | -2 |
| 09 | -4 | -4 | 30 | -8 | -4 | 51 | -4 | -2 |
| 10 | -4 | -4 | 31 | -8 | -4 | 52 | -4 | -2 |
| 11 | -4 | -4 | 32 | -8 | -3 | 53 | -5 | -2 |
| 12 | -4 | -3 | 33 | -9 | -3 | 54 | -5 | -2 |
| 13 | -4 | -3 | 34 | -8 | -3 | 55 | -4 | -1 |
| 14 | -5 | -3 | 35 | -8 | -3 | 56 | -2 | -2 |
| 15 | -5 | -3 | 36 | -9 | -2 | 57 | -3 | -3 |
| 16 | -6 | -5 | 37 | -9 | -3 | 58 | -3 | -3 |
| 17 | -6 | -5 | 38 | -9 | -2 | 59 | -3 | -2 |
| 18 | -6 | -5 | 39 | -8 | -2 | 60 | -3 | -2 |
| 19 | -6 | -5 | 40 | -7 | -3 | 61 | -3 | -3 |
| 20 | -6 | -4 | 41 | -7 | -3 | 62 | -3 | -2 |
| 21 | -7 | -5 | 42 | -7 | -3 | 63 | -4 | -2 |

Table A.1: The inferred shift (in pixels) for each image position in unsynchronized experiment. The shifts are inferred using the shift correction algorithm explained in Subsection 2.2.10.

Appendix B

Experiment Protocols

B.1 Images acquisition

B.2 Placeholder