HELMHOLTZ
ASSOCIATION
Alliance on Systems Biology

HelmholtzZentrum münchen
German Research Center for Environmental Health

TECHNISCHE
UNIVERSITÄT
MÜNCHEN

Computational prediction of cellular states in time-lapse microscopy based on single-cell morphodynamics

---

Felix Buggenthin

November 2015

TECHNISCHE UNIVERSITÄT MÜNCHEN

Wissenschaftszentrum Weihenstephan für Ernährung, Landnutzung und Umwelt

Lehrstuhl M12 (Mathematische Modellierung biologischer Systeme)

Computational prediction of cellular states in time-lapse microscopy based on single-cell morphodynamics

Felix Buggenthin

Vollständiger Abdruck der von der Fakultät Wissenschaftszentrum Weihenstephan für Ernährung, Landnutzung und Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

genehmigten Dissertation.

Vorsitzender:

Univ.-Prof. Dr. D. Frischmann

Prüfer der Dissertation:

1. Univ.-Prof. Dr. F. J. Theis
2. Univ.-Prof. Dr. N. Navab

Die Dissertation wurde am 25.11.2015 bei der Technischen Universität München eingereicht und durch die Fakultät Wissenschaftszentrum Weihenstephan für Ernährung, Landnutzung und Umwelt am 26.07.2016 angenommen.

Danksagungen

Die Fertigstellung dieser Arbeit war nur durch die Unterstützung einiger Personen möglich, bei denen ich mich hiermit herzlich bedanken möchte.

Danke Carsten, für deine großartige Betreuung während meiner gesamten Zeit als Doktorand. Danke auch für deine Ruhe, Hilfsbereitschaft und Erreichbarkeit, ob am Wochenende oder am späten Abend.

Danke Fabian, für die Möglichkeit in den letzten sechs Jahren ein Teil vieler spannender Projekte zu sein und dabei viel von dir und anderen zu lernen.

Danke Timm, für die intensive Zusammenarbeit besonders in meinem Hauptprojekt. Danke auch an die Mitglieder deines Labors, besonders Philipp.

Danke Jörg, Michi Str., Michi Schw. und alle Kollegen des ICB für die vielen schönen und lustigen Momente sowie die gute Arbeitsatmosphäre. Ihr habt dafür gesorgt dass es immer Spaß gemacht hat. Vielen Dank weiterhin Sebastian und Henrik für die anregenden Diskussionen mit euch und die daraus resultierenden Lösungsansätze bei vielen Problemen. Ein weiteres dankeschön geht an Prof. Nassir Navab und Prof. Dimitrij Frishman für die Mitgliedschaft in meiner Prüfungskommission.

Vielen Dank Stephie! Danke für deine Geduld, deine immer wieder aufbauenden Worte und auch unsere fachlichen Diskussionen. Zusammen mit dir waren selbst die stressigsten Phasen der letzten vier Jahre zu bewältigen. In diesem Zuge auch vielen Dank Wolfgang und Helga für eure Unterstützung in den letzten Jahren.

Abschließend ein ganz besonderes Dankeschön an meine Eltern, Ute und Arno und meine Großeltern. Nur durch eure umfassende Unterstützung über meine gesamte Ausbildung und in jeder Lebenslage habt ihr diese Dissertation möglich gemacht.

Abstract

Studying the behavior of single cells with time-lapse microscopy allows to detect heterogeneities that would be masked in conventional population-based experiments. Recently, several studies have shown that morphology and motility can be used as a readout for a cells present or future state in such experiments. However, to quantify and analyze the information in the large amounts of images that are generated in a typical time-lapse experiment, customized computational methods from image processing and machine learning are required.

In this thesis, we developed processing pipelines that identify cells in the brightfield or fluorescence channel of a time-lapse experiment and quantify their morphology. By linking these measurements with the temporal information of automatic or manual tracking approaches, we were able to describe the morphodynamics and motility of single cells but also whole genealogies. In particular we contributed to two biological applications. First, we analyzed the migration behavior of T-lymphocytes under changing environmental influences. Applying a variational Bayesian algorithm that fitted a Gaussian mixture model to the derived cell speeds, we could identify groups of motile and non-motile cells. Our model allowed to exactly determine the mean movement as well as the variance of every group. Also, a regression model of the total morphology space was able to predict a cells future speed. Second, we assessed whether cell morphology and motility can be used as a predictor of hematopoietic lineage choice. To this end, we quantified differentiating hematopoietic stem and progenitor cells in long-term time-lapse microscopy. Following, we developed an artificial neural network that correctly predicted the commitment of a hematopoietic stem and progenitor cell to the megakaryocytic/erythroid or mono-/granulocytic lineage up to three generations before conventional molecular markers were detected.

If complemented with single cell sequencing techniques, our methods could be used to identify novel molecular factors that are involved in hematopoietic differentiation. In the future, our methods could be transferred to other lineage decisions and cell types, which would allow the detailed analysis of cellular features over entire lifetimes and even genealogies and thus, may contribute to the development of novel diagnostic tools for e.g. leukemia or immune system deficiencies.

Zeitraffermikroskopie ermöglicht die Untersuchung von Verhaltensmustern einzelner Zellen in einer Population, welche in konventionellen Populationsstudien nicht detektierbar wären. Eine Reihe von Puplikationen zeigte kürzlich, dass Morphologie und Bewegung der Zellen in solchen Experimenten als Indikatoren für deren momentanen aber auch zukünftigen Status verwendet werden können. Um die große Menge an Bildern aus der Zeitraffermikroskopie automatisch quantifizieren und analysieren zu können, werden computergestützte Methoden der Bildprozessierung und des maschinellen Lernens benötigt, die an die jeweilige Anwendung angepasst sind.

In der vorliegenden Arbeit haben wir Prozessierungspipelines entwickelt, die einzelne Zellen im Durchlicht- oder Fluoreszenzkanal von Zeitrafferfilmen mit hoher Robustheit erkennen und deren Morphologie quantifizieren. Die Verknüpfung dieser Messungen mit automatischen und manuellen Zellverfolgungsmethoden ermöglichte uns, die Morphodynamik und Bewegung von Einzelzellen über mehrere Generationen zu beschreiben. So analysierten wir konkret das Migrationsverhalten von T-Lymphozyten unter unterschiedlichen Umwelteinflüssen. Hier konnten wir durch einen variationellen Bayesianischen Algorithmus ein Gausssches Mischmodell an die gemessenen Zellgeschwindigkeiten anpassen, woraufhin wir Gruppen migrierender und nicht-migrierender Zellen identifizieren konnten. Zusätzlich ermöglichte unser Modell die genaue Bestimmung der mittleren Geschwindigkeit beider Gruppen, sowie deren Varianz. Wir konnten keine Korrelation der Zellbewegung mit einer bestimmten Morphologie nachweisen, jedoch war ein von uns auf den vollständigen Morphologieraum trainiertes Regressionsmodell in der Lage, die zukünftige Geschwindigkeit einer Zelle vorherzusagen. Zum anderen untersuchten wir, ob Zellmorphologie und -bewegung verwendet werden können, um hämatopoetische Linienentscheidungen während der Blutbildung vorherzusagen. Hierfür quantifizierten wir differenzierende hämatopoetische Stamm- und Vorläuferzellen in Langzeit-Zeitraffermikroskopie. Basierend auf diesem Datensatz entwickelten wir ein künstliches neuronales Netz, welches bis zu drei Generationen vor der Nachweisbarkeit von molekularen Markern vorhersagt, ob eine hematopoietische Stamm- oder Vorläuferzelle zur megakaryozytischen/erythroiden oder mono-/granulozytischen Linie ausdifferenziert.

Die Verbindung unserer Methodik mit Techniken zur Einzelzellsequenzierung könnte verwendet werden, um bisher unbekannte molekulare Faktoren zu identifizieren welche die hämatopoetische Linienentscheidung beeinflussen. Unsere Methoden können in Zukunft verallgemeinert und für die Analyse weiterer Linienentscheidungen oder Zelltypen angepasst werden, was die detaillierte Analyse zellulärer Eigenschaften über volle Lebenszyklen und sogar mehrere Generationen ermöglicht. Hierdurch könnte zum Beispiel ein Beitrag zu der Entwicklung neuartiger Methoden in der Erforschung und Behandlung von Leukämie und anderen Erkrankungen des Immunsystems geleistet werden.

Contents

Chapter 1

Introduction

The study of biological cells is a century-old discipline that is crucial for a comprehensive understanding of living organisms. It originated from the invention of the first light microscope by Zacharias Janssen in 1590. This instrument allowed to examine the structural makeup of plants and animals and led to the foundation of cell biology as a discipline in 1838 by Matthias Jacob Schleiden, who realized that plants consist of single cells. In 1839 Theodor Schwann found this theory to be also true for animals. Nowadays, the cell is defined as the smallest living entity of an organism, where one differentiates between unicellular and multicellular species. In the latter case, different cell types in an organism are connected to functional units that serve a specialized purpose, the so-called tissue. For example, the human body consists of several hundreds of different tissues and cells, that form a complex network of interactions contributing to the body's vital functions. The identification of new cell types, their role in the organism and especially their interaction with each other is crucial to better understand multifactorial diseases such as cancer (Singh et al., 2004) and to develop specific clinical treatments (Chattopadhyay et al., 2014). Yet, research in cell biology is a great challenge due to its high complexity and nowadays requires the involvement of different disciplines of the life sciences. In the following sections we will introduce these subfields and highlight some of the key findings in recent years.

1.1 Biological processes at different levels of resolution

The function of a cell is in large parts determined by its molecular makeup (Alberts et al., 2007). Every cell bears a full copy of DNA, a complex molecule that contains all the information that is necessary to build an organism. Its structure was first described byWatson and Crick (1953) and led to the formulation of the central dogma of biology (Crick, 1970). It states that the information on the double-stranded DNA is transcribed into single-stranded RNA, which in turn is translated to proteins. The latter are large biological molecules that consist of one or more chains of amino acids. They perform a multitude of different functions such as catalyzation of metabolic reactions, the replication and repair of the DNA or the formation of canals and pumps that connect different compartments of a cell. DNA, RNA and proteins build a huge and very complex network with a vast array of regulatory pathways. The exhaustive study of all the interactions recently became possible through the complete description of an organism's DNA, the genome. The most important contribution in this context was the full sequencing of the human genome in 2001 (Venter et al., 2001; Lander et al., 2001). It opened up the possibility to analyze the DNA in its entirety, thereby deciphering for example which parts are coding for functional proteins and how many types of different proteins exist (Bernstein et al., 2012). Yet, it has also been found that there are several other intermediate layers of regulation that add to the complexity of an organisms regulatory network, such as miRNAs (Ruvkun, 2001) or alternative splicing (Chow et al., 1977). The challenge still is to assign all functions and regulatory interactions to all the different molecular players, for which a wide variety of methods have been developed.

To study the function of a gene or protein, one approach is to measure its abundance in a specific system or process. A classical and still extensively used technique is western blotting (Towbin et al., 1979; Burnette, 1981). A population of cells is lysed and the lysate is separated by gel-electrophoresis. The proteins of interest are then detected by staining them with antibodies, while the rough quantification of absolute abundance is also possible. Yet, as typically $\sim 10^5$ cells are required to conduct the method and a high amount of manual labor is involved in the protocol, the applicaction of western blotting is limited to small-scale analyses. A method to analyze the expression of thousands of proteins at the same time is the Protein-microarray (Chattopadhyay et al., 2014). Here, antibodies that are known to bind specific proteins are placed on a glas slide in a small-scaled grid. A mixture of proteins is then extracted from a population of cells and put on the slide, which are then bound by their specific antibody. After washing all unbound proteins from the slide, another set of antibodies is added that are labeled by fluorescence dyes. Eventually, the expression of all proteins that have been bound by antibodies can be identified by scanning the fluorescence over the grid coordinates. Typically, microarrays have been used to measure gene expression from mRNA transcripts by reversely transcribing them as cDNA and using primers on the glas slide instead of antibodies (Schena et al., 1995).

To apply the methods discussed above, a large number of cells is required for a single experiment, and the measured protein expression only reflects an average over all cells in the population. Recently however, there is increasing evidence that measuring the population average leads to biased or even incorrect results, as most single cells behave heterogeneously even if they belong to the same cell type (Anselmetti, 2009). In 2002, the Swain lab showed that clonal single cells from *E. coli* bacteria exhibit a strongly heterogeneous expression of a single gene, an effect that is explainable by stochastic noise (Elowitz et al., 2002). In addition, Weinberger et al. (2005) found that protein translation underlies a high level of variation between cells that also has effects on higher levels of regulation. Thus, measuring the expression of genes and proteins in single cells allows to derive population statistics and correctly account for heterogeneity. This approach thereby provides a more precise description of the analyzed system.

Plenty of different techniques have been developed over the past decades to quantify molecular processes on the single-cell level (Hoppe et al., 2014). A technique that finds broad application is flow cytometry, where single cells are analyzed by the detection of fluorescently stained proteins (Dittrich and Göhde, 1969). The method operates by channeling tens of thousands of cells through a very thin water stream separately and non-invasively. The cells are passing a set of lasers, which can excite the fluorophores in the cell. Depending on the application, flow cytometry can be used to sort cells in different categories (fluorescent activated cell sorting, FACS), or to analyze gene expression on the single-cell level (Herzenberg and Sweet, 1976). For example, FACS was successfully used to analyze stem cells with heterogeneous expression levels of the pluripotency factor Nanog (Chambers et al., 2007). The study completely altered the view of the role of Nanog during embryogenesis, an insight that would not have been possible by population studies.

Due to its dependence on the spectrum of visible light, the number of fluorescent dyes that can be analyzed in parallel via state-of-the-art flow cytometers is restricted to 18 channels (Perfetto et al., 2004). Also, as the cells of interest are labeled by antibodies, FACS is restricted to cell types for which the necessary surface proteins have been described. A more involved technique to measure transcriptional abundance is single-cell real time quantitative polymerase chain reaction (RT-qPCR), where the mRNA of up to 96 selected genes of a single cell is reversely transcribed into cDNA and then amplified until sequencing becomes possible (Bustin, 2000). The method is able to report absolute numbers of mRNA transcripts, which is of great importance to derive bio-mathematical models. Yet the cell is destroyed during the process, hindering a subsequent analysis of the same cell at a later time point. Recent publications that involved single-cell RT-qPCR are e.g. Guo et al. (2010), who analyzed the differential expression of 42 genes in cells of the three germ layers during early development of the mouse blastocyst. More recently, Moignard et al. (2015) deciphered parts of the transcriptional programs that are involved in organogenesis by analyzing 40 genes in more than 3000 single cells with blood-forming potential.

Flow cytometry and single-cell RT-qPCR are applicable if the players that are involved in the studied system are known a priori and the set of genes that is to be analyzed can be narrowed down to a few candidates. In order to analyze the full transcriptome of a single cell to identify unknown genes or non-coding RNAs, single-cell RNA sequencing (scRNA-seq) can be used (Mortazavi et al., 2008). The method became feasible with the advent of next generation sequencing, a group of techniques that improved throughput and precision of DNA squencing, accompanied by a great reduction of costs (Wu et al., 2014). The protocol involves the isolation of a single cell, for example by FACS. The RNA is then extracted and reversely transcribed into cDNA. The cDNA is amplified via PCR and a sequencing library is generated, which can then be sequenced by a next-generation sequencing technique. RNA-seq of single cells has been used to describe the transcriptional differences of coding and non-coding RNAs on a genome-wide scale (Shalek et al., 2013; Saliba et al., 2014). In addition, it was found that splicing patterns (Shalek et al., 2013) and allelic random expression (Deng et al., 2014) are widely variable between cells.

It is worth noting that most of the mentioned studies in this section would not have been feasible without the support of computational methods during data postprocessing and analysis. All high-throughput techniques generate huge datasets with high dimensionality (with regard to the number of cells as well as the number of measured dimensions) that have to be processed with appropriate methods to reveal the information of interest. Beside the algorithms that reconstruct the genome from the short reads of next-generation sequencing (Li and Homer, 2010), several algorithms and tools for data assessment have been developed. This includes for example t-SNE that performs a nonlinear dimension reduction, thus revealing structures and patterns in the data that were hidden before (van der Maaten and Hinton, 2008). Along this line, Buettner et al. (2015) proposed scLVM, a method that accounts for the heterogeneity in gene expression of single cells. Another method called diffusion maps added the possibility to assign a pseudotemporal ordering to samples in high-dimensional datasets (Haghverdi et al., 2015).

1.2 Time-lapse microscopy allows to study spatio-temporal dynamics of single cells

The aforementioned methods are useful to study the molecular makeup of single cells at a single time point. However, many biological systems show spatial and temporal variations that cannot be uncovered by analyzing snapshot data alone (see Figure 1.1). A well established method to study the spatio-temporal dynamics of single cells in vitro is time-lapse microscopy (Muzzey and van Oudenaarden, 2009). The technique is based on Étienne-Jules Marey's work in 1891, who analyzed a moving object by taking photographs at evenly spaced temporal intervals (Landecker, 2006). In 1910, Jean Comandon used time-lapse experiments to study the motility of syphilis bacteria under the "ultramicroscope", the only machine at that time that was able to visualize this type of bacteria (Landecker, 2005). The observation of cells from more complex eukaryotic organisms was made possible by the invention of cell culturing techniques by Ross Granville Harrison in 1910. Around the same time, August Koehler and Henry Siedentopf found a way to visualize distinct parts of single cells like the nucleus or cytoplasm by staining them with fluorochromes. The staining of subcellular structures in living cells such as specific proteins became possible by the discovery of the green fluorescent protein, which was first described by Osamu Shomura in 1962 and made usable in molecular biology by Martin Chalfie and Roger Tsien around 1994 (Chalfie et al., 2007). Nowadays, a multitude of fluorescent dyes that span the full spectrum of visible light allow the staining of several genes of interest at the same time (Lichtman and Conchello, 2005). Powerful time-lapse microscopes are commercially available that continuously image hundreds of thousands of single cells and allow to study a broad range of biological processes. This includes cell-cycle dependent drug response (Tung et al., 2011) or the change of morphology after whole-genome knockout experiments (Neumann et al., 2006). In the last decade, the invention of high-throughput long-term time-lapse microscopy made it possible to follow single cells over periods of more than two weeks, a time span that is necessary to observe for example the differentiation of stem cells into mature cells (Schroeder, 2008).

A typical time-lapse microscope comprises several modules (Khodjakov and Rieder, 2006). The cells are illuminated by a system consisting of a light source (i.e. a mercury lamp or LEDs) together with a set of spectral filters that allow to excite fluorophores of different colors. The light passes through the growth chamber, i.e. a plastic or glass slide with several wells or channels, that are filled with culturing medium in which the cells are resolved. The light beam then passes through the objective which is magnifying the image to make the cells visible (typical magnifications are 10x, 20x or 63x). Due to the limited

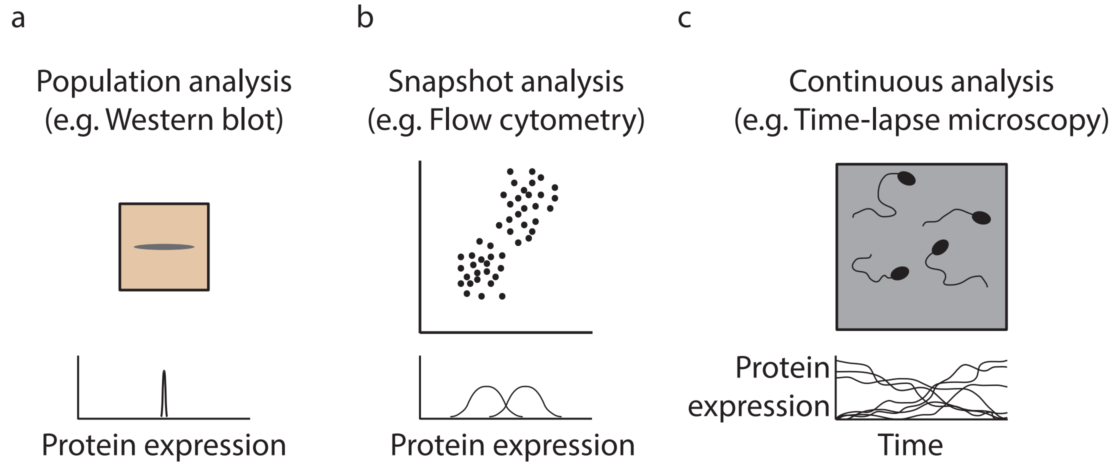


Figure 1.1: Only the analysis of time-resolved expression profiles can reveal temporal dependencies in a biological system. (a) Western blotting only allows to measure mean expression levels of a cell population and involves high amounts of manual labor. (b) If the same protein is quantified at the single-cell level e.g. by flow cytometry or single-cell RT-qPCR, an intricate pattern can be revealed by correlations between two cell types. Yet, the temporal dynamics remain unknown. (b) By continuous single-cell analysis such as time-lapse microscopy the protein expression dynamics can be observed to e.g. figure out whether a bimodal distribution in expression arises from two stable subpopulations of cells or whether all cells start at an intermediate expression level and then increase or decrease expression. Image adapted from Hoppe et al. (2014)

field of view of most microscopes, a single well cannot be covered by a single image. Thus, the growth chamber is moved beneath the objective by a motorized plate to image a grid of overlapping tiles consecutively. Eventually, the image is recorded by a digital camera and stored on the computer. See Figure 1.3 for a general overview.

Depending on the application and experimental setup, modern light microscopes generally produce brightfield and fluorescence images. While the acquisition of brightfield images is less detrimental to cell health, they provide a limited flexibility as only the full cell body can be imaged. In addition, this type of images suffers from a low signal to noise ratio and a bad contrast, which renders an automated analysis very difficult. Fluorescence images on the other hand allow the user to observe only the parts of the cell that are of interest by staining them with specific fluorophores, e.g. the nucleus or a certain protein (see Figure 1.2). They feature a much higher signal to noise ratio and higher contrast than brightfield images, which is of great help for the analysis of detailed cell structures. Also, the application of automated computational quantification pipelines is less challenging on

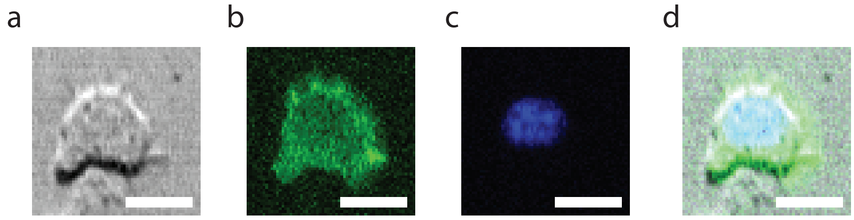


Figure 1.2: Staining of cells with fluorophores, shown for a T-lymphocyte in 20x resolution. (a) Brightfield image of the cell without any staining. (b) Staining of the actin network with LiefAct-GFP. (c) Staining of the nucleus with Hoechst405. (d) Merged image showing all color channels together. Scale bars correspond to 10 µm.

this type of images (see Section 1.4 for a detailed discussion).

There is no single optimal setup for time-lapse microscopy that performs best for every application at hand (Schroeder, 2011). Tuning the multitude of parameter settings that are involved in time-lapse imaging requires a thorough understanding of optics, physics as well as molecular and cell biology, and is always a trade-off between cell health and image quality. For example, a single well of a growth chamber with a coated surface and high volumes of medium might be beneficial for cell health. Yet, the coating and the high medium level can hinder a proper illumination of the sample, thus leading to noisy images that are not suitable for later analysis. Another decision has to be made for light exposure times during image acquisition. While high exposures may lead to clear fluorescence signals that are easily analyzable, the repeated illumination in a time-lapse experiment might lead to increased cell death due to phototoxicity. Also, an objective that provides higher magnification might be helpful to discover specific details, but the limited field of view of the camera then requires to capture several images as a tiled grid in order to cover the full specimen (or well). Last but not least, every part of the microscope has to be of high quality to minimize the probability of one module failing during experiment conduction.

Despite these challenges, time-lapse microscopy is used extensively to gain novel insights into biological processes that were previously only studied in snapshot data. For example, the noise in the timing of meiotic stages in budding yeast was quantified (Nachman et al., 2007). Furthermore, the Elowitz lab conducted time-lapse experiments with sub-cellular resolution and monitored the activation status of the calcium-response reporter Crz1 by localizing the expression of the tagged GFP molecule (Cai et al., 2008). There is also increased effort to study the dynamics of molecules inside single cells. For example the

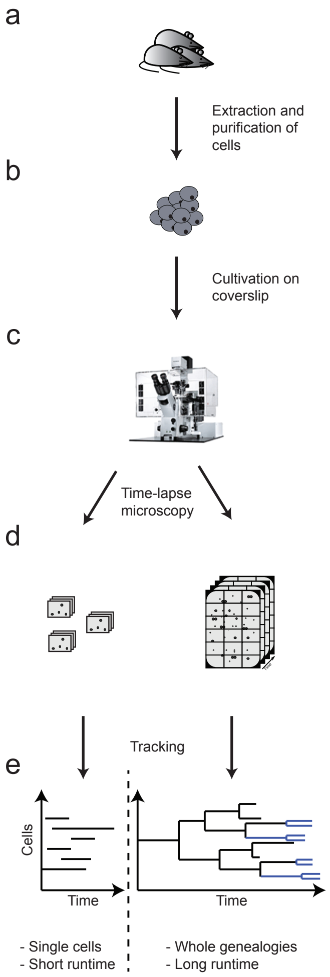


Figure 1.3: Time-lapse microscopy protocol. (a) A population of cells is extracted from the host or a cell line. (b,c) Cells of interest (e.g. blood stem cells or immune cells) are purified by fluorescence activated cell sorting (FACS) and cultivated on a plastic slide. (d) Cells are imaged in short intervals (60-120 seconds are typical) for 1 hour to 8 days. Depending on the application and magnification, the area to be observed can be covered by a single image (single tile of the depicted rectangle), or a grid of several images (all tiles of the depicted rectangle). (e) After the experiment is stopped, single cells or whole genealogies can be followed over time by automated or manual tracking (see Sections 1.4, 2.5.6 and 2.5.2).

Dinger lab developed methods to analyse mRNA localization in neurons (Buxbaum et al., 2015) or the nuclear accessibility of ß-actin via real-time tracking (Smith et al., 2015a). In the blood context, high-throughput long-term microscopy was used to show the existence of the hemogenic endothelium (Eilken et al., 2009) and to proof the instructive potential of cytokines on hematopoietic lineage choice (Rieger et al., 2009) in murine cells.

The insights that were gained by studying the dynamics of single cells on the molecular level also renewed the question to which extent a cell's morphometry is contributing to its function. In the last years several groups have extended their efforts to elucidate the contribution of a cell's shape to its molecular state, also with the help of time-lapse microscopy.

1.3 Cell morphology and motility as a readout of molecular states

The alteration of a cell's shape is a complex process that is regulated by a network of intracellular mechanisms, but also extracellular stimuli (Paluch and Heisenberg, 2009). It was already used as a readout by Rámon y Cajal (1960) to define the function of different cell and tissue types in classical light microscopy studies. The ability of cells to adopt specific shapes is important in a wide range of biological processes. For example, during embryogenesis the cells have to align in different conformations to form various tissues (Yin et al., 2014). Immune cells such as lymphocytes need to elongate their cell body in order to migrate from blood vessels to inflammatory sites (Girard et al., 2012). With the advent of fluorescence microscopy and appropriate markers it became possible to visualize distinct compartments of the cell like the nucleus, the cytoskeleton, or membranes. More recently, processes like spindle formation during mitosis (Held et al., 2010) or the concentration of actin, a protein that is a crucial part of cytoskeleton formation, can be visualized (Riedl et al., 2008). Also, the analysis of cell morphology finds application in cancer diagnostics, as mutated cells often have unnatural shapes and are thus identifiable in histology images (Diamond et al., 1982).

At a first glance, a researcher that visually assesses the morphological phenotypes a cell can adopt under the microscope would describe hundreds of different conformations. However, several studies have shown that the variability can be reduced to a small number of clearly defined classes (Yin et al., 2014). For example the Bakal lab analyzed cell lines from *D. melanogaster* as well as humans and found that generally two to seven classes of distinct cell shapes exist (see Fig. 1.4). Also, Keren et al. (2008) showed that migrating fish keratocytes adopt a small number of different shapes. These findings support the theory that through evolution some genes like actin and tubulin were involved in the emergence of a defined number of advantageous shapes that was most likely driven by biophysical constraints (Yin et al., 2014). In addition, it was found in RNA interference experiments that certain cell populations showed an increase of shape classes that were less abundant in wild type cells.

Many cell types alter their shape dynamically and with regard to their actual program. For example, proliferating hematopoietic progenitor cells stop their locomotion and become perfectly round during mitosis, while they show an elongated drop-like shape and high motility otherwise. Also, fish keratocytes switch their shapes during different migration programs (Keren et al., 2008). In 2010, the Ellenberg lab conducted a genome-scale

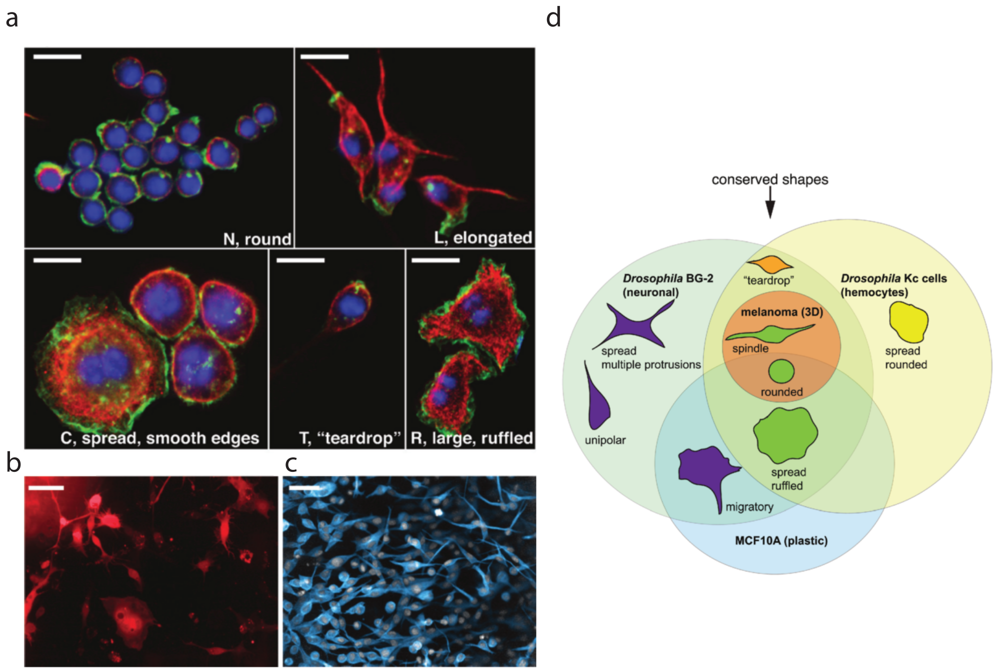


Figure 1.4: Morphological complexity in different cell lines. a) Shapes adopted by wild-type *D. melanogaster* hemocytes. Five shape classes (round, elongated, spread, teardrop, ruffled) could be identified. Scale bars represent 20 µm. b) *D. melanogaster* BG-2 neuronal cells showed six distinguishable shape classes as described in Sailem et al. (2014). Scale bar represents 20 µm. c) WM266.4 melanoma cells adopt only two shape classes, round and elongated. Scale bar represents 50 µm. d) Different cell types can adopt similar shapes. Although the shape space explored by different cell types is diverse, some shapes, such as the rounded or large/flattened shape, are routinely observed. These findings led to the hypothesis that certain cell shapes are, similarly to genes or proteins, conserved throughout evolution. Figure taken with permission from Yin et al. (2014).

high-throughput RNAi screening study of living human cells using time-lapse imaging that allowed the authors to assign novel functions to many genes, such as cell division, migration and survival, by classifying their shape (Neumann et al., 2006, 2010). Recently, Sero et al. (2015) demonstrated in breast epithelial and tumor cells that a cell's shape can also determine its molecular state. The authors found that among other environmental and morphological factors, cell-cell contact, cell area and protrusiveness had an impact on the localization of NF-κB inside the cell. They hypothesized that mechanical and environmental factors can influence molecular signaling pathways and that this finding could partially explain the heterogenous reactions of cells under same chemical conditions.

The discussed findings show that the study of cell morphology in the context of a cell's molecular state becomes more and more important for a full understanding of behavior. Yet, the multitude of shapes that cells can exhibit in time-lapse experiments requires the development of measures that allow to derive meaningful statistics to study the shape space. Most of the discussed examples in this section - morphological analyses as well as molecular studies - have made extensive use of computerized analysis pipelines and quantifications to analyze the huge amounts of data, which we will discuss in the next section.

1.4 Bioimage informatic processing and computational modeling

One bottleneck in the analysis of high-throughput microscopy data is the availability of suitable automatic processing tools. Only with the aid of computational processing it becomes possible to make the huge amount of information that is hidden in the data accessible (Schroeder, 2008). Consider an experimental setup that features a growth chamber that is divided into 80 tiles, where every tile is imaged in intervals of 60 seconds in a single color channel over 10 days. The resulting dataset would consist of over one million images that occupy over one terabyte of hard drive space (assumed a single image saved in tiff-format is around 1 megabyte). Depending of the observed cell type and the amount of cells that were plated out at the beginning of the experiment, the number of single-cell measurements can easily exceed 300 million datapoints (Buggenthin et al., 2013). A trained expert would spend already one working day to postprocess and analyze only a single time point of the experiment, e.g. correcting image illumination, subtracting the background, identifying single cells and quantifying the cellwise fluorescence intensity or shape information (assumed this quantification procedure of a single image takes 6 minutes and the scientist is working 8 hours a day). Also, the manual assessment will most likely introduce a bias into the results (Held et al., 2010). In addition, the high level of customization of experimental setups leads to a great variability of images, even in similar experiments. All these challenges render a proper manual analysis of the acquired data extremely difficult or - in some cases - impossible. Computerized quantification pipelines on the other hand generate reproducible results and can run 24/7 (Swedlow et al., 2009). The same task can be executed in parallel on several machines, which reduces the amount of time that is needed to analyze a full high-throughput experiment.

To automatically process, quantify and analyze the huge amounts of biological imaging

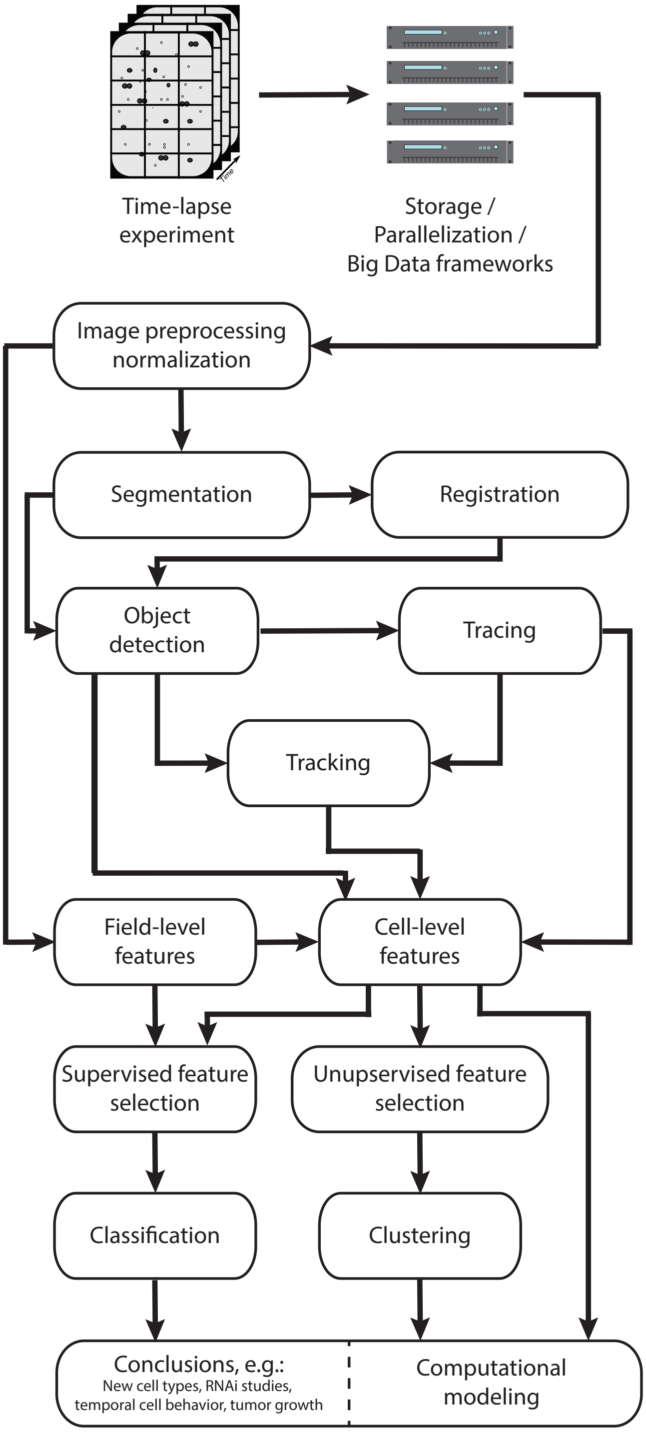


Figure 1.5: Overview of the tasks that are part of bioimage informatics. The discipline covers the development of automated methods from the acquisition of a high-throughput time-lapse experiment, starting from adequate data storage to the full processing, quantification and eventual analysis of all images. Figure adapted from Shariff et al. (2010)

data that are constantly generated, not only for time-lapse microscopy but for all high-throughput imaging data in biology, the discipline of bioimage informatics has recently emerged (Peng, 2008). Residing at the intersection of medical informatics and bioinformatics, it strives to provide methods to derive meaningful information from unstructured image data, but also to conduct statistical analyses that may lead to new biological insights. Contrary to its parent disciplines, bioimage informatics has to cope with the challenges that the high customization in bioimaging brings up. While in classical bioinformatics most methods generate readouts that are structured and stored in a few standardized data formats (e.g. FASTA for DNA sequences or PDB for protein structures), a multitude of different imaging methods exist that are tailored to their own proprietary datasets (Swedlow et al., 2009). In addition, changing one module in a microscope installation can lead to completely different readouts, which renders the development of versatile methods extremely challenging. As a consequence, most algorithms and frameworks in bioimage informatics are to date heavily adapted to their particular application (Murphy, 2014).

A typical analysis pipeline in bioimage informatics consists of several steps (see Figure 1.5. For details about the discussed algorithms and concepts, see Chapter 2). First, all images of an experiment have to be normalized to a level that allows processing of the full dataset without continuous parameter adjustments. This can involve illumination correction methods or noise reduction. Next, the regions of interest like single cells or colonies have to be identified in the image. This is carried out by e.g. segmentation algorithms that group every pixel in the image into foreground and background. The identified regions may then be quantified by different measures that describe shape or texture. If the temporal dynamics have to be be quantified, an automated or semi-automated tracking algorithm can be applied to connect single cells or colonies in adjacent frames of a time-lapse experiment. On the extracted features, basic statistics such as histogram or correlation analyses can be derived. Often more sophisticated machine-learning algorithms are applied to leverage the high-dimensional data, or the data is used to define theoretical models.

Over the last decade, several frameworks and programs for bioimage informatic analyses have been developed that try to tackle non-standardized experimental setups and the problems that arise thereof. Fiji and Icy are powerful java-based image viewers that allow to interactively filter and normalize images from biological experiments (Schindelin et al., 2012; de Chaumont et al., 2012). Especially in Fiji plenty of state-of-the-art methods are available as plugins, also it features a fully featured scripting language. Yet, its dependence on a graphical user interface makes it of limited use to process large amounts of time-lapse data. Examples for applications of Fiji are the reconstruction of huge three-dimensional image stacks (often larger than one terabyte) from light sheet microscopy (Preibisch et al., 2014), or the identification of neuronal lineages in *D. melanogaster* by axon tracing (Cardona et al., 2010).

CellProfiler was built for the easy assembly of parallelizable processing pipelines that can also derive statistical analyses (Carpenter et al., 2006). However, more advanced or very recent algorithms are not covered by the available modules and it provides limited functionality to implement self-written code for the analysis. A success story of CellProfiler is for example the identification of therapeutic targets within human macrophages to stop Ebola virus infections (Sakurai et al., 2015). Also, CellProfiler was used to create a fully automatic quantification toolbox to classify the morphological phenotypes of *C.elegans* worms in brightfield time-lapse microscopy (Wählby et al., 2012).

On the commercial side, Bitplane Imaris is a fully featured software suite that has a semi-automated tracking module and interfaces to include self-written scripts in an analysis pipeline, but is lacking full parallelization and server-based batch processing capabilities. It was for example used in a study to determine a functional hematopoietic stem cell niche via real-time imaging (Xie et al., 2009). A full high-content screening platform named Operetta is available from PerkinElmer. It allows to observe large amounts of cells over time, yet the analysis is limited to the capabilities of the machine, which is for example not able to conduct high-throughput long-term time-lapse imaging. Operetta was for example used to conduct a high-throughput screening study of drugs to decrease the growth of breast cancer tumors in a 3-dimensional environment (Lovitt et al., 2015).

In summary, to analyze modern bioimaging experiments, the involvement of bioimage informatic analyses is nearly always necessary (Myers, 2012). The unbiased quantification of features for the objects of interest allows to conduct robust statistics and greatly increases reproducibility by reducing the amount of time that is needed for a full analysis to a reasonable amount. Several platforms exist that allow to conduct high-throughput experiments, yet every method has their strengths and weakenesses and is thus not always applicable. The importance of bioimage informatics is also reflected by the amounts of papers that were published with this term. In the year 2008, a search with google scholar returned 203 hits, while this number doubled to 467 for a search for publications in the year 2014. However this number does not reflect the numerous specialized analysis pipelines and programs that are published in the supplement of papers describing the biological findings.

Despite the direct biological conclusions that can be drawn from the data that is derived by bioimage informatics, the typically highly quantitative datasets are also used in image-based systems biology and computational modeling. The discipline of systems biology tries to understand an organism in its entirety (Berger et al., 2013). The goal is to generate an integrated view of all regulatory processes over all levels of abstraction, from the interactions of all genes over proteins to organelles and eventually to organismic behavior and biomechanics (Wolkenhauer and Klingmueller, 2004). High-throughput imaging is a way to include the complexity layer of single cells into systems biology. For example, in RNA-mediated interference (RNAi) experiments, libraries of knock-out cell-lines can be generated to conduct genome-wide screens that identify all possible regulators of a general biological process (Sharma and Rao, 2009). The resulting datasets often comprise microscopic images of several hundred wells with different knockout lines that need to be automatically analyzed via bioimage informatics. For example, Knapp et al. (2011) proposed a method to include the population context of a single cell into the analysis of RNAi-screens by identifying the spatial position of every cell in an image and calculating cell densities. The authors thereby improved the performance in measuring virus infection effectivity.

Another way to make use of the wealth of image-based data is to build *in silico* mathematical models and simulations that help to better understand the spatiotemporal dynamics and interactions of single cells (Sbalzarini, 2013). A model can be derived in two different directions. The “bottom-up” approach reproduces cell behavior (like shape changes or movement) from an underlying molecular regulatory network by measuring e.g. protein abundance. This approach was used by Zinzen et al. (2006), who measured the gene expression of transcription factors involved in the development of *D. melanogaster* embryos and defined a model that allowed to analyze the key rate-limiting factor that is involved in localized patterns of gene expression that can be observed in fluorescence microscopy. In contrast, the “top-down” approach seeks to infer a regulatory process by observing cell behavior. For example, Keren et al. (2008) derived a model of the actin network from the shape and speed of fish keratocytes. Once a model is defined and evaluated, it can be used to e.g. test new hypotheses (Chickarmane et al., 2010) or identify previously unknown interactions in regulatory networks (Nakae et al., 2014).

1.5 Biological applications

While the image-based analysis of single cells over time is useful in many biological disciplines (Muzzey and van Oudenaarden, 2009), in this work we focused on two specific applications in the blood context where time-lapse microscopy experiments have a long tradition and led to numerous impactful findings. On the one hand, fluorescence time-lapse microscopy is extensively used to study the migration behavior of T-lymphocytes, a subclass of immune cells (Lämmermann and Germain, 2014). On the other hand, long-term high-throughput time-lapse microscopy is well suited to observe the differentiation of hematopoietic stem cells to mature blood cells (Schroeder, 2008; Hoppe et al., 2014). In the following paragraphs, we will discuss the biological background of both applications in more detail.

1.5.1 The role of T-lymphocytes in the immune system

T-lymphocytes are a subgroup of white blood cells, where they build the adaptive immune response together with B-lymphocytes (Alberts et al., 2002). The "T" in the name is an abbreviation for thymus, their place of maturation, while B-lymphocytes are maturating in the bone marrow. Lymphocytes comprise 20% to 40% of the white blood cells in humans and are playing a fundamental role in the immune system, where their abundance can vary in dependence of an inflammation of the host. Their function is the detection of cell-bound antigens, i.e. foreign substances or microorganisms that could be detrimental for the host. T-lymphocytes bear receptors on their surface that can bind antigens. A large number of T-lymphocyte subpopulations exist, which are characterized by different receptor combinations that are specialized to specific antigens. An exhaustive characterization of the function, behavior and molecular makeup of T-lymphocytes is thus crucial for the treatment of inflammatory and autoimmune diseases (Nourshargh et al., 2010).

T-lymphocyte migration

Throughout their lifetime, T-lymphocytes are scanning the host for cell-bound antigens. They perform this task by constantly migrating from the blood stream to the surrounding tissue, where they stay for a short time and then re-enter the blood through lymphatic vessels (Girard et al., 2012). The varying environments from the suspension-like blood to porous confined tissues that the T-lymphocytes are passing throughout this process require them to switch between different motility strategies, a behavior that is known as "amoeboid migration" (Renkawitz and Sixt, 2010). One hallmark of this behavior is speed. In contrast to e.g. mesenchymal cells that typically migrate with a speed of 0.5 µm/min, lymphocytes can reach velocities up to 20 µm/min (Friedl et al., 2012). Another characteristic for amoeboid migration is that the cells can move autonomously from their environment. Similarly to metastatic tumor cells, lymphocytes are anchorage-independent, meaning that they can survive without being attached to other cells (Renkawitz and Sixt, 2010).

T-Lymphocytes can adopt several migration strategies in response to different environmental signals. A typical observation is a morphological change from a circular to a more complex and often polarized and elongated shape (see Figure 1.6). Among others,

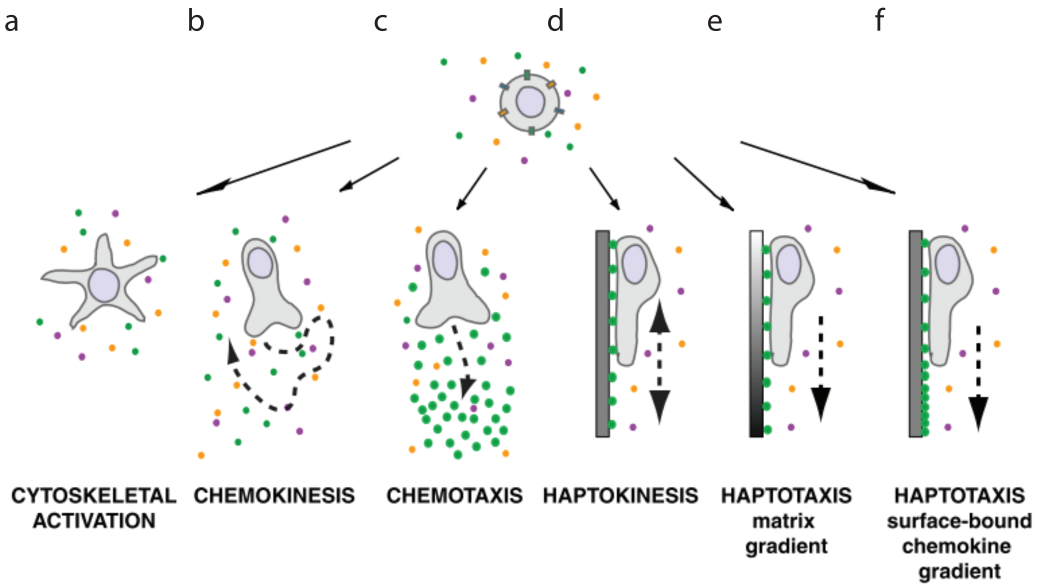


Figure 1.6: Overview of T-lymphocyte migration strategies. T-lymphocytes react to different stimuli with complex intracellular signaling cascades to induce changes in shape or migration. In a homogeneous field of soluble ligands, leukocytes can (a) increase the cytoskeletal activity, leading to morphological shape changes without cell motility or (b) induce self-polarization as prerequisite for non-directed cell migration (chemokinesis). (c) Soluble gradients of external ligands can polarize cells along the gradient and stimulate directed migration (chemotaxis). (d) Upon perceiving homogeneous surface-bound chemokines, lymphocytes can increase cell adhesion, self-polarize, and confine their migration in a non-directed manner along the surface (haptokinesis). Lymphocytes can migrate by two forms of haptotaxis: (e) along a gradient of extracellular matrix and cellular adhesion sites and (f) along a gradient of substrate-bound chemoattractants. Figure taken with permission from Lämmermann and Germain (2014).

these environmental signals consist of cytokines or chemokines, but also growth factors and unconventional stimuli like hydrogen peroxide or electrical fields. A motility program called chemokinesis can be observed if the cell is residing in a homogeneous field of soluble chemokines. There, the cell polarizes in a front-back axis with a broader leading edge and a contractile trailing edge, that is also called uropod (see Figure 1.6b). If the cell is migrating along an external soluble gradient of a chemoattractant in a directed fashion, this is called chemotaxis (see Figure 1.6c). A different set of motility strategies is needed when lymphocytes perceive chemoattractants that are bound to cell surfaces. In this scenario, the cells migrate along the cell membrane in an undirected or directed manner, which is called haptokinesis (see Figure 1.6d) and haptotaxis (see Figure 1.6e,f), respectively.

The role of actin in T-lymphocyte migration

Actin is a structural protein that is part of the cytoskeleton and represents one of the most abundant proteins in eukaryotic cells. Actin proteins are the building blocks of actin filaments, a structure that functions as a scaffold in the cell soma, thereby defining its shape. The cell can change the structure of the actin network rapidly, which results in morphological changes and directed migration, but also intracellular reorganisation (Alberts et al., 2007). Actin has a key role in the execution and switching between different motility programs by contributing to the formation of protrusive cell structures, pseudopods, and leading edges at the cell front (Lämmermann and Germain, 2014). During migration, a cell first builds a protrusion at the intended direction, mediated by actin branches that are built on existing actin filaments (Pollard and Cooper, 2009). Regulated by a network of molecular factors, migration is then achieved by simultaneous assembly and disassembly of actin at the leading and rear edge, respectively (Ridley et al., 2003). The importance of actin is also stressed by its involvement in many diseases. Mutations that affect the proper production of actin proteins can lead to muscular diseases (Olson, 1998) or deafness (Procaccio et al., 2006). Thus, understanding the regulatory network that controls actin production and degradation, but also the effect of changes in the actin filament network is crucial to understand the migration behavior of T-lymphocytes.

Confinement setups to study lymphocyte migration in vitro

Due to their rapid migration and constant transition from the blood stream to tissue, following T-lymphocytes *in vivo* is challenging. Despite increasing efforts to establish imaging techniques that allow to observe inflammatory responses in a living organism (Germain et al., 2012), most insights into T-lymphocyte behavior have been achieved through *in vitro* experiments (Lämmermann and Germain, 2014). A technique that is widely used to mimic T-lymphocyte behavior in 3-dimensional environments is to confine the cells between two planar surfaces at a distance that is smaller than the typical diameter of a single cell and conduct time-lapse microscopy (Malawista and de Boisfleury Chevance, 1997). The setups can be varied in the height of the confinement by custom-made micropillars (2-8 µm are typical) and the migration behavior as well as morphological differences can be observed. Studies with T-lymphocytes in such microchannels of various sizes demonstrated the existence of a confinement optimum for migration. T-lymphocytes migrated the fastest at confinement heights of 7–9 µm, while their speed was significantly decreased in a smaller confinement of 4–5 µm or larger confinement of 12–20 µm (Jacobelli et al., 2010). Also the cell speed was found to be independent of the specific channel coating like integrin ligands, fibronectin or ICAM-1.

Bioimage informatic analysis and computational modeling

To fully understand the different modes of T-lymphocyte migration, the molecular factors that are involved in this process and the environmental dependence, a detailed quantitative analysis of their migration behavior on the cellular and molecular level is necessary. For example, it has been shown in fish keratocytes that a cell's speed can be predicted by its shape and the underlying actin dynamics (Keren et al., 2008). After manually extracting single cell bodies from every frame of a time-lapse experiment the authors quantified the shape and the distribution of actin at the leading edge and built a mathematical model based on these measurements. Similar to this study by Keren et al. (2008), most of our recent understanding of T-lymphocyte migration has been derived from *in vitro* time-lapse microscopy experiments. Yet as typically hundreds of cells have to be analyzed in different environmental conditions (e.g. confinement depths or channel coatings), the manual analysis of these datasets is tedious and does not allow a full quantification of the shape and motility space. Generic software packages for automatic tracking and quantification exist, but are either not well suited for manual postprocessing, lack proper strategies to identify cell shapes or do not scale sufficiently to process large amounts of data (see Section 1.4). Thus, to fully quantify the migration behavior and contribution of molecular players such as actin in thousands of cells in time-lapse microscopy experiments, novel processing and analysis strategies have to be developed that can provide continuous measurements without human interaction.

1.5.2 Hematopoietic lineage choice

The hierarchical differentiation system of hematopoiesis

Studying the phenotypical behavior and molecular mechanisms of T-lymphocytes might provide information for the treatment of inflammatory diseases. However, if the whole immune system is malfunctioning, which is the case in leukemia, other cell types and molecular methods have to be scrutinized. Leukemia is caused by errors in the blood building system, the hematopoiesis (Orkin and Zon, 2008). The most common forms of the disease (i.e. chronic lymphoid leukemia and acute lymphoid leukemia) typically leads to a huge increase of non-functioning white blood cells, with T-lymphocytes being

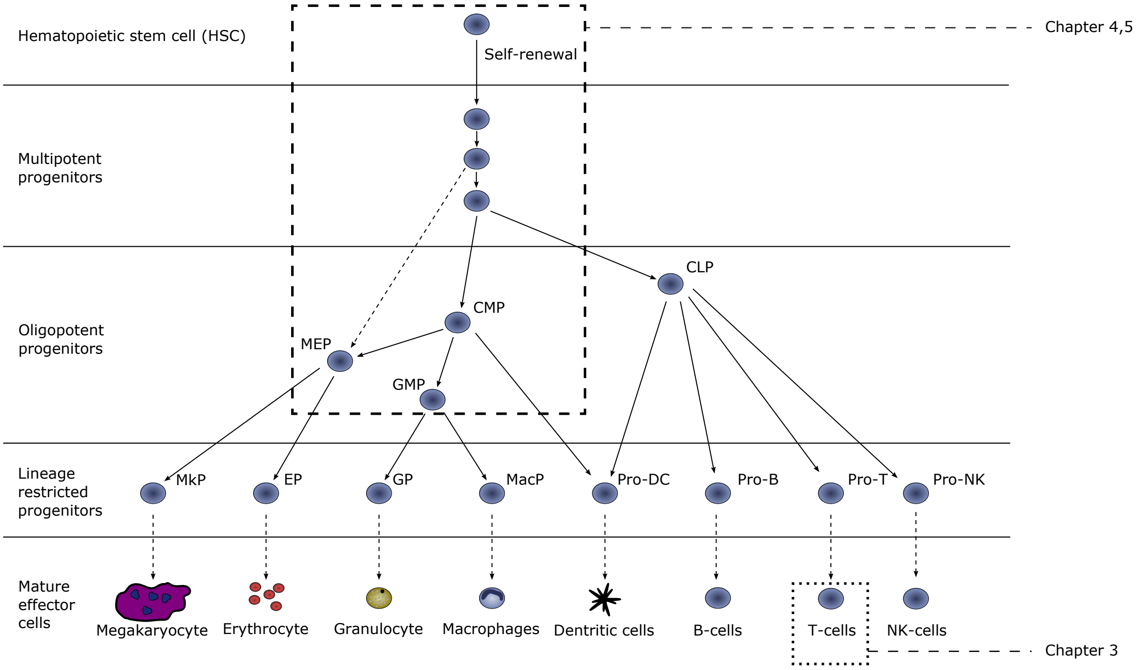


Figure 1.7: Current hierarchical differentiation model of the murine hematopoietic system. The hematopoietic stem cell (HSC) resides at the top of a hierarchy, is able to self-renew but also to give rise to all hematopoietic lineages (i.e. differentiation). HSCs first differentiate into multipotent progenitors (MPPs), i.e. cells that have lost the ability to self-renew but retain multipotency. MPPs are capable of giving rise to common myeloid progenitors (CMPs) and common lymphoid progenitors (CLPs), although recent evidence suggests that MPPs are also able to directly differentiate into megakaryocyte/erythrocyte progenitors (MEPs, dashed gray line) Seita and Weissman (2010). CMPs differentiate into megakarycyte/erythroid progenitors (MEPs) and granulocyte/macrophage progenitors (GMPs). The MEP then differentiates into unipotent progenitors, namely megakaryocytes (MkPs) and erythrocytes (EPs). GMPS differentiate into unipotent granulocyte progenitors (GPs) and macrophage progenitors (MacP). CLPs directly differentiate into unipotent progenitors of mature leukocytes, i.e. dendritic cells (DCs), natural killer cells (NK), B-lymphocytes or T-lymphocytes. Image adapted from (Seita and Weissman, 2010) and Hermann (2009). Dotted gray box: T-lymphocytes play an important role in the innert immune answer. We contributed to better understand the behavior of this cell type in Chapter 3. Dashed gray box: Myeloid lineage choice: A HSC can differentiate into either an MEP or GMP. We contributed to this question in Chapters 4 and 5.

a subpopulation thereof. To date, many molecular causes (e.g. gene mutations that are involved in the process) of different leukemias are not known in detail (Hutter, 2010).

Healthy hematopoiesis is structured hierarchically, with the hematopoietic stem cell (HSC) residing at the top of this hierarchy (Orkin and Zon, 2008). Similar to other stem cells, the HSC is characterized by two exclusive key attributes. On the one hand, a HSC is multipotent, i.e. it is able to give rise to all mature blood cell types throughout several rounds of cell divisions and intermediate cell types, a process called differentiation. On the other hand, HSCs can maintain their population indefinitely by dividing into two daughter cells that posses identical properties, which is called self-renewal. These attributes were first identified by Becker et al. (1963), who studied the effect of freshly implanted bone marrow cells in irradiated mice and later specified for the hematopoietic system by Wu et al. (1968). The hematopoietic system is one of the most extensively studied stem cell systems and the hierarchical differentiation model, i.e. the graph-structured pathways of lineage decisions a HSC can take until differentiating into a mature blood cells, has undergone several revisions over the last decades (Eaves, 2015). While the hierarchical nature of hematopoiesis persisted in most models, the initial concept only accounted for a myeloid and lymphoid lineage that directly emerged from a HSC. Out of these progenitors, the mature cell types were thought to maturate over several cell division rounds. The identification of several subtypes of progenitor cells were lead the current model of the murine hematopoietic system as shown in Figure 1.7.

Current evidence suggests that differentiation of a HSC to mature blood cells arises through a series of cell divisions, where one or both daughters represent a more specialized subtype from the previous one (Seita and Weissman, 2010). First, the progenitors of a HSC lose their ability to self-renew but retain multipotency (multipotent progenitors, MPPs). MPPs differentiate into progenitors either of the myeloid lineage (common myeloid progenitor, CMP) or the lymphoid lineage (common lymphoid progenitor, CLP). CMPs give rise to progenitors of either the MegE lineage (megakaryocyte/erytrocyte progenitor) or the GM lineage (granulocyte/macrophage progenitor, GMP). CLPs differentiate into all lymphocytes, i.e. dentritic cells (DCs), B-lymphocytes, T-lymphocytes, natural killer cells (NK-cells) and all their unipotent progenitors. MEPs are evenutally differentiating into erythrocytes (red blood cells) or megakaryocytes, whereas progeny of GMPs differentiate into the three types of granulocytes as well as macrophages. The process of differentiation in the adult organism primarily takes place in the bone marrow of pelvis, cranium, vertebrae, and sternum, where stem cells are residing in a microenvironment (the so-called niche), that maintains the amount of stem cells and triggers lineage commitment (Orkin and Zon, 2008). Recent studies suggest that HSCs leave their niche and dissolve in the blood stream only if already committed to a lineage. Their abundance is very low, only one in $200,000$ cells that are found is an uncommitted HSC (NIH, 2006).

A hematopoietic stem or progenitor cell (HSPC) is identified by analyzing whether the cell expresses combinations of surface proteins, typically by FACS (see Section 1.1 and Seita and Weissman (2010)). For example, HSCs can be identified by the marker combination $Lin^- \ cKit^+ \ Sca1^+ \ Flk2^- \ CD34^- \ Slamf1^+$ (a "+" denotes marker expression, a "-" denotes the contrary), while a MEP can be identified by $Lin^- \ cKit^+ \ Sca1^+ \ CD34^-$ $FcgR-$ and a GMP is determined by $Lin^- \ cKit^+ \ Sca1^+ \ CD34^+ \ FcgR+$ Seita and Weissman (2010). Thus the *FcgR* receptor (also called *FC gamma*, $FC\gamma$ or CD16/32) can be used to decide if a HSC has differentiated into either the MegE or the GM lineage.

Myeloid lineage choice

Despite the progress in refining the hematopoietic hierarchy and the function of mature blood cells, the molecular mechanisms driving the hematopoietic lineage choice are widely unknown (Hoppe et al., 2016). Hematopoietic differentiation is a highly regulated process where a variety of factors, such as cell to cell signaling or environmental changes, as well as gene regulation through transcription factors are involved Hoppe et al. (2014). For example, it was found that signaling glycoproteins (i.e. cytokines) that are involved in cell growth and differentiation are also playing a role in the regulation of hematopoiesis (Zhu and Emerson, 2002; Rieger et al., 2009; Hoppe, 2008). In addition, certain transcription factors are key intrinsic regulators of fate decisions (Göttgens, 2015).

Two factors that are believed to be involved in the myeloid lineage choice (i.e. commitment of a HSC to either MegE or GM lineage) are PU.1 and GATA1 (see Figure 1.8 and (Hoppe et al., 2016)). Previous studies suggested that these proteins have a cross-antagonistic relationship, resulting in an increase of GATA1 and simultaneous decrease of PU.1 expression in cells of the MegE lineage, whereas the opposite can be observed in GM-committed cells (Back et al., 2005; Nutt et al., 2005). Also, several mathematical models were created that tried to explain the hematopoietic lineage choice (Roeder and Glauche, 2006; Huang et al., 2007; Chickarmane et al., 2009). A recent model proposed by Strasser et al. (2012) used a probabilistic toggle-switch model to study the lineage choice of HSPCs and found that high protein numbers or long-term modifications (e.g. chromatin remodeling) are needed to maintain a stable commitment. Yet, all these findings were made on snapshot or population analyses on the transcription level, thus alternative models could not be ruled out (Hoppe et al., 2016).

To analyze the relationship of PU.1 and GATA1 during differentiation in more detail, Hoppe et al. (2016) used long-term high-throughput time-lapse microscopy as described in Section 1.2. This technique is a powerful tool to study differentiation processes of single cells in unprecedented temporal resolution *in vitro*, as a high frequency of bright-

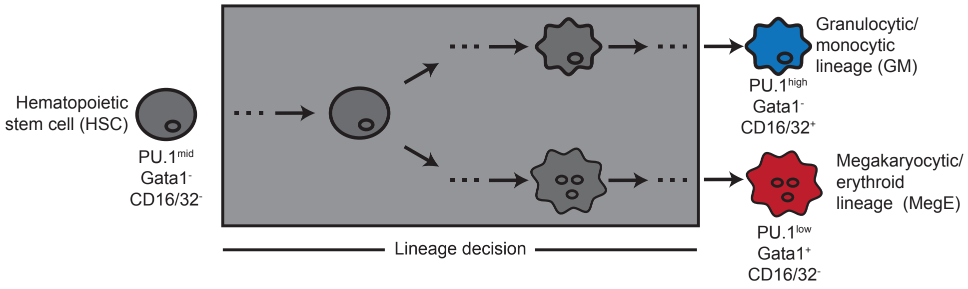


Figure 1.8: Hematopoietic stem and progenitor cells (HSPCs, gray cells) can be annotated as committed by observing cells stained by CD16/32 antibody (GM lineage) or GATA1 expression (MegE lineage). As markers typically identify more specialized cell types, early identification of committed cells is not possible (gray box). However, morphological differences might occur before changes in marker expression.

field imaging (typically on the scale of a few minutes) ensures that moving single cells and cell divisions can be accurately tracked and used for the generation of cellular genealogies. Additionally, fluorescent imaging (due to toxicity typically done on the scale of hours, see Schroeder (2011)) allows the identification of molecular lineage markers or other fluorescent readouts such as transcription factor expression. Using CD16/32 as a marker for GM-committed cells and high GATA1 expression for MegE-committed cells, the authors found that the expression dynamics of PU.1 and GATA1 did not fit to the recent cross-antagonistic model. The authors thus proposed that the "physical PU.1/Gata1 interaction and antagonism is not the core mechanism of GM versus MegE lineage decision making, but rather serves as an execution and/or lock-down mechanism that makes terminal differentiation irreversible" (Hoppe et al., 2016).

Early identification of myeloid lineage choice by morphological differences

The gradual change of PU.1/GATA1 expression before the upregulation of lineage markers such as CD16/32 and GATA1 suggests that the commitment between MegE or GM lineage could happen in any HSPC state, from the HSC over MPPs to CMPs. Yet, the lack of suitable markers makes it impossible to narrow down the time point of lineage commitment any further. However, a rich source of information that is available throughout the full observation of a differentiating cell is its morphological and motility behavior that can be derived directly from brightfield images without the use of molecular markers. The general feasibility of this approach has been shown by Cohen et al. (2010). The authors conducted time-lapse experiments to observe differentiating retinal progenitor cells

(RPCs) and built a bioimage informatic processing pipeline to derive descriptive features, i.e. a cell's speed and morphology over time. Based on this data a classification algorithm (a support vector machine) was trained, revealing that RPCs show distinctive behavior depending on which combination of daughter cell types they were about to generate. This approach achieved a prediction accuracy of 95%. Yet, conducting a similar approach on long-term high-throughput time-lapse experiments holds several challenges. While Cohen et al. (2010) studied RPCs that underwent one division and were barely migrating, differentiating HSPCs divide several times and migrate over the full coverslip at high speeds (Schroeder, 2010). In addition, RPCs show diverse morphodynamics before division, while HSPCs mostly show ellipsoid shapes that are hard to distinguish. Thus, the development of fully automated image processing pipelines that can deal with high-throughput time-lapse microscopy data, but also sophisticated algorithms that can detect even subtle morphological differences between differentially committed cells becomes necessary.

1.6 Research questions and thesis overview

To fully understand complex biological mechanisms of T-lymphocyte migration and hematopoietic lineage choice, the analysis of single cells is crucial. Time-lapse microscopy is a powerful tool to study spatio-temporal dynamics of biological systems on the single-cell level. Yet, to analyze the abundant, often high-dimensional data that is generated in these experiments customized bioimage informatic processing pipelines and statistical methods are required.

The goal of this thesis is to contribute to the question, to which extend the morphodynamics of a single cell allows conclusions about its current or future molecular state. To this end, our task is the development of adequate computational methods to identify single cells in (i) fluorescence time-lapse microscopy experiments observing T-lymphocyte migration and (ii) high-throughput time-lapse microscopy experiments observing differentiating hematopoietic stem cells and to quantify their behavior. This involves the adjustment of experimental setups to generate images that are optimal for automated processing, the filtering of different types of noise, and the correction of background intensities. Furthermore, we need to extract all cell bodies from the images and apply adequate measures to derive the morphology and motility information. Eventually, we have to apply statistical analyses and machine learning models to the derived data in order to detect patterns that allow novel biological insights.

In Chapter 2 we introduce methods and algorithms that we applied or extended to process images from time-lapse microscopy. We also define morphological and motility measures that we have used to quantify the dynamics of single cells. Furthermore, we explain the statistical measures, machine learning techniques and software tools that we have applied to analyze our datasets.

In Chapter 3 we describe the development of an automated tracking and segmentation pipeline to quantify and analyze the morphodynamics and migration behavior of T-lymphocytes in fluorescence time-lapse microscopy experiments with different confinement depths. The challenge in this project was to create a framework that allows to (i) automatically track single cells over the full time-lapse experiment, (ii) derive morphological information in several fluorescence channels and (iii) quantify and statistically analyze the migration behavior of T-lymphocytes and the dependence to shape changes. We apply a variational Bayesian approach to fit a Gaussian mixture model and find two groups of non-migrating and migrating cells in every confinement depth. In addition, we use the quantified shape information to train a nonlinear regression model that predicts a cell's speed at every frame. The developed framework will be an important tool for future analyses within this project, as the multitude of different environmental influences that have to be analyzed in separate experiments demands an automated quantification and analysis. We established the project together with the group of Prof. Michael Sixt from the Intitute of Science and Technology in Vienna, while the author of this thesis established the full image processing pipeline, statistically analyzed the data and built the regression model. A manuscript is currently prepared for publication. It is entitled "A. Reversat*, **F. Buggenthin***, J. Merrin, A. Leithner, I. de Vries, F. J. Theis, C. Marr and M. Sixt. Morphodynamic analysis of T lymphocyte migration in confined microenvironments."

In Chapter 4 we discuss a fully automated and generally applicable segmentation method for brightfield images from long-term high-throughput time-lapse microscopy. The challenge in this project was to create a procedure that detects single cells in brightfield images without user interaction as accurate and fast as possible, as available methods are not able to produce satisfying results on this type of data. The project led to a publication where we demonstrated the robustness and high accuracy in the detection of single cells in hundreds of thousands of brightfield images by predicting population doubling times. The author of this thesis was involved in the change of image acquisition protocol, designed the full segmentation method and conducted the statistical analyses that demonstrated the high performance of the proposed algorithm. The paper is entitled: **F. Buggenthin**, C. Marr, M. Schwarzfischer, P. S. Hoppe, O. Hilsenbeck, T. Schroeder, F. J. Theis. An automatic method for robust and fast cell detection in bright field images from high-throughput microscopy. *BMC Bioinformatics*, 2013.

In Chapter 5 we embed the segmentation algorithm for brightfield images in a fully parallelized and automatized framework that identifies single cells in millions of images from brightfield microscopy and quantifies their morphology and motility. Furthermore, we implement a convolutional neuronal network to predict the differentiation of hematopoietic stem cells into different lineages, based only on the raw pixel information of derived cell patches and a cell's speed. We apply our framework to a set of three long-term high-throughput time-lapse experiments of differentiating hematopoietic stem cells and show that our method predicts the correct lineage choice with high accuracy up to three divisions before molecular markers become detectable, even if the predictions are carried out on a fresh experiment. This project was a joint effort. The author of this thesis created the parallelized and automatized image processing framework and derived the morphological information, performed extensive data cleaning and annotation, parts of the manual tracking and trained the conventional machine learning methods. Florian Buettner designed, implemented, trained and tested the convolutional neural network. This chapter is based on and in parts identical with "**F. Buggenthin***, F. Buettner* , P. S. Hoppe, M. Kroiss, M. Strasser, M. Schwarzfischer, D. Loeffler, K. D. Kokkaliaris, O. Hilsenbeck, T. Schroeder, F. J. Theis and C. Marr, Prospective identification of hematopoietic lineage choice by deep learning. *Under review at Nature Methods*".

In Chapter 6 we summarize the methods and biological insights that arose during the preparation of this dissertation and put it into context of recent publications in the field. We also discuss possible extensions of our methods and suggest new experiments that will build upon our contributions.

1.7 Further scientific contributions

In addition to the publications mentioned above, the author of this dissertation participated in the preparation and publication of several other projects. The contributions were not directly related to the main research questions discussed above, yet resulted in the following co-authorships:

- S. Bardehle, M. Krüger, **F. Buggenthin**, J. Schwausch, J. Ninkovic, H. Clevers, H. J. Snippert, F. J. Theis, M. Meyer-Luehmann, I. Bechmann, L. Dimou and Magdalena Götz. Live imaging of astrocyte responses to acute injury reveals selective juxtavascular proliferation. *Nature Neuroscience*, 2013.

- O. Hilsenbeck*, M. Schwarzfischer*, S. Skylaki, B. Schauberger, P. S. Hoppe, D. Loeffler, K. D. Kokkaliaris, S. Hastreiter, E. Skylaki, A. Filipczyk, M. Strasser, **F. Buggenthin**, J. S. Feigelman, J. Krumsiek, A. J. J. van den Berg, M. Endele, M. Etzrodt, C. Marr, F. J. Theis, T. Schroeder. Software tools for single-cell tracking and quantification of cellular and molecular properties. *Nature Biotechnology*, 2016.

- A. von Streitberg, C. Straube, **F. Buggenthin**, S. Schneider, C. Marr and L. Dimou. Transient loss of NG2-glia homeostasis after acute brain injury. *Under review at Nature Communications.*

- V. Lupperger, **F. Buggenthin**, P. Chapouton, C. Marr, Identification of single neural stem cells in the zebrafish brain. *Submitted to ISBI'16.*

- D. Dragoi, A. Krattenmacher, V. K. Mishra, J. M. Schmidt, U. J. Kloos, L. K. Meixner, H. S. Bartsch, K. Sotlar, S. Hauck, **F. Buggenthin**, D. C. Marr, S. A. Johnsen and C. H. Scheel. Twist1 induces distinct cell states depending on TGFBR1-activation. *Oncotarget*, 2016.

Chapter 2

General methods and definitions

In this chapter we provide formal definitions and background information to all methods and algorithms used in the subsequent chapters 3, 4 and 5. We start by explaining basic concepts of image processing algorithms that are necessary for noise reduction and normalization, followed by algorithms to extract objects from images. We continue by defining measures to quantify morphology and motility of cells. In the last part of this chapter, we discuss how to derive patterns of cell behavior from multidimensional datasets with machine-learning algorithms. We close the chapter by providing an overview of the software and tools that we used in the studies.

2.1 Image processing

In order to derive meaningful information from an image, it is necessary to filter noise and normalize for batch effects. This is especially important when dealing with images from high-throughput or time-lapse microscopy, where the quality can strongly vary throughout the - typically very large - dataset. In this section we introduce mathematical concepts and algorithms from the field of image processing that we used in our projects. If not stated differently, formal definitions and pseudocode descriptions are adapted from Young et al. (1998) and Nixon and Aguado (2012).

2.1.1 Principles of digital imaging

Definition of a digital image

In general, an image can be defined as a continuous two-dimensional function $I(x, y)$ with x and y being two independent variables. A digital image is a sampled and thus discretized representation of I, where $x \in \{1, 2, ..., N_x\}$ and $y \in \{1, 2, ..., N_y\}$ are indices of a two-dimensional matrix $\mathbf{D}$. The entries $\mathbf{D}_{x,y}$ of $\mathbf{D}$ are termed pixels. The range G of discrete values $g \in 0, 1, ..., G$ (here: color or brightness) that the pixels of $\mathbf{D}$ can attain is determined by the data format, which is typically 8-bit (255 values), 16-bit (65536 values) or 32-bit(2^{32} values). As most algorithms in image processing are designed for digital images, we will define all concepts for the discretized case. $\mathbf{D}$ can be assumed to be a monochromatic (i.e. gray-scale) image of $\sim 1000 \times 1000$ pixels with a depth of 8 bit, as this was the predominant type of image we analyzed in our projects. Note that extended formulations of $\mathbf{D}$ exist to describe three-dimensional and/or polychromatic images, or even n-dimensional images with multiple color channels, time points and three spatial dimensions, which were not used in this study.

Noise and artifacts in digital imaging

In digital photography, several parts of an imaging system can contribute to unwanted variations of the brightness or color level in the acquired image that were not part of the original signal. For example, CCD cameras that are typically used in digital microscopy count the number of photons that are hitting the cells of an array of light-sensitive capacitors, where the cells of the array correspond to the pixels of the resulting image (Tompsett et al., 1971). Due to the stochastic processes that are involved in photon production, it cannot be guaranteed that in a given pixel for two consecutive but independent observation intervals, the same number of photons is counted. In addition to stochastic noise, digital imaging systems often suffer from systematic artifacts. This could be dead pixels in CCD cameras or optical distortions due to distracted light beams in a system of lenses.

The degree of noise in an image can range from small specks that are almost not visible (e.g. photos taken with digital cameras under good illumination conditions), to extremely distorted images where the original information that should be recorded is nearly impossible to retrieve (e.g. radioastronomical images). In the following paragraphs we will introduce the most common types of noise and image artifacts that we found in our data

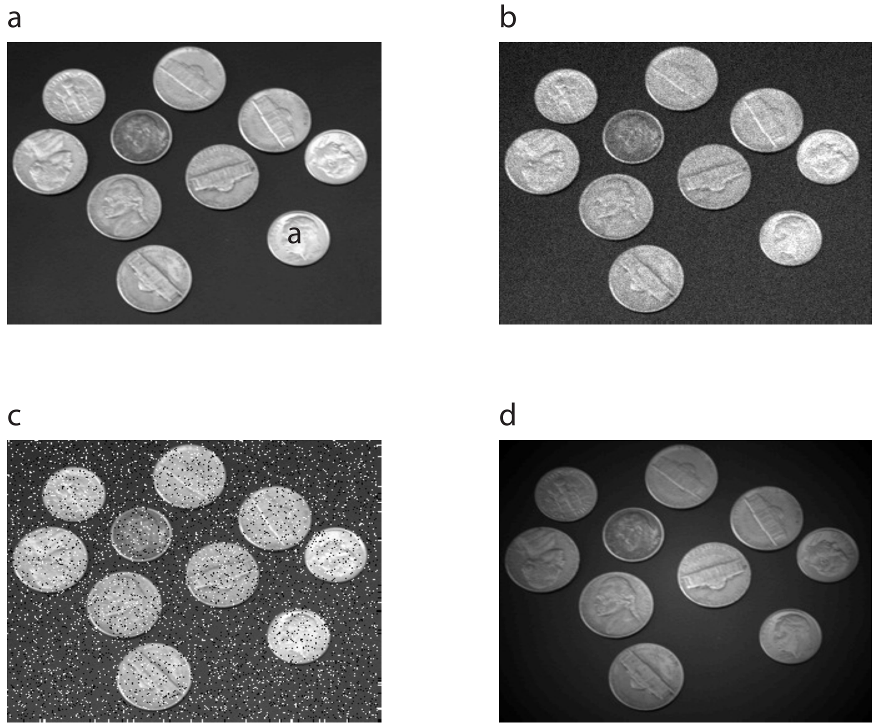


Figure 2.1: Impact of different types of noise or artifacts on image quality. (a) 8-bit grayscale image without noise. The background is evenly colored, details such as contours of the coins are clearly visible (b) Gaussian noise, e.g. caused by heterogeneous photon counts on CCD chips. A stained pattern is covered all over the image. (c) Salt and Pepper noise, typically caused by e.g. dead CCD pixels or bad shielding of the transmission cable. Single pixels or small areas of the image are highly over- or underexposed. (d) Vignetting. Optical aberration causes a uneven, spherical illumination, where the corners of an image appear darker than the center.

sets. See Figure 2.1 for an overview. Methods to correct for the different noise terms are explained in Section 2.1.2.

- **Gaussian noise**: The number of photons that are collected by a CCD at every pixel typically follows a Poisson distribution (Schottky, 1918). Yet, as a high-quality digital camera counts 10^6 photons per pixel per second (Llull et al., 2013), and the Poisson distribution can be approximated by a normal distribution for high photon numbers, the noise can be modeled by a Gaussian distribution $p(g)$, with g being the intensity value. For an 8-bit digital image $\mathbf{D}$ with $g \in [0, 255]$ the Gaussian noise term is defined as

$$p(g) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(g-\mu)^2}{2\sigma^2}} \tag{2.1}$$

where μ is the mean and σ the standard deviation of the Gaussian. Typically, Gaussian noise is regarded as an additive effect to the original signal. Thus it can be simulated by adding an image $\mathbf{D}^{Gauss}$ with i.i.d pixel intensities drawn from a Gaussian distribution to the original image $\mathbf{D}$.

- **Salt-and-pepper noise**: Salt and pepper noise results in single, randomly distributed pixels in an image that are set to the maximum or minimum value of the data format (i.e. 255 or 0 in an 8-bit image). It occurs as the result of defective pixels in a CCD camera or errors in the transmission system, for example a broken or badly shielded transmission cable.
- **Vignetting**: Vignetting is an optical distortion of an image that results in a gradual and circular decrease of the pixel intensities from the center of an image to its borders. While there are different sources of this effect, in microscopy vignetting mostly occurs due to the round lenses that are used in the imaging system. The optical light path illuminates a round area of the specimen, while the CCD records a rectangular image. Vignetting can be circumvented at acquisition by Koehler illumination (Koehler, 1894). However especially in high-throughput long-term time-lapse microscopy this effect occurs frequently.

Signal to noise ratio

The quality of an image is a subjective value that is highly application-dependent. While an extremely noisy radioastronomical image could be sufficient to detect brighter groups of pixels that could represent stars, the same level of noise would destroy an image with an high amount of details, such as a digital photograph. However, a way to quantify the amount of noise that is contributing to a digital image $\mathbf{D}^{\text{noise}}$ by comparing it to an undistorted ground truth image $\mathbf{D}^{\text{org}}$ with dimensions $N \times N$. This measure is called Signal-to-noise ratio (SNR). It is defined as follows:

$$SNR(\mathbf{D}^{\text{noise}}, \mathbf{D}^{\text{org}}) = 10 \cdot log_{10} \left(\frac{\sum\limits_{x=1}^{N} \sum\limits_{y=1}^{N} (\mathbf{D}^{\text{noise}}_{x,y})^2}{\sum\limits_{x=1}^{N} \sum\limits_{y=1}^{N} (\mathbf{D}^{\text{noise}}_{x,y} - \mathbf{D}^{\text{org}}_{x,y})^2} \right) dB \tag{2.2}$$

2.1.2 Image filtering

Several methods exist to enhance the signal to noise ratio of a digital image $\mathbf{D}$ or to filter noise. These steps are necessary in order to derive unbiased measures from images in an experiment that were taken at different time points or locations, but also to compare images between different experiments. In this section we introduce algorithms that we used to filter different types of noise, as well as the basic mathematical concepts that are needed for their formulation.

Convolution

One of the fundamental operations in image processing is to change the pixel values of an image by applying a locally dependent kernel $\mathbf{K}$. The kernel usually is a very small squared image with odd spatial dimensions to ensure a proper positioning. This operation is termed convolution. For a digital image $\mathbf{D}$ and a kernel $\mathbf{K}$ with dimensions $H \times H$, convolution is defined as:

$$\mathbf{D}^{\text{conv}}_{x,y} = \mathbf{D} * \mathbf{K} = \sum_{i=0}^{H-1} \sum_{j=0}^{H-1} \mathbf{K}_{i,j} \cdot \mathbf{D}_{x-i,y-j}, \tag{2.3}$$

i.e. the operation is applied on every pixel $\mathbf{D}_{x,y}$ of $\mathbf{D}$. Depending of the entries (or coefficients) of $\mathbf{K}$, different operations such as smoothing or sharpening can be achieved. Convolution makes it possible to implement operators where the output pixel values represent simple linear combinations of two or more input pixel values. Typical dimensions of $\mathbf{K}$ are 3×3, 5×5 or 7×7, while larger kernel sizes result in slower execution of the operation. In this thesis, we used convolution to build the Gaussian smoothing filter (see Section 2.1.2) and the convolutional layers in neural networks (see Section 2.4.2).

Discrete Fourier transform

The Fourier transform is a method to dissect a signal into its frequency components. This allows to examine the contribution of every frequency to the original signal, for example to filter out noise that often resides in higher frequency ranges. The original formulation of the Fourier transform is referring to a continuous and infinite one-dimensional signal. For digital images that are discrete and non-infinte, the discrete Fourier transform (DCT) has been formulated. For a digital image $\mathbf{D}$ with spatial dimensions $N \times N$, the DCT is written as:

$$\mathrm{F}(\mathbf{D}) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \mathbf{D}_{x,y} e^{-i(\frac{2\pi}{N})(ux+vy)}, \tag{2.4}$$

where u, v are the respective horizontal and vertical frequencies and $i = \sqrt{-1}$ is the complex variable. Consequently, the reverse operation to reconstruct an image from Fourier space is defined as follows:

$$\mathbf{D} = \mathrm{F}^{-1}(\mathrm{F}(\mathbf{D})) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \mathrm{F}(\mathbf{D})_{u,v} e^{i(\frac{2\pi}{N})(ux+vy)}. \tag{2.5}$$

One of the most important properties of the Fourier transform with respect to image processing is that the convolution of two images $\mathbf{D1}$ and $\mathbf{D2}$ corresponds to the multiplication of both images in the Fourier space:

$$\mathbf{D}^{\text{conv}} = \mathbf{D1} * \mathbf{D2} = \mathrm{F}^{\text{-1}}(\mathrm{F}(\mathbf{D1}) \cdot \mathrm{F}(\mathbf{D2})) \tag{2.6}$$

This relationship and the resulting improvement in computation speeds allows for example to apply kernels with larger sizes. Also, the DCT, or more precisely the fast Fourier transform (FFT) was used to realize fast implementations of various image processing algorithms, for example frequency filters (Almeida, 1994) or registration operations (Reddy and Chatterji, 1996). In this dissertation the FFT is found in the implementation of the Gaussian filter as described in section 2.1.2.

Contrast stretching

Depending on the experimental setups, images acquired by digital microscopy often feature pixel intensity distributions that do not span the full range of available values (0-255 in an 8-bit image). A simple technique to enhance the dynamic range of an image is constrast stretching. Given an image $\mathbf{D}$ we can rescale the intensity of every Pixel $\mathbf{D}_{x,y}$ in accoradance to a minimum intensity threshold τ_{min} and a maximum intensity threshold

$\tau_{\max}$:

$$\begin{aligned} \mathbf{D}^{\text{norm}}_{x,y} &= (\mathbf{D}_{x,y} - \tau_{\min}) \frac{(255-0)}{(\tau_{\max} - \tau_{\min})} + 255, \quad \forall\, \mathbf{D}_{x,y} \in \mathbf{D}, \\ &\text{with} \\ \mathbf{D}^{\text{norm}}_{x,y} &= \begin{cases} 0, & \text{if } \mathbf{D}^{\text{norm}}_{x,y} < 0 \\ 255, & \text{if } \mathbf{D}^{\text{norm}}_{x,y} > 255. \end{cases} \end{aligned} \tag{2.7}$$

The thresholds $\tau_{\min}$ and $\tau_{\max}$ could be the minimum and maximum intensity of $\mathbf{D}$, yet this often leads to pixels with outlier intensities. A more robust approach is to use the e.g. 5% and 95% quantiles as the tresholds. The result of this operation is a contrasted image where foreground objects such as cell bodies should be better identifiable. A disadvantage of this method is that it also enhances noise, thus it is important to apply suitable filtering methods to the image before enhancing its contrast.

Gaussian filter

A standard filtering method in image processing is Gaussian blurring, e.g. convolving an image $\mathbf{D}$ with a Gaussian kernel $\mathbf{K}^{\text{gauss}}$. The values of the pixels $\mathbf{K}_{x,y}$ are determined by a two-dimensional Gaussian, which is defined as

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

where μ is the mean and σ is the standard deviation of the Gaussian. The convolution is then written as

$$\mathbf{D}^{\text{blurred}} = \mathbf{D} * \mathbf{K}^{\text{gauss}}. \tag{2.9}$$

The result of this operation is a smoothed version of the input image, where the level of blurring depends on the dimensions of $\mathbf{K}^{\text{gauss}}$ and the standard deviation σ. Low values of σ result in a reduction of stochastic noise (see section 2.1.1), but retain local details of the image. Higher values of σ and larger kernels can be used to remove all details from an image, so that only the background illumination pattern remains. The latter will be useful in background correction approaches of brightfield images. The size of $\mathbf{K}^{\text{gauss}}$ typically depends on σ and is computed as $2 * \text{ceil}(2 * \sigma) + 1$.

As discussed in section 2.1.2, computing a convolution for large images $\mathbf{D}$ and large kernels $\mathbf{K}$ can be very slow. Yet, a Gaussian function has the property that it stays a Gaussian in the frequency space. This can be used to implement a much faster version of the Gaussian filter by first transforming **D** into the Fourier space and then multiplying the resulting frequency image F(**D**) with the kernel $\mathbf{K}^{\text{gauss}}$:

$$\mathbf{D}^{\text{blurred}} = F^{-1}(\mathrm{F}(\mathbf{D}) \cdot \mathbf{K}^{\text{gauss}}). \tag{2.10}$$

Median filter

A disadvantage of a Gaussian filter is that it's not useful for non-stochastic noise such as salt-and-pepper noise. A method that better preserves edges while still effectively removing distortions in images is the median filter. The algorithm changes the intensity value of every pixel in an image **D** by the median intensity of a neighborhood **O**. **O** is related to the kernel **K**, yet as the median filter is a nonlinear operation it cannot be expressed by a convolution. The size of **O** determines the level of blurring, where dimensions of 3×3, 5×5 or 7×7 are common. It is written as:

$$\mathbf{D}^{\text{blurred}} = \text{median}([\mathbf{D}_{x-n/2,y-n/2}, ..., \mathbf{D}_{x+n/2,y+n/2}]). \tag{2.11}$$

Similar to convolution, the above operation is applied to every pixel $\mathbf{D}_{x,y}$ of **D**.

Sobel filter

To analyze an image it is often useful to figure out regions of pixels with homogeneous intensity levels. Typically the border of such a region can be detected by searching for sharp changes of intensity from one pixel to its neighbors, which is called an edge. The sobel filter detects these edges by convolving an image **D** with two kernels $\mathbf{K}^{\text{horizontal}}$ and $\mathbf{K}^{\text{vertical}}$:

$$\mathbf{K}^{\text{horizontal}} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \quad \text{and} \quad \mathbf{K}^{\text{vertical}} = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}. \tag{2.12}$$

The resulting images can be regarded as an approximation for the derivative of **D** in horizontal and vertical direction. The gradient magnitude can then be computed by combining both gradient images:

$$\mathbf{D}^{\text{edge}} = \sqrt{(\mathbf{D} * \mathbf{K}^{\text{horizontal}})^2 + (\mathbf{D} * \mathbf{K}^{\text{vertical}})^2}. \tag{2.13}$$

Laplace of Gaussian filter

In contrast to the sobel filter which detects edges, the Laplace of Gaussian filter (LoG) is a method to detect blobs in images, i.e. regions of pixels that have intensities that are constant or approximately constant. It can for example be used to detect circular objects such as stained nuclei in fluorescence images. An input image $\mathbf{D}$ is first convolved by a Gaussian kernel $\mathbf{G}$:

$$\begin{aligned} \mathbf{G} &= \frac{1}{2\pi s^2} e^{-\frac{x^2+y^2}{2s^2}}, \quad \textit{and} \\ \mathbf{D}^{\text{blurred}} &= \mathbf{D} * \mathbf{G} \end{aligned} \tag{2.14}$$

The difference to a normal Gaussian filter is the scale parameter $s \geq 0$, that determines the level of blurring and is used to create a scale space representation of the image. Afterwards the Laplacian operator is applied, which results in a representation of the image where blobs with an extent of $\sqrt{2t}$ appear as strongly positive values or strongly negative values, depending on their brightness relative to the mean intensity of $\mathbf{D}$. It is defined as the second derivative of $\mathbf{D}$ in horizontal and vertical direction:

$$\nabla^2 \mathbf{D} = \frac{\delta^2 \mathbf{D}}{\delta x^2} + \frac{\delta^2 \mathbf{D}}{\delta y^2}. \tag{2.15}$$

The blobs are then detected in $\mathbf{D}$ by a minimum/maximum search.

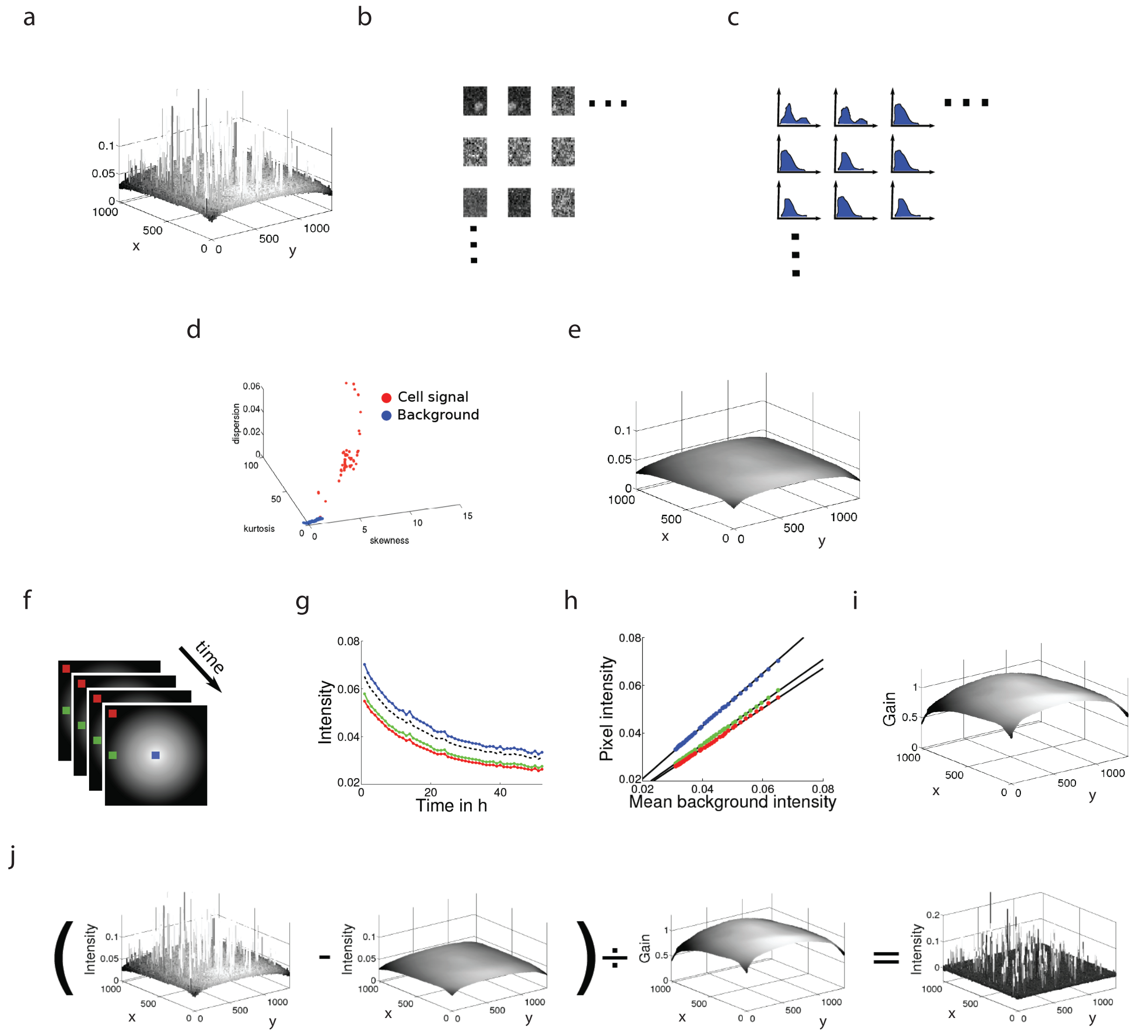
2.1.3 Illumination correction

In applications where it is necessary to accurately quantify e.g. the fluorescence signal of a given image, simple filtering methods often do not suffice to normalize the different sources of noise (Smith et al., 2015b). In this thesis we adapted and applied a method developed for fluorescence time-lapse microscopy by Schwarzfischer et al. (2011) that estimates and corrects the background of a given image.

Machine-learning-based background estimation

The sources of noise affecting a digital image $\mathbf{D}$ at spatial coordinates x and y (i.e. pixels $\mathbf{D}_{x,y}$) at a time point t of a time-lapse experiment can be formalized as follows:

$$\mathbf{D}_{x,y,t} = \mathbf{D}^{\text{signal}}_{x,y,t} \cdot g(x,y) + b(t) \cdot g(x,y) + o(x,y), \tag{2.16}$$


a
b
c
d
e
f
g
h
i
j
0.1
0.05
0
1000
500
x
0 0
500
1000
y
0.06
0.04
0.02
dispersion
100
50
kurtosis
0
5
10
15
skewness
Cell signal
Background
time
0.08
Intensity
20
40
Time in h
Pixel intensity
0.02
0.04
0.06
0.08
Mean background intensity
Gain
1
0.5
0.2

where $\mathbf{D}^{\text{signal}}_{x,y,t}$ is the pure fluorescence signal at a single pixel without any noise, $b(t)$ is a constantly decreasing homogeneous background signal (e.g. autofluorescence of the medium that is affected by photobleaching), $g(x, y)$ is a spatially dependent illumination function (also called gain) and $o(x, y)$ is the camera offset, which is constant over time. This formula can be reordered to reveal the influences of noise to the cell signal at a pixel:

$$\mathbf{D}^{\text{signal}}_{x,y,t} = \frac{\mathbf{D}_{x,y,t} - b(t) \cdot g(x, y) - o(x, y)}{g(x, y)}. \tag{2.17}$$

We define $\mathbf{D}^{\text{bg}}_{x,y,t} = b(t) \cdot g(x, y) + o(x, y)$ as the illuminated background signal including the offset at a pixel $\mathbf{D}_{x,y}$. To estimate $\mathbf{D}^{\text{bg}}_{x,y,t}$, the algorithm from Schwarzfischer et al. (2011) first divides a given digital image $\mathbf{D}$ into overlapping windows (tiles) of a fixed size $J \times J$. For each tile the intensity distribution is then analyzed and statistical moments are computed. Using the supervised machine-learning method random forest (see Section 2.4.2) or the unsupervised clustering method DBscan (see Section 2.4.1), every rectangle is then classified into either covering exclusively background pixels or also foreground pixels (i.e. pixels belonging to cell bodies). All foreground rectangles are discarded. At the center coordinates of every background tile, its respective mean intensity is used as a seed point in a grid to intra- and extrapolate the full background image $D^{bg}_{x,y,t}$. This step is applied to every image independently. To infer the gain the pixel intensities of a single image are analyzed over time and approximated by linear regression:

$$g'(x, y) = c \cdot g(x, y), \tag{2.18}$$

where c is the ordinate intercept that represents $o(x, y)$. In a last step, the corrected

---

Figure 2.2 *(facing page)*: Overview of the machine-learning-based background correction. (a) A graycale fluorescence image $\mathbf{D}$. The background intensity is constantly higher than zero (offset) and peaks in the plot denote cellular signals, obscured by noise. (b) The image is divided into small overlapping image patches (tiles). (c) For each tile, statistical moments of the intensity distribution are calculated. (d) The tiles are classified with regard to showing cell signal (foreground) or only background information. (e) The full background image $\mathbf{D}^{\text{bg}}$ is reconstructed by intra- and extrapolation, using the background tiles as seed points. (f,g) Analyzing the temporal change of single pixels (red,blue and green dot in (f)) over time reveals a constant decrease in intensity (bleaching, red blue and green curve in (g), black dashed line indicates mean intensity of the full image). (h,i) By linear regression, the gain $g'_t(x, y)$ and offset $o(x, y)$ are approximated. (j) Final correction of the original image $\mathbf{D}^{\text{corr}}$. The estimated brackground image $\mathbf{D}^{\text{bg}}$ is subtracted and the result is normalized by the approximated gain and offset. Figure adapted from Schwarzfischer et al. (2011).

relative cellular signal $\mathbf{D}^{\text{corr}}_{x,y,t}$ is calculated by subtracting the estimated background image at every pixel $\mathbf{D}^{\text{bg}}_{x,y,t}$, followed by a normalization with the approximated gain $g'(x,y)$:

$$\mathbf{D}^{\text{corr}}_{x,y,t} = \frac{\mathbf{D}_{x,y,t} - \mathbf{D}^{\text{bg}}_{x,y,t}}{g'(x,y)}, \tag{2.19}$$

with

$$\mathbf{D}^{\text{corr}}_{x,y,t} = \mathbf{D}^{\text{signal}}_{x,y,t} \cdot c^{-1}. \tag{2.20}$$

This normalization leads to a cell signal that is comparable over all images of a time-lapse experiment. Furthermore, the equalized illumination over single images leads to a huge increase in the robustness of segmentation algorithms, as we will show in chapter 4.

Supervised active learning for background estimation

The manual creation of a training set of labeled image patches for background estimation is inherently biased and very time consuming. An approach to reduce these issues is active learning, where a classification model is initialized with a very small training set (Settles, 2009). In our case, a graphical user interface then presents additional unlabeled image patches with borderline classification scores and asks to manually provide a class label. This approach minimizes the manual workload, as the samples that are added to the training set are most informative and thus less samples are needed to derive a well generalizing classifier. In this thesis we are using stochastic query-by-forest, a method proposed by Borisov et al. (2011). The following formulation of the algorithm is adapted from Schwarzfischer (2013). For implementation details, see Borisov et al. (2011).

Algorithm 1: Query-by-forest active learning

1. Build a random forest classifier (see section 2.4.2
2. For each unlabeled data sample x, compute the ensemble disagreement $q(x)$ and sort all remaining n_u unlabeled instances with respect to $q(x)$.
3. Sample the next batch from $x_1, ..., x_{\alpha n_u}$ using the sample probabilities with the following utility scores $L(x) = \frac{q(x)-q_0}{q(x_1)-q_0}$, where $q_0 = q(x_{\alpha n_u})$. By normalizing with the sum of L we get the sample probabilites $p_s(x) = \frac{L(x)}{\sum_x L(x)}$
4. Gather the labels of the newly sampled batch, rebuild the random forest and return to step 2

The ensemble disagreement $q(x)$ for a sample x is defined as the standard deviation of the weighted rare class probabilities for all trees i in the random forest ensemble among the random forest model:

$$q(x) = \mathrm{std}\left(\frac{p_{i,c}(x)/p_c}{\sum_c p_{i,c}(x)/p_c}\right), \tag{2.21}$$

where $c \in [1, 2]$ is the class label of a two-class dataset and $p_c = N_c/N$ is the counts of the target classes in the labeled data.

2.1.4 Image segmentation

In many image processing applications one is interested in informative regions of an image that can be further analyzed. An example for this could be the extraction of human faces in surveillance videos or cells on a microscope slide. The process of subdividing an image into foreground regions of interest (ROI) and background regions is called segmentation (Nixon and Aguado, 2012). The difference to the filter methods discussed in Section 2.1.2 is that the resulting image $\mathbf{D}^{\mathrm{seg}}$ is a binary representation of the original image where pixels $\mathbf{D}^{\mathrm{fg}}_{x,y}$ belonging to ROIs have the value one and all background pixels $\mathbf{D}^{\mathrm{bg}}_{x,y}$ are zero. One simple and yet very effective approach in this field is segmentation by thresholding of pixel intensities.

Otsu thresholding algorithm

The Otsu algorithm relies on the assumption that foreground regions (fg) in an image are always brighter (or darker) than the background (bg), leading to a bimodal distribution of intensity values. The algorithm tries to build two clusters of pixels - foreground and background - by finding an optimal threshold τ that minimizes the variance σ^2 of pixel intensities inside one cluster but maximizes the variance between both clusters. While computing the within-class variance of the two classes and all possible thresholds τ is computationally expensive, the author showed that maximizing the between-class variance leads to the same result (Otsu, 1975). Thus, the algorithm solves the following equation:

$$\sigma^2_{\mathrm{between}}(\tau) = \sigma^2 - \sigma^2_{\mathrm{within}}(\tau) = \omega_{\mathrm{fg}}(\tau)\omega_{\mathrm{bg}}(\tau)\big(\mu_{\mathrm{fg}}(\tau) - \mu_{\mathrm{bg}}(\tau)\big)^2 \tag{2.22}$$

where $\mu_1(\tau) = \big(\sum_0^{\tau} p(i)h(i)\big)$ represents the mean intensity of all pixels in each class, with $h(i)$ being value at the center of the i-th value of the intensity histogram. $\omega_1(\tau) = \sum_0^{\tau} p(i)$ is the class probability. μ_2 and ω_2 are computed in by using all intensities above τ. For a given intensity histogram from an image $\mathbf{D}$ with an intensity range [0,255], The Otsu algorithm performs the following steps:

Algorithm 2: Otsu thresholding algorithm

```
1. Compute probabilities $p(i)$ for all intensities $i$
2. Initilize $\omega_1(0)$, $\omega_2(0)$ and $\mu_1(0)$, $\mu_2(0)$.
3. For every threshold $\tau = [0..255]$
    • Compute $\omega_{\text{fg}}(\tau)$, $\omega_{\text{bg}}(\tau)$ and $\mu_{\text{fg}}(\tau)$, $\mu_{\text{bg}}(\tau)$.
    • Compute $\sigma^2_{\text{between}}(\tau)$.
4. Derive the value of $\tau$ where $\sigma^2_{\text{between}}(\tau)$ is maximal.
5. Output the binarized image $\mathbf{D}^{\text{seg}}$, where all pixels with intensities $< \tau$ are zero and
   all pixels with intensities $\geq \tau$ are one.
```

Maximally stable extremal regions algorithm

The Maximally Stable Extremal Regions (MSER) algorithm is a feature detector, originally designed to find informative regions (descriptors) in two images that show the same object but from different angles or distances (Matas et al., 2004). For a given image $\mathbf{D}$ the algorithm returns a list of nested extremal regions R. A region $r \in R$ in an image is a contiguous subset of pixels which are 4-neighborhood connected. Two pixels are 4-neighborhood connected if both pixels share one edge. A region r^{extremal} is an extremal region if the intensities of the pixels belonging to the boundary of the region are lower than any intensity inside of the region. The algorithm outputs those extremal regions that are maximally stable. The term maximally stable indicates that a region r_i^{extremal} is satisfying a stability criterion $q(i)$, defined as

$$q(i) = \frac{|r_{i-\Delta}^{\text{extremal}} \cdot r_{i+\Delta}^{\text{extremal}}|}{|r_i^{\text{extremal}}|} - \tag{2.23}$$

r_i^{extremal} is maximally stable if $q(i)$ has a local minimum at i. The stability of an extremal region r is the relative area variation of the region R when the intensity level is increased by the parameter Δ. More concrete, an extremal region is maximally stable if the area of the region varies only little when the intensity threshold is increased or decreased by Δ, respectively. Thus, Δ determines how big the contrast between foreground objects and background is. The algorithm performs the following steps to detect MSERs in an 8-bit image **D**:

Algorithm 3: Maximally stable extremal regions algorithm

```
1. Group all pixels in $\mathbf{D}$ in accordance to their intensities (results in 256 bins)
2. Sort all bins in descending order
3. Create an empty binary image $\mathbf{D}^{\text{temp}}$ with all pixels set to False
4. Set the pixels in $\mathbf{D}^{\text{temp}}$ that belong to the bin with intensity 1 to True
5. Save all resulting connected regions r in a list R
6. For every intensity bin [2..255]:
    (a) Set the pixels in $\mathbf{D}^{\text{temp}}$ that belong to the bin to True
    (b) Update growing regions in R
    (c) Evaluate $q(i)$ on every region $r \in R$ to find MSERs
7. Return $\mathbf{D}^{\text{seg}}$ with all MSERs
```

Watershed algorithm

The segmentation algorithms discussed above are useful to identify foreground regions of interest (ROI), however if two or more distinct objects (in our case cells) are clumped together this results in a single foreground region. To split these regions and retrieve the objects of interest, we used the watershedding algorithm as described in Meyer (1994).

Here, an image can be interpreted as a landscape, where pixels with high intensity comprise hills and pixels with low complexity are valleys. This landscape is then inverted and sources of water are placed in each valley (which is now a region of maximum intensity in the original image). The water level is raised, until water of two different sources touches each other or the boundaries of the image are reached. At these watersheds a line is drawn that separates one valley from another.

To assign a seed point (i.e. the valley floor) for the watershed algorithm to a digital image **D**, two different methods exist. The first approach uses distance transformation, where the euclidean distance to the nearest pixel whose intensity value belongs to the background is assigned to each nonzero valued pixel of a binary image. The result is inverted and watershedding is applied. A second approach is called marker based watershedding. This method enhances the fraction of pixels in the original image with maximum intensity to islands, building so-called markers. If the image is inverted the valleys where water sources are put originate in these markers. Note that the marker-based approach can also be applied by detecting ROIs in one image (e.g. stained nuclei) and use the centers of these ROIs as seed points for the watershedding in another image (e.g. the cell soma). In Meyer (1994), the algorithm for watershadding is defined as follows:

Algorithm 4: Watershed algorithm

```
1. Choose a set of seed pixels $\mathbf{D}^{\text{seed}}_{x,y}$ and uniquely label them.
2. For each seed pixel, order neighboring pixels in a priority queue in accordance to
   ther intensity value $g$.
3. Extract the pixel $\mathbf{D}^{\text{lp}}_{x,y}$ with the lowest priority level from the queue. If the
   neighbors of $\mathbf{D}^{\text{lp}}_{x,y}$ that have already been labeled all have the same label, then the
   pixel is labeled with their label.
4. Update the priority queue by adding all non-labeled pixels to the priority queue
   with the adjacent label.
5. Redo step 3 until the priority queue is empty.
```

Evaluation of segmentation quality

To evaluate if an algorithm correctly identified all foreground regions, one approach is to compare the overlap of true-valued pixels in a binary image $\mathbf{D}^{\text{seg}}$ with another binary image $\mathbf{D}^{\text{gt}}$, that represents the perfect segmentation result, i.e. ground truth. A measure for this is the jaccard coefficient, defined as the ratio of the size of the intersection and the union of the pixels in both images (Jaccard, 1912):

$$J(\mathbf{D}^{\text{seg}}, \mathbf{D}^{\text{gt}}) = \frac{|\mathbf{D}^{\text{seg}} \bigcap \mathbf{D}^{\text{gt}}|}{|\mathbf{D}^{\text{seg}} \bigcup \mathbf{D}^{\text{gt}}|}, \tag{2.24}$$

so $J(\mathbf{D}^{\text{seg}}, \mathbf{D}^{\text{gt}}) \in [0, 1]$, where a value of 0 denotes zero overlap of the segmentated regions and 1 denotes a perfect overlap.

2.1.5 Mathematical morphology operations

Depending on the quality of a given image $\mathbf{D}$, the segmented objects in a binary image $\mathbf{D}^{\text{seg}}$ often show rough or fuzzy edges, for example due to insufficient noise filtering. A way to smoothen these objects is applying filters from mathematical morphology. Typically, these algorithms take as input a digital image and a structuring element, e.g. a very small binary image with true-valued pixels in certain shapes (e.g. a disk, square or cross). We used four simple operations to postprocess the binary images in our datasets, namely erosion, dilation, opening, and closing.

Dilation and erosion work by moving a small binary image $\mathbf{S}$ (generally less then 10% of the original image size) over every one-valued pixel of the binary input image $\mathbf{D}^{\text{seg}}$ and examining the intersection between positive pixels in $\mathbf{S}$ and $\mathbf{D}^{\text{seg}}$. $\mathbf{S}$ is called a structuring element and is related to the kernel $\mathbf{K}$ in the convolution operation. The shape of $\mathbf{S}$ is determined by one-valued pixels. A typical shape and size for $\mathbf{S}$ is a diamond of 3 × 3 pixels:

$$\mathbf{K}^{\text{horizontal}} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{2.25}$$

The center of $\mathbf{S}$ is then placed on every one-valued pixel $\mathbf{D}^{\text{fg}}_{x,y}$. We define the one-valued pixels of $\mathbf{S}$ as a set A of points in euclidean space. In the same fashion, we define the set of pixels in $\mathbf{D}$ that are covered by $\mathbf{S}$ as B.

The dilation δ is then defined as

$$\delta(A, B) = \bigcup_{b \in B} A. \tag{2.26}$$

Thus, the dilation operation results in the growth of all foreground regions in dependence of the size and shape of $\mathbf{S}$. The complement to this operation is erosion ϵ. It is defined as:

$$\epsilon(A, B) = \{b | \forall a \in A, b + a \in B\}, \tag{2.27}$$

Consequently, erosion of a binary image results in a decrease of all foreground regions. By nesting the two operations in alternating order, we get morphological opening γ, defined as

$$\gamma(\mathbf{D}^{\text{seg}}) = \delta(\epsilon(\mathbf{D}^{\text{seg}})), \tag{2.28}$$

and morphological closing ϕ, defined as

$$\phi(\mathbf{D}^{\text{seg}}) = \epsilon(\delta(\mathbf{D}^{\text{seg}})). \tag{2.29}$$

These operations posses some interesting properties (Nixon and Aguado, 2012). For example, a dilation operation on a binary image will first enhance all foreground regions and thus filling e.g. holes in segmented cell objects, but also connect adjacent cells to form a single region. The erosion on the other hand will then disconnect the cells again by reducing the regions but will not change filled holes as there is no edge to work on anymore.

2.1.6 Circular hough transform

The circular hough transform (CHT) finds objects with circular shapes but incomplete boundaries in grayscale or binary images. It is thus useful to detect centers of biological cells in microscopic images, even if these cells reside in larger colonies that are difficult to separate. The detected center coordinates in turn can then be used as seed points for the watershedding algorithm (see above).

The CHT is a specialization of the Hough transform, a method to detect straight lines in images (Hough, 1959). The idea of the hough transform is to find pixels of an image $\mathbf{D}$ that are lying on the same line. A line equation in the polar system can be written as:

$$y = \big(-\frac{\cos(\theta)}{\sin(\theta)}\big)x + \big(\frac{r}{\sin(\theta)}\big), \tag{2.30}$$

and rewritten as:

$$r = x \cdot \cos(\theta) + y \cdot \sin(\theta), \tag{2.31}$$

where r and θ are the polar coordinate parameters and x and y are spatial coordinates of a pixel. The general hough transform is then defined as depicted in Algorithm 5.

To detect circles, formula 2.30 is replaced by the following definition:

$$\begin{aligned} x &= a + R\,\cos(\theta), \\ y &= b + R\,\sin(\theta), \end{aligned} \tag{2.32}$$

where a and b are the center coordinates of a circle and R its radius, that is typically known and thus fixed. It is worth noting that the implementation we used in this thesis

Algorithm 5: General Hough transform (HT)

```
1. Find edges in a digital image $\mathbf{D}$, for example by sobel filtering or thresholding,
   resulting in a binary image $\mathbf{D}^{\text{seg}}$
2. For each true-valued pixel $\mathbf{D}^{\text{seg}}_{x,y}$, compute the sinusoid such that $r > 0$ and
   $0 < \theta < 2\pi$, called hough space
3. Find intersections for curves of different pixels in the hough space (e.g. points are
   residing on the same line)
4. Construct an image $\mathbf{D}^{\text{hough}}$ showing only the lines above a threshold $\tau$
```

is capable of detecting circles in an image in a range of radii.

2.1.7 Distance transform

The distance transform is a method that is typically applied on a binary image $\mathbf{D}^{\text{seg}}$, for example after a thresholding operation (see Section 2.1.4). It assigns the distance of every foreground pixel $\mathbf{D}^{\text{fg}}_{x_1,y_1}$ (i.e. with a value of one) to its closest background pixel $\mathbf{D}^{\text{bg}}_{x_2,y_2}$ (i.e. with a value of zero). As a metric we here used the euclidean distance:

$$dist(\mathbf{D}^{\text{fg}}_{x1,y1}, \mathbf{D}^{\text{bg}}_{x2,y2}) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \tag{2.33}$$

The result of this operation is a grayscale image $\mathbf{D}^{\text{dist}}$, where every foreground pixel has as a value the distance to the center of its region.

2.2 Automated cell tracking

The methods and algorithms that we described in the previous section identify all objects (e.g. cells) in a given image. To derive temporal dynamics of the objects from a time-lapse movie, the position of every object has to be determined in every frame and then linked to a full trajectory. As the consecutive tracking of thousands of objects over hundreds of frames is tedious and labour-intensive, computerized approaches have been developed that automatize the tracking (Meijering et al., 2006, 2009, 2012).

In general, two different algorithmic concepts exist to track single cells in time-lapse experiments (Meijering et al., 2009). The first approach performs object identification and tracking in a single step by fitting a deformable mathematical model to the object of interest in the first frame of a time-lapse experiment (Zimmer et al., 2002). The model is then updated for every consecutive frame, resulting in a progression of model parameters for a full trajectory. The second approach decouples object identification and object tracking. First, objects are identified frame-by-frame, for example by image segmentation. Second, an algorithm tracks cells in all frames of a time-lapse experiment consecutively (Xinghua Lou et al., 2011). Note that in this dissertation we used a software that implemented a two-step approach (see Section 2.5.2). For a comparison of state-of-the-art cell tracking algorithms, see (Maška et al., 2014).

2.2.1 Deformable mathematical model (Single-step approach)

A typical approach for a deformable model is representing each object by a separate level-set function $\phi_c()$, $c = 1, ..., N$, where N is the number of objects (Dzyubachyk et al., 2010). The optimal segmentation in every frame is achieved by minimizing an energy functional E, which depends on image features such as the intensity gradient, but also includes criteria like the smoothness of the contour. For every pixel of a two-dimensional image $\mathbf{D}_{x,y}$, the energy functional E is evaluated:

$$E(\phi_c, ..., \phi_N) = \sum_{c=0}^{N} \int\int_{\Omega_c} - \log p(g(\mathbf{D}_{x,y})|\Omega_c)\ dxdy + a + \text{Length}(\delta\Omega), \tag{2.34}$$

where Ω_0 is the background of $\mathbf{D}$. $\Omega_c = \mathbf{D}_{x,y} : \phi_c(\mathbf{D}_{x,y}) > 0, c = 1, ..., N$ are the object regions such that $\Omega = \bigcup_{c=0}^{N} \Omega_c$ is the image domain, $\delta\Omega$ is the boundary between foreground and background regions and a is a positive parameter. $p(g(\mathbf{D}_{x,y}))$ is the conditional probability that voxel $\mathbf{D}_{x,y}$ with intensity $g(\mathbf{D}_{x,y})$ belongs to region Ω_c.

To start the tracking, the level-set functions have to be initialized by either drawing manual contours around the objects or providing an initially segmented image at frame 1. The contours are then evolved by adapting the result of a frame at time point $t - 1$ to time point t.

While the deformable model approach allows for the analysis of the object behavior over time and provides a continuous mathematical description of shape changes below pixel resolution and natively models object splitting and merging, it has several downsides. First, if the cells are moving too fast or the temporal resolution is too low, the contour of the previous time point might not overlap anymore with the object in the present time-point, leading to a premature ending of cell tracks or erroneous shapes. Also, as every cell has to be initialized separately, cells that are entering the field of view during the time-lapse experiment are difficult to add to the model. Second, the correct fit of the contour to the cell shape is heavily parameter dependent and can require a separate set of parameters for every cell in the experiment. Also, the adaptation of one contour per cell is computationally intensive, which makes the application of deformable model approaches to large amounts of cells (approximately more than 30 cells) or long time-lapse experiments (approximately more than 200 frames) difficult or necessitates the usage of powerful computation clusters.

2.2.2 Linear assignment problem (Two-step approach)

A highly cited tracking approach that decouples object detection and object linking (i.e. two-step approach) is the algorithm proposed by (Jaqaman et al., 2008). First, a set S of object positions (x, y) in every frame t is determined, for example by an LoG filter (see Section 2.1.2) or the center of a segmented object (see Section 2.1.4) (object detection step, see step 0 in Figure 2.3a). Next an optimal linking for the objects in S over the full time range has to be found, which the authors formulated as a linear assignment problem (LAP, Jonker and Volgenant (1987)). The LAP models every potential linking operation l by a cost matrix $\mathbf{M} : C \times R$. The goal is to find a combination of object links that minimizes the sum of costs for a given matrix of links $\mathbf{A} : C \times R$:

$$\hat{A}_{\text{arg min}} = \sum_{r=1}^{R} \sum_{c=1}^{C} \mathbf{A}_{r,c} \mathbf{M}_{r,c}, \tag{2.35}$$

such that

$$\sum_{r=1}^{R} \mathbf{A}_{r,c} = 1 \text{ and } \sum_{c=1}^{C} \mathbf{A}_{r,c} = 1. \tag{2.36}$$

The authors defined two separate linking operations that were both formulated as LAPs (see steps 1 and 2 in Figure 2.3a). First, every object is linked mutually exclusive to one object in its preceding and following frame (see the cost matrix $\mathbf{M}_1$ in Figure 2.3b), penalized by the cost function ℓ. If no matching object is found in the next frame, the track is stopped (cost function d). Similarly, if an object has no match in a preceding frame, this is regarded as a track start (cost function b). The cost functions d and b are inferred from the tracking information available up to the source frame t. They are defined as 1.05× the maximal cost of all previous links (see Jaqaman et al. (2008) for details). A track is stopped if all potential links of a particle in frame t to particles in frame $t+1$ had

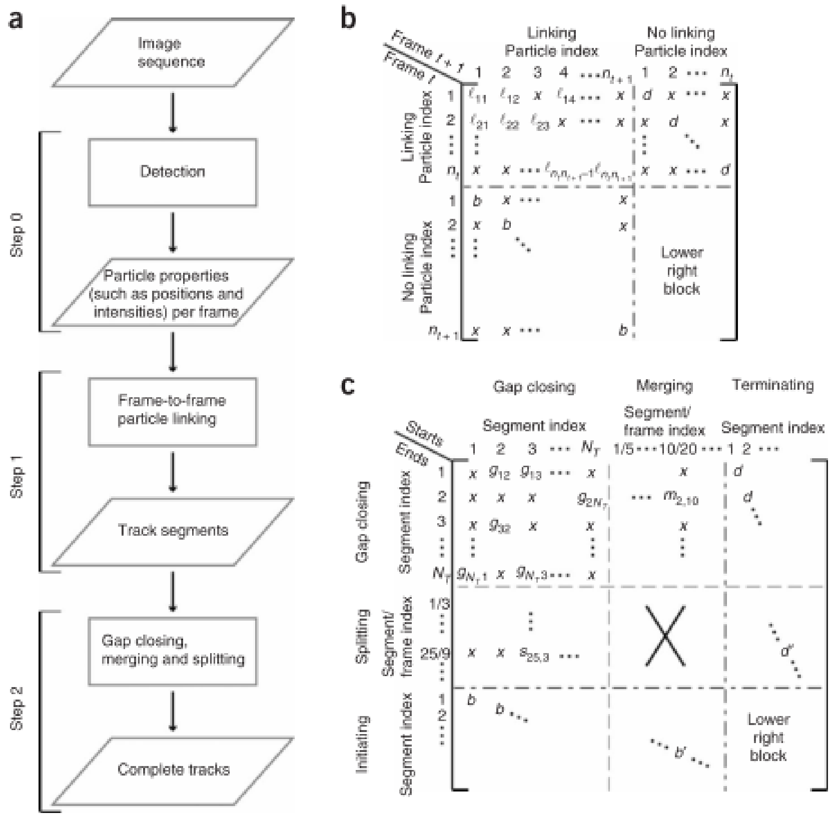


Figure 2.3: Two-step tracking approach with global optimization as proposed by Jaqaman et al. (2008). (a) Flow chart of the full procedure. Step 0: The positions for all objects are determined in every frame, e.g. by a LoG filter (see Section 2.1.2) or the center of a segmented object (see Section 2.1.4). Step 1: Every object in one frame is linked to a single object in the following frame or ended, resulting in a set of locally optimal track segments that might only be part of a full track. Step 2: To account for object splitting (or merging) events, as well as temporally missing objects (gaps in the track), the set of track segments is connected to each other by a globally optimal. Both steps are implemented as Linear assignment problems (Jonker and Volgenant, 1987). Figure taken with permission from Jaqaman et al. (2008)

costs larger than this value. Similarly, a track was started if a particle in frame t, lost all possible linkings to a particle in frame $t-1$.

This procedure results in a set of short tracks (i.e. track segments). Yet, the causes for a track end or start can not be modelled by this simple approach. For example, an object could temporally diappear or moved out of the field of view for a few frames. Also, a newly started track could result from an object splitting event. Thus in a second step, the found track segments are linked to each other by the the cost matrix $\mathbf{M}_2$ (see Figure 2.3c) with the cost functions $\ell()$ (linking), $g()$ (gap closing), $m()$ (merging) and $s()$ (splitting):

$$\ell_{rc} = \delta_{rc}^2, \tag{2.37}$$

where δ_{rc} is the distance between particles r and c,

$$g_{uv} = \delta_{uv}^2, \tag{2.38}$$

where δ_{uv} is the distance between the end of track segment u and the start of track segment v,

$$m_{RC}, s_{uv} = \begin{cases} \delta_{rc}^2 \times \rho_{uv}, \ \rho_{uv} > 1 \\ \delta_{rc}^2 \times \rho_{uv}^{-2}, \ \rho_{uv} < 1 \end{cases}, \tag{2.39}$$

where δuv is the distance between the end or start of track segment u and the middle point of track segment v,and ρ_{uv} is the ratio of the intensities $\mathbf{A}_u$ (of track segment u)and $\mathbf{A}_v$ (of track segment v) before and after merging or splitting:

$$\rho_{uv}(\text{merge in frame t}) = \frac{A_v(t)}{A_u(t-1) + A_v(t-1)}, \tag{2.40}$$

$$\rho_{uv}(\text{split in frame t}) = \frac{A_v(t)}{A_u(t-1) + A_v(t-1)}. \tag{2.41}$$

While the first linking step in this algorithm is a greedy approach that is globally optimal in space, the second linking step of acquired track segments is globally optimal in space and time. It models cell splitting events and also integrates new track starts, for example by cells entering the field of view. In contrast to the deformable model approach, the evaluation of the cost matrices is computationally less demanding than updating a level-set equation for every object, which allows the tracking of more cells and longer time periods (more than 100 cells per frame or over 500 frames are possible). The more flexible object detection step allows to apply state-of-the-art segmentation algorithms which makes the tracking of cells e.g. in brightfield images possible where level-set algorithms are hard to parameterize. One downside of this approach is however that no continuous evolution of the cell shape can be modeled, as is the case with the deformable model approach.

2.3 Quantification of cell morphology and motility

In order to derive patterns in data from high-throughput time-lapse microscopy, it is helpful to reduce the abstract information encoded in single pixels of an image to more descriptive and intuitively interpretable measures. We compiled a set of algorithms that derive features from regions of an image that were previously identified to show foreground objects (i.e. cells). These features comprise morphological information such as shape and texture of an object, but also time-dependent measures such as motility or directionality and persistence. Most of the discussed features have already been used to describe the shape of and texture of cells in microscopy experiments (Held et al., 2010; Cohen et al., 2010; Smith et al., 2009). Apart from the more recent Ray features (Smith et al., 2009), they are also implemented in image processing programs such as CellProfiler (see Section 2.5.3).

2.3.1 Basic measures

We calculated the following basic shape measures for every foreground region r in a set of regions R in a given binary image $\mathbf{D}^{\text{seg}}$. A region r is a set of spatially connected pixels $r_i, i \in \{1, 2, ..., |r|\}$ with value one in $\mathbf{D}^{\text{seg}}$. We define the convex hull c as

$$c = \left\{ \sum_{i=1}^{|r|} a_i r_i \middle| (\forall i : a_i \geq 0) \wedge \sum_{i=1}^{|r|} a_i = 1 \right\} \tag{2.42}$$

where a_i is a coefficient . Note that in the MATLAB image processing toolbox of the the convex hull we used in this dissertation, $p \in \mathbb{N}^+$ is a set of pixels that describes the smallest convex polygon that can be fitted around the region r (Mathworks, 2015).

- **Area**:
  The number of pixels in r:

$$f^{\text{A}} = \sum_{i=1}^{|r|} r_i. \tag{2.43}$$

  $f^{\text{A}} \in \mathbb{N}^+$.

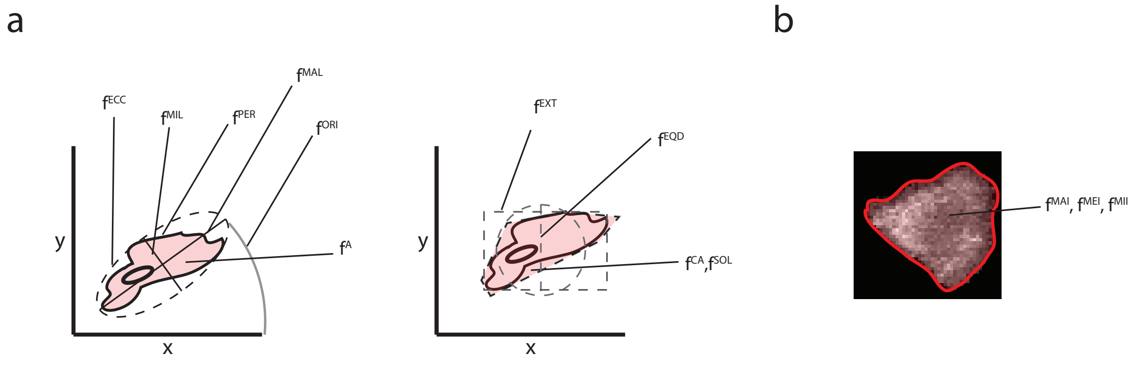


Figure 2.4: Overview of the 13 basic measures. (a) Measures computed on the binary image $\mathbf{D}^{\mathrm{seg}}$. (b) Pixel intensity statistics computed on the original image $\mathbf{D}$.

- **Convex area**:

  The number of pixels in the convex hull c.

$$f^{\mathrm{CA}} = \sum_{i=1}^{|c|} c_i. \tag{2.44}$$

  $f^{\mathrm{CA}} \in \mathbb{N}^+$.

- **Solidity**:

  The proportion of the pixels in the convex hull that are also part of r:

$$f^{\mathrm{SOL}} = \frac{f^{\mathrm{A}}}{f^{\mathrm{CA}}}. \tag{2.45}$$

  $f^{\mathrm{SOL}} \in [0, 1]$.

- **Equivalence diameter**:

  The diameter of a circle with the same amount of pixels as r:

$$f^{\mathrm{EQD}} = \frac{2}{\sqrt{\pi}} * \sqrt{f^{\mathrm{A}}}. \tag{2.46}$$

  $f^{\mathrm{EQD}} \in \mathbb{R}+^+$.

- **Extent**:

  The ratio of pixels in r to a set of pixels bb in the bounding box, which is the smallest rectangular image patch that covers the whole region r:

$$f^{\mathrm{EXT}} = \frac{|r|}{|bb|}, \tag{2.47}$$

with

$$bb = [\min_{i\in r}(r_{xi}), ..., \max_{i\in r}(r_{xi}); \min_{i\in r}(r_{yi}), ..., \max_{i\in r}(r_{yi})]. \tag{2.48}$$

$f^{\mathrm{EXT}} \in [0,1]$.

- **Major and minor axis length**:
  The length of the major (minor) axis of an ellipse fitted to r in pixels:

$$\begin{aligned} f^{\mathrm{MAL}} &= 2\sqrt{2}\sqrt{u_{xx} + u_{yy} + \sqrt{\left((u_{xx} - u_{yy})^2 + 4 * u_{xy}^2\right)}}, \\ f^{\mathrm{MIL}} &= 2\sqrt{2}\sqrt{u_{xx} + u_{yy} - \sqrt{\left((u_{xx} - u_{yy})^2 + 4 * u_{xy}^2\right)}}, \end{aligned} \tag{2.49}$$

  with

$$\begin{aligned} u_{xx} &= \frac{\sum_{i=1}^{|r|}(x_i^2)}{|r|} + 1/12, \\ u_{yy} &= \frac{\sum_{i=1}^{|r|}(y_i^2)}{|r|} + 1/12, \\ u_{xy} &= \frac{\sum_{i=1}^{|r|}(x_i y_i)}{|r|}, \end{aligned} \tag{2.50}$$

  where, x_i and y_i are the coordinates for a pixel $r_i \in r$ in x- and y-direction, respectively. $f^{\mathrm{MAL}}$and$f^{\mathrm{MIL}} \in \mathbb{N}^+$.

- **Eccentricity**:
  The ratio of the distance between the foci of an ellipse fitted to r and its major axis length f^{MAL}, approximated as:

$$f^{\mathrm{ECC}} = \sqrt{1 - \left(\frac{f^{\mathrm{MIL}}}{f^{\mathrm{MAL}}}\right)^2}. \tag{2.51}$$

$f^{\mathrm{ECC}} \in [0, 1]$, where 0 denotes a perfectly round (circular) region and 1 represents a line segment.

- **Orientation**:
  The angle between the x-axis and the major axis of the ellipse in degrees:

$$f^{\mathrm{ORI}} = \left(\frac{180}{\pi}\right) \mathrm{atan}\left(\frac{n}{d}\right), \tag{2.52}$$

with

$$\begin{aligned} n &= \begin{cases} u_{yy} - u_{xx} + \sqrt{(u_{yy} - u_{xx})^2 + 4 * u_{xy}^2} & \text{, when } u_{yy} > u_{xx} \\ 2u_{xy} & \text{else} \end{cases} \\ &\text{and} \\ d &= \begin{cases} 2u_{xy} & \text{, when } u_{xx} > u_{yy} \\ u_{xx} - u_{yy} + \sqrt{(u_{xx} - u_{yy})^2 + 4 * u_{xy}^2} & \text{else} \end{cases} . \end{aligned} \tag{2.53}$$

$f^{\mathrm{ORI}} \in [0, 360]$.

- **Perimeter**:
  The distance around the boundary of r. Given a list b of pixels that constitute the boundary of r in clockwise direction, the perimeter f^{PER} of r is defined as:

$$f^{\mathrm{PER}} = \sum_{i=1}^{|b|-1} \sqrt{s_{xi} + s_{yi}}, \tag{2.54}$$

with s_x and s_y being the lists of differences between adjacent pixels in x- and y-direction:

$$s_{xi} = b_i + 1 - b_i,\ i = \{1, 2, ..., |b|\}, \tag{2.55}$$

with s_y being computed analogously and $|s| = |b| - 1$. $f^{\mathrm{PER}} \in \mathbb{R}^+$.

In addition, we computed three features that statistically describe the intensity of a group of pixels in the original 8-bit image $\mathbf{D}$ with pixel intensities $g \in [0, 255]$ defined by the region r in $\mathbf{D}^{\mathrm{seg}}$:

- **Maximum intensity**:
  of the set of pixels belonging to r:

$$f^{\mathrm{MAI}} = \max_{r_i \in r}(g(r_i)). \tag{2.56}$$

$f^{\mathrm{MAI}} \in [0, 255]$.

- **Average intensity**:
  Of the set of pixels belonging to r:

$$f^{\mathrm{MEI}} = \frac{1}{|r|} \sum_{i=1}^{|r|} g(r_i). \tag{2.57}$$

$f^{\mathrm{MEI}} \in \mathbb{R}^+$.

- **Minimum intensity**:
  Of the set of pixels belonging to r:

$$f^{\mathrm{MII}} = \min_{r_i \in r}(g(r_i)). \tag{2.58}$$

$f^{\mathrm{MII}} \in [0, 255]$.

2.3.2 Shape

The measurements defined above are well suited for a basic analysis of identified objects in an image. However, while f^{A} and f^{ECC} provide a rough estimate about an object's shape, they do not account for a more complex morphology. For example, we can construct a circular and a star-shaped object that both have the same values for f^{A} and f^{ECC}. Thus, we introduce more sophisticated shape measures in the following paragraphs.

Ray features

The Ray feature set was introduced by Smith et al. (2009) to describe deformed or irregular shapes of objects in images. For a given digital image $\mathbf{D}$, the algorithm first detects all edges in the image, for example by a Sobel filter (see Section 2.1.2). The resulting binary image $\mathbf{D}^{\mathrm{edge}}$ is then analyzed by evaluating the function e():

$$\mathbf{e}_\theta = \mathrm{e}(\mathbf{D}^{\mathrm{edge}}, \mathbf{m}, \theta). \tag{2.59}$$

The function returns the spatial coordinates $\mathbf{e}_\theta$ of the closest pixel belonging to an edge in the image $\mathbf{D}^{\mathrm{edge}}$ to the location $\mathbf{m}$ in direction θ. Thus, $\mathbf{m}$ and $\mathbf{e}_\theta$ can be regarded as vectors that point to different locations in the image. Based on e(), the following measures can be computed:

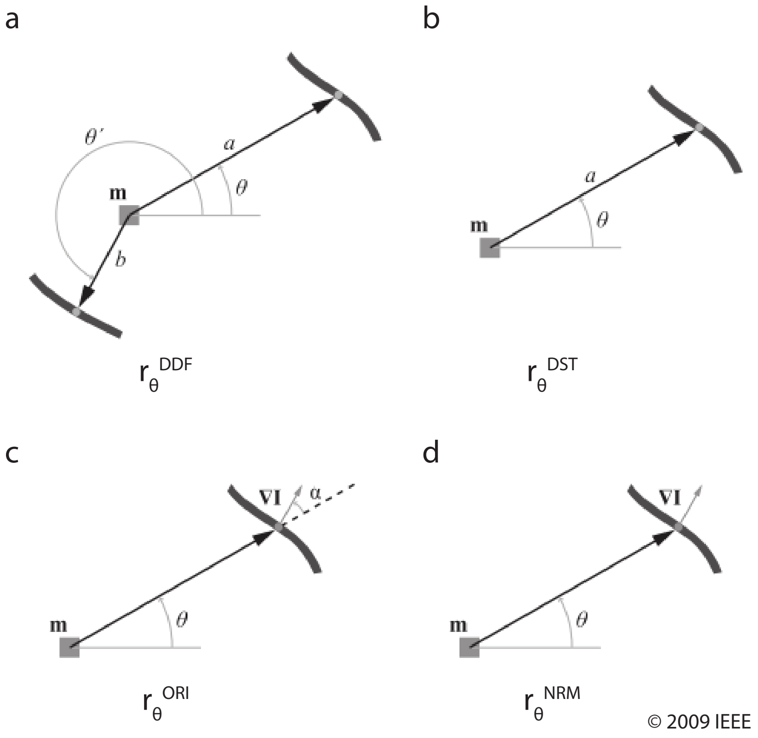


Figure 2.5: Overview of the four ray features. (a) Distance difference (b) Distance (c) Orientation (d) Norm. For mathematical details see the main text. Figure adapted from Smith et al. (2009)

- **Distance**:

$$r_\theta^{\mathrm{DST}} = ||\mathbf{e}_\theta - \mathbf{m}||, \tag{2.60}$$

where $||\cdot||$ is the euclidean vector norm. The feature computes the absolute distance to the closest edge point in orientation θ. It is not invariant to scaling an thus provides information about the object size. $r_\theta^{\mathrm{DST}} \in \mathbb{R}+$.

- **Distance difference**:

$$r_\theta^{\mathrm{DDF}} = \frac{||\mathbf{e}_\theta - \mathbf{m}|| - ||\mathbf{e}_{\theta'} - \mathbf{m}||}{||\mathbf{e}_\theta - \mathbf{m}||}, \tag{2.61}$$

where $\theta' = \theta - 120°$ and $||\cdot||$ is the euclidean vector norm. The feature compares the relative distance from a given spatial coordinate to the nearest edge in two orientations. Due to the normalization term this feature is invariant to scaling. $r_\theta^{\mathrm{DDF}} \in \mathbb{R}+$.

- **Orientation**:

$$r_\theta^{\mathrm{ORI}} = \frac{\nabla \mathbf{D}_e^{\mathrm{edge}}}{||\nabla \mathbf{D}_e^{\mathrm{edge}}||} \cdot (\cos\theta, \sin\theta)^\top, \tag{2.62}$$

  where $\nabla \mathbf{D}_e^{edge}$ denotes the gradient of $\mathbf{D}^{edge}$ at point $\mathbf{c}$. This feature analyzes the edge orientation at all angles θ and is thus useful to differentiate convex shapes from concave shapes. Thus, an averaged value close to one for r^{ORI} is an indicator for a closed convex shape (see below). $r_\theta^{ORI} \in [0, 1]$.

- **Norm**:

$$r_\theta^{\mathrm{NRM}} = ||\nabla \mathbf{D}_e^{\mathrm{edge}}||. \tag{2.63}$$

  The norm feature tests the image intensity at the specified edge point. When applied on the binary image $E(x, y)$, this feature can also be used to test the closedness of a shape. $r_\theta^{\mathrm{NRM}} \in \mathbb{R}^+$.

For a single image $\mathbf{D}$ we evaluated all formulas for $\theta \in [0, 30, ..., 330]$ with $\mathbf{m}$ being fixed to the center of the image patch. We then computed the final ray features by averaging over all values per feature, with N being the amount of different values:

$$r^* = \frac{1}{N} \sum_{\theta=0}^{330} r_\theta^*. \tag{2.64}$$

2.3.3 Texture

Basic intensity measures such as mean intensity average out patterns in the identified objects that could be useful for a later analysis. For instance, biological cells that reside in different phases of the cell cycle at the time point of image acquisition often show differences in their intensity patterns. A way to quantify these differences is to compute statistical features based on the single pixels of an object and their spatial dependencies. Two well established approaches are the feature sets described by Haralick et al. (1973) and Tamura et al. (1978). In the field of bioimage informatics, these features have already been used to e.g. classify cell types (Boland et al., 1998; Boland and Murphy, 2001; Held et al., 2010). The features are already implemented in Software for bioimage analysis such as CellProfiler (see Section 2.5.3), which increases their usability.

Haralick features

Given a grayscale image $\mathbf{D}$ with a range of intensity values $G \in [0, 255]$ (in the 8-bit case), the general idea in this approach is to quantify how often a pixel with e.g. intensity 1 is found adjacent to a pixel with e.g. intensity 2. For this purpose the authors introduced a so-called gray-tone spatial dependence matrix (or co-occurence matrix) $\mathbf{M} : G \times G$:

$$\mathbf{M}_d = \begin{pmatrix} 0,0 & 0,2 & \cdots & 0,G \\ 2,1 & 2,2 & \cdots & 2,G \\ \vdots & \vdots & \ddots & \vdots \\ G,0 & G,2 & \cdots & G,G \end{pmatrix}, \tag{2.65}$$

where d is the direction of pixel adjacency. In the two-dimensional case, $\mathbf{M}$ can be computed in four directions d (horizontal, vertical, left and right diagonal). An entry $\mathbf{M}_{d,i,j}$ of the matrix can be considered to be the probability that a pixel with intensity i is adjacent to a pixel with intensity j. $\mathbf{M}_{d,i,j}$ is derived by counting the occurences of such a pixel relationship over a given image $\mathbf{D}$ and dividing this value by the number of total comparisons in $\mathbf{M}$. The following definitions are needed before we can define the Haralick features:

- N_G: the number of values in G.
- $\mathbf{M}_{d,i,j}$: probability of finding a pixel with intensity i next to a pixel with intensity j in direction d
- $\mathbf{M}_{d,i,*}$: marginal probability for pixel with intensity i, computed as $\mathbf{M}_{d,i,*} = \sum_{j=1}^{N_G} \mathbf{M}_{d,i,j}$.
- $\mathbf{M}_{d,*,j}$: marginal probability for pixel with intensity j, computed as $\mathbf{M}_{d,j,*} = \sum_{i=1}^{N_G} \mathbf{M}_{d,i,j}$.
- $\mathbf{M}_{d,*,*,k}$: $\sum_{i=1}^{N_G} \sum_{i=0}^{N_G} \delta_{i+j,k} \mathbf{M}_{d,i,j}$, with $k = 2, 3, ..., 2N_G$.
- $\mathbf{M}^{-}_{d,*,*,k}$: $\sum_{i=0}^{N_G} \sum_{i=0}^{N_G} \delta_{|i-j|,k} \mathbf{M}_{d,i,j}$, with $k = 0, 1, ..., N_G - 1$.
  where the Kronecker delta function is defined as
  $$\delta_{m,n} = \begin{cases} 1 & \text{when } m = n \\ 0 & \text{when } m \neq n \end{cases}$$
- $H_{d,i,*} = -\sum_{i=1}^{N_G} \mathbf{M}_{d,i,*} \log(\mathbf{M}_{d,i,*})$ (entropy of $\mathbf{M}_{d,i,*}$)
- $H_{d,*,j} = -\sum_{j=1}^{N_G} \mathbf{M}_{d,*,j} \log(\mathbf{M}_{d,*,j})$ (entropy of $\mathbf{M}_{d,*,j}$)

- $H_{d,*,*} = -\sum_{i=1}^{N_G}\sum_{j=1}^{N_G}\mathbf{M}_{i,j}\log(\mathbf{M}_{i,j})$
- $H1_{d,*,*} = -\sum_{i=1}^{N_G}\sum_{j=1}^{N_G}\mathbf{M}_{i,j}\log(\mathbf{M}_{i,*}\mathbf{M}_{*,j})$
- $H2_{d,*,*} = -\sum_{i=1}^{N_G}\sum_{j=1}^{N_G}\mathbf{M}_{i,*}\mathbf{M}_{*,j}\log(\mathbf{M}_{i,*}\mathbf{M}_{*,j})$

The set of 14 statistical measures is then computed for every co-occurence matrix $\mathbf{M}_d$:

- **Angular Second Moment**

$$h_d^{ASM} = \sum_{i=1}^{N_G}\sum_{j=1}^{N_G}(\mathbf{M}_{d,i,j})^2 \tag{2.66}$$

- **Contrast**:

$$h_d^{CON} = \sum_{k=0}^{G-1} k^2\mathbf{M}^{-}_{d,*,*,k} \tag{2.67}$$

- **Correlation**

$$h_d^{COR} = \frac{1}{\sigma_{d,i,*}\sigma_{d,*,j}}\sum_{i=1}^{N_G}\sum_{j=1}^{N_G}\Big((ij)\cdot\mathbf{M}_{d,i,j} - \mu_{d,i,*}\mu_{d,*,j}\Big) \tag{2.68}$$

  where $\mu_{d,i,*}$,$\mu_{d,*,j}$,$\sigma_{d,i,*}$,$\sigma_{d,*,j}$ are the means and standard deviations of the marginal probabilities of the i-th row of $\mathbf{M}$ and the j-th column of $\mathbf{M}$.

- **Sum of Squares Variance**

$$h_d^{SSV} = \sum_{i=1}^{N_G}\sum_{j=1}^{N_G}(i-\mu)^2\mathbf{M}_{d,i,j} \tag{2.69}$$

- **Inverse Difference Moment**:

$$h_d^{IDM} = \sum_{i=1}^{N_G}\sum_{j=1}^{N_G}\frac{1}{1+(i-j)^2}\mathbf{M}_{d,i,j} \tag{2.70}$$

- **Sum Average**

$$h_d^{SAV} = \sum_{k=2}^{2N_G} k\mathbf{M}_{d,*,*,k} \tag{2.71}$$

- **Sum Entropy**

$$h_d^{SEN} = -\sum_{k=2}^{2N_G}\mathbf{M}_{d,*,*,k}\log(\mathbf{M}_{d,*,*,k}) \tag{2.72}$$

- **Sum Variance**

$$h_d^{SVA} = \sum_{k=2}^{2N_G} (k - h_d^{SEN})^2 \mathbf{M}_{d,*,*,k} \tag{2.73}$$

- **Entropy**

$$h_d^{ENT} = -\sum_{i=1}^{N_G} \sum_{j=1}^{N_G} \mathbf{M}_{d,i,j} \log(\mathbf{M}_{d,i,j}) \tag{2.74}$$

- **Difference Variance**

$$h_d^{DVA} = \sum_{k=0}^{N_G-1} k^2 \mathbf{M}^-_{d,*,*,k} \tag{2.75}$$

- **Difference Entropy**

$$h_d^{DEN} = -\sum_{k=0}^{N_G-1} \mathbf{M}^-_{d,*,*,k} \log(\mathbf{M}^-_{d,*,*,k}) \tag{2.76}$$

- **Information Measure of Correlation 1**

$$h_d^{\mathrm{IMC1}} = \frac{h_d^{ENT} - H1_{d,*,*}}{\max(H_{i,*}, H_{j,*})} \tag{2.77}$$

- **Information Measure of Correlation 2**

$$h_d^{\mathrm{IMC2}} = \big(1 - \exp\big(-2\big(H2_{d,*,*} - h_d^{E}\big)\big)\big)^{\frac{1}{2}} \tag{2.78}$$

- **Maximal Correlation Coefficient**
  h_d^{MCC} = Square root of the second largest eigenvalue of Q, where

$$Q_{d,i,j} = \sum_{k=1}^{N_G} \frac{\mathbf{M}_{d,i,k} \cdot \mathbf{M}_{d,k,j}}{\mathbf{M}_{d,i,*} \mathbf{M}_{d,*,j}} \tag{2.79}$$

Note that while the authors suggested the final measures to be computed by averaging over all values for the four co-occurence matrices, we used the implementation from CellProfiler1.0 that computes the features only for a single direction (right adjacency) to save computation time (Carpenter et al., 2006).

All Haralick features are $\in \mathbb{R}^+$.

Tamura

A substantially different approach to quantify texture in an image was undertaken by Tamura et al. (1978). The authors defined a set of features and compared their numerical description of images with the visual perception of human subjects in psychological experiments. Eventually they found three features that could resemble the classification of images from the human subjects very well. Following the definition and implementation given by Howarth and Rüger (2004), the Tamura features are defined as:

- **Coarseness**
  This feature analyses textural patterns in a given image $\mathbf{D}$ with dimensions $N \times N$ at different scales and then tries to find the largest representation of this pattern. The feature is computed by moving a squared window of size $2^K \times 2^K$ over the image and computing the average intensity of the window for each pixel $\mathbf{D}_{x,y}$ for a predefined interval of $k = [2^1, ..., 2^k, ...2^K]$:

  $$A_k = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} \mathbf{D}_{i,j} \frac{1}{2^2 k}. \tag{2.80}$$

  Next, differences between adjacent but non-overlapping windows are computed for every pixel $\mathbf{D}_{x,y}$ in $\mathbf{D}$ in both horizontal and vertical directions d:

  $$\begin{aligned} E_k^{\text{horizontal}}(\mathbf{D}_{x,y}) &= |A_k(x + 2^{k-1}, y) - A_k(x - 2^{k-1}, y)|, \\ E_k^{\text{vertical}}(\mathbf{D}_{x,y}) &= |A_k(x, y + 2^{k-1} - A_k(x, y - 2^{k-1}|. \end{aligned} \tag{2.81}$$

  At each pixel, the value of k where E is maximal is detected:

  $$s_{x,y}^{\text{texture}} = \underset{k=1...K}{\text{argmax}} \max_{d=\text{horizontal,vertical}} E_k^d(\mathbf{D}_{x,y}). \tag{2.82}$$

  The final coarseness measure $t^{\text{COR}}(\mathbf{D})$ is then computed by averaging over all optimal k for each pixel. $t^{\text{COR}} \in \mathbb{N}^+$.

  $$t^{\text{COR}} = \frac{1}{N^2} \sum_{x=1}^{N} \sum_{y=1}^{N} 2^{s_{x,y}^{texture}}. \tag{2.83}$$

- **Contrast**

  This statistic evaluates the dynamic range of gray levels in an image by measuring the mean μ and the standard deviation σ of a digital image $\mathbf{D}$ with dimensions $N \times N$. The contrast is then defined as:

$$t^{\text{CON}} = \frac{\sigma}{(\alpha_4)^{\frac{1}{4}}}, \quad \text{with } \alpha_4 = \frac{\mu_4}{\sigma^4}, \tag{2.84}$$

  where $t^{\text{CON}} \in \mathbb{R}^+$ and μ^4 is the fourth moment about the mean. It is computed as

$$\mu^4(\mathbf{D}) = \frac{1}{N^2} \sum_{x=1}^{N} \sum_{y=1}^{N} (\mathbf{D}_{x,y} - \mu)^4. \tag{2.85}$$

- **Directionality**

  The directionality analyses the presence of different directional of textures in an image $\mathbf{D}$. To calculate the diractionality in vertical and horizontal direction, the derivatives $\Delta^{\text{horizontal}}$ and Δ^{vertical} are computed by convolving $\mathbf{D}$ with the kernels K_h and K_v, respectively:

$$\begin{aligned}\Delta^{\text{horizontal}} &= \mathbf{D} * \mathbf{K}^h = \text{ with } \mathbf{K}^h = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}, \\ \Delta^{\text{vertical}} &= \mathbf{D} * \mathbf{K}^v = \text{ with } \mathbf{K}^v = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}.\end{aligned} \tag{2.86}$$

  Next, the angle θ is computed for every pixel $\mathbf{D}_{x,y}$:

$$\theta_{x,y} = \frac{\pi}{2} + tan^{-1} \frac{\Delta^{\text{vertical}}}{\Delta^{\text{horizontal}}} \tag{2.87}$$

  The values θ are then grouped into a histogram H_d of n bins (here, n=125) and the Tamura directionality $t^{\text{DIR}} \in \mathbb{R}^+$ can then be caluculated as the sum of second moments around each peak from valley to valley, normalized by the number of pixels.

2.3.4 Motility

We derived the morphological features discussed above for every frame of a time-lapse experiment separately. To observe patterns in the temporal dimension we computed an object's movement behavior (also called migration or motility) with three measures. To compute these measures we used a series of observations (e.g. a cell under the microscope that is followed over time) at time points $t \in i, ..., T$ that can be generated by e.g. an automatically tracked cell over time as described in Section 2.2.

Speed

The speed m_t^{SPE} of an object o with spatial coordinates x and y at time point t is defined as

$$m_{o,t}^{\text{SPE}} = \sqrt{(x_{o,t} - x_{o,t-1})^2 + (y_{o,t} - y_{o,t-1})^2} \frac{1}{\Delta t}, \tag{2.88}$$

where Δt is the absolute difference in time for two consecutive frames and $m_{o,0}^S$ is not defined. We computed $m_{o,t}^S$ for all pairs of adjacent frames for every object. $m_{o,t}^S \in \mathbb{R}+$. Note that the unit of $m_{o,t}^S$ can either be px/sec or $\mu m/sec.$ depending on the application and the used calibration of the experimental setup.

Direction and persistence

The direction of an object o at time point t is computed by the two-parametric arctangent of its movement vector :

$$m_{o,t}^{\text{DIR}} = \text{atan2}(x_{o,t-1} - x_{o,t}, y_{o,t} - y_{o,t-1}). \tag{2.89}$$

Similar to the mathematical artangent function, the method returns the angle of a vector that is described by a point in (x, y) in a plane. Additionally it considers the signs of the inputs in order to compute the appropriate quadrant of derived angle. Thus, $m_{o,t}^D \in [-180, 180]$.

Based on the direction, the persistence $m^{\text{PER}} \in [-180, 180]$ of an object o at time point $t + 1$ is then computed by

$$m_{o,t+1}^{\text{PER}} = m_{o,t+1}^{\text{DIR}} - m_{ot}^{\text{DIR}}. \tag{2.90}$$

2.4 Machine learning

The field of machine learning comprises the development and application of algorithms that explore datasets to find informative structures or patterns. The found information is then used to construct computational models that can classify new data points or predict future progressions of processes. In the following sections we will introduce the methods we used to process and analyze the huge amounts of data that we derived from time-lapse experiments in the different projects. If not stated differently, formal definitions and pseudocode descriptions are adapted from Hastie et al. (2009) and Maimon and Rokach (2010).

2.4.1 Unsupervised machine learning

In unsupervised machine learning, the task is to find pattern or structures in often high-dimensional datasets that have no label information (Bishop, 2007). A typical application of unsupervised learning is cluster analysis, i.e. grouping the data in clusters in accordance to some metric, e.g. the euclidean distance. In this thesis we used two different approaches of unsupervised learning, which we will discuss in the following paragraphs.

DBscan

DBscan is is a clustering algorithm that exploits the different levels of density in dataset (Ester et al., 1996). This approach leads to several advantages in comparison to other highly used clustering algorithms such as k-means. For example. DBscan does not require the user to provide the number of clusters that should be detected, as the algorithm natively detects the amount of clusters that is most likely. In addition it can detect arbitrarily shaped clusters.

The density of the dataset D in accordance to two datapoints p and q, with a distance metric $dist(p, q)$ is described by following definitions (for details see Ester et al. (1996)).

- **Eps-neighborhood:**

  $$N_{\mathrm{Eps}}(p) = q \in D | dist(p, q) \leq Eps, \tag{2.91}$$

  i.e. for every point p in a Cluster C there is a point q in C such that p resides inside of $N_{eps}(q)$. In addition, $N_{eps}(q)$ is required to contain at least $MinPts$ points.

- **Directly density reachable:**
  A point p is directly density reachable if

  $$p \in N_{\mathrm{Eps}}(q) and |N_{\mathrm{Eps}}| \geq MinPts \tag{2.92}$$

- **Density reachable:**
  A point p is density reachable from a point q if a chain $p_1, ..., p_n$ exists, while $p_1 = q, p_n = p$ such that p_{i+1} is directly density reachable from p_i

- **Density connected:**
  A point p is density connected to a point q if there is a point o such that p and q are density-reachable from o.

A cluster C is then defined by two properties:

- All points p inside C are mutually density connected
- A point q that is density reachable from any point p of C is part of C as well

For a parameterset $D, eps, MinPts$, the algorithm is then defined as:

Algorithm 6: DBscan

```
1. $C = 0$
2. For each $p$ in $D$
   (a) if $p$ was already visited, skip it
   (b) get the set of points $N$ that lie within the eps-neighborhood of $p$
   (c) if $|N| < MinPts$, mark $p$ as NOISE. Otherwise, open a new cluster $C$ and for
       each point $q$ in $N$
       • if $q$ was already visited, skip it
       • get the set of points $M$ that lie within the eps-neighborhood of $q$
       • if $|M| \geq MinPts$, join $M$ and $N$
       • if $q$ was not assigned to any cluster, add $q$ to $C$
```

Variational Bayesian approach to fit Gaussian mixture models

When analyzing a dataset of univariate or multivariate measurements, it is often unclear whether the data is structured into one or many components (i.e. subpopulations or clusters). A way to assess this question is to fit a probabilistic model to the dataset which assumes that the observed measurements emerged from one or more probability distributions. An approach that is widely used is the Gaussian mixture model (GMM).

Given a dataset D with $|D|$ observations, we assume that K components exist that are following Gaussian distributions with different parameters. The parameter vector is written as $\theta_{k=1,\dots,K} = \{w_{k=1,\dots,K}, \mu_{k=1,\dots,K}, \sigma^2_{k=1,\dots,K}\}$, where $\mu_{k=1,\dots,K}$ and $\sigma^2_{k=1,\dots,K}$ are the mean and the variance of component k and $w_{k=1\dots K}$ is the mixing coefficient of every component k, respectively. Additionally we denote a single observation as $o_{d=1,\dots,|D|}$. Finally, we define $p(o|\theta)$ as the probability distribution of o that is parametrized by θ. Thus, the GMM is defined as

$$p(o|\theta) = \sum_{k=1}^{K} w_k N(o; \mu_k, \sigma^2_k). \tag{2.93}$$

The parameters θ are typically fitted by expectation maximization (EM). Starting with an uniformly distributed prior w and a provided number of components K, the membership of every $o_{d=1,\dots,|D|}$ is adjusted. Next μ_k, σ^2_k and w_k are recomputed for every component, resulting in a shift of the distributions and thus a reassignment of every datapoint to the respective component becomes necessary. After some iterations the system converges to a local optimum, resulting in the fitted model parameters θ and w. A downside of this approach is that the number of components has to be provided a priori.

An approach to fit K together with the other parameters is to put the GMM in a Bayesian formulation. In this setting, the task is to infer the posterior distribution $p(\theta, z|x)$, where $z = \{z_{dk}\}$ is a indicator variable such that $z_{dk} = 1$ if observation o_d belongs to the k^{th} component and 0 otherwise. The model is then written as a full data likelihood:

$$p(o, z|\theta) = \prod_{d=1}^{|D|} \prod_{k=1}^{K} \{w_j N(o_d; \mu_k, \sigma^2_k)\}^{z_{ij}}, \tag{2.94}$$

As the direct inference of $p(o, z|\theta)$ is intractable, it has to be approximated. A computationally less demanding way is to approximate p by a variational distribution $q(\theta, z|o)$ (Variational Bayes, see Bishop (2007)). As this distribution factorizes over the parameters θ and z, it can be rewritten as $q_\theta(\theta|o) \times q_z(z|o)$. with the joint distributions defined as $p(o, z, \theta) = p(o, z|\theta), p(w)p(\mu|\sigma^2)p(\sigma^2)$. In addition, the priors are defined as :

$$\begin{aligned} p(w) &= \text{Dirichlet}\big(w; \alpha_1^{(0)}, \dots\alpha_k^{(0)}\big), \\ p(\mu|\sigma^2) &= \prod_{k=1}^{K} N\big(\mu_k; m_k^{(0)}, (\beta_k^{(0)}, \sigma^2_k)\big), \\ p(\sigma^2) &= \prod_{k=1}^{K} \text{Gamma}\big(\sigma^2_k; \frac{1}{2}v_k^{(0)}, \frac{1}{2}\phi_k^{(0)}\big), \end{aligned} \tag{2.95}$$

where $\alpha^{(0)}, \beta^{(0)}, m^{(0)}, v^{(0)}, \phi^{(0)}$ are predefined parameters. The posteriors are then derived as:

$$\begin{aligned} q_w(w) &= \text{Dirichlet}\big(w; \alpha_1, ...\alpha_k\big), \\ q_{\mu|\sigma^2}(\mu|\sigma^2) &= \prod_{k=1}^{K} N\big(\mu_k; m_k, (\beta_k, \sigma_k^2)\big), \\ q_{\sigma^2}(\sigma^2) &= \prod_{k=1}^{K} \text{Gamma}\big(\sigma_k^2; \frac{1}{2}v_k, \frac{1}{2}\phi_k\big), \end{aligned} \tag{2.96}$$

By using this Bayesian formulation of the GMM, redundant components are eliminated as the posterior over the component weights $w_{k=1,...,K}$ becomes sparse and only the number of components that is required to fit the data is retained. In addition, as the parameters are probability distributions itself, we receive a confidence over the component weights $w_{k=1,...,K}$ on how likely the memberships of every datapoint to a given component is. Figure 2.6 exemplifies the fitting procedure by the Variational Bayes approach. For a detailed discussion of the method, see (Bishop, 2007).

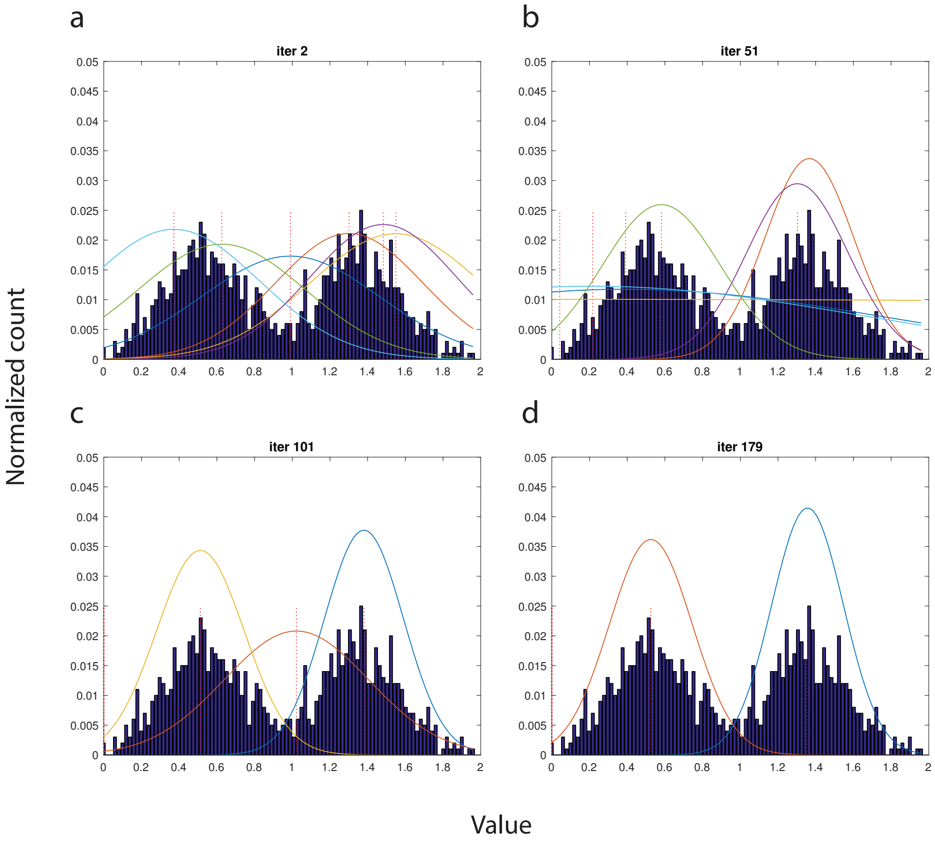


Figure 2.6: Variational Bayesian mixture of K = 6 Gaussians applied a univariate set of samples that were drawn from two different Gaussian distributions. Lines correspond to the different components $k \in K$, dotted red lines represent the mean σ_k for every component. (a) 2 iterations. The initial components are visible, i.e. Gaussian distributions with equal standard deviations but different means. (b) 51 iterations 3 of the 6 initial components are close to being dropped out because of redundancy. (c) Two components fit the data quite well, the standard deviation broadens more and more. (d) Convergence. Only two components "survived" the fitting process, all others were dropped out.

2.4.2 Supervised machine learning

The goal of supervised machine learning is to generate a computational model that is able to assign a continuous (e.g. weight of a person) or discrete (e.g. patient is sick or healthy) target output y to a given sample data point of $\mathbf{x}$. While y is a scalar with $\text{dom}(y)$ being the range of values that y can attain, $\mathbf{x}$ is a vector of N features f, i.e. numerical values that describe the sample $\mathbf{x}$ in a high-dimensional space. To achieve this goal, the model parameters are typically trained by providing a matrix of sample data points $\mathbf{X} : M \times N$ together with a vector $\mathbf{y}$ of known target outputs. The rows of $\mathbf{X}$ denote independent data points and columns represent the features. Training is then achieved by adjusting the parameters (or weights) of the model to minimize an error ϵ, that compares the predicted output $\hat{y}_i$ of the model with the known true values y_i for every training sample $\mathbf{x}_i$ with $i = 1, ..., M$. If $dom(y) \in \mathbb{R}$ we speak about a regression problem. If $dom(y)$ is an amount of discrete values we call the task classification. In this thesis we used different approaches for supervised learning, which we will describe in the following paragraphs.

Decision tree

The decision tree algorithm consists of a rooted tree-structured graph where each interior node represents a feature f_j with $j = 1, ..., N$ and the leaves correspond to the possible output values dom(y) (i.e. $y \in \mathbb{R}$ in the continuous case, and $y = \{0{,}1\}$ in the discrete binary case).

The decision tree is constructed top down (e.g. from root to leaves), at each branch (or node) of the tree a training data set $\mathbf{T} \subset \mathbf{X}$ is split into partitions $R_{1,...,|R|}$ in accordance to a single feature f_j. The feature f_j is chosen by an evaluation function that determines the homogeneity of the resulting partitions (e.g. R_1 and R_2) after the split. One measure for this is the Gini index, which computes the probability distributions of dom(y). It is defined as:

$$\text{Gini}(\mathbf{y}_T, \mathbf{T}) = 1 - \sum_{c \in dom(y)} \left(\frac{|\sigma_{y=c}\mathbf{T}|}{|\mathbf{T}|} \right)^2, \tag{2.97}$$

where $\mathbf{y}_T$ is the vector of target vectors for the samples in $\mathbf{S}$ and $|\cdot|$ is the finite cardinality. Thus, for each value $c \in dom(y)$, Gini's index computes its relative abundance. A low Gini index is indicating high variable importance and vice versa. The evaluation criterion to select an attribute a is then defined as

$$\text{GiniGain}(f_j, \mathbf{T}) = \text{Gini}(\mathbf{y}_T, \mathbf{T}) - \sum_{k \in \mathbb{R}} \frac{|\sigma_{f_j=k}\mathbf{T}|}{|\mathbf{T}|} \cdot \text{Gini}(\mathbf{y}, \sigma_{f_j=k}, \mathbf{T}) \tag{2.98}$$

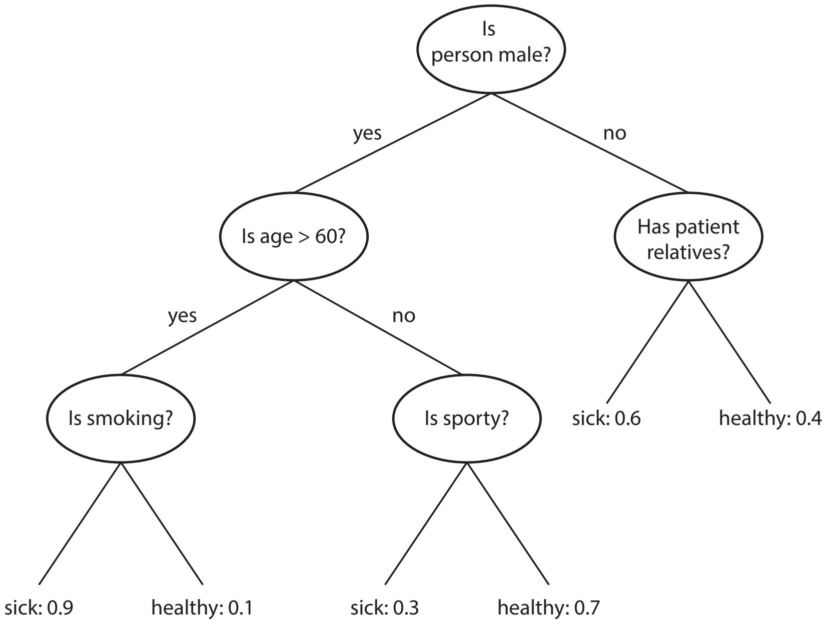


Figure 2.7: Example for a decision tree. At every branch point, the data set is split in accordance to a single feature and a certain split point. The split can be binary (male?) or numerical (age < 60?). If a leaf is reached, the model reports the predicted class (here: sick or healthy)

Ensemble learning

A challenge in the training of decision trees is to find a proper pruning strategy that prevents the overfitting of the model to the training data. One way to tackle this problem is to use ensemble learning, where many different models are trained on the data set and predictions are aggregated, which often results in improved predictive power. Two well-known methods for ensemble learning are on the one hand boosting (Schapire et al., 1998), where successive models are given an extra weight to samples incorrectly predicted by earlier predictors and in the end a weighted vote is taken for prediction. On the other hand bagging (Breiman, 1996) constructs each model independently using a bootstrap sample of the data set and a majority vote over all trees is taken to generate the final prediction score.

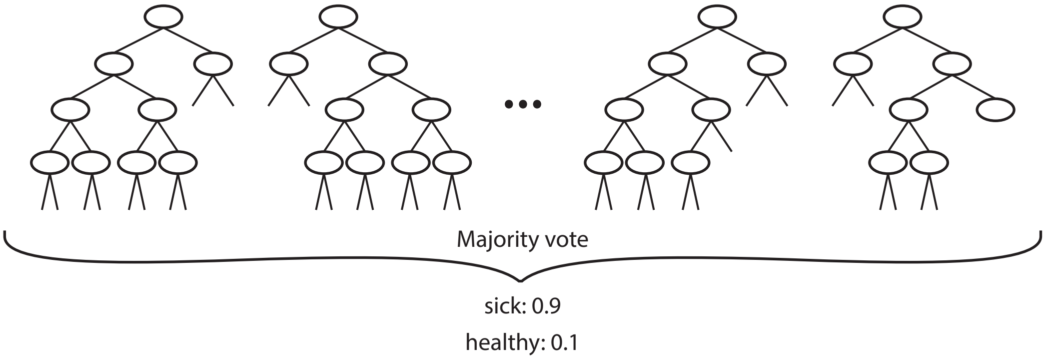


Figure 2.8: Schematized view of a random forest model. An ensemble of decision trees is constructed, where each tree is trained only on a (randomly drawn) subset of the data and feature space. Improvement in generalization (i.e. predicting unseen data samples) is achieved by taking the majority vote of all predicted class labels.

Random forest

Random forests are a variant of bagging (Breiman et al., 1984). A subset of features is randomly drawn and a decision tree is grown based on this subset. This is repeated a predefined amount of times (usually some hundred trees suffice for a good classification performance), resulting in a “forest” of decision trees, each based on a subset of the original features. Each sample in the training set is then classified by every decision tree and the most probable output is chosen by majority vote. A useful feature of random forests is the inherent computation of feature importances. These measures are derived by averaging the changes in the split criterion over the entire ensemble of grown trees. Changes in the split criterion are computed by estimates of input feature importance for every decision tree by summing changes in the risk due to splits on every feature. At each node, the risk is estimated as node impurity. This risk is weighted by the node probability. Variable importance associated with this split is computed as the difference between the risk for the parent node and the total risk for the two children. Subsequently, the given probability to which a sample is assigned a class can be used to examine borderline cases and to improve the training procedure.

Support vector machine

The support vector machine (SVM) is a mathematical model that tries to find a hyperplane in a high-dimensional space that separates the samples $\mathbf{s}_i$ in a training set $\mathbf{S} \subset \mathbf{X}$ with

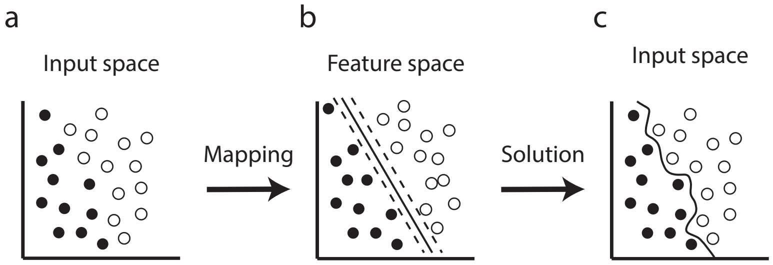


Figure 2.9: Schematized view of SVM training. (a) a dataset is not linearly separable in the feature space (here: 2 dimensions). (b) After transforming the data in a higher-dimensional feature space, for example by a radial basis function (RBF) kernel, the data becomes linearly separable. A maximum margin hyperplane is fitted with respect to some support vectors, i.e. vectors on both sides of the margin that have the closest distance to the margin. (c) After retransforming the data to the original feature space, we receive a non-linear classifier.

different target outputs y_i, where $i = 1, ..., |S|$ (Cortes and Vapnik, 1995). As most datasets are not separable linearly, the authors suggested to transform the data into an even higher-dimensional space, where linear discrimination becomes possible by applying a kernel function $\phi()$. The support vector machine then solves the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C\sum_{i=1}^{|S|} \xi_i, \\ \text{subject to} \quad & y_i(\mathbf{w}^\top \phi(\mathbf{s}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \tag{2.99}$$

Here, training vectors $\mathbf{s_i} \in \mathbf{S}, i = 1, .., |S|$ are mapped into a higher-dimensional space by the kernel $\phi()$. SVM then finds a linearly separating hyperplane with the maximal margin in this higher dimensional space by optimizing the weights $\mathbf{w}$ and the bias b, where ξ_j is a measure that denotes the distance of wrongly classified samples to the hyperplane and $C > 0$ is a penalty parameter. A kernel $\phi()$ that is used extensively and has been shown to produce reasonable results for a wide range of applications ins the radial basis function (RBF), defined as:

$$\phi_{\text{RBF}}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma ||\mathbf{x}_i - \mathbf{x}_j||^2), \tag{2.100}$$

with $\gamma > 0; 1 < i < j < |S|$ and $\phi(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$.

Convolutional neural network

The convolutional neural network (CNN) is an approach of supervised machine learning on image-based datasets that recently was improved by a row of breakthrough improvements and was thus dominating several image classification challenges (Krizhevsky et al., 2012). The principles underlying CNNs were already published 35 years ago (Fukushima, 1980), yet it took the power of modern GPU computing to unravel their full potential. Nowadays CNNs provide the benchmark models for many large-scale image classification challenges, as shown for example in Krizhevsky et al. (2012). The greatest strength of CNNs is their ability to derive feature maps directly from the raw pixel information of an image, which renders the need to derive hand-crafted features in the image obsolete. In the following we will introduce the conceptual parts of a CNN that are needed to understand its application in chapter 5. An excellent recent review which highlights the most important publications in the field is available from LeCun et al. (2015). The theory underlying convolutional neural networks can also be found in the classical multilayer perceptrons (MLP). A layer of nodes in a directed graph with several input edges but a single output edge is fully connected to another layer by a nonlinear activation function. Similar to other classification methods, a multilayer perceptron is trained by presenting the network a set of samples with known class label. The classification error ϵ of a validation set is then minimized by adjusting the weights of the edges in the graph by backpropagation Rumelhart et al. (1986).

Convolutional layer CNNs aim to classify a digital image $\mathbf{D}$ with pixels $\mathbf{D}_{x,y}$ by resembling the architecture of neurons found in human vision (Hubel and Wiesel, 1968). The idea is to reduce the huge parameter space (and thus computational burden) that arises when every pixel of an image is used as an input of a fully connected MLP by adding a locally connected layer. A convolutional layer consists of a set of nodes that apply linear filters to sub-regions of the image, which is conceptually equivalent to a convolution as defined in section 2.1.2. After adding a bias term and applying a non-linear activation function to each filter, a feature map is obtained, defined as

$$h^k_{x,y} = \psi((W^k * \mathbf{D}_{x,y}) + b_k), \tag{2.101}$$

where $k \in K$ is the number of filters, ψ is the activation function, W are the weights and b is the bias for each filter k, respectively. Note that the idea of using locally connected networks for image classification is based on the assumption that natural images are stationary, meaning that statistics of one part of the image are the same as in another part.

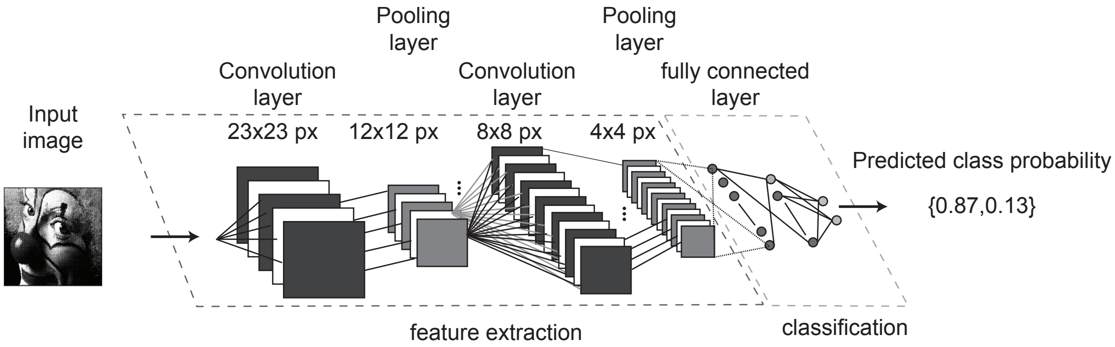


Figure 2.10: Exemplary architecture of a convolutional layered neuronal network. An input image is first passed through several alternating convolutional and max pooling layers. The layers are not fully but locally connected, which reduces the computational burden and the number of weights that have to be optimized during training. The convolutional layers lead to an abstract representation of the original image. This procedure takes local textural patterns into account and the pooling layers also introduce some sort of shift invariance. After 7 or 8 convolutional or pooling layers, the resulting filters are fed in a conventional neuronal network with fully connected layers. The nework then reports a classification score (here: binary).

Thus, locally learned features are valid througout the whole image.

Pooling layer The number of features after running an image through a convolutional layer is still too large to train a deep network. For example, for an image with size 128×128 px and 500 learned filters over sub-regions of size 10×10 px, the number of features for single filter after convolution would be $(128 - 10 + 1) \cdot (128 - 10 + 1) = 14161$. This results in a total of $14161 * 500 = 7080500$ features. However, due to the stationarity property of natural images it is possible to divide the image into non-overlapping tiles of predefined size, and apply simple statistics such as averaging or taking the maximum. In addition to a great reduction of features that are propagated through the network, pooling provides a form of translational invariance, which is beneficial to the classification performance of the final network (Zeiler and Fergus, 2013).

Activation function The activation function of every node in a neural network decides if this node produces output ("fires") or not. Using non-linear functions allows to model complex relationshops in a given data set. The classical activation functions used in neural networks are the sigmoid or hyperbolic tangent function:

$$\begin{aligned} \text{Sigmoid: } f(x) &= \frac{1}{1+\exp(x)}, \\ \text{Hyperbolic tangent: } f(x) &= \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \end{aligned} \tag{2.102}$$

Recently, Nair and Hinton (2010) showed that another activation function, called rectified linear function, solves some problems of the sigmoind and tanh functions such as vanishing gradient (not mentioned here) and in addition allows to efficiently train deep networks without pre-training the weights. It is defined as:

$$f(x) = \max(0, x). \tag{2.103}$$

Dropout To train a model that performs well on a given test set, ensemble methods as discussed in section 2.4.2 can largely improve the prediction performance. However, the complex and long-lasting training procedure of even a single CNN (depending on the size of the dataset and the used architecture but usually between many hours and several days) prohibits the simultaneous training of hundreds of these models. A very efficient workaround of this problem is called dropout and was first formulated by Hinton et al. (2012). The algorithm works by setting the weight of randomly chosen nodes to zero during the training process. This leads to a different sampling of the network architecture every time a training input is presented, but the network is still trained as a whole. This technique is comparable to ensemble learning, but the training time is much faster as only one model hast to be learned.

Stochastic Gradient Descent Stochastic Gradient Descent (SGD) is a technique to effectively train a deep neural network. As described in Bottou (2010), SGD follows the same principles as standard gradient descent while being drastically simplified. Given the tuple z of a data sample x and its true class label y, $z = (x, y)$, a loss function $l(\hat{y}, y)$ evaluates whether the prediction corresponds to the true label y. The task is to find a function $f_w(x)$ with the weight vector w parameterizing f, thus minimizing the loss $Q(z, w) = l(f_w(x), y)$, averaged on all samples x. The SGD algoirhtm is then written as

$$w_{t+1} = w_t - \gamma_t \nabla_w Q(z_t, w_t), \tag{2.104}$$

where z_t is a randomly picked data sample. The main hyper-parameter for gradient descent based methods is the learning rate γ. This factor determines the step size that is taken in the direction of the gradient. Unfortunately, this parameter is highly problem dependent and needs to be adjusted during training. The later we are in the trainings process, the smaller the adjustments to the parameters should be. This can be solved by introducing an additional hyper-parameter for momentum α, which controls the decay of the learning rate after each iteration through the training set (also referred to as epoch). It is given by

$$\begin{aligned} v &= \alpha v + \gamma \nabla_w Q(z_t, w_t) \\ w &= w - v \end{aligned} \tag{2.105}$$

where v is the velocity vector with same dimensions as the parameter vector w.

2.4.3 Model evaluation

Evaluation methods for classification models

In supervised machine learning, a model performs well if it correctly predicts the class labels most samples that were not part of the training set. In applications with binary class labels (e.g. positive and negative), several basic measures can be derived from the prediction of a model. Positive samples that are correctly classified as such are called true positives (TP), whereas correctly classified negative samples are called true negatives (TN). Furthermore, falsely classified positive samples are called false negatives (FN), and falsely classified negative samples are called false positives (FP). Based on these definitions the following measures can be computed:

- **Precision**:
  The precision measures the portion of the classified samples that were correct. It falls in the range from [0, 1], with 1 being the best score. Precision is denoted as

  $$\text{prec} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.106}$$

- **Recall / true positive rate**:
  Recall is a measure to determine the portion of the samples were assigned the correct class. It falls in the range [0, 1], with 1 being the best score. Recall is denoted as

  $$\text{rec} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.107}$$

- **False positive rate**:

$$\text{rec} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{2.108}$$

- **Specificity / true negative rate**: The specificity determines the fraction of correctly identified negative samples

$$spec = \frac{\text{TN}}{\text{TN} + \text{FN}} \tag{2.109}$$

- **F1-measure**:
The F_1-measure combines precision and recall in a single score. Its values are fall within the interval [0, 1], where 1 denotes a perfect classifier and 0.5 stands for random guessing. The F_1-measure is denoted as

$$\text{F}_1 = 2 \cdot \frac{\text{prec} \cdot \text{rec}}{\text{prec} + \text{rec}} \tag{2.110}$$

- **Macro- and micro averaged $\mathbf{F_1}$-measure**:
In multi-class classification tasks, the measures described above only give information about the performance on one class (normally the positive). In order to obtain a single measure that accounts for performance on both classes, it is necessary to apply macro and micro statistics. For example, the F_1-measure can be calculated for each class and then averaged, which is called macro-averaging. In contrast, the true positives, false positives and false negatives can be summed up for all classes and precision, recall and F_1-measure are calculated afterwards. For a set of class labels C = c1, ..., cn, the micro-averaged F_1-measure is defined as:

$$\begin{aligned}
\text{TP}' &= \sum_{i=1}^{|C|} \text{TP}(c_i), \\
\text{FP}' &= \sum_{i=1}^{|C|} \text{FP}(c_i), \\
\text{prec}' &= \frac{T'}{\text{TP}' + \text{FN}'}, \\
\text{rec}' &= \frac{\text{TP}'}{\text{TP}' + \text{FN}'}, \\
\text{micro}_{F_1} &= 2 \cdot \frac{\text{prec}' \cdot \text{rec}'}{\text{prec}' + \text{rec}'}
\end{aligned} \tag{2.111}$$

and the macro-averaged F_1-measure is defined as the mean of all class labels C_i:

$$\text{macro}_{F_1} = \frac{1}{|C|} \sum_{i=1}^{|C|} F_1(c_i). \tag{2.112}$$

The two procedures bias the results differently - micro-averaging tends to overemphasize the performance on the class with the most samples, while macro-averaging over-emphasizes the performance on the class with the fewest samples.

- **Receiver-operator characteristic and AUC**:
  For classification methods that return a continuous score instead of a categorical class label, the receiver operating characteristic (ROC) can be used to determine the best tradeoff between the true positive rate (TPR) and the false positive rate (FPR). The ROC-analysis is carried out by analyzing all possible thresholds of the classification score after which a sample is regarded as the positive or the negative class. The analysis can be visualized by plotting the resulting TPR and FPR against each other. The resulting curve can be used in two ways. First, the point on the curve where the TPR is highest and the FPR is lowest is the optimal threshold for the classifier. Second, by analyzing the area under the curve (AUC), we obtain an easily interpretable value in the interval of $[0, 1]$ that represents how well the classifier performs overall. An AUC of 0.5 represents random guessing, whereas a AUC of higher than 0.8 can be regarded as a well performing model.

Evaluation methods for regression models

While classification models can be evaluated by doing statistics over the correctly and erroneously predicted values, the continuous target variable in a regression model renders different approaches necessary. Similar to the evaluation of a fitted curve, one typically assesses the residual error of the function, or even the correlation of the predicted values to the true values. In this thesis we used the RMSE and the R^2 score to evaluate regression models.

- **Root mean squared error**:
  A very popular measure to evaluate the error rate of a regression model is the root

mean squared error (RMSE). It is defined as

$$RMSE = \sqrt{\frac{\sum\limits_{i=1}^{n}(p_i - l_i)^2}{n}}, \tag{2.113}$$

where l_i is the true value and p_i is the predicted value for a sample i. The disadvantage of this measure is that it is only comparable between different regression models that wer fitted to the same data. It does not provide a general idea if the model works well or not.

- **$\mathbf{R^2}$ score**:
  The R^2 score reports the percentage of variance that is explained by the trained regression model. A value of one denotes a perfect fit and smaller values denote worse fits. It is defined as

  $$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum\limits_{i}(p_i - \bar{p})^2}{\sum\limits_{i}(y_i - \bar{p})^2}, \tag{2.114}$$

  where l_i is the true value and p_i is the predicted value for a sample i. $\bar{p}$ is the mean of the observed data, denoted as $\bar{p} = \frac{1}{n}\sum\limits_{i=1}^{n} p_i$.

Cross-validation

Cross-validation is a popular method to evaluate the generalization performance of a trained model (Devijver and Kittler, 1982). The data set is split into k stratified batches of equal size and $k-1$ batches are used to train the model. The performance measures are then evaluated on the remaining batch that the model has not seen before. This procedure is repeated k times in a round-robin fashion. By doing statistics on the computed performance measures for every fold, we can see if our model is well trained or overfitted to the training data. For classification methods that utilize hyperparameters such as SVMs, Hastie et al. (2009) pointed out that two rounds of cross-validation have to be carried out in order to prevent overfitting:

The error estimates from step 2c are then accumulated over all K folds to produce the cross-validation estimate of prediction error.

Algorithm 7: K-fold cross validation

```
1. Divide the samples into K cross-validation folds at random
2. For each fold $k = 1, 2, ..., K$
    (a) Find a subset of "good" predictors that show fairly strong (univariate)
        correlation with the class labels, using all of the samples except those in fold k.
    (b) Using just this subset of predictors, build a multivariate classifier, using all of
        the samples except those in fold k.
    (c) Use the classifier to predict the class labels for the samples in fold k.
```

2.5 Software tools

In the following we list the tools and frameworks that we used throughout the course of this thesis.

2.5.1 ImageJ / FIJI

ImageJ is a widely used software for the analysis of biological images (Rasband, 2012). It features a multi-stack imageviewer that enables the visulization of datasets with up to 5 dimensions (x × y × z × time × color channels), also as a 3D rendering. In addition, a multitude of plugins are available, while new plugins are actively developed by a huge open-source community. ImageJ also has frameworks and wrappers for most scripting languages, e.g. Python, that enables batch scripting.

FIJI is a distribution of ImageJ that was specifically designed for the analysis of biological image-based datasets (Schindelin et al., 2012). It features the core ImageJ routines along with a set of powerful and carefully tested features for image processing, registration, segmentation and noise reduction.

ImageJ and FIJI are written in Java and are fully open source.

2.5.2 Trackmate

Building on the algorithm developed by Jaqaman et al. (2008), Trackmate provides a powerful open-source framework to automatically track single cells in time-lapse microscopy experiments. It is capable of modeling cell division events, a feature that most other tools for automatic tracking are lacking. Furthermore, Trackmate features a graphical user interface that allows to easily filter erroneous tracks or to manually correct them. Being a plugin for ImageJ, it is possible to script full pipelines that involve sophisticated image preprocessing steps followed by the autoatracking in a single application. Also, its xml-based data format is easy to parse in order to further analyse or process the results from autotracking in programs other than FIJI.

2.5.3 CellProfiler

CellProfiler is program to batch-process and analyze data from high-throughput imaging experiments citepCarpenter2006. It consists of GUI where the user can easily build a pipeline by choosin from a list of available modules. The modules cover the full range of a bioimage informatic analysis, yet its modular structure and predefined parameter sets for every module render more sophisticated or specialized analyses difficult.

CellProfiler is written in Python and fully open-source.

2.5.4 Matlab

Matlab is a scripting language that is based on Java and thus platform independent (Mathworks, 2015). Being used extensively in the fields of engineering, computer vision and machine learning, many recently published methods come with Matlab implementations. Matlab's greatest strength is numerical computing and matrix manipulations. Powerful plotting functions, toolboxes for image processing and statistical analysis and the interactive GUI make it a perfect tool for rapid prototyping.

2.5.5 Background Correction Tool

The method for background estimation and correction described in sections 2.1.3 and 2.1.3 is implemented as a Matlab tool featuring a full graphical user interface. For details please see the dissertation of Schwarzfischer (2013).

2.5.6 TTT

TTT is a program to manually follow (i.e. track) single cells and their progeny in time-lapse microscpy experiments (Rieger et al., 2009). The key feature of TTT is its feature-rich GUI and the strong code base, that allows to track full genealogies of e.g. differentiating (and thus proliferating) stem cells over long time spans of at least 14 days. As these genealogies can comprise tens of thousands of cells, most other available tracking programs fail in this task. TTT is closed-source and written in C++.

Chapter 3

Quantitative analysis of T-lymphocyte migration modes

In this chapter we describe the development of an automated pipeline to quantify and analyze the migration behavior of T-lymphocytes in time-lapse experiments of a murine model cell line. These cells are robust against photo-toxicity, which allowed the acquisition of fluorescence images of both nucleus and cell soma at every frame. This made it possible to use available state-of-the-art algorithms for cell detection and autotracking. In addition, the cells did not divide throughout the period of observation. Thus the challenge of our automated analysis was to follow highly motile non-dividing T-lymphocytes in multiple time-lapse experiments and statistically analyze their morphodynamics over time. We conducted the project in close collaboration with Dr. Anne Reversat and Prof. Dr. Michael Sixt from IST Austria. A manuscript is currently prepared for publication. It is entitled "A. Reversat*, **F. Buggenthin***, J. Merrin, A. Leithner, I. de Vries, F. J. Theis, C. Marr and M. Sixt. Morphodynamic analysis of T lymphocyte migration in confined microenvironments."

We identified four key modules that the pipeline had to satisfy and could not be established with standard software in order to derive the necessary information: (i) Detection of cell somata and nuclei in separate channels and for every frame of the experiment, (ii) automatic tracking of single cells over the full experimental time span, (iii) quantification of a set of interpretable features that describe cell morphology and motility, and (iv) quantification of actin expression inside the cell soma. The pipeline should show high robustness for a fixed parameter set that can be chosen once for an experimental setup. Furthermore, it should have the ability to manually assess automatically tracked

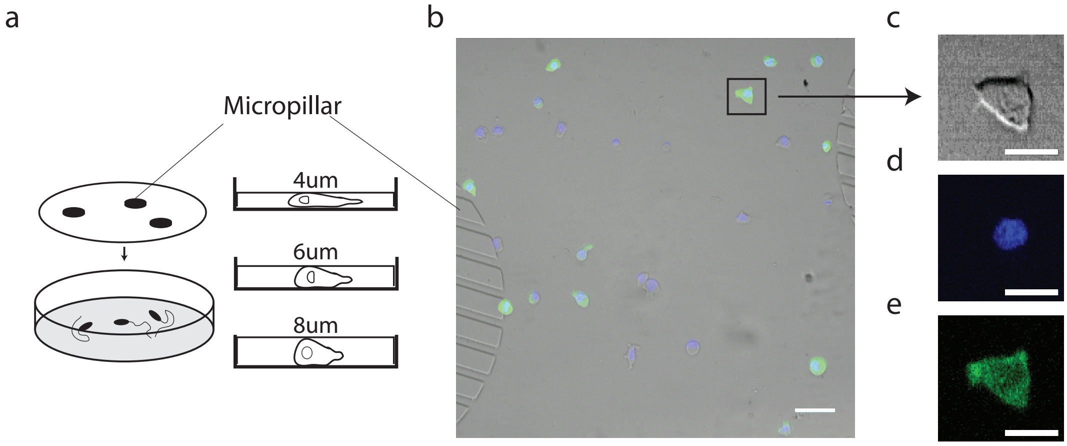


Figure 3.1: Experimental setup and data structure. (a) A micropillar of different heights is confining the cells on the coverslip. (b) Example of a single frame of size 1000x1000 px with 3 color channels. Scale bar respresents 40 µm. (c,d,e) Cutouts from (b) showing a single cell in the brightfield (c), Hoechst405 (d) and LifeAct-GFP channel (e). Scale bars represent 15 µm.

cell trajectories and correct or discard them if necessary. As the experimental setup will be changed throughout the experiments to acquire best results for different conditions, the pipeline should be easily adaptable to changes in the experimental setup to allow for the analysis of new datasets.

3.1 Experimental setup and used datasets

We developed our pipeline based on a set of image stacks from fluorescence time-lapse microscopy experiments. Here, a T-cell lymphoma cell line was used to model wild type lymphocytes (Malherbe et al., 2000). The cells were transfected with LifeAct-GFP, to reveal filamentous actin, and stably selected. Before each experiment, the nucleus of every cell was stained with Hoechst405 (Invitrogen).

The cells were plated out on a customized coverslip with with either adhesive coating (Fibronectin, BSA) or non-adhesive coating (PLL-PEG). Fibronectin is adhesive via Integrins and other molecules present on T cells, BSA binds non-specifically to those molecules. A micropillar with differing heights of $8, 6$ and 4 µm was used to mimic confinement *in vitro* as described in Le Berre et al. (2014) (see Figure 3.1a). Cells were imaged in intervals of 30 seconds for 60 minutes with a time-lapse light microscope (Nikon) using a

Experiment	Condition	Confinement (µm)	Replica	Seeding Density
140609	Fibronectin	8	1	low
140609	Fibronectin	8	2	low
140609	Fibronectin	6	1	low
140609	Fibronectin	6	2	low
140609	Fibronectin	4	1	low
140609	Fibronectin	4	2	low
150429	BSA	6	1	high
150429	BSA	6	2	high
150429	PEG/BSA	6	1	high
150429	PEG/BSA	6	2	high

Table 3.1: Overview of the full dataset used in this study. Low seeding density was 22 ± 8.7 cells per 0.4 mm^2 and high seeding density was 59.5 ± 12.87 cells per 0.4 mm^2, respectively.

20x objective. One bright-field image and two fluorescence images for Hoechst and GFP staining were acquired, respectively. An image stack consisted of 121 frames (1000×1000 px, 1 px = 0.4 µm^2) in 16-bit format, for every color channel (see Figure 3.1b-d). All time-lapse experiments were conducted by Dr. Anne Reversat, Institute of Science and Technology Austria (IST Austria).

The full dataset comprised 11 image stacks from two experiments, one with a low seeding density (22 ± 8.7 cells per 0.4 mm^2) and one with a high seeding density (59.5 ± 12.87 cells per 0.4 mm^2) with differing conditions (see Table 3.1 for details). Processing of all image stacks took a total of 3 hours. The segmentation and tracking modules were executed sequentially on a standard laptop (Core i7-3520M @ 2.90 GHz, 12GB RAM, 250GB SSD), while the quantification module was executed as parallellized jobs for single cells on a computation cluster (the average node architecture was equal to an Intel Xeon 2GHz, 4GB RAM running a 64bit linux-based operating system).

3.2 A pipeline for automatic quantification and analysis of T-lymphocyte morphology and motility

To process images from fluorescence time-lapse experiments automatically and robustly, several design decisions have to be taken that typically are a tradeoff between measurement precision, computation speed and the amount of manual adjustment. For example, model-based methods to detect objects in images such as active contours and level set algorithms are able to describe a cells shape with a precision below pixel resolution, but are also highly parameter dependent and computationally demanding (Meijering, 2012). The same class of algorithms can be used for accurate and automatic tracking, but typically only a few cells can be followed at the same time because of the computational burden (Ambühl et al., 2011) (see Section 2.2). In addition, an autotracking method that conducts cell identification and tracking in a single step is not well suited for parallelization. As in time-lapse microscopy experiments many thousand images have to be processed in reasonable time with as less user interaction as possible, it is necessary to devise a robust and modular pipeline architecture where every part can be executed automatically and in parallel.

In this project we chose algorithms for the different modules of our pipeline with respect to these demands. A schematic overview of the pipeline is depicted in Figure 3.2. In a first step, we preprocessed every frame in every channel of the image stack to filter noise and correct illumination differences. Next, we separated the channels to detect cell and nuclei outlines separately. Simultaneously, we followed the nuclei over time by an autotracking approach to create single cell trajectories. We then used the detected outlines in every channel together with the tracking information to derive a set of 40 features that describe the shape (f^* and r^* measures, see Section 2.3), as well as texture (t^* and f^* measures) and motility (m^* measures) of every cell. We will discuss the steps executed in every module in the following paragraphs. For details about the used algorithms and mathematical concepts, see Chapter 2.

3.2.1 Image preprocessing

In many time-lapse experiments the cells are transfected with fluorescent compounds such as green fluorescent protein (GFP) to label structures of interest. The varying efficiency of the transfection protocol leads to heterogeneous levels of marker expression in individual cells that render the robust identification of e.g. nuclei difficult. Thus, for a given fluorescence image stack, we adaptively enhanced the contrast of every image frame and every channel by histogram equalization such that 0.4 % of the pixels were saturated, resulting in an increase of overall contrast and the intensity of very dim cells. Next, we applied a median filter with a neighborhood $\mathbf{O}$ of size 5×5 to remove noise from the images (for details see Section 2.1.1 and Section 2.1.2).

3.2.2 Autotracking and optional manual correction

To retain a high modularity and robustness of our pipeline, we decided to use a two-step autotracking approach as provided by the ImageJ plugin Trackmate (Jaqaman et al., 2008;

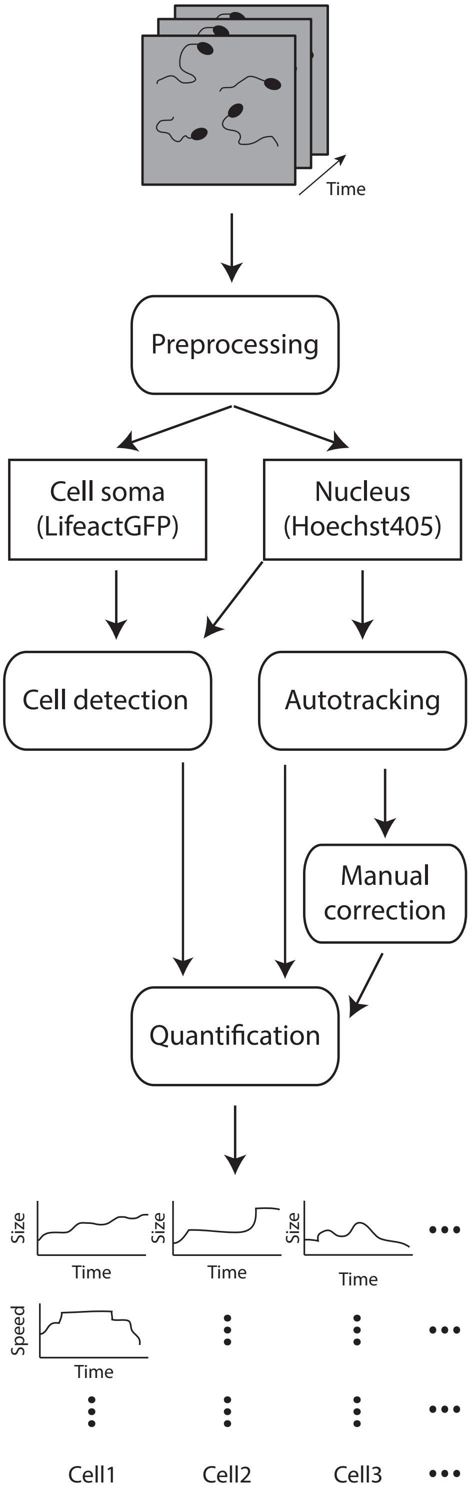


Figure 3.2: Schematic view of the developed automated analysis pipeline. The pipeline takes a 4-dimensional image stack (2 fluorescence channels, 121 time points) as input. First, every channel of the stack is preprocessed to filter noise and correct illumination differences. Next, the channels are separated and outlines for cells and nuclei are detected by segmentation. Using the images of observed nuclei, all cell centers are automatically detected and then tracked over time. If needed, the resulting cell trajectories can then be manually assessed and corrected. Eventually, a set of 40 features that describe the shape (f^* and r^* measures, see Section 2.3), as well as texture (t^* and f^* measures) and motility (m^* measures) of every cell over time are derived.

Schindelin et al., 2012). The method first detects the centers of all nuclei in fluorescence images by a Laplace of Gaussian (LoG) filter (see Section 2.1.2). Next, a tracking algorithm that is able to account for missing time points and cell divisions is applied (see Section 2.2). The algorithm first links objects between adjacent frames by a locally optimal approach, thereby creating several smaller track segments. Next, the segments are evaluated by a global optimization procedure. Given a cost matrix that compares possible gap closures (i.e. merging of two track segments), splitting events (i.e. cell division) or termination (i.e. cell death or loss), the configuration of all tracks that minimize the total cost is derived. The full procedure can be executed as a batch job without user interaction. Still, all tracks can be manually evaluated by an easy to use graphical user interface, allowing to delete erroneous tracks or even to reassign single frames or whole track segments.

3.2.3 Cell detection

We separated nuclei and cell somata in the respective fluorescence images from the background by applying the thresholding-based segmentation algorithm maximally stable extremal regions (MSER) for all images in the respective channel (Matas et al., 2004). We applied mathematical closing to fill holes in segmented foreground objects and filtered objects with too small areas (typically smaller than 100 px). The resulting binary image showed the masks of all foreground objects. To split clumped cells we used the coordinates that were detected by the LoG filter operation in the autotracking step as seeds for a watershedding operation.

Quantification of morphology and motility

By combining the spatial coordinates from autotracking with the detected outlines of nuclei and cell somata we built image patches of a fixed size with mass centered cell nucleus and soma in the middle of the patch, respectively. We then quantified the morphology of the object for both fluorescence channels by a set of 92 features describing shape and texture. The set included basic measurements such as a cell's area or its eccentricity, but also more sophisticated shape descriptors such as Ray features (Smith et al., 2009). To describe textural patterns, we derived the simple pixel intensity statistics, as well as the feature sets defined by Haralick (Haralick et al., 1973) and Tamura (Tamura et al., 1978) (21 textural features in total). To describe the motility of a cell we computed its speed, the direction and its persistence for all frames. For a detailed discussion of all features, see Chapter 2.

Implementation

We implemented the segmentation and quantification modules of the pipeline in Matlab and the autotracking in Java. The full pipeline is executed by a single Matlab script where all parameters can be set. A batch script then scans a given folder for all image stacks and automatically executes the ImageJ plugin Trackmate that conducts the autotracking. Afterwards, the resulting XML file bearing the tracking information is converted into a Matlab structure. In the mean time, every channel is separately segmented in Matlab. Together with the tracking information, the morphology and motility features are then quantified, where every cell is processed in parallel on a computing cluster. Eventually the results are parsed in a Matlab structure that eases further statistical analyses.

3.2.4 Pipeline parameters

We used the following parameters in the different modules of our pipeline to process the dataset:

- Preprocessing

Parameter	Value
Percentage of saturated pixels	4%

- Autotracking

Parameter	Value
Radius	5px
Intensity threshold	250
Median filtering	False
Subpixel accuracy	False
Allow track splitting	True
Allow track merging	False
Maximum splitting distance	15 px
Maximum frame gap	12 frames
Maximum gap closing distance	20 px
Maximum linking distance	15 px

- Cell detection
  Due to the different object sizes and fluorescence intensities, we used different parameter sets for the object detection module to segment cell nuclei and somata in the respective channels:

Parameter	Value (Nuclei)	Value (Somata)
Delta	2	1
Minimum Object Size	100 px	20 px
Maximum Object Size	5000 px	2000 px
Maximum Variation	1	1

- Quantification
  No special parameters were necessary in this module. For implementation details of the measured features, see Section 2.3.

It is worth noting that we used the same set of parameters for all image stacks, illustrating the robustness of our pipeline.

3.2.5 Evaluation

Depending on the application and the desired readout, a method that automatically detects and tracks cells in time-lapse experiments can be evaluated by different measures (Chenouard et al., 2014; Maška et al., 2014). For example, a simple analysis of cell motility does not require the method to derive accurate cell shapes. On the other hand, if one is only interested in cell shape dynamics but not in detailed motility statistics the pipeline does not have to detect exact cell centers. In this project we were interested in both, accurate cell motility and shape measurements of single cells over time. Thus we decided to evaluate the pipeline in both ways, but did not account for missing measurements in single time points of a cell track, as we were able to interpolate or discard these errors in a later curation step.

To evaluate an autotracking method, one typically computes the disagreement to a ground truth by a globally optimal overlap between both track sets. As the dataset in this project was lacking a proper ground truth, we manually assessed every autotrack and decided if it was correct or showing errors. We marked an autotrack as wrong if one or more of the following criteria applied: (i) Premature track end due to loss of cell, (ii) misassignment of a single frame, leading to a crossing of two cell tracks, (iii) larger jumps in adjacent frames due to erroneous cell detections and (iv) falsely detected splitting events due to debris or touching cells. Our method generated smooth and continuous autotracks for image stacks with low (see Figure 3.3a) and high seeding densities (see Figure 3.3b). The manual assessment of every cell in the dataset revealed that 95 % and 90 % of all tracks were correctly assigned in stacks with low and high seeding densities, respectively.

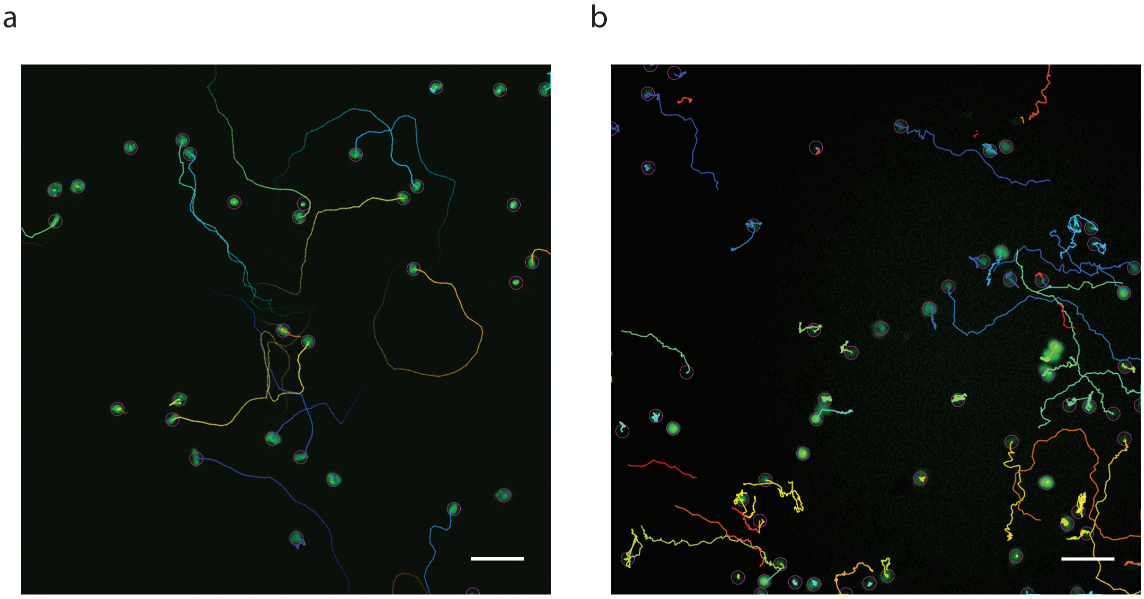


Figure 3.3: Autotracking shows high accuracies on image stacks with two different seeding densities. (a,b) Visualization of tracked single cells over the full time span (121 frames, 30 seconds intervals, 60 minutes duration) for low seeding density (22 ± 8.7 cells per 0.4mm^2) and a high seeding density (59.5 ± 12.87 cells per 0.4 mm^2).

In a next step we assessed how accurate our pipeline detected the shape of cell nuclei and somata in the two color channels. We created a set of 30 representative images by taking the first, middle and last frame from every stack and every channel, respectively. We manually segmented each image to produce a ground truth of optimal segmentations. Additionally we applied a simple automatic segmentation approach using Otsu's algorithm with mild postprocessing steps to see how its performance compared with our proposed method. We then computed the overlap of manually and automatically generated segmentation approaches by the Jaccard distance, a measure frequently used in computer vision to evaluate segmentation results (see Section 2.1.4).

We found that in the segmentation results for cell nuclei our proposed method as well as the simple method reported good overlap with the ground truth in the low seeding density ($0.68 \pm 0.20\%$ and $0.77 \pm 0.13\%$, respectively. Mean±s.d., n=18 images, see Figure 3.4a) and the high seeding density ($0.78 \pm 0.05\%$ and $0.83 \pm 0.07\%$, respectively. Mean±s.d., n=12 images, see Figure 3.4b). Yet, our proposed method outperformed the simple approach in terms of robustness at segmenting cell somata in the dataset with low seeding density ($0.67 \pm 0.10\%$ and $0.63 \pm 0.45\%$. Mean±s.d., n=18 images, see Figure 3.4a), while both methods performed equally well on images with high seeding density ($0.67 \pm 0.09\%$

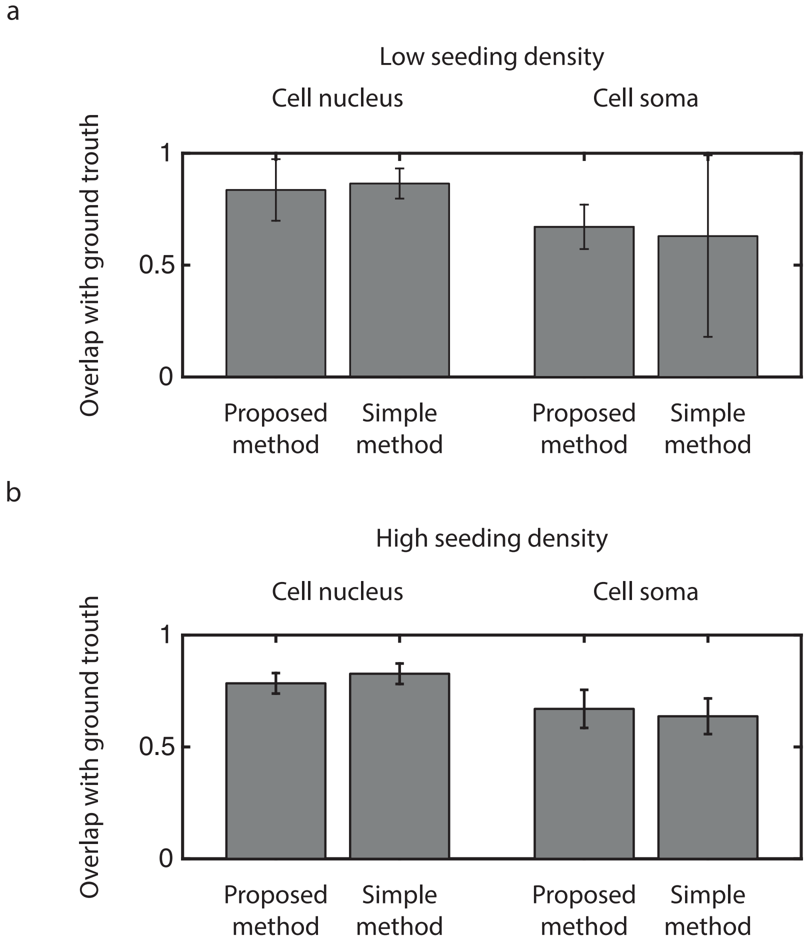


Figure 3.4: Automatic segmentation yields high overlap with manually generated ground truth. (a) Lower seeding density, n=18 images. (a, Cell nucleus) Percentage of overlap of our proposed method (0.68 ± 0.20%, mean±s.d.) and the simple method with Otsu's algorithm (0.77 ± 0.13%, mean±s.d.) with the ground truth. Both methods achieve equal results while the simple method performs slightly better. (a, Cell soma) Percentage of overlap of our proposed method (0.67 ± 0.10%, mean±s.d.) and the simple method (0.63 ± 0.45%, mean±s.d.) with the ground truth. While our method performed equally well on all images, the simple method reported erroneous segmentation results in one half of the images, but performed well on others (thus the high standard deviation). This is evidence of the higher robustness of our proposed method. (b) Higher seeding density, n=12 images. (b, Cell nucleus) Percentage of overlap of our proposed method (0.78±0.05%, mean±s.d.) and the simple method (0.83 ± 0.07%, mean±s.d.) with the ground truth. (b, Cell soma) Percentage of overlap of our proposed method (0.67 ± 0.09%, mean±s.d.) and the simple method (0.64 ± 0.08%, mean±s.d.) with the ground truth. Both methods perform equally well.

and $0.64 \pm 0.08\%$,respectively. Mean±s.d., n=12 images, see Figure 3.4b). These results emphasize the importance of robustness in a segmentation approach. While the simple method we used here yielded equal results on most images, even slight changes in the image quality led to completely wrong results.

Taken together, we showed that our automated pipeline robustly detected cell nuclei and somata in the respective fluorescence channels and also correctly created autotracks for over 90% of the observed cells. These results allowed us to statistically analyse the quantified motility and shape dynamics in T-lymphocytes.

3.3 Single T-lymphocytes react heterogeneously to in vitro confinement

We analyzed the migration behavior of single T-lymphocytes when exposed to different levels of confinement, as discussed in Section 1.5.1. We used the quantified speed in experiment “140906”, as it contained time-lapse image stacks with 8 µm, 6 µm and 4 µm confinement (see Table 3.1). We did not include experiment “150429” in this analysis because it featured only a single confinement height and and different coatings, which rendered a comparison of both experiments impossible.

At a micropillar height of 8 µm with n=24 cells, we could barely observe any migration (0.03 ± 0.03 µm/sec, mean±s.d., see Figure 3.5a). The cells were moving back and forth without leaving their position. We could observe only a single cell that was moving slowly and seemingly undirected. In the time-lapse movies with a micropillar height of 6 µm with n=34 cells we could observe a slight increase of motility (0.04 ± 0.03 µm/sec, mean±s.d., see Figure 3.5b), also the motile cells showed very straight directionality. Still, around 50 % of the cells did not move at all. At a micropillar height of 4 µm with n=54 cells, at least 60 % of the cells showed high speed with the typical elongated morphology (0.08 ± 0.06 µm/sec, mean±s.d., see Figure 3.5c). Yet, 40% of the cells did again not exhibit any motility and adopted a round morphology while moving back and forth.

We hypothesized that there might exist several subpopulations of cells that respond differently to the micropillar heights. We assessed this idea by fitting Gaussian mixture models (GMMs) with a varying number of Gaussians per model to the distributions of mean cell speeds per micropillar height and evaluated the model that fitted best with a variational Bayes expectation maximization approach using the pmtk3 toolbox developed by Murphy (2012) (see Figure 3.5d-i). The Bayesian approach found two subpopulations of cells in each confinement condition, which we termed resting (i.e. non-motile) and motile, depending on their averaged cell speed. At 8 µm confinement we found 18 cells (75%) to belong to the resting population with an averaged speed of 0.02 ± 0.01 µm/sec, while 6 cells (25%) where grouped into the motile population with an averaged speed of 0.07 ± 0.03 µm/sec. We made the same finding in the experiments with 6 µm/sec confinement, yet the ratios between two groups were nearly equal. We found 18 cells (53%) in the resting population with an averaged speed of 0.01 ± 0.01 µm/sec and 16 cells (47%) with an averaged speed of 0.07 ± 0.03 µm/sec in the motile condition. The differences between both subpopulations were most striking in experiments with 4 µm confinement. Here, 21 cells (39%) in resting population showed an averaged speed of 0.02 ± 0.02 µm/sec, where 33 cells (61%) were highly motile with averaged speeds of 0.12 ± 0.03 µm/sec.

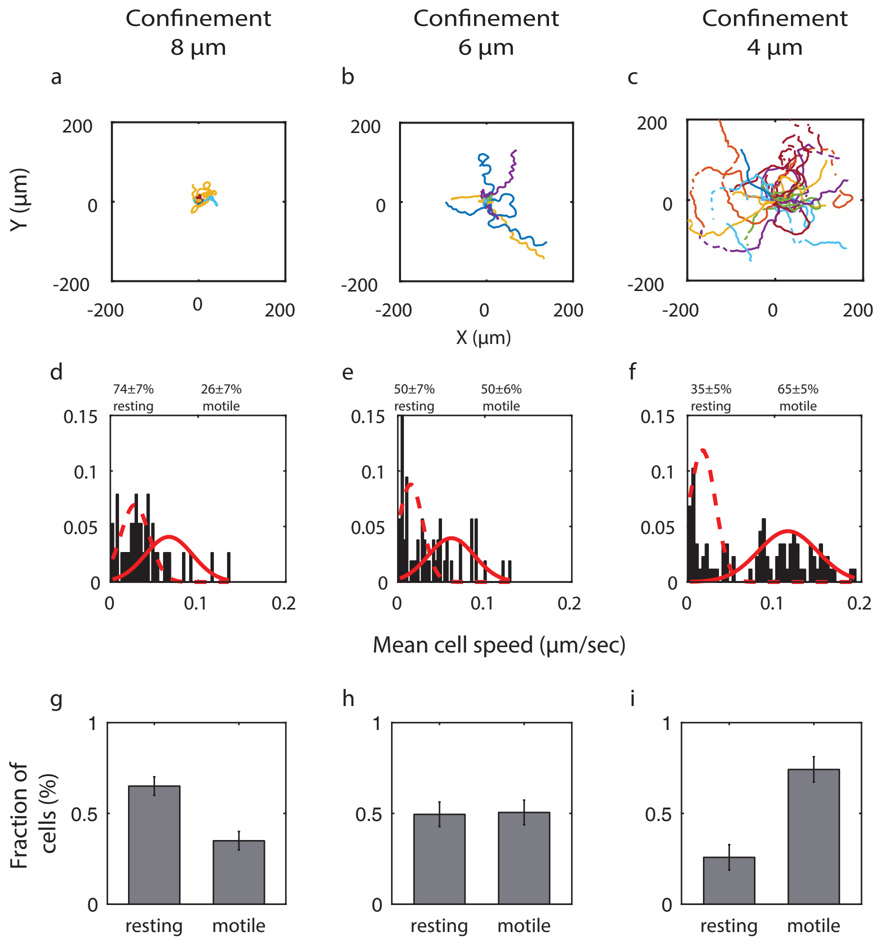


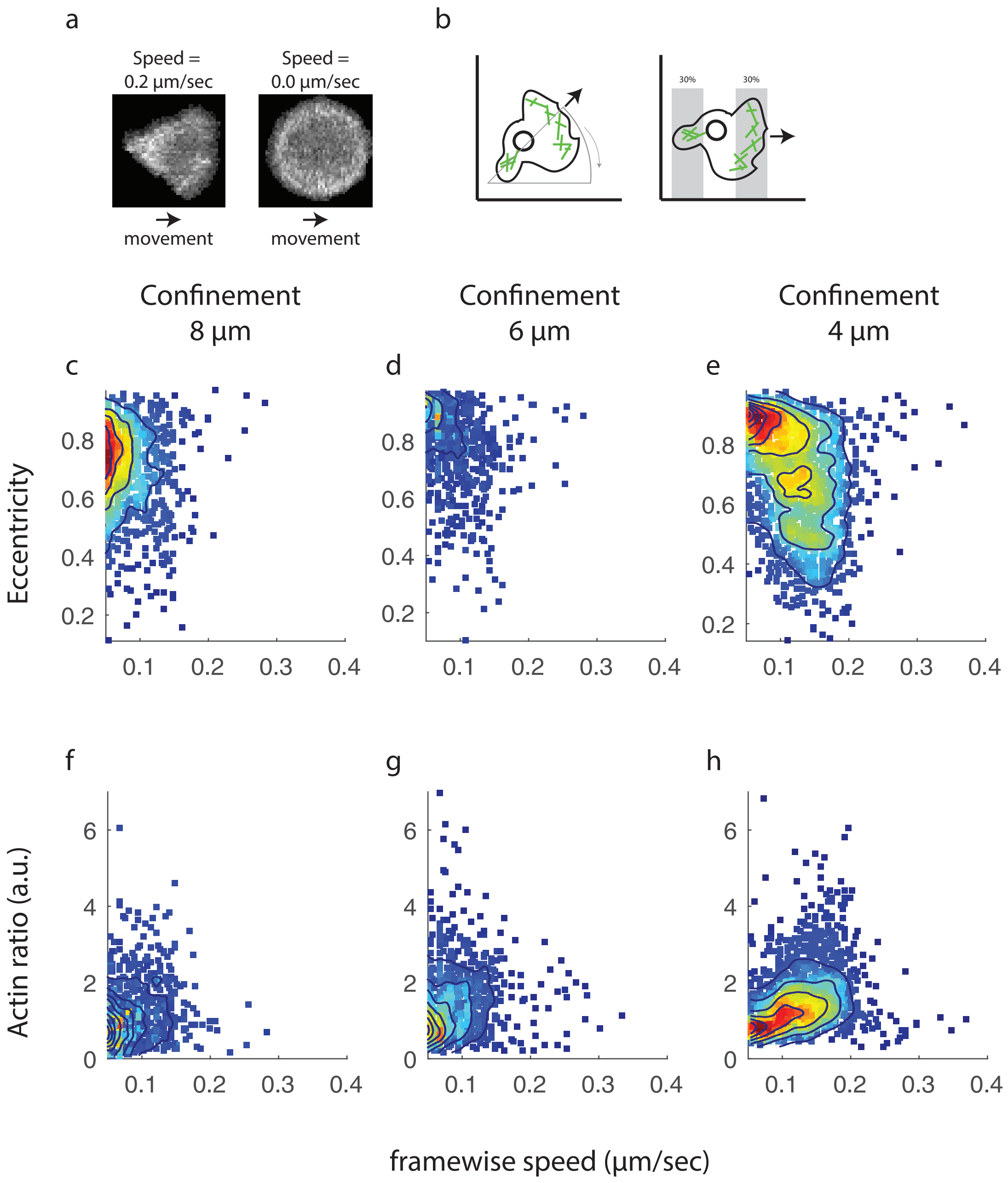
Figure 3.5: Analysis of T-lymphocyte migration speed reveals motile and resting subpopulations. (a,b,c) Trajectory plot of all cells in the different confinement conditions. Starting coordinates of all cells are set to (0,0). (a) 8 µm confinement, n=24 cells. The cells show only very little speed. One cell (yellow trajectory) observably moves in an undirected fashion, without leaving its starting area. (b) 6 µm confinement, n=34 cells. The cells migrate, directed speed is observable. 50% of the cells do not show any speed. (c) 4 µm confinement, n=54 cells. The cells migrate further, yet around 20% of the cells do not show any speed. (d,e,f) A variational Bayesian approach to fit a Gaussian mixture model reveals two subpopulations in the different confinements. A resting population and a motile population was detected in all conditions. (d) 8 µm confinement. Speed of resting population: 0.02±0.01µm/sec, n=18 cells (74%±7%). Speed for motile population: 0.07±0.03µm/sec, n=6 cells (26±7%). (e) 6 µm confinement. Speed of resting population: 0.01 ± 0.01µm/sec, n=18 cells (50 ± 5%). Speed for motile population: 0.07 ± 0.03µm/sec, n=16 cells (47%). (f) 4 µm. Speed of resting population: 0.02 ± 0.02µm/sec, n=21 cells (35 ± 5%) . Speed for motile population: 0.12 ± 0.03µm/sec, n=33 cells (65 ± 5%). (g,h,i) Mean and standard deviation of 10000 samples from a derichlet distribution to determine the most likely fraction of resting and motile cells for every condition, respectively.

3.4 Correlations of T-lymphocyte morphology with speed

As discussed earlier, it is known from previous studies that migrating cells exhibit an elongated soma, while resting cells are nearly perfectly circular (see Figure 3.6a). A measure of elongation is eccentricity, where a value of one would represent a straight line and a value of zero would be a circular object (see Section 2.3). We analyzed whether cells that were classified as motile in the different confinement conditions showed a correlation of eccentricity with cell speed on a frame-by-frame basis (see Figure 3.6c-e). Interestingly, we found no correlation in 8 µm confinement (Pearson's $r = -0.06$, $p = 0.07$, $n = 848$ frames) and a mildly negative correlation between eccentricity and speed in 6 µm (Pearson's $r = -0.28$, $p < 0.00001$, $n = 848$ frames) and 4 µm (Pearson's $r = -0.26$, $p < 0.00001$, $n = 1569$ frames) confinement, respectively.

Migrating cells are pushed forward by the reorganization of their actin network in the cell soma (see section 1.5.1). A first manual assessment of the actin intensity in the cell with a speed of 0.2µm/sec already revealed a heterogeneous intensity pattern, while in the resting cell actin was distributed more homogeneously around the cell membrane (see Figure 3.6a). We tested whether we could find a correlation between the actin ratio at the leading edge of a cell and its speed. For every frame, we computed a cell's direction of migration and rotated the fluorescence image such that it was parallel to the x-axis (see Figure 3.6b, first panel). We then computed the integrated fluorescence intensity of 30% of the cell soma at the front and rear of the cell (see Figure 3.6b, second panel). The ratio of actin at the front and rear of the cell showed no correlation at a confinement of 8 µm (Pearson's $r = 0.11$, $p = 0.001$, $n = 848$ frames) and 6 µm (Pearson's $r = 0.10$, $p = 0.001$, $n = 848$ frames), and a weak correlation at 4 µm (Pearson's $r = 0.28$, $p < 0.00001$, $n = 1569$ frames).

Figure 3.6 *(facing page)*: Migrating cells show only mild correlation of speed with eccentricity or actin polarization. (a) Fluorescence image patches showing a cell with a speed of 0.2 µm/sec with an elongated shape in the first panel. Actin seems polararized at the front and the rear of the cell. In contrast, a resting cell in the second panel is perfectly round and actin is homogeneously distributed around the cell membrane. (b) Strategy to compute the ratio of actin at the front and the rear of the cell. The direction of migration is detected and the cell is rotated to move parallel to the x-axis. Then, the fluorescence intensity of the first and last 30% of the cell soma are quantified. The ratio is then computed by dividing the front intensity by the rear intensity. (c-e) Density plot of cell speed (µm/sec) vs. Eccentricity for different confinement conditions. Dots denote single frames. (c) 8 µm confinement. Pearson's $r = -0.06$, $p = 0.07$. (d) 6 µm confinement. Pearson's $r = -0.28$, $p < 0.00001$. (e) 4 µm confinement. Pearson's $r = -0.26$, $p < 0.00001$. (f-h) Density plot of cell speed (µm/sec) vs. Actin ratio (a.u.). (f) 8 µm confinement. Pearson's $r = 0.11$, $p = 0.001$. (b) 6 µm confinement. Pearson's $r = 0.10$, $p = 0.001$. (c) 4 µm confinement. Pearson's $r = 0.28$, $P < 0.00001$.


a
Speed =
0.2 µm/sec
Speed =
0.0 µm/sec
movement
movement
b
30%
30%
Confinement
8 µm
Confinement
6 µm
Confinement
4 µm
c
d
e
Eccentricity
0.2
0.4
0.6
0.8
0.1
0.2
0.3
0.4
f
g
h
Actin ratio (a.u.)
0
2
4
6
framewise speed (µm/sec)

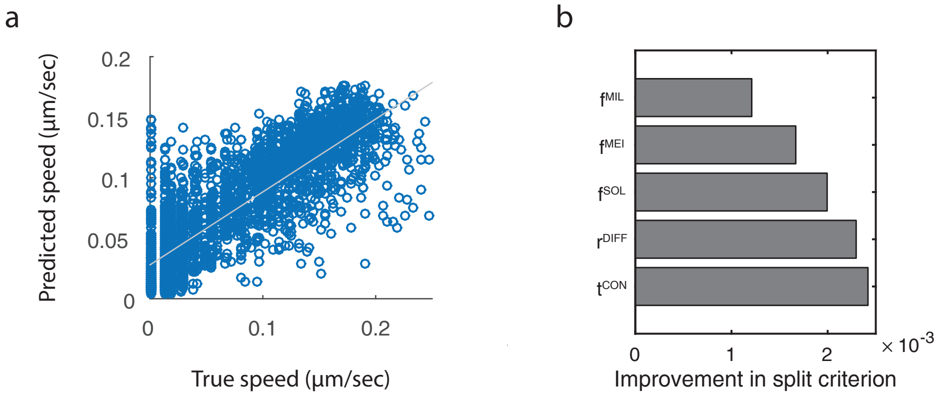


Figure 3.7: A cell's speed is predictable only from its shape and texture by a nonlinear regression model. a) Scatter plot of predicted speed (y-axis) versus measured motility speed (x-axis). The values show a correlation coefficient of 0.82, an RMSE of 0.04 and an R^2 score of 0.67. b) The most important features reported by the random forest regression are t^{CON} (Tamura contrast, texture), r^{DIFF} (Ray difference, shape), t^{SOL} (Solidity, shape), f^{MEI} (Mean intensity, texture) and f^{MIL} (Minor axis length, shape)

.

3.5 T-lymphocyte speed is predictable by morphology

A strength of our quantification pipeline is that it computes a multitude of different measures that describe a cell's shape and texture. As we could not find correlations in univariate analyses, we tested if cell speed could be predicted by a multivariate nonlinear model, namely random forest regression (Breiman et al., 1984). We used all cells from the experiments with a micropillar height of 4 µm and all shape features (40 in total). We used the framewise speed as our target and evaluated the fitted model by its out of bag error. The technique is comparable to a classical cross validation (see Section 2.4.2). After a training round with 200 trees, we found the predicted speed values to highly correlate with the true values ($R_{corr} = 0.82$, see Figure 3.7a). With an RMSE of 0.04 and an R^2 score of 0.67, the model predicted the speed very well. By evaluating the features that had the most predictive value in the trained model, we found the features t^{CON} (Tamura contrast, texture), r^{DIFF} (Ray difference, shape), t^{SOL} (Solidity, shape), f^{MEI} (Mean intensity, texture) and f^{MIL} (Minor axis length, shape) to have the most predictive value (see Figure 3.7b and Section 2.3 for details).

3.6 Conclusion

In this chapter we described a fully automated modular pipeline for the processing of multichannel time-lapse fluorescence experiments, that tracks single cells while also quantifying their morphology and motility robustly and fast. It allows the manual assessment and correction of cell tracks, but is also capable being executed autonomously on a computation cluster. The modular design of the pipeline makes it easy to change parts and adapt them to different experimental setups and applications.

We applied the pipeline to a set of time-lapse experiments observing the motility of T-lymphocytes with different seeding densities in two fluorescence channels showing nuclei through Hoechst 405 and cell somata by lifeAct-GFP. Without changing the parameters, at least 90% of the cell trajectories were correctly and automatically tracked. Our segmentation module showed a good overlap with the manually determined ground truth and was more robust than a simple thresholding algorithm. By fitting the most likely Gaussian mixture model via a Bayesian approach to the derived averaged speed of T-lymphocytes, we found that that two subpopulations of resting and motile cells exist in all confinement heights of 8 µm, 6 µm and 4 µm, respectively. We showed that the fraction of motile cells increased to 25%, 53% and 61% as the micropillar height was decreased. Also, our analysis revealed that with an averaged speed of 0.12µm/sec under 4 µm confinement, the cells were nearly twice as fast as under 6 µm and 8 µm confinement, which both showed an averaged speed of 0.07 µm/sec. The fact that cells were migrating the fastest under 4 µm confinement is contradicting previous studies, where it was shown that a height of $7-9$ µm is optimal (Jacobelli et al., 2010). However, as the the experimental setup could differ, e.g. in micropillar construction or coating (in our case: fibronectin), this finding needs to be further validated. Also, the reasons why we always found two subpopulations of resting and motile T-lymphocytes needs to be further analyzed. Yet, we could rule out that cells in the resting population are simply dead, as the cells still expressed the LifeAct-GFP marker and showed changes in their morphology, but did not leave their position.

Contradicting to our expectations, we could not find a correlation of a cell's eccentricity and its speed or its front/rear ratio of actin concentration. The exact reasons for this have to be further analyzed. For example we could observe that migrating cells often feature a broad leading edge followed by an elongated but thin rear. This could lead to an overestimation of a cell's roundness and render the usage of eccentricity as a feature useless. This hypothesis is also supported by the fact that we could predict a cell's speed by a multivariate nonlinear regression model, but did not find eccentricity in the most predictive features. Yet, we found the shape featuers ray difference, solidity and minor axis lengths to be predictive for a cell's speed. As especially the ray difference can capture more complex morphologies than just the level of roundness, it will be worthwile to analyze this feature in more detail.

Our pipeline is at the moment not capable to segment brightfield images and derive cell shapes from this channel. In future analyses this could be possible by e.g. testing different objectives that enhance the contours of the cells during image acquisition. Furthermore, we could implement more sophisticated preprocessing steps and learning-based segmentation algorithms that can recognize incomplete cell boundaries in images with bad contrast, as proposed earlier (Liu et al., 2012; Theriault et al., 2011). The possibility to segment cells in the brightfield channel will be of help to better understand the actin distribution in the cell soma. As we are deriving cell shape from the LifeAct-GFP channel at the moment, we have to rely on a proper transfection of this marker, or we measure wrong cell shapes or entirely miss full cells.

The cell motility statistics we derived where based on a single experiment with small amounts of T-lymphocytes. As our pipeline was built for fully automated quantification, we plan to extend our dataset by thousands of cells. This will increase the statistical power of our findings and may lead to rare subpopulations which we could not identify with the available data. We also will expand our project to analyse the behavior of T-lymphocytes under different coatings of the micropillar and coverslip. Last but not least, we will incorporate more sophisticated analyses to describe the actin distribution at the cell front and rear during migration. We will for example compare the model-based computational analysis of fish keratocytes by Keren et al. (2008).

Taken together we believe that the high robustness of our automatic quantifications ant the promising first analyses will be of great help road to a full understanding of T-lymphocyte migration.

Chapter 4

Robust cell detection in high-throughput brightfield microscopy

This chapter is based on and in part identical with Buggenthin et al. (2013).

In Chapter 3 we used established image processing approaches to extract centers and somata of single cells in the respective fluorescence images. Yet, to extract this information from brightfield images that are acquired in high-throughput long-term time-lapse microscopy, the available methods did not suffice (see Sections 1.2 and 1.5.2).

In this chapter we present a novel method that is able to robustly segment and analyze cells with ellipsoid morphology from high-throughput brightfield microscopy in a time efficient manner without user interaction. The procedure comprises two steps: (i) Image acquisition is adjusted to obtain optimal brightfield image quality for automatic processing. (ii) A concatenation of fast performing image processing algorithms robustly identifies single cells in each image. We applied the method to a time-lapse movie consisting of 315,000 images that captured differentiating hematopoietic stem cells on the full coverslip over 6 days. We evaluated the accuracy of our method by comparing the number of identified cells with manual counts. Our method is able to segment images with varying cell density and different cell types without parameter adjustment and clearly outperforms a standard approach. By computing population doubling times, we were able to identify three growth phases in the stem cell population throughout the whole movie, and validated our result with cell cycle times from single-cell tracking. We conducted the project in close collaboration with the group of Prof. Dr. Timm Schroeder from D-BSSE Basel (ETH Zurich).

4.1 Previously available methods for high-throughput segmentation

In the last couple of years, several computational methods for the automatic processing of high-throughput microscopy experiments have been proposed. For example, Fenistein et al. (2008) developed an automatic method for the segmentation of cell nuclei in fluorescence images for different cell lines in dilution experiments and reported an average cell recognition rate of 95%. Knapp et al. (2011) employed a method to identify single cells in two-channel RNAi screens and used this information to improve the statistical power of the analysis. Both applications demonstrated the feasibility of automatic high-throughput image processing methods on large amounts of fluorescent images. The framework *CellProfiler* (Carpenter et al., 2006) is a great example of how automated image analysis can be made accessible for a broad range of users, not only specialists. The intuitive GUI and wealth of different implemented methods has led to frequent usage (2400 citations, as of March 2015), where most applications relate to the analysis of fluorescence images.

However, the development of an automated processing method for high-throughput brightfield experiments is more demanding than in the fluorescence case and holds several challenges. Cells imaged by brightfield microscopy exhibit heterogeneous intensity levels and are often badly contrasted. In addition, differences in illumination over time and across the cell culture plate hamper the ability to specify a global set of parameters for cell detection algorithms over the whole experiment. This prevents the application of available automatic image processing frameworks, which are mostly developed to perform well on fluorescent images. Despite the large amount of methods that are implemented in frameworks like *CellProfiler*, the available algorithms for illumination correction and segmentation do not perform well enough to achieve satisfying results on many high-throughput brightfield microscopy experiments.

By employing active contour and level set methods, many issues of cell segmentation in brightfield or phase contrast images have already been solved (Tse et al., 2009; Li et al., 2010). For example, Ambühl et al. (2011) demonstrated the very accurate tracking of a single cell in phase-contrast microscopy images. Ali et al. (2011) developed a method that combined out of focus image acquisition and segmentation by level sets to identify outlines of adherent cells. However, these approaches are computationally expensive and often highly parameter-depended, which prevents the application in high-throughput image processing, where millions of objects have to be processed in reasonable time and without user interaction.

4.2 A method for automated detection of cells in brightfield high-throughput microscopy

For the development of a method to analyze high-throughput microscopy data, it is especially important to incorporate algorithms that are (i) robust against heterogeneities between images that are processed and (ii) able to process single images in the range of seconds up to a few minutes at maximum in order to finish a full experiment in reasonable time. In this work, we chose the algorithms used in every step according to these requirements. The complete procedure is visualized in Figure 4.1.

4.2.1 Image acquisition

The first step in our method concerns image acquisition. Adapted from Selinummi et al. (2009), we recorded every image with the microscope's auto focus set 18μm below the optimal focal plane. In the case of brightfield image acquisition with a 10x fluar objective (Zeiss), the change of the focal plane resulted in enhanced contrast of single cells, yet with a loss of textural complexity (see Figure 4.1a). The cell body was evenly illuminated and much darker than the background. In addition, every cell exhibited a bright halo that is supporting the identification of touching cells.

4.2.2 Background correction

After all images were acquired in the proposed manner, differences in illumination across the images had to be resolved. We used an adapted version of the method proposed by Schwarzfischer et al. (2011) (see Section 2.1.3). This machine learning based algorithm estimates the background for every image, using a grid of image patches that are classified as showing only background or a mixture of background and foreground pixels. In comparison to standard correction methods like Gaussian filtering that is parametrized on the average foreground object size, the machine learning based method is able to estimate the background more robustly. As shown in Figure 4.1b, every cell body was clearly separated from the background. The halo surrounding each cell was corrected, yet clumped cells still

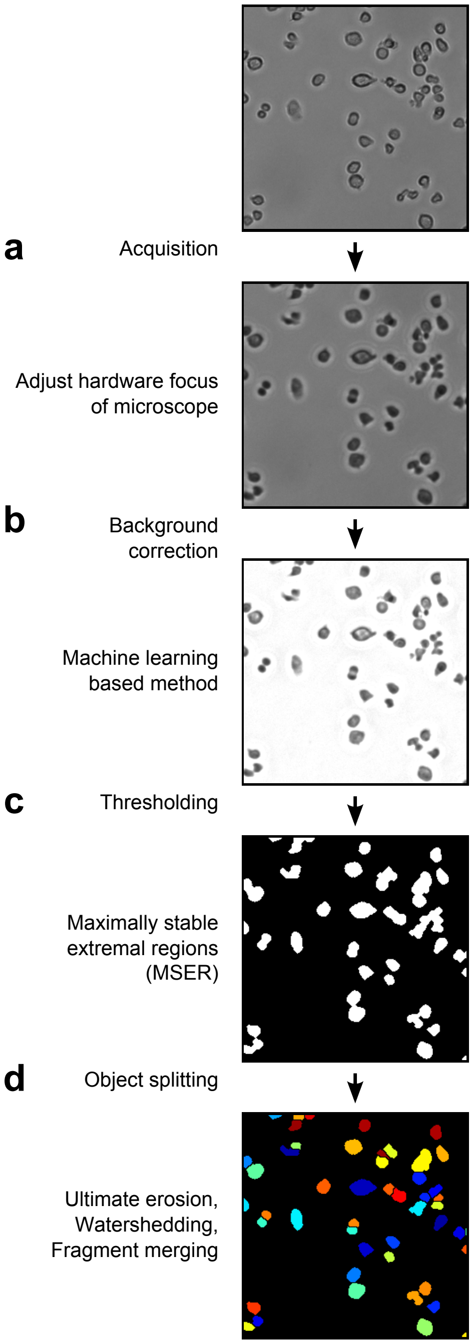


Figure 4.1: Flow chart of the proposed method. The results of each step are exemplified on a brightfield image of hematopoietic progenitor cells. (a) The image is acquired with the focus set 18 µm below the optimal focal plane to enhance contrast of cells. (b) The inhomogeneously illuminated background is corrected by a machine learning based approach to resolve differences in illumination across different locations on the cell culture plate and over time. (c) Foreground objects are identified by maximally stable extremal region (MSER) detection. (d) Splitting of clumped cells. Maxima of cells are identified by ultimate erosion and split by watershedding. Over-segmented cell bodies are reconstructed by merging of too small neighboring regions.

exhibited a change in illumination at their touching edges. Due to the time-consuming feature calculation during machine learning, this algorithm was occupying nearly 50% of total computing time for a single image, which was in our case 30 to 50 seconds on a standard laptop (Intel Core i7 dual-core, 2.8GHz, 8GB RAM, Windows 7 64bit).

4.2.3 Thresholding

In the next step, all foreground objects had to be separated from the background. In our method, we used the maximally stable extremal regions (MSER) algorithm (Matas et al., 2004) (see Section 2.1.4). An advantage compared to thresholding methods such as Otsu's algorithm is its robustness in segmentation when there are inhomogeneities in object illumination or huge differences of cell densities between different images. As shown in Figure 4.1c, MSER correctly identified nearly all cell bodies. The used implementation of MSER has linear time complexity, thus it is able to process a single image (i.e. 1388×1040 pixels) in milliseconds (Nistér and Stewénius, 2008).

4.2.4 Object splitting

Eventually it was necessary to split clusters of multiple cells that were segmented as a single foreground object (i.e. under-segmentation). We used a two-step approach consisting of an initial marker-based watershedding (see Section 2.1.4), followed by merging of cells that were erroneously split into fragments (i.e. over-segmentation).

In this step (see Figure 4.1d), the earlier conducted out of focus acquisition was very advantageous: the homogeneous illumination of cell bodies and the slightly brighter interfaces of touching cells simplified the task to find cell centers, which then served as seed points for the watershed algorithm. Secifically, the raw input image $\mathbf{D}$ was inverted resulting in local intensity maxima residing at cell centers. All local maxima were extracted to a binary mask $\mathbf{D}^{segmax}$, which was done calculating the regional maxima of the Euclidean distance transform of the cell mask $\mathbf{D}^{seg}$ that was returned from the MSER algorithm. In a next step, the distance transform $\mathbf{D}^{dist}$ of the foreground mask $\mathbf{D}^{seg}$ was computed. The final transformed image $\mathbf{D}^{transformed}$ was derived by imposing $\mathbf{D}^{segmax}$ on $\mathbf{D}^{dist}$. Watershedding was applied on $\mathbf{D}^{transformed}$, resulting in the mask of identified and split objects $\mathbf{D}^{watershed}$.

To reduce over-segmentation, a rule-based split & merge procedure of small regions from *CellProfiler* 1.0 was applied (Carpenter et al., 2006). A list of adjacent neighbors for all objects in $BW_{watershed}$ that are likely to be over-segmented (i.e. very small objects with high eccentricity) was computed. For each neighbor the following criteria were evaluated: (i) For the pixels residing on the interface of the evaluated object and its neighbor, the likelihood to belong to the background or to the foreground was computed. Foreground and background were represented as Gaussian distributions, where mean and variance are derived from the image (i.e. pixels that were classified as foreground and background by the thresholding step). (ii) The eccentricity for the merged object was calculated. The evaluated object and its neighbor were merged if the interface pixels were more likely to belong to the foreground and if the merged object's eccentricity was lower than an empirically determined value (here: 0.7). Depending on the number of cells in an image this step occupies 10 to 50 seconds of processing time.

4.2.5 Implementation

All methods were implemented using MATLAB version 8.0.0.783 (R2012b) with the additional packages image processing toolbox 8.1 and statistics toolbox 8.1. If MATLAB code was available for the cited methods, this code was used. For MSER thresholding, a C++ implementation with linear time complexity was used. To speed up computation times, the data set was split into junks of images and processed on a computation cluster (sun grid engine version 6.2u5). The average node architecture was equal to an Intel Xeon 2GHz, 4GB RAM running a 64bit linux-based operating system.

4.3 Large-scale application shows high robustness and cell detection accuracy

We applied our method on a time-lapse experiment of murine hematopoietic stem cells (HSCs) under conditions that promote differentiation towards myeloid cells. For a review of blood cell differentiation in mice, see for example Orkin and Zon (2008). All wetlab experiments where conducted by Dr. Philipp S. Hoppe from the D-BSSE at the ETH Zurich.

To obtain population of HSCs as pure as possible, femurs, tibiae and ilia of a healthy mouse strain on C57Bl/6 background with no discernible phenotype were removed from 14 weeks old mice and the bone marrow was extracted. Using fluorescence activated cell sorting (FACS), the stem cells were then purified by combining two protocols described

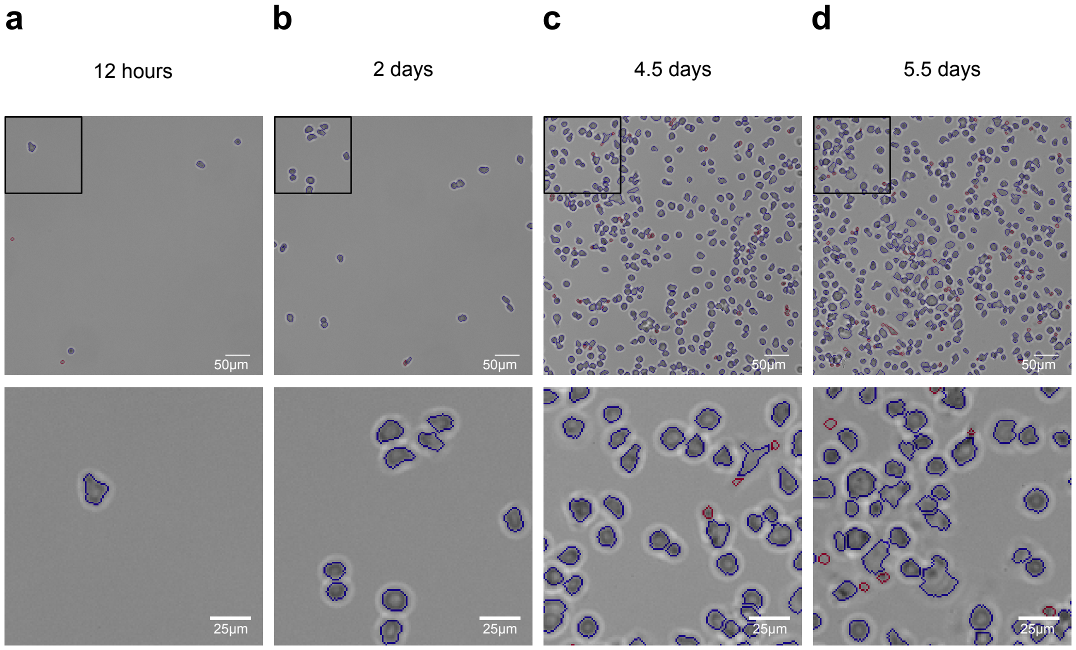


Figure 4.2: Manual examination of segmentation results, shown at exemplary image patches over the whole time span of a 6 day time-lapse experiment of differentiating hematopoietic stem cells. Blue outlines: Segmented objects regarded as cells. Red outlines: Objects unlikely to represent cells (size <50 px and eccentricity >0.99). First row: 500x500px image patch, second row: 150x150px image patch. (a) 12 hours after experiment start. Very few cells are populating the field of view. Cell outlines are correctly segmented. Erroneous measurements originate in debris in the image. (b) 2 days after experiment start. The number of cells is slightly increased, still the object density is very sparse. Pairs of clumped cells can be identified, which are correctly split by the method. (c) 4.5 days after experiment start. More complex cell morphologies arise that lead to errors in segmentation. The field of view becomes more and more crowded, complicating the identification of single cells. Small artifacts that are a result of over-segmented cells or fragments of dead cells are filtered by size. (d) 5.5 days after experiment start. Most cells are differentiated and different morphologies can be found. Segmentation errors are observed more frequently, especially for adherent cells with elongated shape.

in Kiel et al. (2005) and Osawa et al. (1996). According to the original publications, the fraction of true HSCs that are received by this method is between 40%-60% .

Directly after sorting, the cells were plated out on a plastic slide (μ-slide VI coated with Fibronectin, Integrated BioDiagnostics GmbH, Munich, Germany) with two physically separated wells in serum-free medium (StemSpan SFEM, StemCell Technologies) supplied with cytokines that only promote differentiation towards myeloid cells. All animal experiments were performed in compliance with the institutional guidelines of the Helmholtz Center Munich and the regulations of the State of Bavaria.

Due to the cameras limited field of view, both wells were subdivided into 33 overlapping tiles (fields of view). Each field of view corresponded to an image of 1388 × 1040 px (1 px = 1.0238 µm), that was saved in 8-bit png format. Imaging was conducted using a Cell Observer microscope (Zeiss) surrounded by an incubator to maintain a constant temperature of 37°C. Images were obtained using a 0.63x TV-adapter (Zeiss) and an AxioCam HRm camera (Zeiss), with a 10x fluar objective (Zeiss). Each field of view was imaged in intervals of ∼2.3 minutes for 6 days. Automatic focusing was achieved using a hardware autofocus (Zeiss) which was set to 18 µm below the optimal focal plane to acquire slightly blurred images. The complete data set comprised a total of 315,942 images (4787 time points * 66 fields of view) and occupied ∼500 gigabytes of hard drive space. The wells were sparsely covered with cells at experiment start, yet as the experiment was stopped all wells were densely populated by hematopoietic progenitors and differentiated cells (see Figure 4.2).

Complete processing of the full data set occupied 72 hours, using 150 cores of a computer cluster. The average node architecture was equal to an Intel Xeon 2GHz, 4GB RAM running a 64bit linux-based operating system. Complete processing of a single image with average cell density (see Figure 4.2) lasted 100 seconds. To account for small debris or fragments of dead cells that were erroneously segmented by our method, we discarded all foreground objects with a size < 50 pixels and an eccentricity > 0.99, as well as objects that were touching the image border.

The final test set comprised ∼315,000 raw images with the according object masks and computed background corrections, covering ∼270,000,000 identified objects.

Parameters

For the dataset that was used in this work, the method was initialized using the following parameter set (one setting for all images of the data set):

- Background correction

Parameter	Value
Tile dimensions	30 px
Overlap	15 px
eps	0.1
MinPts	6

- Cell detection

Parameter	Value
Delta	5
Minimum Object Size	30 px
Maximum Object Size	4000 px
Maximum Variation	1

- Object splitting

Parameter	Value
Maximum Eccentricity	0.7
Minimum Object Size	30 px
Maximum Object Size	1000 px

Evaluation

The performance of a segmentation approach can be measured in different ways depending on the analyses that are intended after processing. For the development of automatic tracking approaches it is necessary to identify single cells with high accuracy. Especially cells that stick together shortly after division or clusters of multiple cells need to be split correctly. For population analysis or simple cell counting it suffices to detect the number of cells in each image with high accuracy. Here, we manually determined the total number of cells after 12 hours, 2 days, 4.5 days and 5.5 days at two randomly chosen fields of view per well. To manually assess the segmentation quality we used the java based image processing tool ImageJ 1.47K with the plugin CellCounter (Abramoff et al., 2004).

We evaluated if a cell was (i) correctly segmented, (ii) missed, (iii) over-segmented or (iv) under-segmented. Next, we computed the average accuracy, specificity and sensitivity of cell detection based on the number of true positives (complete cell bodies, the largest fragment of over-segmented cells and one cell per under-segmented object), false positives (dirt and cell fragments) and false negatives (missed cells, remaining cells in under-segmented objects). In addition, we calculated the mean and the according standard deviation of cell densities (cells per mm^2) over all fields of view at the given time point.

After 12 hours (Figure 4.2a), all fields of view were sparsely covered with cells (cell density $5.6 \pm 2.4\ ^1/_{mm^2}$). Despite the high fraction of correctly segmented cells (92%), debris in the examined fields of view that was falsely identified as a cell by our method lead to a decrease in accuracy ($83 \pm 11\%$). The low number of cells at this early time point resulted in a high variability (11%) between the fields of view.

At day 2 (Figure 4.2b), the number of cells increased to a density of $16.2 \pm 7.1\ ^1/_{mm^2}$. Pairs of clumped cells appeared. 82% of all cells were correctly segmented, only very few cells were missed or over-segmented (11%) and under-segmentation was not detected. The cell detection accuracy was $82 \pm 3\%$.

At day 4.5 (Figure 4.2c), the number of cells across the examined fields of view was significantly larger ($741.6 \pm 250.6\ ^1/_{mm^2}$). We observed cells that were clumped together in large clusters and first differentiated cells with more complex morphology were found. Cells with round shapes were correctly identified in most cases (92%), especially round clusters of cells were under-segmented (1%) and cells with elongated shape were over-segmented (4%). Most over-segmented cells were still only counted once since smaller fragments were discarded by the filtering step (see methods). 1% of the cells were missed, mostly because of bad contrast or direct contact to the image border. Due to the large increase in cell number, debris did not significantly contribute to a drop in cell recognition accuracy anymore. The accuracy at this time point was $95 \pm 1\%$.

At day 5.5 (Figure 4.2d), fields of view of both wells were very densely populated by cells ($1.2*10^3 \pm 0.22*10^3\ ^1/_{mm^2}$) that were exhibiting a variety of shapes. With 1.2%, the fraction of missed cells was even reduced compared to the time point examined before. Over- and under-segmented cells were observed more frequently (6% and 1%, respectively), yet most cells were correctly segmented (90%). The amount of debris was increased, mostly due to clumps of fragments of dead cells. Yet, the cell detection accuracy was very high ($92 \pm 3\%$).

Sample images showing the cell densities at different time points for our method are given in Figure 4.2. The object quantification is summed up in Table 4.1.

To demonstrate the superior robustness of our method, we conducted the same manual evaluation with the results from a pipeline of methods available in *CellProfiler* (Carpenter et al., 2006). We used version 2.0 (r11710) of the software and created a pipeline for automatic processing of the brightfield images that we used in the manual evaluation. The

Experiment time	**12 h**	**2 d**	**4.5 d**	**5.5 d**
Number of manually counted cells	37	90	5837	7414
Correct	34(91.9%)	74(82.2%)	5356(91.8%)	6638(89.5%)
Missed	1(2.7%)	4(4.4%)	78(1.3%)	86(1.2%)
Over-segmented	2(5.4%)	10(11.1%)	230(3.9%)	411(5.5%)
Under-segmented	0(0.0%)	0(0.0%)	46(0.8%)	87(1.2%)
Debris	6	7	1	32
True positives	36	84	5632	7136
False positives	6	13	96	329
False negatives	1	6	205	278
Accuracy	0.83 ± 0.11	0.82 ± 0.03	0.95 ± 0.02	0.92 ± 0.03
Specificity	0.86 ± 0.11	0.87 ± 0.06	0.98 ± 0.01	0.96 ± 0.03
Sensitivity	0.97 ± 0.06	0.94 ± 0.04	0.96 ± 0.02	0.96 ± 0.01

Table 4.1: Manual evaluation of segmentation results. Two randomly chosen fields of view per well were quantified for 12 hours, 2 days, 4.5 days and 5.5 days, respectively. In each field of view, the number of true cells was counted. All segmented objects were classified as correct, over-, under-segmented, or debris. Accuracy, Sensitivity and Specificity of cell detection were calculated based on true positives (complete cell bodies, the largest fragment of over-segmented cells and one cell per under-segmented object), false positives (dirt and cell fragments) as well as false negatives (missed cells, cells in under-segmented objects). Note that we deliberately keep differences in the total number of counted cells at different experiment times, since these impact on the standard deviation of accuracy, specificity and sensitivity.

following modules were sequentially called for each image: Correct Illumination (Gaussian filter, Average object size: 60 px), Apply threshold (Otsu global), Identify primary objects (Typical diameter of cells: 5 to 50 px, splitting method: Intensity, method to draw dividing lines: Shape), Convert objects to image (saved binary mask). Parameters were optimized according to a single image of the evaluation set of day 4.5. In the case of long-term time-lapse experiments the constant change of cell density and illumination, as well as the acquisition of different fields of view makes Otsu's method the best choice out of the algorithms that are available in CellProfiler. We applied the CellProfiler pipeline on the identical set of out-of-focus images and optimized the parameters of each module based on a single image of day 4.5. For a graphical comparison of the cell detection accuracy of our method against the *CellProfiler* pipeline, see Figure 4.3.

At 12 hours, the *CellProfiler* pipeline produced highly heterogeneous results. The used thresholding algorithm performed well on images of 2 fields of view but produced completely mis-segmented images on the others, leading to a low cell detection accuracy ($24 \pm 31\%$). This was most likely due to errors in the clumped cell splitting step.

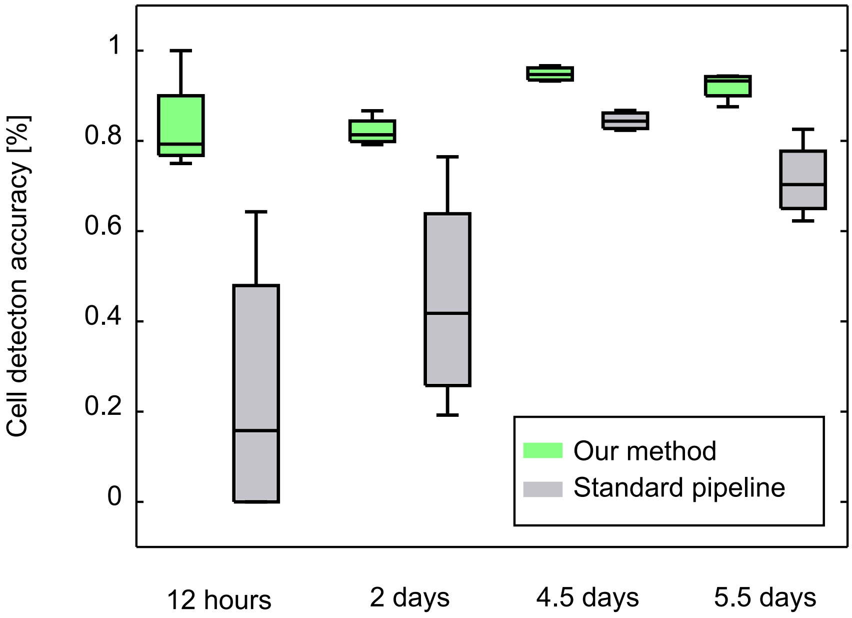


Figure 4.3: Comparison of manually evaluated cell detection accuracy as described in Table 4.1 between our method (green boxplots) and the *CellProfiler* pipeline (gray boxplots). Especially at the two early time points, CellProfiler performs not very robust on the different fields of view. Note that the pipeline was parametrized to perform best on images at day 4.5. Thus, the pipeline might be able to perform well on images on the early time points, but is not robust enough with the given parameter settings.

For images taken at day 2, the *CellProfiler* pipeline performance increased (45 ± 25%). Yet the accuracy was rather low and less robust across different fields of view (25%).

At 4.5 days, the increased cell density lead to an improvement in the cell detection accuracy (84 ± 2%), with a huge decrease of standard deviation. Still, 3% of the cells were missed completely and 9% were under-segmented.

At the last manually evaluated time point of 5.5 days, cell detection accuracy of the CellProfiler pipeline decreased to 71 ± 9%. This was mainly because of the high fraction of missed (20%) and under-segmented (6%) cells.

Taken together, our method showed high robustness in cell detection and low over- and under-segmentation over the full experiment range. Even at very late time points where the wells were very densely covered, the cell detection accuracy was satisfying (∼92%). The out-of-focus acquisition improves the overall segmentation accuracy of our method: Applied to a comparable in-focus movie, the segmentation accuracy dropped to 70% due to over-segmentation of badly contrasted cells and complex cell texture.

As shown in Figure 4.3, our method clearly outperformed the standard *CellProfiler* pipeline. Note that the low cell detection accuracy in the early time points does not necessarily mean that *CellProfiler* in general is not able to segment this type of images (i.e. very few cells). Still, the combination of algorithms performed less robustly on images with different cell densities, given the parameter set that we optimized for images with medium cell density (i.e. day 4.5).

Finally, we would like to note that our pipeline achieved similar robust results (segmentation accuracy ∼85%) in a second long-term high-throughput experiment.

4.4 Population doubling time derived from cell counts

A possible use-case in the analysis of high-throughput time-lapse experiments is the control of cell proliferation. Due to photo toxicity or different medium conditions, cells could die early or exhibit deviating proliferation rates (Schroeder, 2011), which would introduce errors in later analyses that are conducted on the data set.

Here, we first analyzed the mean cell density over 66 fields of view over the full experiment time span (blue line in Figure 4.4a). We found that the number of cells increased monotonously until a plateau roughly at day 5. We compared the results with the manually quantified numbers of cells as shown in Table 4.1 and found them to reside within the standard deviation of the number of objects. From our accuracy estimation in Table reftab1, we conclude that the plateau is not due to a failure of our method, but resulted from biological or experimental reasons. One explanation could be the differentiation and thus post-mitotic state of the hematopoietic cells, but also a depletion of the medium. In addition, the high density of cells could lead to an arrest in population growth.

Plotting the growth curves in log scale (see Figure 4.4b) revealed three different phases of population dynamics. At the beginning of the experiment, the number of cells increased sub-exponentially. Between approximately 2 and 4.5 days a clearly exponential increase with an average doubling time of 10-12 hours was observed. The population stops to grow exponentially and reaches a plateau after ∼4.5 days.

Based on the cell counts resulting from our image processing method, we derived the population doubling time. Due to the high temporal resolution of ∼2.3 minutes between measurements, the population doubling time could be estimated by computing the differ-

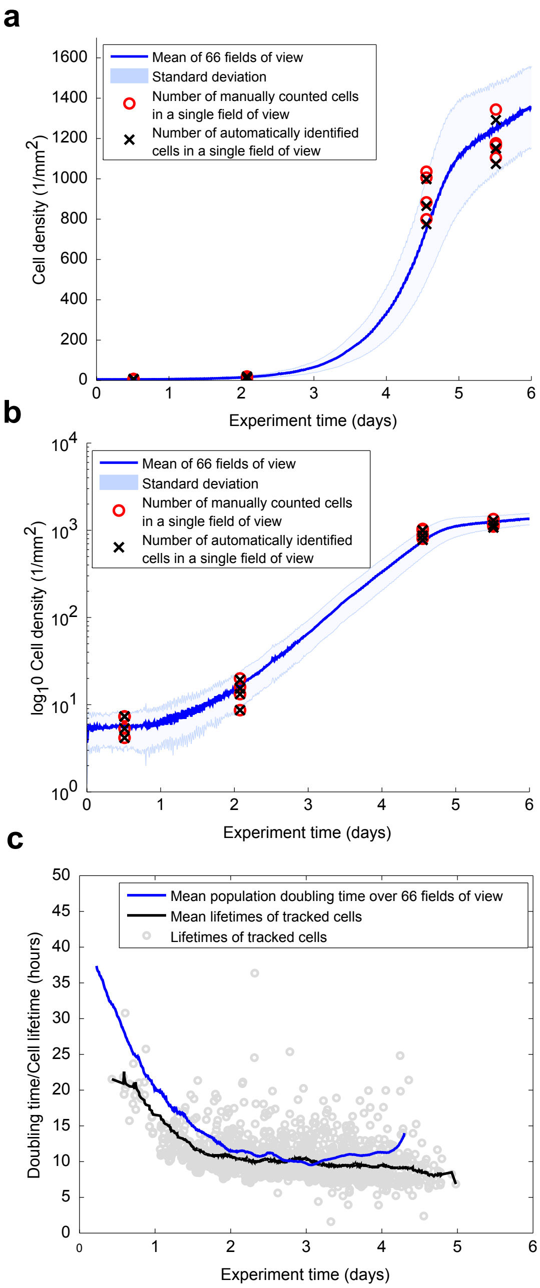


Figure 4.4: Whole-movie analysis of population growth rates and doubling times. (a) Mean cell densities over 66 fields of view (blue line) and according standard deviation (light blue patch) per mm^2 covering the full experiment time range. Red circles indicate the manually determined number of cells in 4 randomly chosen fields of view at 12 hours, 2 days 4.5 days and 5.5 days as described in Table 4.1. (b) Increase of cells plotted in log-scale. A non-exponential growth phase could be identified until day 2. Between day 2 and day 5 the number of cells increased exponentially. (c) Population doubling times (blue line) per time point and cell cycle times of ∼1600 manually tracked cells with mean lifetimes from 0.5 to 5 days (gray circles). The doubling time decreased until day 2, where it roughly stabilized around 10 to 12 hours. The cell cycle times coincided with the derived doubling times, indicating a correct automatic derivation.

ence of each time point and the time point where the cell number had doubled, respectively (see blue line in Figure 4.4c). The doubling time decreased from ∼40 hours and stabilized after day 2 until day 4 at around 10 to 12 hours.

To validate the estimated doubling times, we tracked 1600 cells manually using our inhouse developed software TTT (Rieger et al., 2009). As shown in Figure 4.4c, the cell cycle times of tracked cells that were born between 0 and 4.5 days (gray circles) show the same trend, decreasing from ∼20 hours to 9-11 hours in the exponential growth phase.

4.5 Conclusion

In this contribution, we described a fully automated method for processing of high-throughput brightfield microscopy experiments, that relies on the combination of optimized image acquisition and a concatenation of image processing algorithms that identify cells in in a robust yet time efficient manner. Using the same parameter set for all images, we applied the method on a 6 day time-lapse movie of differentiating hematopoietic stem cells and achieved a cell detection accuracy of at least 82%, which outperformed a pipeline of algorithms available in *CellProfiler*. We demonstrated the application of the results generated by our method by computing population doubling times based on the increasing number of cells over the whole experiment in time and space. We compared the results to the cell cycle times of 1600 manually tracked cells and showed that the automatically derived doubling times coincided with the manually tracked cell cycle times.

The full data set of ∼315,000 images was processed within ∼72 hours. Note that this value was achieved by parallel computing on 150 cores of a computation grid. However, the code used in this work was not optimized for speed. Using implementations in C++ or Java that are optimized for fast computation, the processing time could be further improved. This would allow on-line processing of a time-lapse or high-content experiment during acquisition, which offers powerful options. For example, a researcher could check population doubling times, and thus cell health during a time-lapse experiment, or acquisition could be stopped automatically when a certain number of cells is reached in the experiment.

The robustness of our method relies on the out of focus acquisition of brightfield images, which results in very well defined cell outlines but also in the loss of textural complexity for single cells. However, by acquiring an additional image with an optimally set focus in every interval, the quantification of morphological features such as shape and texture becomes feasible. Together with the high accuracy in cell detection of our method, this will support the development of automatic tracking approaches in time-lapse microscopy. For both time-lapse and high-content screening experiments, morphological quantification of millions of cells in one experiment allows the application of machine learning methods to classify, e.g., dying and surviving cells after drug treatment or the fate of differentiating stem cells.

Our method will be improved and adapted in the future. A promising avenue is the extension of the MSER segmentation algorithm to include more cellular features, like eccentricity or size.

Another possible improvement in our method is the splitting of clumped cells. Many methods have been developed in the past, e.g. ellipse fitting that is well suited to split nuclei or cells with round morphology (Bai et al., 2009). Unfortunately, the restrictive assumptions in this method do not allow more complex cell shapes that may emerge during a long-term movie of differentiating cells. We showed that our method performs well at the segmentation of hematopoietic stem and progenitor cells, which show round morphology. Still, the marker based watershedding we used is flexible enough to also cover more complex cell shapes that are appearing later in the differentiation process. Li et al. proposed a method based on gradient flow tracking and showed that it performs well on fluorescent images with hundreds of stained nuclei that are densely packed and are thus exhibiting different morphologies (Li et al., 2008). Another approach could include the development of a robust and fast performing level set evolution method. This class of algorithms has already been shown to perform very well on complex cell shapes (Li et al., 2010), however the computional complexity hinders an application in a high-throughput context. An approach that was already applied on high-throughput screens is to iteratively learn the different cell shapes of a given cell type or system in an experiment (Jones et al., 2009). Due to the modular structure of our method, the extension with algorithms that are able to split cells with a more complex morphology is easily possible. In the time-lapse experiment that was used in this work, these improvements could specifically enhance the cell detection accuracy for differentiated blood cells at the end of the experiment.

In summary, we believe that the high overall robustness in cell detection as well as the fast processing speed of our method will be of great service for the analysis of high-throughput microscopy experiments.

Chapter 5

Prospective identification of hematopoietic lineage choice without molecular labeling

In this chapter we set out to develop a computational method that predicts the commitment of HSCs to GM or MegE lineage in high-throughput time-lapse microscopy experiments as discussed in Section 1.5.2. In contrast to the experiments used in Chapter 3 that observed non-dividing fluorescently stained cells over a time span of 60 minutes in a single field of view, the time-lapse experiments used here comprised a duration of up to 8 days with highly proliferating and differentiating hematopoietic stem cells that migrated freely under a grid of 72 fields of view (i.e. single images). The longer time of observation, the brightfield acquisition as well as the much higher number of cells (due to proliferation) introduced a couple of obstacles that we had to overcome to complete the task, for which we extened the developed methodology from Chapters 3 and 4. We conducted the project in close collaboration with the group of Prof. Dr. Timm Schroeder from D-BSSE Basel (ETH Zurich).

We defined five milestones in order to complete the project: (i) Creation of three independent long-term time-lapse experiments with appropriate image quality as discussed in Chapter 4, (ii) generation of at least 100 hematopoietic pedigrees with annotated lineage commitment to build the dataset, (iii) robust identification of cell somata in the brightfield channel for every frame and field of view, (iv) quantification of morphology and motility for full cell trajectories in the generated pedigrees and (v) development of a machine learning method that would derive descriptive patterns from quantified morphology and motility information. The final model would have to be able to predict a cells most likely lineage decision on a fresh experiment that was not used for training. Finally, to be able to predict lineage choice on-line, the model should require as few observations of a single cell as possible for a correct prediction.

This chapter is based on and in part identical with the manuscript "**F. Buggenthin***, F. Buettner*, M. Kroiss, M. Strasser, P. Hoppe, M. Schwarzfischer, T. Schroeder, F.J. Theis* and C. Marr*. Prospective identification of hematopoietic lineage choice without molecular labeling. *Under review at Nature Methods*". Note that this project was a joint effort. The author of this thesis created the paralellized and automatized image processing framework and derived the morphological information, performed extensive data cleaning and annotation, parts of the manual tracking and trained the conventional machine learning methods. Florian Buettner designed, implemented, trained and tested the convolutional neural network.

5.1 Experimental setup

The time-lapse experiments that were the basis of this study were generated by Dr. Philipp S. Hoppe from the D-BSSE at the ETH Zurich following the protocol described in the next paragraphs. Note that in comparison to the protocol we used in Chapter 4, some steps such as the culture medium or acquisition intervals were tweaked or changed. For more details about wetlab materials and methods, see (Hoppe et al., 2016).

5.1.1 Purification of primary murine hematopoietic stem cells

Pelves, femurs, tibiae, humeri and vertebrae were extracted from a transgenically modified mouse line with knocked-in yellow (eYFP) and red (mCHERRY) fluorescent proteins at the gene sequences of PU.1 and GATA1, respectively. After isolation, the cells were incubated with anti-CD16/32 antibody. HSCs were then purified with flow cytometry to 40%-60% following a combination of protocols described in Kiel et al. (2005) and Osawa et al. (1996). Directly after sorting, cells were plated out on a plastic slide (µ-slide VI coated with Fibronectin, Integrated BioDiagnostics GmbH, Munich, Germany) with two physically separated wells in serum-free medium (StemSpan SFEM, StemCell Technologies) supplied with cytokines that only promote differentiation towards myeloid cells. All animal experiments were performed in compliance with the institutional guidelines of the Helmholtz Zentrum München and the regulations of the State of Bavaria.

5.1.2 Time-lapse microscopy

For each experiment, two wells of the plastic slide were subdivided into $72-78$ overlapping fields of view. Each field of view corresponded to an image comprising 1388 × 1040 pixels that was saved in 8-bit png format (1 px was equal to 1.0238 µm). Images were acquired using Cell Observer microscopes (Zeiss), equipped with a 0.63x TV-adapter (Zeiss), Axio-CamHRm cameras (Zeiss) and 10x Fluar objectives (Zeiss). Microscopes were surrounded by an incubator to keep a constant temperature of 37°C, cells were maintained in 5% CO2. Each field of view was imaged in intervals of 60-120 seconds (brightfield channel), 25-40 minutes (PU.1eYFP and GATA1mCHERRY channels) and 120-240 minutes (CD16/32 channels) for up to 8 days. Automatic focusing was achieved using a hardware autofocus (Zeiss) which was set to 18 µm below the optimal focal plane to acquire slightly blurred images optimal for cell detection (Selinummi et al., 2009). Three time-lapse experiments were used in this study, comprising a total size of 1TB of disc space.

5.2 A pipeline to quantify morphology and motility in hematopoietic genealogies

To identify all cell centers and somata outlines in the brightfield images of every frame and every field of view in all experiments, we extended our previously developed segmentation method as described in Chapter 4 to a full quantification pipeline. The changes and improvements to the method are specified in the different modules in the following paragraphs.

5.2.1 Illumination correction

To resolve illumination differences in single fields of view of the plastic slide and over time, we normalized the background of all images using the method from Schwarzfischer (2013) as described in Section 2.1.3. In difference to the original method, we used a random forest model that we trained with an active-learning approach to classify whether pixels belong to foreground or background. We trained classifiers to estimate the background image $\mathbf{D}^{\mathrm{bg}}$ for both fluorescent channels and the brightfield channel independently, but used the same classifiers for all experiments. The corrected image $\mathbf{D}^{\mathrm{corr}}$ was then derived by pixelwise division of the raw image $\mathbf{D}$ through the background image:

$$\mathbf{D}^{\text{corr}} = \frac{\mathbf{D}}{\mathbf{D}^{\text{bg}}}. \tag{5.1}$$

Finally, we enhanced $\mathbf{D}^{\text{corr}}$ by contrast stretching as described in Section 2.1.2 and normalized the interval of intensities to [0 255]. We computed a single background image for the first frame of every field of view which we used also for all subsequent frames, respectively. While this approach resulted in a great reduction of total computing time, we could not detect a decrease on the robustness of cell identification (see Section 5.2.7).

5.2.2 Identification of cell somata

We used Maximally Stable Extremal Regions (MSER) to separate foreground objects (i.e. cells) from background in all brightfield images. We filled holes in foreground regions, i.e. background pixels with no connection to an image border by morphological erosion. Additionally, we resolved open cell boundaries and single pixels connecting two adjacent foreground regions by applying two rounds of morphological opening and closing with a disk-shaped structuring element. Foreground objects that were too small or featured very elongated shapes (most likely fragments of the coverslip border) were discarded.

We filtered the resulting binary mask $\mathbf{D}^{\text{seg}}$ for cell fragments and dirt particles and discarded all foreground objects with an area smaller than 50 pixels or with an eccentricity greater than 0.99. We determined both values empirically, based on randomly drawn images of the data set. In addition, all objects that were touching an image border were removed.

5.2.3 Manual tracking of genealogies

Several studies have shown that to date no automatic tracking algorithm exists that robustly generates trajectories of differentiating hematopoietic stem cells from high-throughput brightfield microscopy (Schroeder, 2011; Meijering et al., 2009). Yet, as the Trackmate program we used in Chapter 3 produced accuracies of over 90% for short-term fluorescence time-lapse experiments, we evaluated the tracking performance on image stacks of the same length and size (i.e. 60 minutes duration, single field of view), for long-term high-throughput time-lapse microscopy.

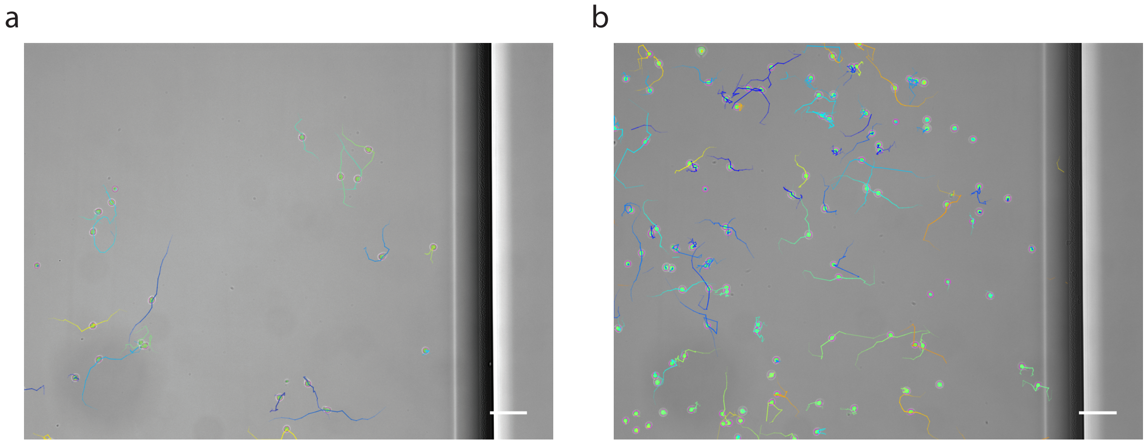


Figure 5.1: Autotracking correctness decreases in dependence of cell density and experiment length. (a) Visualization of tracked single cells with a low cell density (22 ± 8.7 cells per 0.1mm^2). The percantage of correctly identified trajectories was at 73 ± 5% over three stacks. (b) Single-frame visualization of tracked single cells for high cell density (59.5 ± 12.87 cells per 0.1 mm^2). The percentage of correctly identified trajectories dropped to 15 ± 5% over three stacks. Traces are shown for 10 time points. Scale bars represent 100 µm.

Applying Trackmate on a set of 3 imagestacks with a density of 22 ± 8.7 cells per 0.1 mm^2 and 59.5 ± 12.87 cells per 0.1 mm^2 revealed that 73 ± 5% and 15 ± 5% were correctly tracked for the low and high cell density, respectively (see Figure 5.1). This example shows that even for this extremely reduced level of complexity, sophisticated tracking algorithms that are capable of detecting cell divisions and powerful enough to track hundreds of cells simultaneously are not performing good enough to track whole genealogies in long-term high-throughput time-lapse experiments fully automatically. We thus followed a manual tracking approach that is well established in our labs by using the custom written software TTT (see Section 2.5.6) to built a pedigree of a hematopoietic stem cell and their progeny (Eilken et al., 2009; Rieger et al., 2009; Hoppe et al., 2016). We stopped tracking if i) a cell divided after a lineage marker was identified, ii) the cell died, or iii) a cell was lost (cell moved beneath the coverslip border or obscured exact position due due to insufficient temporal resolution). Note that our results also showed that while an autotracking approach might not perform well enough to track entire pedigrees over long time spans and in high cell densities, a short tracking interval together with low cell densities could suffice to at least partially track single cells with sufficient accuracies.

5.2.4 Mapping of somata to track coordinates

Before we were able to quantify the morphology and motility of full cell trajectories, a mapping of the tracking coordinates to the identified somata per frame and cell was necessary. Due to inaccuracies of the manual tracking process, the track points were deviating up to 100 pixels from the true cell center. We thus automatically identified the center of the nearest cell soma for all frames in every cell track of a pedigree by the following iterative search procedure.

For every tracked coordinate, we identified the nearest foreground object (i.e. cell) in $\mathbf{D}^{\text{seg}}$ by a window around the coordinate with an initial size of 30 × 30 px (the approximate radius of a cell) that was iteratively extended by 10 × 10 px. If an object was found, a binary image patch $\mathbf{P}^{\text{cell}}$ of 41 × 41 px around the object was extracted from $\mathbf{D}^{\text{seg}}$ and the procedure was continued. If no cell was identified after 12 iterations the frame of the track was skipped and labeled as missing.

In the extracted image patch $\mathbf{P}^{\text{cell}}$, we separated large objects that presumably represented mitotic cells or cell clumps by applying a modified version of marker based watershedding. While we applied watershedding to the full image in Chapter 4, the procedure described here reduced the computational burden. Also, instead of using ultimate erosion to define the seed points for watershedding, we identified ellipsoid structures with a maximum radius of 30 pixels (the approximate radius of a hematopoietic progenitor cell) by applying a circular Hough transform. After computing the distance transform $\mathbf{P}^{\text{dist}}$ of the binary patch, all detected centroid pixels that were found by the hough transform in $\mathbf{P}^{\text{dist}}$ were set to zero and watershedding was applied, resulting in the binary image with splitted objects $\mathbf{P}^{\text{watershed}}$. If a foreground object was split the image patch $\mathbf{P}^{\text{watershed}}$ was rearranged to the retained cells center of mass and all pixels not belonging to the identified cell were set to zero.

5.2.5 Quantificaton of cell trajectories

We derived a set of 66 features (14 basic measurements, 3 Ray features, 13 Haralick texture features, 2 Gabor wavelet features, 5 Tamura features, cell speed, 27 Zernike moments, 2 Gabor wavelets) from every extracted image patch $\mathbf{P}^{cell}$ (see Section 2.3 for details). If a fluorescence image was available at the given frame, we quantified the fluorescence signal by summing up all pixels within the segmented cell in the normalized fluorescence image. To normalize the amount of fluorescence, we divided the summed up signal by cell size.

To be able to filter quantification errors (clumped cells, dirt falsely identified as cell, cells lost due to border contact, over-segmented cell fragments) in a later data cleaning step, we fitted a B-spline to the quantified cell size over time by using the FDA toolbox (Ramsay et al., 2009).

5.2.6 Pipeline parameters

We used the following set of parameters for the processing of all experiments:

- Background correction

Parameter	Value
Tile dimensions	30 px
Overlap	15 px
Number of trees	1000
Class probability prior	0.5

- Detection of cell somata

Parameter	Value
Delta	40
Minimum Object Size	20 px
Maximum Object Size	4000 px
Maximum Variation	1
Morphological closing	1 px
Morphological opening	1 px
Filter objects (eccentricity)	> 0.99
Filter objects (size)	$< 30px$

- Mapping of somata to cell tracks

Parameter	Value
Initial size of search window	30 px
Growth of search window	10 px
Search Iterations	12
Size of extracted image patch	41 × 41px
Maximum Eccentricity	0.7
Minimum Object Size	30 px
Maximum Object Size	1000 px
Cell radius for CHT	25 px

5.2.7 Evaluation of single-cell quantification

We evaluated the general performance of our cell identification approach by quantifying and analyzing the cell size during cell cycle with a fitted B-spline function to the full single cell trajectory using the FDA toolbox (Ramsay et al., 2009). Since cell size can be expected to smoothly grow over a cell's lifetime, large residuals would suggest errors in cell identification (see Figure 5.2a-c). We thus computed the standard squared error for a single-cell trajectory as

$$\mathrm{SSE}(c) = \sum_{n=1}^{N} \big(g_{c,n} - f_c(n)\big)^2, \tag{5.2}$$

where N is the number tracked time points for cell c, $g_{c,n}$ is the measured size of the cell at time point n and $f_c(n)$ is the respective value of the B-spline. We then determined the contribution of the SSE to the quantified cell size:

$$\mathrm{ERR}_{\mathrm{size}} = \left(\mathrm{SSE}(c)\frac{N}{\sum_{n=1}^{N} f^{\mathrm{A}}_{c,n}}\right)100 \tag{5.3}$$

where f^{A} is the cell area as defined in Section 2.3. We found that only $\sim 5\%$ of the cells in our dataset exceeded an average cell size error of 25%, indicating an overall robust cell identification (see Figure 5.2).

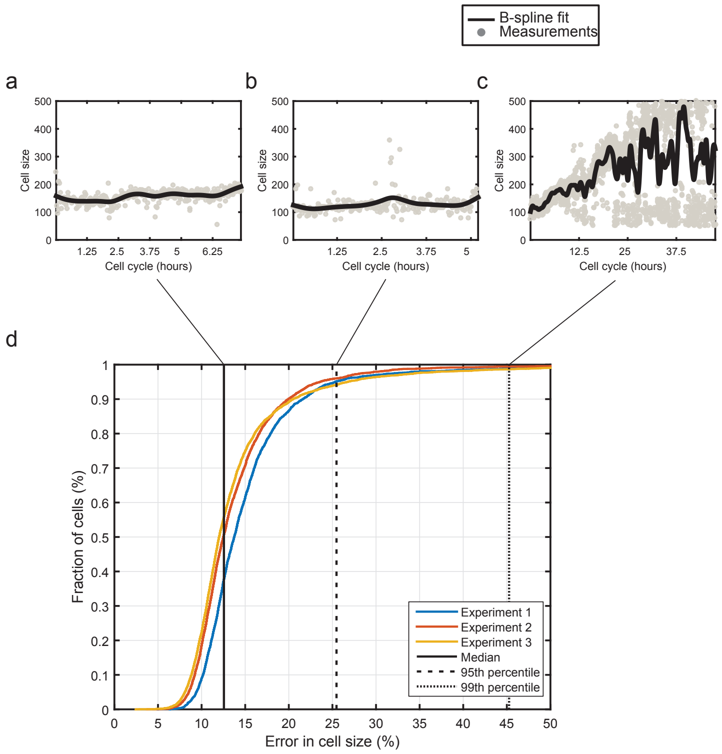


Figure 5.2: Quality and robustness of cell identification. (a) Cells with a size error of 10% – 20% show very few outliers and smooth size quantifications (cell taken from the Median, solid black line in (d)). (b) Cells with an error higher than 25% have a few measurement errors (most likely due to under-segmentation), yet the B-spline fit is smooth and the majority of residues is small (cell at 95th percentile, dashed black line in (d)). (c) If cell identification is erroneous, a high variance in cell size quantification results in a size error of more than 45% (cell at 99th percentile, dotted black line in (d)), which is unlikely for normal cell growth. (d) Fraction of all cells per error in cell size (root mean square error of B-spline fit divided by mean cell size), for each experiment (blue, red and yellow lines). For less than 5% of all cells in our dataset, the amount of cell size that was explainable by error was higher than 25% (dashed black line).

5.2.8 Annotation of lineage commitment

Based on the quantified concentration dynamics of CD16/32 and GATA1mCHERRY of a full branch (i.e. an annotated cell and all their predecessors) we annotated whether a cell was differentiated, using a custom-written user interface. We computed the concentration by dividing the summed up pixelwise fluorescence intensity of all pixels P in the mask of cell c at time point t in channel ch:

$$\operatorname{conc}(c, t, ch) = \sum_{p=1}^{P} \mathrm{Fl}_{t,ch}(x_{c,p}, y_{c,p}) \frac{1}{f^{\mathrm{A}}_{c,t}}, \tag{5.4}$$

where f^{A} is the cell area as defined in Section 2.3. We annotated lineage commitment when the respective lineage marker was detectable in the fluorescent channel (CD16/32 for GM and GATA1mCHERRY for MegE) and assigned all tracked cells to one of three categories (see Figure 5.3: i) "annotated" cells with clear marker expression, ii) "latent" cells with no immediate marker expression but an expression in a subsequent generation, and iii) "unknown" cells with no marker expression in current or subsequent generations.

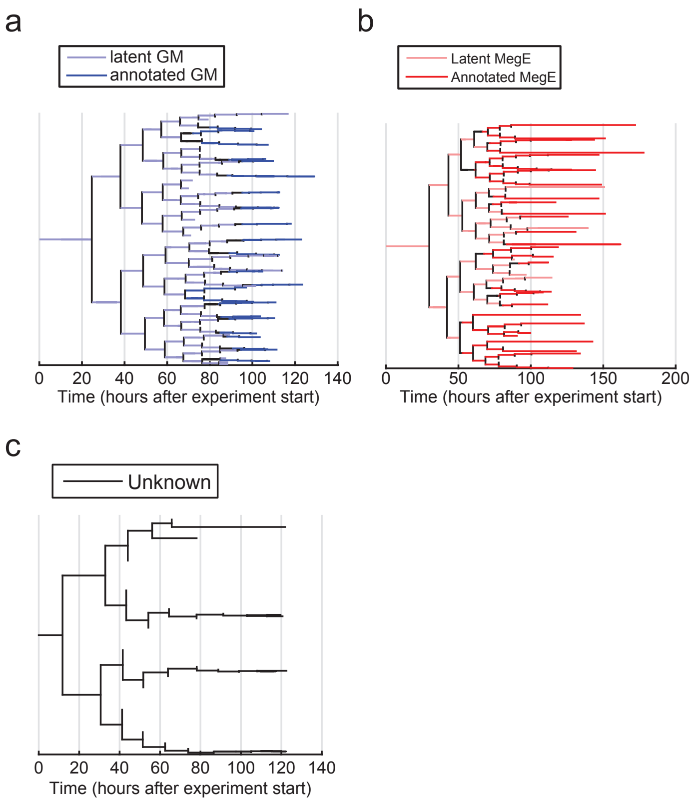


Figure 5.3: Lineage annotation and dataset overview. A single hematopoietic stem cell at the root of a genealogy gives rise to an exponentially growing number of successors via repeated cell division. (a) Genealogy featuring cells committed to the granulocytic/monocytic lineage (GM) with annotated (dark blue lines) or latent (light blue lines) CD16/32 antibody onset. Cells in branches with no annotated onset but annotated sister branches were also labeled as latent. (b) Genealogy featuring cells committed to the megakaryocytic/erythroid lineage (MegE), with annotated (dark red lines) or latent (light red lines) GATA1mCHERRY expression and/or characteristic megacaryocytic morphology (large cell body, multiple nuclei). Cells residing in branches with no annotated onset but with annotated sister branches are also labeled as latent.

5.3 Dataset overview

After the full procedure of manual tracking, automatic idenfication of cell bodies and semi-automatic lineage annotation, we received robust morphological quantifications for entire branches of a hematopoietic pedigree for GM (see Figure 5.4a) and MegE committed cells (see Figure 5.4b).

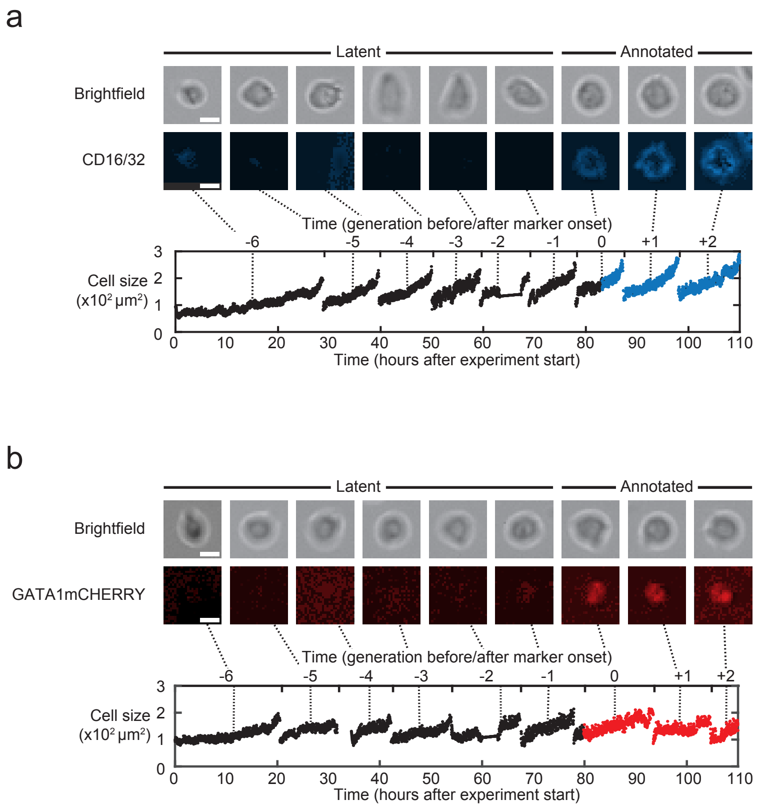


Figure 5.4: Exemplary image patches of a full branch of single cells committed to either GM (a, upper row) or MegE (b, upper row) lineage. Differences in morphology are hardly visible before a marker onset (latent state), and even after a lineage is annotated. (a,b, lower rows) Result of the automatic image processing pipeline (see supplementary Note 1 for details). The high accuracy of cell identification in all brightfield frames for a tracked branch allows robust quantification of morphology and motility dynamics for both GM- (b, lower row) and MegE-committed (c, lower row) cells, demonstrated here for cell size. Scale bars represent 10µm.

After filtering for all unknown cells (containing both uncommitted cells and committed cells for which the markers had not yet switched on), the dataset to train and evaluate our method comprised 4402 single cells ($\sim 1,700,000$ image patches) with annotated or latent marker onset (34% MegE and 66% GM). See Table 5.1 for a detailed overview of the full dataset.

# Genealogies	MegE	GM	GEMMeg	Unknown	Sum
Experiment1	16	17	4	20	57
Experiment2	7	24	5	21	57
Experiment3	20	14	1	1	36
Sum	43	55	10	42	150

# Cells	MegE	GM	latent MegE	latent GM	latent GEMMeg	Unknown	Sum
Experiment1	248	386	206	367	21	612	1840
Experiment2	315	510	124	286	15	932	2182
Experiment3	341	489	236	696	1	137	1900
Sum	904	1385	566	1349	37	1681	5922

# image patches	MegE	GM	latent MegE	latent GM	latent GEMMeg	Unknown	Sum
Experiment1	71726	86696	56715	88529	6144	177873	487683
Experiment2	169186	201550	71595	141736	10347	528700	1123114
Experiment3	149833	224077	115799	310385	960	65933	866987
Sum	390745	512323	244109	540650	17451	772506	2477784

Table 5.1: Detailed listing of the number of genealogies, single cells and single cell observations (i.e. image patches) with defined annotations that were used in this study. Note that a genealogy can feature a mixture of the annotations explained in Figure 5.3, called GEMMeg.

5.4 A machine learning method to predict the hematopoietic lineage decision

The information in our derived dataset spanned a highly complex, nonlinear spatiotemporal input space. Thus in a next step we developed a machine learning method to robustly and efficiently predict the lineage choice of a given single cell. Recapitulating our initial requirements for the method, we focused on the following capabilities: (i) General applicability, i.e. training on other cell types or processes should be possible, (ii) on-line prediction, i.e. prediction of a partially tracked cell during experiment duration and (iii) Report of feature importance, i.e. which measure such as size, eccentricity or speed is most influential on model performance.

5.4.1 General design considerations

We decided to develop a method that, given a single cell trajectory should predict the most likely lineage choice. As the derived dataset contained a subset of cells with an annotated lineage based on identified marker expression we chose to employ a supervised classification method with a binary outcome, i.e. is the cell committed to MegE or GM lineage?. Note that we deliberately did not include a “uncommitted” or “undecided” class in the model, since the time-lapse experiments were designed in a way that all cells eventually differentiated into GM or MegE lineage, even if a marker could not be identified until the experiment was stopped. Thus, our dataset lacked a labeled subset of "uncommitted" cells and training on this class was not possible.

Single-cell vs. whole pedigree information As discussed earlier, an algorithm that automatically and correctly tracks a full hematopoietic pedigree does not exist, i.e. during on-line prediction correctly tracked predecessors of a cell of interest would not be available (see Section 5.2.3). Yet, our tests also showed that short tracks of single cells might be accurate enough to predict its lineage choice. We thus decided to only incorporate the information about a single cell trajectory into the classification model, thereby discarding information about predecessor, sister or niece cells.

Raw pixel information vs. high-level features In general, there are two approaches to train a classification model on an image-based dataset. On the one hand, the raw pixel information has to be transformed to high-level features that describe the shapes or textural patterns in the image, and the model is then trained based on a vector of these features per data point. This approach was followed in several well performing models that were used for e.g. the general prediction of different cell types in fluorescence microscopy (Shamir et al., 2008), the classification of cell cycle phases (Held et al., 2010) or the prediction of differentiation decision of retinal progenitor cells (Cohen et al., 2010). Yet, the set of features has to be designed manually, which often involves a thorough understanding of the studied cell type or system that is not always available. In addition, the derivation of high level features adds computation time to the processing pipeline. As the computation of some features such as Zernike moments is computationally demanding, this approach would lead to a slower on-line evaluation. We thus decided to train our method based on the raw pixel information. This approach allowed a general application of the model even for different cell types and the on-line prediction would be faster.

Incorporating temporal dependencies Standard classifications methods such as support vector machines, random forests or neural networks regard the set of samples that are to be classified to be i.i.d., meaning that there are no dependencies of one sample to each other. In our case however, the single frames that comprised a cell trajectory were temporally dependent on each other. On the one hand, using a classification method that natively models the temporal dependencies such as a Hidden Markov model (HMM) would potentially lead to a better model performance, as was shown by Held et al. (2010). The authors first trained a SVM to predict the cell cycle stage of a given fluorescence time-lapse microscopy image frame, then used a HMM to incorporate the temporal information of a full cell trajectory and thereby increased the model performance. On the other hand, requires a single trajectory to change its state during a single cell cycle, which was not the case in our application. In addition, if a cell would be only partially tracked during an on-line prediction, the model would not be applicable.

We chose to develop a two-stage approach, that is similar to the the method described in (Held et al., 2010). First, a classification method should predict lineage commitment of a given raw image patch on a frame-by-frame basis. In a second step, the predicted class for every frame of a cell trajectory should then be averaged to get a single prediction per cell. This procedure is also comparable to ensemble learning, as discussed in Section 2.4.2.

Note that in a side project we used recurrent neural networks (RNNs) to directly integrate the temporal dependence of single frames for a given cell trajectory in the model. The results are discussed in detail in the thesis of Kroiss (2014). Yet, this approach was only possible by using the measured high-level features, not raw pixel information.

5.4.2 Convolutional neural network

A class of machine learning algorithms that satisfied all of our design considerations were convolutional neural networks (CNN, see Section 2.4.2). The network consists of several layers that can extract meaningful patterns from the raw pixel information of an input image. In addition, the architecture of such a method is flexible enough to introduce new layers that allow to incorporate information that is not directly related to the raw image, such as cell movement. We built a model from the LeNet family, which performed well in several recent image classification challenges (Lecun et al., 1998; Ciresan et al., 2011; Krizhevsky et al., 2012; Erhan et al., 2013; Girshick et al., 2014).

The CNN is structured in two parts of processing layers. First, convolutional layers extract information from spatially connected input nodes such that the neural network can learn local patterns from the image. These local, automatically generated image features are iteratively passed through several connected convolutional layers, resulting in increasingly global representations of the images (see Figure 5.5). Each convolutional layer is followed by a non-linear activation function. We chose Rectified Linear Units (ReLU), which have been shown to introduce non-linearities without suffering from the vanishing gradient problem (Nair and Hinton, 2010). In addition, we used max-pooling layers reducing variance and increasing translational invariance by computing the maximum value of a feature over a region (Ranzato et al., 2007) and dropout layers. Next, a fully connected layer is introduced to avoid overfitting. By using this archtitecture we followed largely Ciresan et al. (2011), where is shown that this combination of layers results in

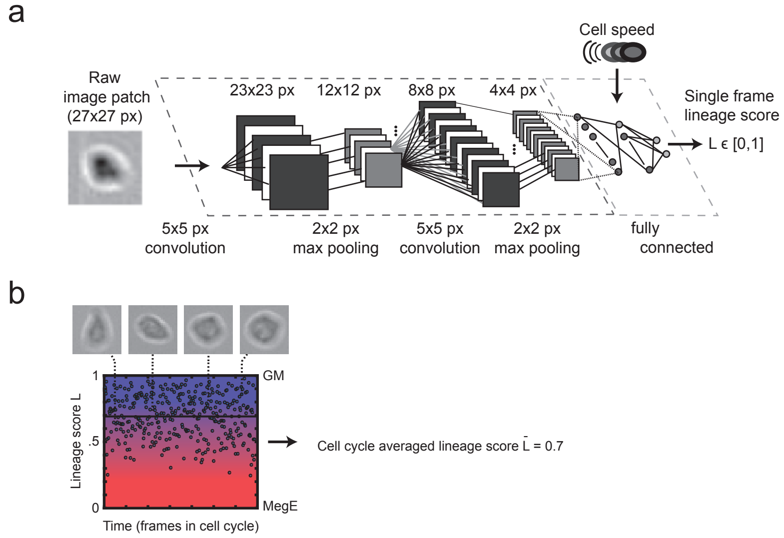


Figure 5.5: Machine learning architecture. (a) For each frame of a tracked cell, the raw image patch and the instantaneous cell speed are fed into a convolutional neural network (CNN, see Section 2.4.2). Every frame is assigned a lineage score L between 0 and 1 (0 = MegE, 1 = GM, 0.5 = unsure). (b) To improve classification performance, we average all lineage scores L for all frames of a cell cycle, resulting in a lineage score $\bar{L}$ for every cell in the dataset. The shown prediction profile is shown exemplarily for the cell at generation -3 in Figure 5.4a.

fast training times and good performance on a variety of image classification data sets. As a convolutional net allows no direct inclusion of features other than pixel information, we introduced a concatenation layer which combines spatial features with cell speed (see Figure 5.5a).

We used a standard softmax loss function and trained the network using stochastic gradient descend with stratified batches of 128 images (for details see Section 2.4.2). We initialized all weights in the network using the Xavier algorithm which automatically determines the scale of the initialization based on the number of input- and output nodes randomly (Glorot and Bengio, 2010). We used standard values for the base learning rate (0.01), momentum (0.9) and the weight decay (inverse decay with $\gamma = 0.0001$ and $power = 0.75$. We then multiplied the base learning rate after i iterations with $(1 + \gamma i^{-power})$ (Jia et al., 2014).

In order to classify individual cells as committed to either lineage, we first generated a lineage score L for each image patch and then averaged all scores resulting in a lineage

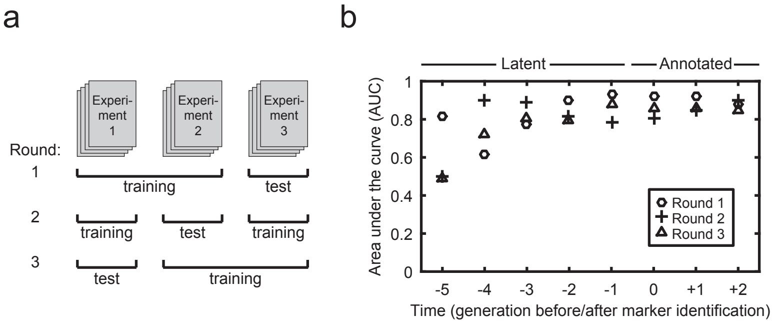


Figure 5.6: Train-test procedure to evaluate our method. (a) We used two experiments for training while one experiment was left out to assess generalization quality of the learned model. We repeated this procedure 3 times in a round-robin fashion, and received a lineage score L for every latent and annotated cell in the dataset. (b) Area under the receiver operating characteristics curve (AUC; 1.0 = perfect classification, 0.5 = random guessing) to determine the performance the trained models. Annotated cells (generations 0,+1,+2) and latent cells up to 3 generations before a marker onset (generations -3,-2,-1) show good AUCs higher than 0.77 (n=3 experiments, 4204 single cells). Rounds refer to the individual cross-validation runs from (a).

score $\bar{L}$ for every cell (see Figure 5.5b).

5.4.3 Training and evaluation

In order to avoid over-fitting we divided the training data into a training and validation set and optimized the weights until the performance on the validation set started to degrade (early stopping). All images were normalized to mean zero and unit variance, allowing for online predictions of cells directly during experiment duration, without having to normalize for possible batch effects.

To assess the generalization power of our model to reliably predict a cell's putative lineage choice in independent experiments, we trained the CNN on 2 experiments and tested the resulting model on the third experiment; we repeated this procedure 3 times in a round-robin fashion (see Figure 5.6a) and evaluated the performance of the trained model by the area-under-the-curve (AUC) of the receiver-operating characteristic (see Figure 5.6b).

Our method achieved high AUCs of 0.86 ± 0.02 (mean ± s.d., $n_E = 3$ rounds) on annotated cells, indicating that morphology and speed suffice to detect the lineage choice of hematopoietic progenitor cells. Interestingly, the reported AUCs for latent cells were

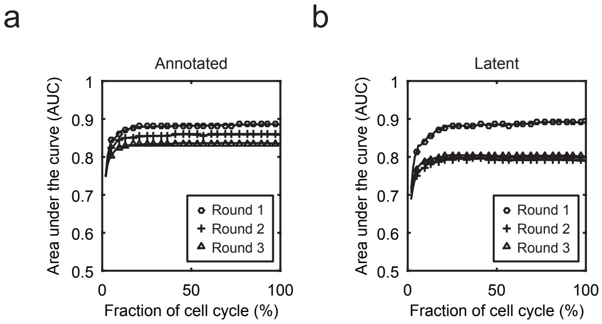


Figure 5.7: Average AUCs when only (contiguous) subsets of the available image patches are used to compute the averaged lineage score L. AUCs of the trained models for each train-test round reach a plateau when after using the first 25% of a cell cycle for averging the lineage score. This holds true for all annotated (h) as well as all latent cells from 3,2 and 1 generations before marker identification (i), respectively.

also high (0.78 ± 0.04, mean ± s.d., $n_E = 3$), suggesting that latent cells are morphologically different before an identifiable marker expression (Supplementary Fig. 3a). We further investigated this finding by analyzing AUCs for every generation separately (Fig. 1g). AUCs stayed at comparable levels from one to three generations before an annotated marker onset ($0.86 \pm 0.07, 0.83 \pm 0.06$ and 0.81 ± 0.06, mean ± s.d., $n_\mathrm{E} = 3$). At four and five generations before the marker onset the decline of AUCs (0.75 ± 0.14 and 0.60 ± 0.19, mean ± s.d., $n_\mathrm{E} = 3$) suggested that the difference in morphology and speed did not longer suffice to correctly identify GM- and MegE-committed cells.

To assess the performance of our CNN when a cell is not tracked over its full cell cycle, we computed AUCs using a growing subset of all available lineage scores per cell to compute the averaged lineage score L. Consistently for all three train-test rounds, improvement of AUCs reached a plateau after using 25% of the cell cycle from annotated (see Figure 5.7a) and latent (see Figure 5.7b) cells 3,2 and 1 generation before the identified marker onset, respectively.

5.4.4 Comparison to feature-based classification methods

We compared the performance of our CNN to a random forest model and a support vector machine (SVM). In addition, we evaluated algorithmic information theoretic prediction, a method that was specifically developed to predict self-renewing versus terminal divisions of retinal progenitor cells (AITP, Cohen et al. (2010)). AITP uses the full trajectory of

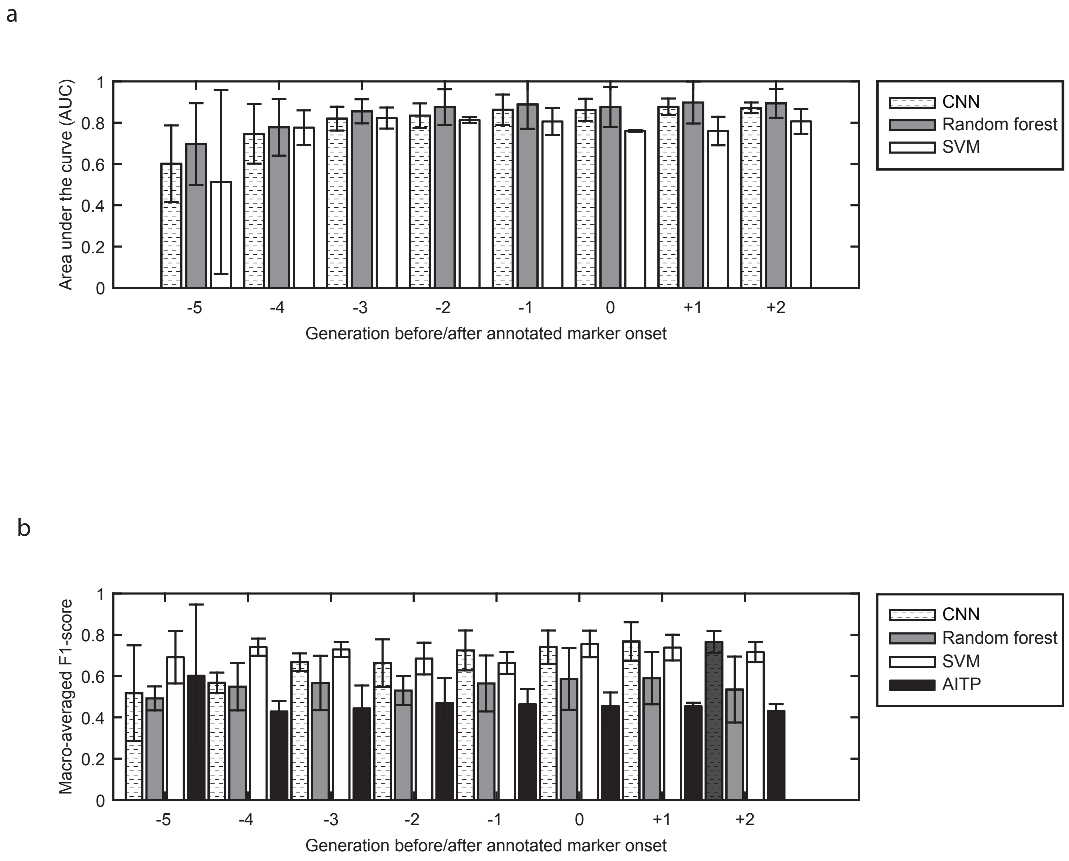


Figure 5.8: CNNs outperform feature-based methods in regard of calibration. (a) Area under the curve (AUC) for generations before and after an annotated marker onset for our CNN (dark gray bars), random forest (light gray bars) and support vector machine (white bars). Our method outperforms the SVM on annotated cells and is on par on latent cells, whereas random forest reports higher AUCs throughout the whole range of generations, yet with higher standard deviations (error bars). (b) Macro-averaged F1-scores (averaged lineage score threshold: 0.5) for generations before and after an annotated marker onset for our deep learning method (dark gray bars), random forest (light gray bars) and support vector machine (white bars), as well as AITP (black bars). Due to better calibration, our method outperforms random forest and AITP. SVM reports similar scores as our method on latent cells, but performs worse on cells with annotated onset.

a given cell to compute the normalized compression distance, a measure from algorithmic information theory, on multiple quantization levels.

A random forest classifier was trained with 200 trees and evaluated by out-of-bag prediction. In addition, we chose BudgetedSVMs (Djuric et al., 2013) with the Pegasos algorithm and a radial basis function kernel as a support vector machine framework that was able to deal with the millions of single-cell measurements in our dataset. We used a grid search with 5-fold cross-validation for every train-test combination to determine optimal hyperparameters. The best-performing model was then used for predicting the testset. Note that the hyperparameters for SVM had to be determined for every train-test run individually. We trained both methods with a set of 65 morphological features and cell speed (see above). We applied the same train-test procedure for model evaluation as for the CNN.

AITP was trained as described in Cohen et al. (2010), using a set of 6 features for each cell (movement, net movement, movement direction, area and eccentricity of fitted convex hull). As AITP was not able to process the the full dataset in a single run, we generated three subsets ($n_c = 400$ cells) for every train-test round, which we evaluated separately. We used the averaged evaluation results for comparison. As the used version of AITP reported class labels and no prediction scores, we used the macro-averaged F1-score for performance evaluation. It is worth noting that in contrast to all other methods, AITP inherently uses the full cell trajectory for training and prediction.

While our method outperformed the support vector machine on annotated cells and was on par on latent cells, we found the reported AUCs for random forest to be higher on both sets (see Figure 5.8a). However, the CNN achieved considerably higher macro-averaged F1-scores as the random forest and AITP (see Figure 5.8b), indicating a poorer calibration of these methods. This suggests that the CNN yields more reliable results when applied to new experiments that were not part of the training procedure.

Cell speed, size and mean intensity are most informative for the prediction of hematopoietic lineage decision

A downside of the convolutional neural network is that the automatically generated abstract feature layers that are used as features to train the model are hard to interpret biologically. Yet, knowing which features are most important for lineage prediction could support experimentalists in the design of novel experiments to study hematopoietic differentiation (Schroeder, 2011). Given the comparable performance on our data set we chose to evaluate the feature importance reported by the trained random forest models (see Section 2.4.2 for methodological details). We found that cell speed and simple morphological features (maximal/mean pixel intensity and cell size) are most important for correct classification (see Figure 5.9).

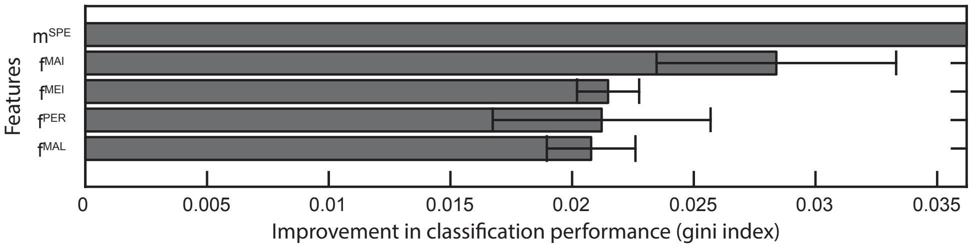


Figure 5.9: Importance of derived morphological features for classification performance. Gini importance for 4 most important morphology features as well as cell speed as reported by the random forest model, averaged over three train-test rounds (see Section 2.3 for details). Cell speed and simple morphological measurements (minimum and maximum intensity, perimeter, major axis length) have the most impact on classification performance. More sophisticated shape (zernike moments, ray features) or texture measurements (tamura and haralick features) are mostly irrelevant.

5.5 PU.1 expression of cells predicted to be MegE- and GM-committed agrees with prior knowledge

A possible application of our method is stopping a time-lapse experiment at any time before lineage markers are identifiable, measuring the expression of key master regulators and eventually assigning a lineage choice to single cells in accordance to L. This would allow to compare expression patterns of differentially committed cells. Here we analyzed PU.1eYFP, a transcription factor that was upregulated in GM-annotated cells (see Figure 5.10a) and downregulated in MegE-annotated cells (see Figure 5.10b), as expected from prior knowledge. In contrast, PU.1eYFP showed an intermediate distribution in cells with no annotated marker expression (see Figure 5.11a). If our proposed method was capable of reliably predicting lineage choice before approaches based on molecular markers, we should be able to in-silico stratify all cells in a stopped experiment (due to missing marker expression only “unknowns” and “latents”) into GM- or MegE-committed cells. These groups in turn, are expected to differentially express PU.1eYFP. Thus we classified every latent and unknown cell of experiment 3 into two groups by analyzing if L was above (GM) or below (MegE) a threshold of 0.5 (see Figure 5.11b). We found the two groups to differentially express PU.1eYFP from 3 generations after experiment start onwards ($P<0.05$ in gen. 3, $P<0.01$ in gen 4, $P<0.001$ in gen. 5-8, unpaired wilcoxon rank-sum test, see Figure 5.11c) with at least 70 cells per generation (Fig. 2e). As only 2 ± 1% (mean ± SD, nE=3) of GM and 15 ± 8% (mean ± SD, nE=3) of MegE marker onsets were annotated earlier than four generations after experiment start, our method is superior

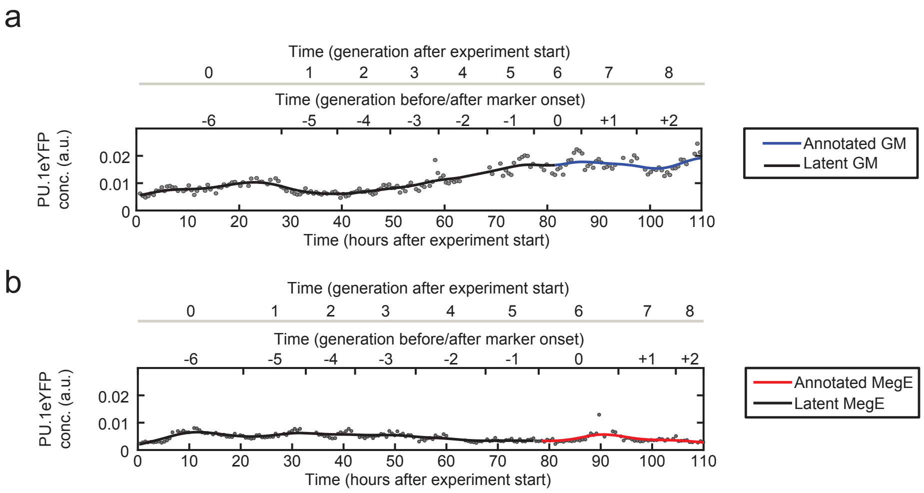


Figure 5.10: PU.1 expression of cells predicted to be MegE- and GM-committed agrees with prior knowledge. (a) Increase of PU.1eYFP concentration from intermediate to high level for a branch with annotated GM marker onset (blue line). (b) Decrease of PU.1eYFP concentration from intermediate to low level for a branch with annotated MegE marker onset (red line). Shown PU.1eYFP levels (black dots) are the sum of cellular fluorescence intensity at every time point, divided by cell size and fitted by a B-spline (black/colored line for latent and annotated lineage, respectively).

than the lineage identification with molecular markers.

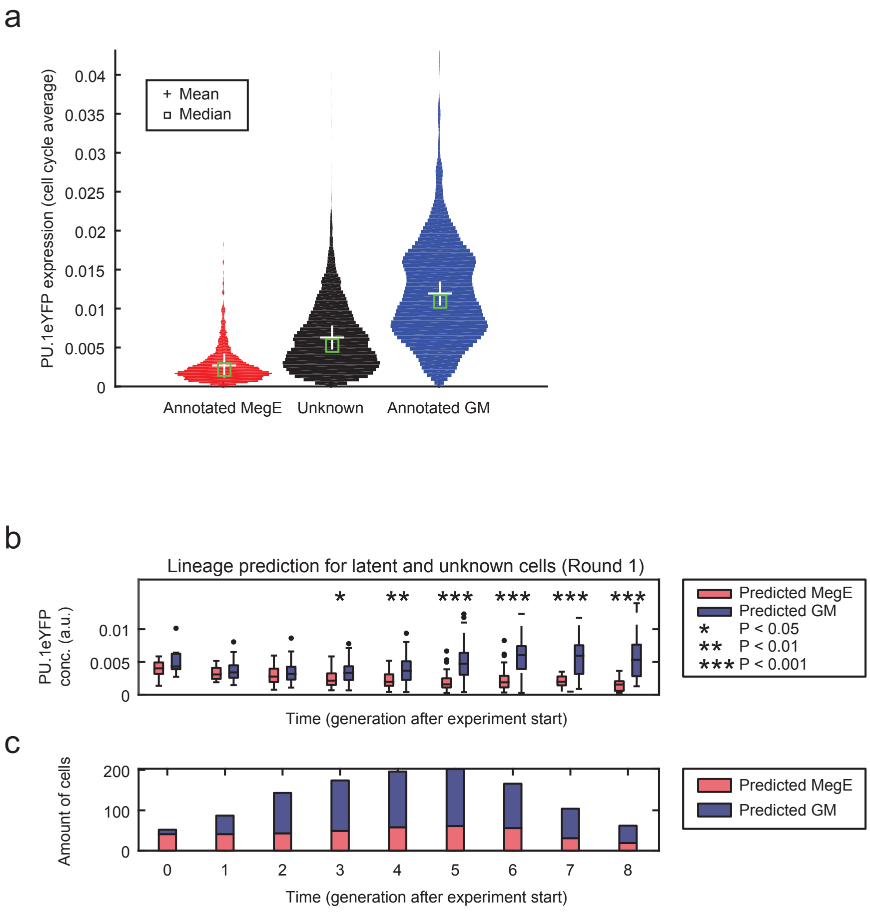


Figure 5.11: Figure 2: Subsets of cells with differential PU.1eYFP concentration can be distinguished 3 generations after experiment start without molecular labeling. (c) Cells without marker expression (unknown or latent fate, black patch) show an intermediate PU.1eYFP concentration distribution in comparison to cells with annotated marker onset for GM (blue patch) and MegE (red patch) lineage and are thus not differentiable without further labeling. (d) Concentration of PU.1eYFP for “unknown” and “latent” cells in generations after experiment start, subdivided into predicted GM (blue boxes) and MegE (red boxes) lineage, as reported by cell cycle averaged lineage score L. Observed trends in PU.1eYFP concentration are in good agreement with behaviour expected from literature. PU.1eYFP concentration is significantly different (black stars, P¡0.05 in gen. 3, P¡0.01 in gen 4, P¡0.001 in gen. 5-8, unpaired wilcoxon rank-sum test) between the two predicted groups. (f) Numbers of predicted GM (blue bars) and MegE (red bars) committed cells for every generation after experiment start. Generations with significantly different groups cover at least 70 single cells.

Implementation

Single-cell identification and quantification was implemented using MATLAB (R2014a). Code from Junior et al. (2009) was used to compute histograms of oriented gradients. All quantifications were parallelized on single-cell level and processed on a computation cluster (sun grid engine version 6.2u5). The average node architecture was equal to an Intel Xeon 2GHz, 4GB RAM running a 64bit linux-based operating system. Random forest classification was conducted with the python-based scikit-learn package (v0.15). The support vector machine was trained using the code provided with the original publication(CITE). AITP was trained using the latest version (april 1st, 2014) from the website of the authors after slight adaptation of input/output functionality to fit our data. To implement the deep neural network, we used the Caffe framework and trained it on a standard PC equipped with an Intel Core i7-4770 CPU, 32GB working memory and a 6GB Geforce GTX Titan Black graphics card.

5.6 Conclusion

Our method prospectively identifies lineage choice in hematopoietic progenitors and performs robustly on three independent time-lapse experiments. For future experimental applications, the brightfield-based prediction frees fluorescent channels that are currently used for lineage marker annotation. Moreover, the differential expression of PU.1eYFP in implies that our methods can be used to identify master regulators of lineage choice when large-scale expression profiling is performed. The use of convolutional neural networks matches the large amount of image data emerging in time lapse microscopy. Compared to other machine learning methods, our CNN approach is fast , independent of a cell-type specific, curated feature set and requires no high-level feature calculation. It is thus very versatile and can be applied to analyze branching processes in in biological systems where suitable feature sets are unknown, shown in other applications before (Mnih et al., 2015).

Chapter 6

Summary and Outlook

In this Chapter, we summarize the developed methods and biological insights we gained throughout our studies. We also critically discuss the limitations of our methodology and provide ideas for further improvements and future applications.

6.1 Summary

The goal of this thesis was the development of methods from bioimage informatics tailored to the application on time-lapse microscopy experiments and the analysis of morphodynamics of single cells. We contributed to the question, whether these non-molecular characteristics can be used to predict cellular state changes of T-lymphocytes and hematopoietic stem cells.

6.1.1 Analysis of T-lymphocyte migration

In a first project described in Chapter 3, we analyzed changes in the migration of T-lymphocytes in three independent experiments with different cellular confinement depths. Based on short sequences from fluorescence time-lapse microscopy we used a concatenation of available image filters and the MSER algorithm to detect nuclei and cell somata. While we found that the performance of our segmentation approach generally was on par with simple Otsu thresholding, we could show that our method was more robust in cases where very few foreground objects (i.e. cells) were present in the image. This finding is especially important for the analysis of time-lapse experiments, where the number of cells but also the image quality can hugely vary over the full experiment duration. A less robust method would lead to systematic errors, e.g. a high fraction of missed cells at the end of the experiment. In a next step, we adapted the tracking framework Trackmate to automatically generate trajectories of single cells over the full experiment with a correctness of over 95%. While the high amount of correctly tracked cells sufficed for the analysis we conducted in this thesis, an important feature of our pipeline is the possibility to manually assess the automatically generated trajectories if necessary. This allows a researcher to discard tracks of e.g. dead cells that would otherwise lead to erroneous measurements.

We analyzed the migration behavior of T-lymphocytes in the different confinement depths (i.e. 8 µm,6 µm and 4 µm) by fitting a Gaussian mixture model with a variational Bayesian approach (VB-GMM), which allowed us to determine the number of cellular subpopulations that are present in the data. Interestingly, we found two groups of motile and non-motile cells in every condition, although with varying abundances, suggesting that cells respond heterogeneously to the confined environment. While these groups could also have been detected manually by a human observer, our fully Bayesian approach allowed to exactly determine the average speed and the standard deviation for every group together with a confidence estimate. This quantitative measure adds a new level of detail to the analysis and could be useful when differences in the migration behavior of the analyzed cells are more subtle.

The detected cell somata together with the tracked cell trajectories allowed us to automatically determine a set of morphological features, as well as the distribution of actin inside the cell. In contrast to literature, neither a cell's eccentricity nor the amount of actin at the leading edge correlated strongly with cell speed. A technical explanation for this observation is that the level of precision of the eccentricity and actin measurement did not suffice to account for the complex shapes that T-lymphocytes adopt over time. Yet, we think it is worthwhile to continue this analysis with more experiments and an increased number of tracked cells, as the missing correlation could also be a consequence of a too small dataset. In a last analysis, we trained a regression model on the quantified morphology per cell to predict its movement on a frame-to-frame basis. While the model predicted the speed of fast and intermediate moving cells very well, the accuracy decreased for slow or non-moving cells. Since our VB-GMM approach identified two different groups of cells, one could split the dataset in accordance to motile and non-motile behavior and train two regression models separately for the group of motile and non-motile cells, respectively.

Taken together, the developed pipeline automatically quantifies the morphology and motility of T-lymphocytes in fluorescence time-lapse experiments with high accuracy and thereby enables to generate datasets with high cell counts, which will eventually increase the statistical power with new experiments. Together with Dr. Anne Reversat from the Sixt lab we started to conduct and analyze time-lapse microscopy series that feature more levels of confinement (i.e. 2µm or 10µm), but also different coating patterns and substrates of the coverslip to study T-lymphocyte migration under non-adhesive and adhesive environments. Our pipeline will be of great help to automatically process the high amount of time-lapse experiments that will arise due to the increased number of combinations of environmental factors and could lead to findings such as under-represented groups of cells that are not detectable by manual analyses alone.

6.1.2 Prediction of hematopoietic lineage choice

In the second project of this dissertation described in the Chapters 4 and 5, we assessed the feasibility to use the morphodynamics of single cells as a predictor for myeloid lineage choice of differentiating hematopoietic stem cells (HSCs) in long-term high-throughput time-lapse experiments. The long experiment duration and the fast cell motility made it necessary to acquire brightfield images at intervals of 60 seconds for cell tracking, as a high frequency of fluorescence images would have been toxic for the cells. As it was much harder to derive meaningful information such as cell shapes from this type of images as from a fluorescent image and reliable algorithms for robust cell detection in high-throughput experiments were not available, we developed and published our own algorithm. The steps that had the most impact on robustness of cell identification (at least 83% of all cells in a field of view at every frame) were on the one hand the adaptation of the microscopes focal plane during image acquisition. This increased the contrast of the cells to a level that spatial and temporal variances in illumination did not affect the segmentation performance. On the other hand, the robust MSER thresholding algorithm allowed us to process the complete dataset (i.e. 3 experiments, more than 1,000,000 images) in a single run with a single set of parameters. Finally, we could quantify the morphodynamics of hematopoietic stem and progenitor cells throughout the full differentiation process by coupling the derived cell outlines with manually tracked genealogies.

In a next step, we used the derived image patches with centered single cells to train a convolutional neural network (CNN) that automatically identified the important differences in the raw pixel information to discriminate between MegE- and GM-committed cells. Additionally, we introduced a concatenation layer to the classical network layout that allowed us to also use cell speed as a feature for model training. We showed that our method robustly detected the lineage commitment of single cells up to three generations before the expression of molecular lineage markers, even when the prediction was done on a completely unknown experiment. This implies that a model - once trained on a set of annotated data - can be applied on newly conducted experiments without generating a ground truth for every particular experiment. While we could not find a significant improvement in the prediction performance of our CNN in comparison to classical machine learning methods such as random forest and support vector machine that we trained with the derived morphological features, we believe that the higher robustness in terms of calibration and the faster computation of prediction scores will be viable in the application of the method in the experimental day to day work and will also allow on-line applications. Also, the CNN can be applied to different cell types where a manually derived set of morphological features might not be available. We hope that this will enable the conduction of high-throughput screening approaches, to e.g. test the influence of different growth factors or drugs on hematopoietic differentiation.

As a side note, our method also allowed to automatically and continuously quantify the expression of PU.1 and GATA1, which are thought to play a major role during differentiation by their fluorescence signals. Given that the currently established procedure to postprocess this data involves a manual inspection of every quantified time point per fluorescence channel (see the dissertation of Schwarzfischer (2013)), we believe that our automated technology can save a great amount of time in the analysis of gene expression dynamics on the single-cell level.

The in our opinion most important biological implication of the results we received in this project is that hematopoietic stem and progenitor cells show subtle differences in their morphodynamics during differentiation that allow to predict whether a cell is committed to the MegE or GM lineage. This means that cells might be committed to a certain lineage much earlier than what was expected previously. Also, the morphological differences could be used to study the molecular state of cells committed to different lineages at every time point during experiment conduction, for example via single cell sequencing (see Section 6.2.2). To demonstrate this possible use-case of our technology, we analyzed the differences in PU.1 expression of cells that were classified to be committed to MegE or GM lineage by our CNN. We found that PU.1 expression behaves as described in literature. Given a mouse line that fluorescently expresses another transcription factor of which the role in hematopoietic lineage choice is unclear, our methodology can be used to group single cells in accordance to their differentiation status, thereby revealing differences or patterns in gene expression that would otherwise not be detectable.

It is worth noting that both image processing pipelines are in large parts executable in parallel on a computation cluster, which greatly decreased the computation time for the detection of single cells and the quantifaction of morpholgy and motility. To that end, we designed and used a load balancing script that splits up the images of a long-term high-throughput time-lapse experiment in equally sized chunks and distributes it over a predefined number of grid notes. The same is possible for the quantification of morphological features on an extracted image patch for every time point of a tracked cell trajectory. In our case, the time required for processing was reduced up to 500-fold, which made many of the analyses and especially parameter tuning possible in the first place.

6.2 Outlook

While the methods we developed throughout the preparation of this dissertation were strongly tailored to the analysis of T-lymphocyte migration and the prediction of hematopoietic lineage choice, several modules in the created pipelines could be used with little adaptations for the analysis of a multitude of different biological experiments based on time-lapse microscopy. In the following paragraphs we will thus first disucss potential methodological extensions that could be beneficial for further analysis of time-lapse experiments. Second, we will provide ideas on how to design follow-up studies that use our methodology to answer biological questions.

6.2.1 Methodological extensions

Identification of single cells in multi-focal brightfield microscopy

A downside of the approach we described in Chapter 4 is the loss of textural detail in brightfield images due to a shift in the optimal focal plane during acquisition. This could be resolved by acquiring brightfield images in several focal planes, including the plane with optimal textural details (Selinummi et al., 2009). While the 72 fields of view that had to be scanned in short acquisition intervals of $\sim$ 60 seconds per frame did not allow to to pursue this approach in the experiments we used in this dissertation, this could be done by either using a motorized stage that scans the plate faster or by reducing the size of the coverslip. Also, for cell types that do not require high acquisition intervals due to slower motility (e.g. embryonic stem cells), the amount of brightfield images in different focal planes could be increased. This procedure could also help to correctly identify already differentiated and thus adherent cells that are residing in a different focal plane, which the algorithm we presented in Chapter 4 often segmented incorrectly.

Autotracking in high-throughput time-lapse microscopy

After parameter adjustment, the autotracking software Trackmate we used in Chapter 3 returned over 95% of correctly tracked trajectories in fluorescence time-lapse experiments. However, the correctness of the algorithm heavily decreased in the brightfield case (see Section 5.2.3). Initial tests that we have conducted throughout our studies suggest that Trackmate is capable of tracking full genealogies, as the underlying tracking algorithm from Jaqaman et al. (2008) can detect cell division events and also fills gaps in a cell trajectory that resulted from e.g. incorrectly segmented cells. We identified the following challenges to be resolved until reliable autotracks of full genealogies can be realized. First, the accuracy of cell detection in brightfield images has to be increased. While our proposed algorithm reported accuracies higher than 80% and split cells correctly after a division event, we found problems when cells were clumped together or became adherent and thus moved to a different focal plane. Yet, a correct splitting of cells especially in clumps of three or more cells is crucial to derive correct genealogies. In addition, in order to allow a global optimization of the tracklet linking (see Section 2.2) the amount of data points has to be reduced, as a matrix of $\sim 300,000,000$ single-cell measurements can not be optimized computationally. We therefore suggest to split this matrix into smaller overlapping chunks both temporally (e.g. intervals of 24 hours as this would represent two cell cycles) and spatially (as cells typically show fast motility but stay in one quadrant of the coverslip). This would create locally optimal solutions that can be connected to a full result afterwards. As errors in this process are inevitable, a measure of quality for the tracked genealogy would have to be introduced. Criteria that could be incorporated are the number of cells in close proximity for the tracked cell at every frame, the quality of segmentation or cell identification and also the time that has passed until the experiment was started (due to cell proliferation tracking accuracy decreases with a rising number of cells on the coverslip) (Meijering et al., 2009; Chenouard et al., 2014; Maška et al., 2014). These quality measure would also make it possible to manually assess parts of the tracked genealogy (e.g. using the user interface provided by Trackmate) that are problematic, without having to screen the full tree.

Convolutional recurrent neural network to predict lineage commitment

The convolutional neural network we developed in Chapter 5 is not able to inherently process time-resolved data. While the averaging approach we used to build the lineage score for a given input cell reported high AUCs and also has the advantage that cells can have arbitrary lengths (e.g. due to tracking errors), the temporal dependencies in a cell cycle could improve the methods overall performance. We thus propose to combine our results with the insights gathered with recurrent neural networks on this dataset as described in Kroiss (2014). A convolutional recurrent neural network could be generated to predict the lineage commitment not only of single cells, but also branches and genealogies. While the training of this class of models could be problematic due to the high model complexity, recent advances might help to tackle the upcoming challenges (Pinheiro and Collobert, 2013; Kalchbrenner and Blunsom, 2013; Donahue et al., 2014).

6.2.2 Biological/experimental extensions and follow-up studies

Prediction of HSC lineage commitment in microfluidic devices

In Chapter 5 we used long-term high-throughput time-lapse microscopy experiments to develop our processing pipelines and conduct the analysis. While this technology has proven to be a valuable tool to study the molecular and morphological dynamics of hematopoietic lineage choice there also exist several downsides. For example, the large coverslips that are necessary to culture the cells and the high motility rates of HSPCs render the development of an reliable autotracking algorithm extremely difficult (Meijering et al., 2006), thus the manual generation of pedigrees and annotation of lineage commitment will remain a huge bottleneck in the analysis of such experiments. In addition, analyzing gene expression by transgenically fused fluorophores is not well suited to study many molecular players in parallel. First, generating a transgenic animal (in our case: mice) is very challenging and can take years (Schroeder, 2010). Second, the number of fluorophores that can be analyzed simultaneously is restricted to ~ 10 color channels. Also, generating a knock-in model requires that a potential player is known beforehand, thus new players can not be detected.

A recently emerged experimental technique that has the potential to overcome these limitations are microfluidic devices (Sackmann et al., 2014). Here, single cells are maintained in small chambers on a tiny plastic chip that are connected by several channels and valves that allow to precisely control the environmental conditions such as the amount of growth factors. If a cell divides one daughter is channeled into another chamber, which allows to easily keep track of all successors of a single cell without tracking. While this approach has been shown for yeast cells by Huberts et al. (2013), unpublished results by the group of Timm Schroeder (D-BSSE at the ETH Zurich) suggest that this technique will also be applicable to observe hematopoietic differentiation. In addition, the recently presented commercial platform Polaris$^{\mathrm{TM}}$ developed by Fluidigm claims to combine microfluidic culturing of single cells in trap chambers, coupled with a downstream single-cell sequencing approach to analyze the expression of thousands of proteins at any given time point during differentiation. However, before the technique will be fully applicable for the long-term analysis of differentiating hematopoietic stem cells, several challenges have to be solved. This includes keeping the less robust stem cells healthy inside the microchambers, but also robustly triggering the switching of one daughter cells to a new chamber after a cell division. Also, the acquisition of brightfield images needs to be adjusted for microfluidic divices, as the microchannels in the field of view can lead to image artifacts that can hinder the correct identification of single cells.

Along this line, our developed methodology could be of great help in determining different time-points in the hematopoietic differentiation process at which the observed cells will be forwarded to further profiling, e.g. a sequencing pipeline. The convolutional neural network detects the lineage commitment of hematopoietic stem and progenitor cells up to three generations before conventional molecular markers, based solely on the pixel-based morphology information and speed that could also be acquired during the culturing step in a microfluidic chamber. Given that the prediction performance reached a plateau when we averaged the lineage score over at least 25% of a cell's full cell cycle (corresponding to ~ 90 individual measurements per cell in our experiments), and the fast frame-wise prediciton in milliseconds per frame, our method could be a great addition to the Polaris$^{\mathrm{TM}}$ pipeline.

Large-scale comparison of cell shape space

The robustness and processing speed of our methodology allows the automated identification of cells in millions of microscopy images, the generation of single-cell image patches as well as the extraction of features and classification with a convolutional neural network. A possible follow-up project could e.g. be an exploration of the total shape space in a high-throughput screening approach of brightfield images, where hundreds of different cell types are observed under the miscroscope. Comparable studies were already done in the Bakal lab for D. melanogaster hemocytes and neuronal cells (Yin et al., 2014). The authors found classes of conserved shapes, but also discuss that an exhaustive analysis of the shape space of a multitude of different cell types is still lacking. Along this line, Neumann et al. (2010) conducted a phenotypic profiling of all $\sim 21,000$ human protein-coding genes in fluorescence time-lapse experiments observing knock-out cell lines by quantifying their morphology. For an evaluation whether the CNN performs better than a feature-based machine learning method, the software WNTCHRM that computes an ensemble of 2500 morphological features could be used (Shamir et al., 2008). While to this date we are not aware of a high-throughput dataset that would be suitable for this project, databases like Morphobase (Futamura et al., 2012) or Morphobank (O'Leary and Kaufman, 2012) that are providing images of a multitude of different cell types are constantly growing. Also, publicly available datasets from already published projects like Neumann et al. (2010) or (Held et al., 2010) could serve as a proof-of-concept study to demonstrate the feasibility of the project to a potential collaborator.

In summary, we believe that the methods we developed throughout the preparation of this dissertation will provide vital support in the automated analysis of experiments that observe cell morphology and motility and thus enable the design of high-throughput experiments to study a multitude of different environmental influences or molecular processes in parallel. The analysis of cell morphology and motility can lead to discoveries in many biological processes, as we demonstrated with the analysis of T-lymphocyte migration and the prediction of hematopoietic lineage commitment. New experimental techniques such as microfluidics open up a large number of potentially interesting studies, and we are excited how the automated quantification of cell morphology will further contribute to the analysis of these experiments.

Bibliography

M. D. Abramoff, P. J. Magalhaes, and S. J. Ram. Image processing with ImageJ. *Biophotonics international*, 11(7):36–43, 2004.

B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Lymphocytes and the Cellular Basis of Adaptive Immunity.* Garland Science, 2002.

B. Alberts, A. Johnson, W. Peter, L. Julian, R. Martin, R. Keith, and O. Nigel. *Molecular biology of the cell, 5th edition*, volume 5. Taylor & Francis Ltd., 2007.

R. Ali, M. Gooding, T. Szilágyi, B. Vojnovic, M. Christlieb, and M. Brady. Automatic segmentation of adherent biological cell boundaries and nuclei from brightfield microscopy images. *Machine Vision and Applications*, 23(4):607–621, may 2011.

L. Almeida. The fractional Fourier transform and time-frequency representations. *IEEE Transactions on Signal Processing*, 42(11):3084–3091, 1994.

M. E. Ambühl, C. Brepsant, J.-J. Meister, A. B. Verkhovsky, and I. F. Sbalzarini. High-resolution cell outline segmentation and tracking from phase-contrast microscopy images. *Journal of microscopy*, oct 2011.

D. Anselmetti. *Single Cell Analysis: Technologies and Applications.* John Wiley & Sons, 2009.

J. Back, D. Allman, S. Chan, and P. Kastner. Visualizing PU.1 activity during hematopoiesis. *Experimental Hematology*, 33(4):395–402, apr 2005.

X. Bai, C. Sun, and F. Zhou. Splitting touching cells based on concave points and ellipse fitting. *Pattern Recognition*, 42(11):2434–2446, nov 2009.

A. J. Becker, E. A. McCulloch, and J. E. Till. Cytological demonstration of the clonal nature of spleen colonies derived from transplanted mouse marrow cells. *Nature*, 197 (4866):452–454, feb 1963.

B. Berger, J. Peng, and M. Singh. Computational solutions for omics data. *Nature reviews. Genetics*, 14(5):333–46, may 2013.

B. E. Bernstein, E. Birney, I. Dunham, E. D. Green, C. Gunter, and M. Snyder. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414): 57–74, sep 2012.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2007.

M. V. Boland and R. F. Murphy. A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells. *Bioinformatics*, 17(12):1213–1223, dec 2001.

M. V. Boland, M. K. Markey, and R. F. Murphy. Automated recognition of patterns characteristic of subcellular structures in fluorescence microscopy images. *Cytometry*, 33(3):366–375, nov 1998.

A. Borisov, E. Tuv, G. Runger, I. Guyon, G. Cawley, G. Dror, V. Lemaire, and A. Statnikov. Active batch learning with stochastic query by forest. *JMLR:Workshop and Conference Proceedings*, 2011.

L. Bottou. Large-Scale Machine Learning with Stochastic Gradient Descent. *Proceedings of COMPSTAT*, pages 177–186, 2010.

L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

L. Breiman, J. Friedman, C. J. Stone, and R. Olshen. *Classification and Regression Trees.* Chapman and Hall/CRC, 1984.

F. Buettner, K. N. Natarajan, F. P. Casale, V. Proserpio, A. Scialdone, F. J. Theis, S. A. Teichmann, J. C. Marioni, and O. Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature Biotechnology*, 33(2):155–160, jan 2015.

F. Buggenthin, C. Marr, M. Schwarzfischer, P. S. Hoppe, O. Hilsenbeck, T. Schroeder, and F. J. Theis. An automatic method for robust and fast cell detection in bright field images from high-throughput microscopy. *BMC Bioinformatics*, 14(1):297, 2013.

W. Burnette. "Western Blotting": Electrophoretic transfer of proteins from sodium dodecyl sulfate-polyacrylamide gels to unmodified nitrocellulose and radiographic detection with antibody and radioiodinated protein A. *Analytical Biochemistry*, 112(2):195–203, apr 1981.

S. A. Bustin. Absolute quantification of mRNA using real-time reverse transcription polymerase chain reaction assays. *Journal of molecular endocrinology*, 25(2):169–93, oct 2000.

A. R. Buxbaum, Y. J. Yoon, R. H. Singer, and H. Y. Park. Single-molecule insights into mRNA dynamics in neurons. *Trends in Cell Biology*, jun 2015.

L. Cai, C. K. Dalal, and M. B. Elowitz. Frequency-modulated nuclear localization bursts coordinate gene regulation. *Nature*, 455(7212):485–490, 2008.

R. S. Cajal. Studies on Vertebrate Neurogenesis. *CC.Thomas Springfield*, 1960.

A. Cardona, S. Saalfeld, I. Arganda, W. Pereanu, J. Schindelin, and V. Hartenstein. Identifying neuronal lineages of Drosophila by sequence analysis of axon tracts. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 30(22): 7538–53, jun 2010.

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

M. Chalfie, Y. Tu, G. Euskirchen, W. W. Ward, and D. C. Prasher. Green fluorescent protein as a marker for gene expression. *Science*, 263:7036–7040, 2007.

I. Chambers, J. Silva, D. Colby, J. Nichols, B. Nijmeijer, M. Robertson, J. Vrana, K. Jones, L. Grotewold, and A. Smith. Nanog safeguards pluripotency and mediates germline development. *Nature*, 450(7173):1230–4, dec 2007.

P. K. Chattopadhyay, T. M. Gierahn, M. Roederer, and J. C. Love. Single-cell technologies for monitoring immune systems. *Nature immunology*, 15(2):128–35, feb 2014.

N. Chenouard, I. Smal, F. de Chaumont, M. Maška, I. F. Sbalzarini, Y. Gong, J. Cardinale, C. Carthel, S. Coraluppi, M. Winter, A. R. Cohen, W. J. Godinez, K. Rohr, Y. Kalaidzidis, L. Liang, J. Duncan, H. Shen, Y. Xu, K. E. G. Magnusson, J. Jaldén, H. M. Blau, P. Paul-Gilloteaux, P. Roudot, C. Kervrann, F. Waharte, J.-Y. Tinevez, S. L. Shorte, J. Willemse, K. Celler, G. P. van Wezel, H.-W. Dan, Y.-S. Tsai, C. O. de Solórzano, J.-C. Olivo-Marin, and E. Meijering. Objective comparison of particle tracking methods. *Nature Methods*, 11(3):281–289, jan 2014.

V. Chickarmane, T. Enver, and C. Peterson. Computational modeling of the hematopoietic erythroid-myeloid switch reveals insights into cooperativity, priming, and irreversibility. *PLoS computational biology*, 5(1):e1000268, jan 2009.

V. Chickarmane, A. H. K. Roeder, P. T. Tarr, A. Cunha, C. Tobin, and E. M. Meyerowitz. Computational morphodynamics: a modeling framework to understand plant growth. *Annual review of plant biology*, 61:65–87, jan 2010.

L. T. Chow, R. E. Gelinas, T. R. Broker, and R. J. Roberts. An amazing sequence arrangement at the 5' ends of adenovirus 2 messenger RNA. *Cell*, 12(1):1–8, oct 1977.

D. Ciresan, U. Meier, and J. Masci. Flexible, high performance convolutional neural networks for image classification. *International Joint Conference on Artificial Intelligence*, pages 1237–1242, 2011.

A. R. Cohen, F. L. A. F. Gomes, B. Roysam, and M. Cayouette. Computational prediction of neural progenitor cell fates. *Nat Methods*, 7(3):213–218, 2010.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

F. H. C. Crick. Central Dogma of Molecular Biology. *Nature*, 227(5258):561–563, aug 1970.

F. de Chaumont, S. Dallongeville, N. Chenouard, N. Hervé, S. Pop, T. Provoost, V. Meas-Yedid, P. Pankajakshan, T. Lecomte, Y. Le Montagner, T. Lagache, A. Dufour, and J.-C. Olivo-Marin. Icy: an open bioimage informatics platform for extended reproducible research. *Nature methods*, 9(7):690–6, jul 2012.

Q. Deng, D. Ramsköld, B. Reinius, and R. Sandberg. Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science (New York, N.Y.)*, 343 (6167):193–6, jan 2014.

P. A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach.* Prentice Hall, 1982.

D. A. Diamond, S. J. Berry, C. Umbricht, H. J. Jewett, and D. S. Coffey. Computerized image analysis of nuclear shape as a prognostic factor for prostatic cancer. *The Prostate*, 3(4):321–332, 1982.

W. Dittrich and W. Göhde. Notizen: Impulsfluorometrie bei Einzelzellen in Suspensionen. *Zeitschrift für Naturforschung B*, 24(3):360–361, jan 1969.

N. Djuric, L. Lan, S. Vucetic, and Z. Wang. BudgetedSVM: a toolbox for scalable SVM approximations. *The Journal of Machine Learning Research*, 14(1):3813–3817, jan 2013.

J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. *Arxiv*, nov 2014.

O. Dzyubachyk, W. A. van Cappellen, J. Essers, W. J. Niessen, and E. Meijering. Advanced level-set-based cell tracking in time-lapse fluorescence microscopy. *IEEE transactions on medical imaging*, 29(3):852–67, mar 2010.

C. J. Eaves. Hematopoietic stem cells: concepts, definitions and the new reality. *Blood*, 125(17):2605–2613, mar 2015.

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

M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain. Stochastic gene expression in a single cell. *Science (New York, N.Y.)*, 297(5584):1183–6, aug 2002.

D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable Object Detection using Deep Neural Networks. *Arxiv*, 2013.

M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, volume 1996, pages 226–231. AAAI Press, 1996.

D. Fenistein, B. Lenseigne, T. Christophe, P. Brodin, and A. Genovesio. A fast, fully automated cell segmentation algorithm for high-throughput and high-content screening. *Cytometry. Part A : the journal of the International Society for Analytical Cytology*, 73 (10):958–64, oct 2008.

P. Friedl, E. Sahai, S. Weiss, and K. M. Yamada. New dimensions in cell migration. *Nature reviews. Molecular cell biology*, 13(11):743–7, nov 2012.

K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4): 193–202, apr 1980.

Y. Futamura, M. Kawatani, S. Kazami, K. Tanaka, M. Muroi, T. Shimizu, K. Tomita, N. Watanabe, and H. Osada. Morphobase, an encyclopedic cell morphology database, and its use for drug target identification. *Chemistry & biology*, 19(12):1620–30, dec 2012.

R. N. Germain, E. A. Robey, and M. D. Cahalan. A Decade of Imaging Cellular Motility and Interaction Dynamics in the Immune System. *Science*, 336(6089):1676–1681, jun 2012.

J.-P. Girard, C. Moussion, and R. Förster. HEVs, lymphatics and homeostatic immune cell trafficking in lymph nodes. *Nature reviews. Immunology*, 12(11):762–73, nov 2012.

R. Girshick, J. Donahue, T. Darrell, U. C. Berkeley, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Cvpr'14*, pages 2–9, 2014.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 9:249–256, 2010.

B. Göttgens. Regulatory network control of blood stem cells. *Blood*, 125(17):2614–2620, mar 2015.

G. Guo, M. Huss, G. Q. Tong, C. Wang, L. Li Sun, N. D. Clarke, and P. Robson. Resolution of cell fate decisions revealed by single-cell gene expression analysis from zygote to blastocyst. *Developmental cell*, 18(4):675–85, apr 2010.

L. Haghverdi, F. Buettner, and F. J. Theis. Diffusion maps for high-dimensional single-cell analysis of differentiation data. *Bioinformatics*, pages btv325–, may 2015.

R. M. Haralick, Dinstein, and K. Shanmugam. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3:610 – 621, 1973.

T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani. *The elements of statistical learning*. Springer, 2009.

M. Held, M. H. a. Schmitz, B. Fischer, T. Walter, B. Neumann, M. H. Olma, M. Peter, J. Ellenberg, and D. W. Gerlich. CellCognition: time-resolved phenotype annotation in high-throughput live cell imaging. *Nature methods*, aug 2010.

A. Hermann. *Analysis of asymmetric division of hematopoietic stem cells by continuous single cell observation*. PhD thesis, Ludwig-Maximilians-Universitaet Muenchen, 2009.

L. A. Herzenberg and R. G. Sweet. Fluorescence-activated cell sorting. *Scientific American*, 234(3):108–17, mar 1976.

G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *Arxiv*, jul 2012.

P. Hoppe. Der Einfluss von Zytokinen auf die Linienentscheidung myeloider Vorläuferzellen – Erkenntnisse aus der Einzelzellanalyse. Master's thesis, Ludwig-Maximilians-Universität München Fakultät für Biologie, 2008.

P. Hoppe, M. Schwarzfischer, D. Loeffler, K. Kokkaliaris, O. Hilsenbeck, N. Moritz, M. Endele, A. Filipczyk, A. Gambardella, N. Ahmed, M. Etzrodt, D. Coutu, M. Rieger, C. Marr, M. Strasser, B. Schauberger, I. Burtscher, O. Ermakova, A. Bürger, H. Lickert, C. Nerlov, F. Theis, and T. Schroeder. Early myeloid lineage choice is not initiated by random PU.1 to GATA1 protein ratios. *Nature*, 2016.

P. S. Hoppe, D. L. Coutu, and T. Schroeder. Single-cell technologies sharpen up mammalian stem cell research. *Nature Cell Biology*, 16(10):919–927, oct 2014.

P. V. C. Hough. Machine Analysis Of Bubble Chamber Pictures. In *Proceedings, 2nd International Conference on High-Energy Accelerators and Instrumentation, HEACC 1959*, volume C590914, pages 554–558, 1959.

P. Howarth and S. Rüger. Evaluation of Texture Features for Content-Based Image Retrieval. In *Proceedings of Image and Video Retrieval 2004*, volume 3115, pages 326—-334, 2004.

S. Huang, Y.-P. Guo, G. May, and T. Enver. Bifurcation dynamics in lineage-commitment in bipotent progenitor cells. *Developmental biology*, 305(2):695–713, may 2007.

D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195(1):215–243, mar 1968.

D. H. E. W. Huberts, S. Sik Lee, J. Gonzáles, G. E. Janssens, I. A. Vizcarra, and M. Heinemann. Construction and use of a microfluidic dissection platform for long-term imaging of cellular processes in budding yeast. *Nature protocols*, 8(6):1019–27, jun 2013.

J. J. Hutter. Childhood leukemia. *Pediatrics in review / American Academy of Pediatrics*, 31(6):234–41, jun 2010.

P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, feb 1912.

J. Jacobelli, R. S. Friedman, M. A. Conti, A.-M. Lennon-Dumenil, M. Piel, C. M. Sorensen, R. S. Adelstein, and M. F. Krummel. Confinement-optimized three-dimensional T cell amoeboid motility is modulated via myosin IIA-regulated adhesions. *Nature immunology*, 11(10):953–961, 2010.

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

Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. In *Proceedings of the ACM International Conference on Multimedia - MM '14*, pages 675–678, New York, New York, USA, nov 2014. ACM Press.

T. R. Jones, A. E. Carpenter, M. R. Lamprecht, J. Moffat, S. J. Silver, J. K. Grenier, A. B. Castoreno, U. S. Eggert, D. E. Root, P. Golland, and D. M. Sabatini. Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning. *Proc Natl Acad Sci U S A*, 106(6):1826–1831, feb 2009.

R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, dec 1987.

O. L. Junior, D. Delgado, V. Goncalves, U. Nunes, and O. Ludwig. Trainable classifier-fusion schemes: An application to pedestrian detection. In *2009 12th International IEEE Conference on Intelligent Transportation Systems*, pages 1–6. IEEE, oct 2009.

N. Kalchbrenner and P. Blunsom. Recurrent Convolutional Neural Networks for Discourse Compositionality. *Arxiv*, jun 2013.

K. Keren, Z. Pincus, G. M. Allen, E. L. Barnhart, G. Marriott, A. Mogilner, and J. A. Theriot. Mechanism of shape determination in motile cells. *Nature*, 453(7194):475–80, may 2008.

A. Khodjakov and C. L. Rieder. Imaging the division process in living tissue culture cells. *Methods (San Diego, Calif.)*, 38(1):2–16, jan 2006.

M. J. Kiel, O. H. Yilmaz, T. Iwashita, O. H. Yilmaz, C. Terhorst, and S. J. Morrison. SLAM family receptors distinguish hematopoietic stem and progenitor cells and reveal endothelial niches for stem cells. *Cell*, 121(7):1109–21, jul 2005.

B. Knapp, I. Rebhan, A. Kumar, P. Matula, N. A. Kiani, M. Binder, H. Erfle, K. Rohr, R. Eils, R. Bartenschlager, and L. Kaderali. Normalizing for individual cell population context in the analysis of high-content cellular screens. *BMC bioinformatics*, 12(1):485, jan 2011.

A. Koehler. New Method of Illimination for Phomicrographical Purposes. *Journal of the Royal Microscopical Society*, 14, 1894.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

M. Kroiss. *Using deep neural networks to predict the lineage choice of hematopoietic stem cells from time-lapse microscopy images.* Master thesis, Technische Univeristät München, 2014.

T. Lämmermann and R. N. Germain. The multiple faces of leukocyte interstitial migration. *Seminars in immunopathology*, feb 2014.

H. Landecker. Cellular Features: Microcinematography and Film Theory. *Critical Inquiry*, 31(4):903–937, 2005.

H. Landecker. Microcinematography and the History of Science and Film. *Isis*, 97(1): 121–132, mar 2006.

E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, R. Funke, D. Gage, K. Harris, A. Heaford, J. Howland, L. Kann, J. Lehoczky, R. LeVine, P. McEwan, K. McKernan, J. Meldrim, J. P. Mesirov, C. Miranda, W. Morris, J. Naylor, C. Raymond, M. Rosetti, R. Santos, A. Sheridan, C. Sougnez, N. Stange-Thomann, N. Stojanovic, A. Subramanian, D. Wyman, J. Rogers, J. Sulston, R. Ainscough, S. Beck, D. Bentley, J. Burton, C. Clee, N. Carter, A. Coulson, R. Deadman, P. Deloukas, A. Dunham, I. Dunham, R. Durbin, L. French, D. Grafham, S. Gregory, T. Hubbard, S. Humphray, A. Hunt, M. Jones, C. Lloyd, A. McMurray, L. Matthews, S. Mercer, S. Milne, J. C. Mullikin, A. Mungall, R. Plumb, M. Ross, R. Shownkeen, S. Sims, R. H. Waterston, R. K. Wilson, L. W. Hillier, J. D. McPherson, M. A. Marra, E. R. Mardis, L. A. Fulton, A. T. Chinwalla, K. H. Pepin, W. R. Gish, S. L. Chissoe, M. C. Wendl, K. D. Delehaunty, T. L. Miner, A. Delehaunty, J. B. Kramer, L. L. Cook, R. S. Fulton, D. L. Johnson, P. J. Minx, S. W. Clifton, T. Hawkins, E. Branscomb, P. Predki, P. Richardson, S. Wenning, T. Slezak, N. Doggett, J. F. Cheng, A. Olsen, S. Lucas, C. Elkin, E. Uberbacher, M. Frazier, R. A. Gibbs, D. M. Muzny, S. E. Scherer, J. B. Bouck, E. J. Sodergren, K. C. Worley, C. M. Rives, J. H. Gorrell, M. L. Metzker, S. L. Naylor, R. S. Kucherlapati, D. L. Nelson, G. M. Weinstock, Y. Sakaki, A. Fujiyama, M. Hattori, T. Yada, A. Toyoda, T. Itoh, C. Kawagoe, H. Watanabe, Y. Totoki, T. Taylor, J. Weissenbach, R. Heilig, W. Saurin, F. Artiguenave, P. Brottier, T. Bruls, E. Pelletier, C. Robert, P. Wincker, D. R. Smith, L. Doucette-Stamm, M. Rubenfield, K. Weinstock, H. M. Lee, J. Dubois, A. Rosenthal, M. Platzer, G. Nyakatura, S. Taudien, A. Rump, H. Yang, J. Yu, J. Wang, G. Huang, J. Gu, L. Hood, L. Rowen, A. Madan, S. Qin, R. W. Davis, N. A. Federspiel, A. P. Abola, M. J. Proctor, R. M. Myers, J. Schmutz, M. Dickson, J. Grimwood, D. R. Cox, M. V. Olson, R. Kaul, N. Shimizu, K. Kawasaki, S. Minoshima, G. A. Evans, M. Athanasiou, R. Schultz, B. A. Roe, F. Chen, H. Pan, J. Ramser, H. Lehrach, R. Reinhardt, W. R. McCombie, M. de la Bastide, N. Dedhia, H. Blöcker, K. Hornischer, G. Nordsiek, R. Agarwala, L. Aravind, J. A. Bailey, A. Bateman, S. Batzoglou, E. Birney, P. Bork, D. G. Brown, C. B. Burge, L. Cerutti, H. C. Chen, D. Church, M. Clamp, R. R. Copley, T. Doerks, S. R. Eddy, E. E. Eichler, T. S. Furey, J. Galagan, J. G. Gilbert, C. Harmon, Y. Hayashizaki, D. Haussler, H. Hermjakob, K. Hokamp, W. Jang, L. S. Johnson, T. A. Jones, S. Kasif, A. Kaspryzk, S. Kennedy, W. J. Kent, P. Kitts, E. V. Koonin, I. Korf,

D. Kulp, D. Lancet, T. M. Lowe, A. McLysaght, T. Mikkelsen, J. V. Moran, N. Mulder, V. J. Pollara, C. P. Ponting, G. Schuler, J. Schultz, G. Slater, A. F. Smit, E. Stupka, J. Szustakowski, D. Thierry-Mieg, J. Thierry-Mieg, L. Wagner, J. Wallis, R. Wheeler, A. Williams, Y. I. Wolf, K. H. Wolfe, S. P. Yang, R. F. Yeh, F. Collins, M. S. Guyer, J. Peterson, A. Felsenfeld, K. A. Wetterstrand, A. Patrinos, M. J. Morgan, P. de Jong, J. J. Catanese, K. Osoegawa, H. Shizuya, S. Choi, Y. J. Chen, and J. Szustakowki. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, feb 2001.

M. Le Berre, E. Zlotek-Zlotkiewicz, D. Bonazzi, F. Lautenschlaeger, and M. Piel. Methods for two-dimensional cell confinement. *Methods in cell biology*, 121:213–29, jan 2014.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, may 2015.

C. Li, C. Xu, C. Gui, and M. D. Fox. Distance regularized level set evolution and its application to image segmentation. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 19(12):3243–54, dec 2010.

G. Li, T. Liu, J. Nie, L. Guo, J. Chen, J. Zhu, W. Xia, A. Mara, S. Holley, and S. T. C. Wong. Segmentation of touching cell nuclei using gradient flow tracking. *Journal of microscopy*, 231(Pt 1):47–58, jul 2008.

H. Li and N. Homer. A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in bioinformatics*, 11(5):473–83, sep 2010.

J. W. Lichtman and J.-A. Conchello. Fluorescence microscopy. *Nature methods*, 2(12): 910–919, 2005.

A.-A. Liu, K. Li, and T. Kanade. A semi-Markov model for mitosis segmentation in time-lapse phase contrast microscopy image sequences of stem cell populations. *IEEE transactions on medical imaging*, 31(2):359–69, feb 2012.

P. Llull, X. Liao, X. Yuan, J. Yang, D. Kittle, L. Carin, G. Sapiro, and D. J. Brady. Coded aperture compressive temporal imaging. *Optics express*, 21(9):10526–45, may 2013.

C. J. Lovitt, T. B. Shelper, and V. M. Avery. Evaluation of chemotherapeutics in a three-dimensional breast cancer model. *Journal of cancer research and clinical oncology*, 141 (5):951–9, may 2015.

O. Maimon and L. Rokach. *Data mining and knowledge discovery handbook*. Springer, 2nd edition, 2010.

S. E. Malawista and A. de Boisfleury Chevance. Random locomotion and chemotaxis of human blood polymorphonuclear leukocytes (PMN) in the presence of EDTA: PMN in close quarters require neither leukocyte integrins nor external divalent cations. *Proceedings of the National Academy of Sciences of the United States of America*, 94(21): 11577–11582, 1997.

L. Malherbe, C. Filippi, V. Julia, G. Foucras, M. Moro, H. Appel, K. Wucherpfennig, J.-C. Guéry, and N. Glaichenhaus. Selective Activation and Expansion of High-Affinity CD4+ T Cells in Resistant Mice upon Infection with Leishmania major. *Immunity*, 13 (6):771–782, dec 2000.

M. Maška, V. Ulman, D. Svoboda, P. P. Matula, P. P. Matula, C. Ederra, A. Urbiola, T. España, S. Venkatesan, D. M. W. Balak, P. Karas, T. Bolcková, M. Streitová, C. Carthel, S. Coraluppi, N. Harder, K. Rohr, K. E. G. Magnusson, J. Jaldén, H. M. Blau, O. Dzyubachyk, P. Křížek, G. M. Hagen, D. Pastor-Escuredo, D. Jimenez-Carretero, M. J. Ledesma-Carbayo, A. Muñoz-Barrutia, E. Meijering, M. Kozubek, and C. Ortiz-de Solorzano. A benchmark for comparison of cell tracking algorithms. *Bioinformatics (Oxford, England)*, 30(11):1609–17, jun 2014.

J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

T. Mathworks. *MATLAB documentation*. The MathWorks Inc., Natick, Massachusetts, 2015.

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

E. Meijering. Cell Segmentation: 50 Years Down the Road [Life Sciences]. *IEEE Signal Processing Magazine*, 29(5):140–145, sep 2012.

E. Meijering, O. Dzyubachyk, I. Smal, and W. A. van Cappellen. Tracking in cell and developmental biology. *Semin Cell Dev Biol*, 20(8):894–902, 2009.

E. Meijering, O. Dzyubachyk, and I. Smal. Methods for cell and particle tracking. *Methods in enzymology*, 504(February):183–200, jan 2012.

F. Meyer. Topographic distance and watershed lines. *Signal Process.*, 38(1):113–125, 1994.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, feb 2015.

V. Moignard, S. Woodhouse, L. Haghverdi, A. J. Lilly, Y. Tanaka, A. C. Wilkinson, F. Buettner, I. C. Macaulay, W. Jawaid, E. Diamanti, S.-I. Nishikawa, N. Piterman, V. Kouskoff, F. J. Theis, J. Fisher, and B. Göttgens. Decoding the regulatory network of early blood development from single-cell gene expression measurements. *Nature Biotechnology*, 33(3):269–276, feb 2015.

A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature methods*, 5(7):621–8, jul 2008.

K. Murphy. *Machine Learning: a Probabilistic Perspective.* Springer, 2012.

R. F. Murphy. A new era in bioimage informatics. *Bioinformatics (Oxford, England)*, 30 (10):1353, may 2014.

D. Muzzey and A. van Oudenaarden. Quantitative time-lapse fluorescence microscopy in single cells. *Annual review of cell and developmental biology*, 25:301–27, jan 2009.

G. Myers. Why bioimage informatics matters. *Nature methods*, 9(7):659–60, jul 2012.

I. Nachman, A. Regev, and S. Ramanathan. Dissecting Timing Variability in Yeast Meiosis. *Cell*, 131(3):544–556, nov 2007.

V. Nair and G. E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807—-814, 2010.

K. Nakae, Y. Ikegaya, T. Ishikawa, S. Oba, H. Urakubo, M. Koyama, and S. Ishii. A statistical method of identifying interactions in neuron-glia systems based on functional multicell Ca2+ imaging. *PLoS computational biology*, 10(11):e1003949, nov 2014.

B. Neumann, T. Walter, J. K. Hériché, J. Bulkescher, H. Erfle, C. Conrad, P. Rogers, I. Poser, M. Held, U. Liebel, C. Cetin, F. Sieckmann, G. Pau, R. Kabbe, A. Wünsche, V. Satagopam, M. H. A. Schmitz, C. Chapuis, D. W. Gerlich, R. Schneider, R. Eils, W. Huber, J. Peters, A. A. Hyman, R. Durbin, R. Pepperkok, and J. Ellenberg. Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. *Nature*, 464(7289):721–727, apr 2010.

B. Neumann, M. Held, U. Liebel, H. Erfle, P. Rogers, R. Pepperkok, and J. Ellenberg. High-throughput RNAi screening by time-lapse imaging of live human cells. *Nature Methods*, 3(5):385–390, 2006.

NIH. Regenerative Medicine, 2006.

D. Nistér and H. Stewénius. Linear Time Maximally Stable Extremal Regions. In D. Forsyth, P. Torr, and A. Zisserman, editors, *08 Proceedings of the 10th European Conference on Computer Vision: Part II*, volume 5303 of *Lecture Notes in Computer Science*, pages 183–196, Berlin, Heidelberg, oct 2008. Springer Berlin Heidelberg.

M. Nixon and A. Aguado. Feature Extraction & Image Processing for Computer Vision, Third Edition. In *Feature Extraction & Image Processing for Computer Vision, Second Edition*. Academic Press, 2012.

S. Nourshargh, P. L. Hordijk, and M. Sixt. Breaching multiple barriers: leukocyte motility through venular walls and the interstitium. *Nature reviews. Molecular cell biology*, 11 (5):366–78, may 2010.

S. L. Nutt, D. Metcalf, A. D'Amico, M. Polli, and L. Wu. Dynamic regulation of PU.1 expression in multipotent hematopoietic progenitors. *The Journal of experimental medicine*, 201(2):221–31, jan 2005.

M. O'Leary and S. Kaufman. MorphoBank 3.0: Web application for morphological phylogenetics and taxonomy. *Wiley*, 2012.

T. M. Olson. Actin Mutations in Dilated Cardiomyopathy, a Heritable Form of Heart Failure. *Science*, 280(5364):750–752, may 1998.

S. H. Orkin and L. I. Zon. Hematopoiesis: an evolving paradigm for stem cell biology. *Cell*, 132(4):631–644, feb 2008.

M. Osawa, K.-i. Hanada, H. Hamada, and H. Nakauchi. Long-Term Lymphohematopoietic Reconstitution by a Single CD34-Low/Negative Hematopoietic Stem Cell. *Science*, 273 (5272):242–245, jul 1996.

N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11:285–296, 1975.

E. Paluch and C.-P. Heisenberg. Biology and physics of cell shape changes in development. *Current biology : CB*, 19(17):R790–9, sep 2009.

H. Peng. Bioimage informatics: a new area of engineering biology. *Bioinformatics*, 24(17): 1827–1836, sep 2008.

S. P. Perfetto, P. K. Chattopadhyay, and M. Roederer. Seventeen-colour flow cytometry: unravelling the immune system. *Nature reviews. Immunology*, 4(8):648–55, aug 2004.

P. H. O. Pinheiro and R. Collobert. Recurrent Convolutional Neural Networks for Scene Parsing. *Arxiv*, jun 2013.

T. D. Pollard and J. A. Cooper. Actin, a central player in cell shape and movement. *Science (New York, N.Y.)*, 326(5957):1208–12, nov 2009.

S. Preibisch, F. Amat, E. Stamataki, M. Sarov, R. H. Singer, E. Myers, and P. Tomancak. Efficient Bayesian-based multiview deconvolution. *Nature methods*, 11(6):645–8, jun 2014.

V. Procaccio, G. Salazar, S. Ono, M. L. Styers, M. Gearing, A. Davila, R. Jimenez, J. Juncos, C.-A. Gutekunst, G. Meroni, B. Fontanella, E. Sontag, J. M. Sontag, V. Faundez, and B. H. Wainer. A mutation of beta -actin that alters depolymerization dynamics is associated with autosomal dominant developmental malformations, deafness, and dystonia. *American journal of human genetics*, 78(6):947–60, jun 2006.

J. O. Ramsay, G. Hooker, and S. Graves. *Functional data analysis with R and MATLAB*. Springer Verlag, 2009.

M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun. Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, jun 2007.

W. S. Rasband. ImageJ: Image processing and analysis in Java. *Astrophysics Source Code Library*, -1:06013, jun 2012.

B. S. Reddy and B. N. Chatterji. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 5(8):1266–71, jan 1996.

J. Renkawitz and M. Sixt. Mechanisms of force generation and force transmission during interstitial leukocyte migration. *EMBO reports*, 11(10):744–50, oct 2010.

A. J. Ridley, M. A. Schwartz, K. Burridge, R. A. Firtel, M. H. Ginsberg, G. Borisy, J. T. Parsons, and A. R. Horwitz. Cell migration: integrating signals from front to back. *Science (New York, N.Y.)*, 302(5651):1704–9, dec 2003.

J. Riedl, A. H. Crevenna, K. Kessenbrock, J. H. Yu, D. Neukirchen, M. Bista, F. Bradke, D. Jenne, T. A. Holak, Z. Werb, M. Sixt, and R. Wedlich-Soldner. Lifeact: a versatile marker to visualize F-actin. *Nature methods*, 5(7):605–7, jul 2008.

M. A. Rieger, P. S. Hoppe, B. M. Smejkal, A. C. Eitelhuber, and T. Schroeder. Hematopoietic cytokines can instruct lineage choice. *Science*, 325(5937):217–218, jul 2009.

I. Roeder and I. Glauche. Towards an understanding of lineage specification in hematopoietic stem cells: a mathematical model for the interaction of transcription factors GATA-1 and PU.1. *Journal of theoretical biology*, 241(4):852–65, aug 2006.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, oct 1986.

G. Ruvkun. Molecular biology. Glimpses of a tiny RNA world. *Science (New York, N.Y.)*, 294(5543):797–9, oct 2001.

E. K. Sackmann, A. L. Fulton, and D. J. Beebe. The present and future role of microfluidics in biomedical research. *Nature*, 507(7491):181–9, mar 2014.

H. Sailem, V. Bousgouni, S. Cooper, and C. Bakal. Cross-talk between Rho and Rac GTPases drives deterministic exploration of cellular shape space and morphological heterogeneity. *Open biology*, 4(1):130132, jan 2014.

Y. Sakurai, A. A. Kolokoltsov, C.-C. Chen, M. W. Tidwell, W. E. Bauta, N. Klugbauer, C. Grimm, C. Wahl-Schott, M. Biel, and R. A. Davey. Two-pore channels control Ebola virus host cell entry and are drug targets for disease treatment. *Science*, 347 (6225):995–998, feb 2015.

A.-E. Saliba, A. J. Westermann, S. A. Gorski, and J. Vogel. Single-cell RNA-seq: advances and future challenges. *Nucleic Acids Research*, pages gku555–, jul 2014.

I. F. Sbalzarini. Modeling and simulation of biological systems from image data. *BioEssays : news and reviews in molecular, cellular and developmental biology*, 35(5):482–90, may 2013.

R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5): 1651–1686, 1998.

M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science (New York, N.Y.)*, 270(5235):467–70, oct 1995.

J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona. Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7):676–82, jul 2012.

W. Schottky. Über spontane Stromschwankungen in verschiedenen Elektrizitätsleitern. *Annalen der Physik*, 362(23):541–567, 1918.

T. Schroeder. Imaging stem-cell-driven regeneration in mammals. *Nature*, 453(7193): 345–351, 2008.

T. Schroeder. Hematopoietic stem cell heterogeneity: subtypes, not unpredictable behavior. *Cell stem cell*, 6(3):203–207, 2010.

T. Schroeder. Long-term single-cell imaging of mammalian stem cells. *Nature Methods*, 8 (4):S30—-S35, apr 2011.

M. Schwarzfischer, C. Marr, J. Krumsiek, P. Hoppe, T. Schroeder, and F. Theis. Efficient fluorescence image normalization for time lapse movies. In *Proc. Microscopic Image Analysis with Applications in Biology*, page 5, Heidelberg, Germany, 2011.

M. Schwarzfischer. *Quantification and analysis of single-cell protein dynamics in stem cells using time-lapse microscopyle.* Dissertation, Technische Universität München, 2013.

J. Seita and I. L. Weissman. Hematopoietic stem cell: self-renewal versus differentiation. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 2010.

J. Selinummi, P. Ruusuvuori, I. Podolsky, A. Ozinsky, E. Gold, O. Yli-Harja, A. Aderem, and I. Shmulevich. Bright field microscopy as an alternative to whole cell fluorescence in automated analysis of macrophage images. *PloS one*, 4(10):e7497, jan 2009.

J. E. Sero, H. Z. Sailem, R. C. Ardy, H. Almuttaqi, T. Zhang, and C. Bakal. Cell shape and the microenvironment regulate nuclear translocation of NF- B in breast epithelial and tumor cells. *Molecular Systems Biology*, 11(3):790–790, mar 2015.

B. Settles. Active learning literature survey. Technical report, University of Wisconsin–Madison, 2009.

A. K. Shalek, R. Satija, X. Adiconis, R. S. Gertner, J. T. Gaublomme, R. Raychowdhury, S. Schwartz, N. Yosef, C. Malboeuf, D. Lu, J. J. Trombetta, D. Gennert, A. Gnirke, A. Goren, N. Hacohen, J. Z. Levin, H. Park, and A. Regev. Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature*, 498(7453):236–40, jun 2013.

L. Shamir, N. Orlov, D. M. Eckley, T. Macura, J. Johnston, and I. G. Goldberg. Wndchrm - an open source utility for biological image analysis. *Source code for biology and medicine*, 3:13, jan 2008.

A. Shariff, J. Kangas, L. P. Coelho, S. Quinn, and R. F. Murphy. Automated image analysis for high-content screening and analysis. *Journal of biomolecular screening*, 15 (7):726–34, aug 2010.

S. Sharma and A. Rao. RNAi screening: tips and techniques. *Nature immunology*, 10(8): 799–804, aug 2009.

S. K. Singh, C. Hawkins, I. D. Clarke, J. A. Squire, J. Bayani, T. Hide, R. M. Henkelman, M. D. Cusimano, and P. B. Dirks. Identification of human brain tumour initiating cells. *Nature*, 432(7015):396–401, nov 2004.

C. S. Smith, S. Preibisch, A. Joseph, S. Abrahamsson, B. Rieger, E. Myers, R. H. Singer, and D. Grunwald. Nuclear accessibility of -actin mRNA is measured by 3D single-molecule real-time tracking. *The Journal of Cell Biology*, 209(4):609–619, may 2015a.

K. Smith, A. Carleton, and V. Lepetit. Fast ray features for learning irregular shapes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 397–404, 2009.

K. Smith, Y. Li, F. Piccinini, G. Csucs, C. Balazs, A. Bevilacqua, and P. Horvath. CIDRE: an illumination-correction method for optical microscopy. *Nature methods*, advance on, mar 2015b.

M. Strasser, F. J. Theis, and C. Marr. Stability and multiattractor dynamics of a toggle switch based on a two-stage model of stochastic gene expression. *Biophysical journal*, 102(1):19–29, jan 2012.

J. R. Swedlow, I. G. Goldberg, K. W. Eliceiri, and O. M. E. Consortium. Bioimage informatics for experimental biology. *Annu Rev Biophys*, 38:327–346, 2009.

H. Tamura, S. Mori, and T. Yamawaki. Textural Features Corresponding to Visual Perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(6):460–473, 1978.

D. H. Theriault, M. L. Walker, J. Y. Wong, and M. Betke. Cell morphology classification and clutter mitigation in phase-contrast microscopy images using machine learning. *Machine Vision and Applications*, pages 1–15, jun 2011.

M. Tompsett, G. Amelio, W. Bertram, R. Buckley, W. McNamara, J. Mikkelsen, and D. Sealer. Charge-coupled imaging devices: Experimental results. *IEEE Transactions on Electron Devices*, 18(11):992–996, nov 1971.

H. Towbin, T. Staehelin, and J. Gordon. Electrophoretic transfer of proteins from polyacrylamide gels to nitrocellulose sheets: procedure and some applications. *Proceedings* *of the National Academy of Sciences of the United States of America*, 76(9):4350–4, sep 1979.

S. Tse, L. Bradbury, J. W. Wan, H. Djambazian, R. Sladek, and T. Hudson. A combined watershed and level set method for segmentation of brightfield cell images. In *Proceedings of SPIE*, volume 7259, pages 72593G–72593G–10. SPIE, feb 2009.

Y.-C. Tung, A. Y. Hsiao, S. G. Allen, Y.-s. Torisawa, M. Ho, and S. Takayama. High-throughput 3D spheroid culture and drug testing using a 384 hanging drop array. *The Analyst*, 136(3):473–8, mar 2011.

L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(9):2579–2605, 2008.

J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, J. D. Gocayne, P. Amanatides, R. M. Ballew, D. H. Huson, J. R. Wortman, Q. Zhang, C. D. Kodira, X. H. Zheng, L. Chen, M. Skupski, G. Subramanian, P. D. Thomas, J. Zhang, G. L. Gabor Miklos, C. Nelson, S. Broder, A. G. Clark, J. Nadeau, V. A. McKusick, N. Zinder, A. J. Levine, R. J. Roberts, M. Simon, C. Slayman, M. Hunkapiller, R. Bolanos, A. Delcher, I. Dew, D. Fasulo, M. Flanigan, L. Florea, A. Halpern, S. Hannenhalli, S. Kravitz, S. Levy, C. Mobarry, K. Reinert, K. Remington, J. Abu-Threideh, E. Beasley, K. Biddick, V. Bonazzi, R. Brandon, M. Cargill, I. Chandramouliswaran, R. Charlab, K. Chaturvedi, Z. Deng, V. Di Francesco, P. Dunn, K. Eilbeck, C. Evangelista, A. E. Gabrielian, W. Gan, W. Ge, F. Gong, Z. Gu, P. Guan, T. J. Heiman, M. E. Higgins, R. R. Ji, Z. Ke, K. A. Ketchum, Z. Lai, Y. Lei, Z. Li, J. Li, Y. Liang, X. Lin, F. Lu, G. V. Merkulov, N. Milshina, H. M. Moore, A. K. Naik, V. A. Narayan, B. Neelam, D. Nusskern, D. B. Rusch, S. Salzberg, W. Shao, B. Shue, J. Sun, Z. Wang, A. Wang, X. Wang, J. Wang, M. Wei, R. Wides, C. Xiao, C. Yan, A. Yao, J. Ye, M. Zhan, W. Zhang, H. Zhang, Q. Zhao, L. Zheng, F. Zhong, W. Zhong, S. Zhu, S. Zhao, D. Gilbert, S. Baumhueter, G. Spier, C. Carter, A. Cravchik, T. Woodage, F. Ali, H. An, A. Awe, D. Baldwin, H. Baden, M. Barnstead, I. Barrow, K. Beeson, D. Busam, A. Carver, A. Center, M. L. Cheng, L. Curry, S. Danaher, L. Davenport, R. Desilets, S. Dietz, K. Dodson, L. Doup, S. Ferriera, N. Garg, A. Gluecksmann, B. Hart, J. Haynes, C. Haynes, C. Heiner, S. Hladun, D. Hostin, J. Houck, T. Howland, C. Ibegwam, J. Johnson, F. Kalush, L. Kline, S. Koduru, A. Love, F. Mann, D. May, S. McCawley, T. McIntosh, I. McMullen, M. Moy, L. Moy, B. Murphy, K. Nelson, C. Pfannkoch, E. Pratts, V. Puri, H. Qureshi, M. Reardon, R. Rodriguez, Y. H. Rogers, D. Romblad, B. Ruhfel, R. Scott, C. Sitter, M. Smallwood, E. Stewart, R. Strong, E. Suh, R. Thomas, N. N. Tint, S. Tse, C. Vech, G. Wang, J. Wetter,

S. Williams, M. Williams, S. Windsor, E. Winn-Deen, K. Wolfe, J. Zaveri, K. Zaveri, J. F. Abril, R. Guigó, M. J. Campbell, K. V. Sjolander, B. Karlak, A. Kejariwal, H. Mi, B. Lazareva, T. Hatton, A. Narechania, K. Diemer, A. Muruganujan, N. Guo, S. Sato, V. Bafna, S. Istrail, R. Lippert, R. Schwartz, B. Walenz, S. Yooseph, D. Allen, A. Basu, J. Baxendale, L. Blick, M. Caminha, J. Carnes-Stine, P. Caulk, Y. H. Chiang, M. Coyne, C. Dahlke, A. Mays, M. Dombroski, M. Donnelly, D. Ely, S. Esparham, C. Fosler, H. Gire, S. Glanowski, K. Glasser, A. Glodek, M. Gorokhov, K. Graham, B. Gropman, M. Harris, J. Heil, S. Henderson, J. Hoover, D. Jennings, C. Jordan, J. Jordan, J. Kasha, L. Kagan, C. Kraft, A. Levitsky, M. Lewis, X. Liu, J. Lopez, D. Ma, W. Majoros, J. McDaniel, S. Murphy, M. Newman, T. Nguyen, N. Nguyen, M. Nodell, S. Pan, J. Peck, M. Peterson, W. Rowe, R. Sanders, J. Scott, M. Simpson, T. Smith, A. Sprague, T. Stockwell, R. Turner, E. Venter, M. Wang, M. Wen, D. Wu, M. Wu, A. Xia, A. Zandieh, and X. Zhu. The sequence of the human genome. *Science (New York, N.Y.)*, 291(5507):1304–51, feb 2001.

C. Wählby, L. Kamentsky, Z. H. Liu, T. Riklin-Raviv, A. L. Conery, E. J. O'Rourke, K. L. Sokolnicki, O. Visvikis, V. Ljosa, J. E. Irazoqui, P. Golland, G. Ruvkun, F. M. Ausubel, and A. E. Carpenter. An image analysis toolbox for high-throughput C. elegans assays. *Nature methods*, 9(7):714–6, jul 2012.

J. D. Watson and F. H. C. Crick. Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. *Nature*, 171(4356):737–738, apr 1953.

L. S. Weinberger, J. C. Burnett, J. E. Toettcher, A. P. Arkin, and D. V. Schaffer. Stochastic gene expression in a lentiviral positive-feedback loop: HIV-1 Tat fluctuations drive phenotypic diversity. *Cell*, 122(2):169–182, 2005.

O. Wolkenhauer and U. Klingmueller. No TitlSystems Biology: From a Buzzword to a Life Sciences Approache. *BioForum Europe*, 2004.

A. M. Wu, J. E. Till, L. Siminovitch, and E. A. McCulloch. Cytological evidence for a relationship between normal hemotopoietic colony-forming cells and cells of the lymphoid system. *The Journal of experimental medicine*, 127(3):455–64, mar 1968.

A. R. Wu, N. F. Neff, T. Kalisky, P. Dalerba, B. Treutlein, M. E. Rothenberg, F. M. Mburu, G. L. Mantalas, S. Sim, M. F. Clarke, and S. R. Quake. Quantitative assessment of single-cell RNA-sequencing methods. *Nature methods*, 11(1):41–6, jan 2014.

Y. Xie, T. Yin, W. Wiegraebe, X. C. He, D. Miller, D. Stark, K. Perko, R. Alexander, J. Schwartz, J. C. Grindley, J. Park, J. S. Haug, J. P. Wunderlich, H. Li, S. Zhang,

T. Johnson, R. A. Feldman, and L. Li. Detection of functional haematopoietic stem cell niche using real-time imaging. *Nature*, 457(7225):97–101, jan 2009.

F. A. H. Xinghua Lou, X. Lou, and F. A. Hamprecht. Structured Learning for Cell Tracking. *Advances in Neural Information Processing Systems*, pages 1296–1304, 2011.

Z. Yin, H. Sailem, J. Sero, R. Ardy, S. T. C. Wong, and C. Bakal. How cells explore shape space: A quantitative statistical perspective of cellular morphogenesis. *BioEssays*, pages n/a–n/a, sep 2014.

I. T. Young, J. J. Gerbrands, and L. J. van Vliet. *Fundamentals of Image Processing (v.2.3)*. University of Delft, Delft, Netherlands, 2.2 edition, 1998.

M. D. Zeiler and R. Fergus. Stochastic Pooling for Regularization of Deep Convolutional Neural Networks. *Arxiv*, page 9, jan 2013.

J. Zhu and S. G. Emerson. Hematopoietic cytokines, transcription factors and lineage commitment. *Oncogene*, 21(21):3295–3313, 2002.

C. Zimmer, E. Labruyère, V. Meas-Yedid, N. Guillén, and J.-C. Olivo-Marin. Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool for cell-based drug testing. *IEEE transactions on medical imaging*, 21(10):1212–21, oct 2002.

R. P. Zinzen, K. Senger, M. Levine, and D. Papatsenko. Computational models for neurogenic gene expression in the Drosophila embryo. *Current biology : CB*, 16(13):1358–65, jul 2006.